

Editor: Dana Simian

**Proceedings of
The International Conference on
Applied Informatics**

ICDD

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

October 8 - 10, 2020, Sibiu, Romania

Editor DANA SIMIAN

**IMAGINATION, CREATIVITY,
DESIGN, DEVELOPMENT**

Proceedings of the International Conference
on Applied Informatics
ICDD

**October 08th – 10th, 2020
Sibiu, Romania**

Lucian Blaga University of Sibiu

Lucian Blaga University of Sibiu, 2020

Editor Dana Simian

All papers in this volume were peer review by two independent reviewers

ISSN-L 2069 – 864X

Associated Editor Laura Florentina Stoica

Proceedings of the International Conference on Applied
Informatics, ICDD
October 08th – 10th, 2020, Sibiu, Romania

Copyright @ 2020 All rights reserved to editors and authors

Preface

This volume contains refereed papers presented within the International Conference on Applied Informatics, “Imagination, Creativity, Design, Development“ - ICDD 2020, which was held between October 08th – 10th, at the Faculty of Sciences, Lucian Blaga University of Sibiu, Romania.

The conference is mainly addressed to young researchers from all over the world. The conference gives the participants the opportunity to discuss and present their research on informatics and related fields (like computational algebra, numerical calculus, bioinformatics, etc.). The conference welcomes submissions of original papers on all aspects of informatics and related fields ranging from new concepts and theoretical developments to advanced technologies and innovative applications. Specific topics of the conference included but are not restricted to: Algorithms and Data Structures, Graph Theory and Applications, Formal Languages and Compilers, Cryptography, Modelling and Simulation, Computer Programming, Computer Vision, Computer Graphics, Game Design, Data Mining, Distributed Computing, Artificial Intelligence, Service Oriented Applications, Networking, Grid Computing, Mobile Operating Systems, Scientific Computing, Software Engineering, Bioinformatics, Robotics, Computer Architecture, Evolutionary Computing, Multimedia Systems, Internet Communication and Technologies, Web Applications.

The conference has brought together participants from 4 countries (Bulgaria, Germany, Romania and Serbia).

We thank all the participants for their interesting talks and discussions. We also thank the members of the scientific committee for their help in reviewing the submitted papers and for their contributions to the scientific success of the conference.

December 2020

Dana Simian
Conference Chair

Motto:

“There are no limits, only your imagination”

Scientific committee

Kiril Alexiev - Bulgarian Academy of Sciences, Bulgaria
Vsevolod Arnaut - Moldova State University, Republic of Moldova
Alina Barbulescu - Ovidius University of Constanta, Romania
Arndt Balzer - University of Applied Sciences, Würzburg-Schweinfurt, Germany
Lasse Berntzen - Buskerud and Vestfold University College, Norway
Peter Braun - University of Applied Sciences, Würzburg-Schweinfurt, Germany
Amelia Bucur - Lucian Blaga University of Sibiu, Romania
Stelian Ciurea - Lucian Blaga University of Sibiu, Romania
Nicolae Constantinescu - University of Craiova, Romania
Daniela Danciulescu - University of Craiova, Romania
Oleksandr Dorokhov - Kharkiv National University of Economics, Ukraine
George Eleftherakis - The University of Sheffield International Faculty, City College
Thessaloniki, Greece
Ralf Fabian - Lucian Blaga University of Sibiu, Romania
Stefka Fidanova - Bulgarian Academy of Sciences, Bulgaria
Ulrich Fiedler - Bern University of Applied Science, Switzerland
Adrian Florea - Lucian Blaga University of Sibiu, Romania
Teresa Gonçalves - University of Evora, Portugal
Andrina Granić - University of Split, Croatia
Katalina Grigorova - University of Ruse, Bulgaria
Daniel Hunyadi - Lucian Blaga University of Sibiu, Romania
Saleema JS - Chris University, Bangalore, India
Milena Lazarova - Technical University of Sofia, Bulgaria
Lixin Liang - Tsinghua University, Beijing, China
Suzana Loskovska - "Ss. Cyril and Methodius" University in Skopje, Republic of
Macedonia
Rossitza S. Marinova - Concordia University of Edmonton, Canada
Gabriela Moise - Petroleum-Gas University of Ploiesti, Romania
G. Jose Moses - Raghu Engineering College Visakhapatnam, Andhra Pradesh, India
Mircea Musan - Lucian Blaga University of Sibiu, Romania
Mircea Iosif Neamtu - Lucian Blaga University of Sibiu, Romania
Grażyna Paliwoda-Pękosz - Cracow University of Economics, Poland
Camelia Pinteá - Technical University Cluj-Napoca, Romania
Antoniú Pític - Lucian Blaga University of Sibiu, Romania
Alina Pític - Lucian Blaga University of Sibiu, Romania
Cristina Popirlan - University of Craiova, Romania
Anca Ralescu - University of Cincinnati, United States of America
Mohammad Rezai - Sheffield Hallam University, United Kingdom
José Saias - University of Evora, Portugal
Livia Sangeorzan - Transilvania University of Brasov, Romania

Soraya Sedkaoui - Khemis Miliana University, Algeria
Dana Simian - Lucian Blaga University of Sibiu, Romania
Lior Solomovich - Kaye Academic College of Education, Israel
Ansgar Steland - RWTH Aachen University, Germany
Florin Stoica - Lucian Blaga University of Sibiu, Romania
Laura Florentina Stoica - Lucian Blaga University of Sibiu, Romania
Detlef Streitferdt - Ilmenau University of Technology, Germany
Grażyna Suchacka - University of Opole, Poland
Jolanta Tańcula - University of Opole, Poland
Milan Tuba - Singidunum University of Belgrade, Serbia
Anca Vasilescu - Transilvania University of Brasov, Romania
Dana Vasiloaica - Institute of Technology Sligo, Ireland
Sofia Visa - The College of Wooster, United States

Contents

Vector Embeddings for textual data in Business Intelligence	9
Eugen Becker, Daniel Wagner	
Tampering detection for in-vehicle systems	20
Roland Bolboacă	
Data classification and applications	27
Stela Caramihai, Marina-Larisa Indrean	
Proximity marketing, personalized offers and banking, all on mobile	33
Arina Ioana Cazacu	
Educational Data Mining using Supervised Learning Techniques	48
Alexandru-Mihail Crăciun	
DICOM image segmentation	61
Valentin-Gabriel Crăciun, Matei-Florin Gaură	
Approaches for reducing the number of intersections between the components of software architectures	68
Ligia – Izabela Crăciunescu	
Cashout	86
Alexandru Dancau, Paul-Robert Ceolca	
3D Head Reconstruction via Volumetric Regression	94
Andreea Dogaru	
Garbage Collector	104
Răzvan Gheorghe Filea	
Improving automatic meter reading using data generated from unpaired image-to-image translation	114
Andreas Filingner	
Towards an Industrial Recommendation System for Quality Improvement: Comparison of Python and C++ Implementations in an Edge- and Cloud-Computing Environment	124
Alexander M. Frühwald, Steffen Kastner, Anna-Maria Schmitt, Simon Haas, Leonhard Hösch, Lars Fichtel, Christian Bachmeir	
Bayesian networks with applicability in sonoluminescence	138
Bogdan-George Gros	
Automation of the Examination Timetabling	146
Janik Hemrich, Stella Konieczek, Justin Seegets	

Towards a comprehensive attack framework against commercial and private UAV Leonhard Hösch, Max Arndt, Lars Fichtel, Alexander M. Frühwald, Vitaliy Schreibmann, Helena Schmiedl, Andreas Schütz, Christian Bachmeir	159
Centralized management web app for real estate developers and HOAs	176
Felix Husac	
Automatic recognition of acoustic musical chords	182
Răzvan-Cosmin Linca	
Collection of software interfaces	196
Madalina Marinescu	
MARC (Monitored Automated Remote Car)	214
Christian Melchior	
Solving critical section problems by using a control thread	220
Milan Savić	
Software application for extracting information from Romanian identity cards	228
Constantin-Marius Stanciu	
Access control system based on QR Codes	238
Sebastian Stoica	
Analysis of the Operating Costs of a Decentralized App in the Ethereum Blockchain	250
André Stollberger, Tobias Fertig, Andreas E. Schütz, Karsten Huffstadt, Nicholas H. Müller	
IoT moisture monitoring system for improving indoor plants' growing conditions and optimizing maintenance routines	262
Minodora Suilea, Teodora Popa, Iuliana Buruiana	
Stellar Pointer	270
Eduard-Traian Ștefănescu	
Using Machine Learning in retail industry. A case study on Lidl fresh food market	284
Römer Walter	
List of authors	292
Sponsors	298

Vector Embeddings for textual data in Business Intelligence

Eugen Becker, Daniel Wagner

Abstract

Vector embeddings for textual data describes numerous methods to convert unstructured text data into a structured form. Especially in the field of Business Intelligence, analysis tools play an important role. Since most machine learning algorithms, like k-Nearest-Neighbor or k-Means, need numerical data as input, text data must be converted into a numerical form. Vector embeddings are a popular approach to achieve this, and over time many efficient methods have been developed. This paper provides an overview of Word2Vec, Doc2Vec and BERT, which create a vector representation of words and documents.

1 Introduction

Data is the basis of all strategies, plans, reports and at the end of the day all decisions in a company. Therefore, the economic success of a company often depends on the evaluation of its collected data and information. In the age of Big Data, where storage is becoming increasingly cheaper and data collection is more and more automated, huge amounts of data are created. According to a report by Splunk Inc. published in 2019, on a global study involving 1,300 IT managers from seven different countries, 55% of the data collected is not used in companies. This data is known as Dark Data[22]. What companies are becoming more aware of today is that Dark Data also contains valuable knowledge. However, it is usually an enormous amount of unstructured data that is neglected, because often the knowledge or the necessary analysis tools to process this kind of data are still missing. Text documents are ubiquitous in Business Intelligence[10], since many applications still produce or record textual data. This paper takes a closer look at the data analysis of textual data. For this purpose, we present methods to preprocess and convert unstructured text data into a structured form, in order to be able to evaluate it with known data analysis methods. We would like to point out that there are many different techniques, but we will focus on a very specific methodology, the so-called vector embeddings. Words, sentences or whole documents are represented as vectors in order to be evaluated by data mining algorithms. Vector embeddings for textual data are still being researched and developed, which speaks for its great benefit.

2 Vector Embeddings

Machines and their algorithms cannot process textual data very well. So, the data must be transformed into a computer friendly format, numbers or vectors. These vectors cannot be compared with the known vectors from geometry. They have a much higher number of dimensions. Therefore, they can be better compared with a list or array of numbers, in which the quantity of numbers corresponds to the dimensionality. There are many methods to convert text into such vectors. The simplest way to create a word vector is *One-Hot-Encoding*. It generates a vector with a dimensionality that corresponds to the number of unique words in the given text corpus. Each dimension represents a unique word. One-Hot-Encoding generates vectors with only one “1”, representing the encoded word and “0”’s representing the other

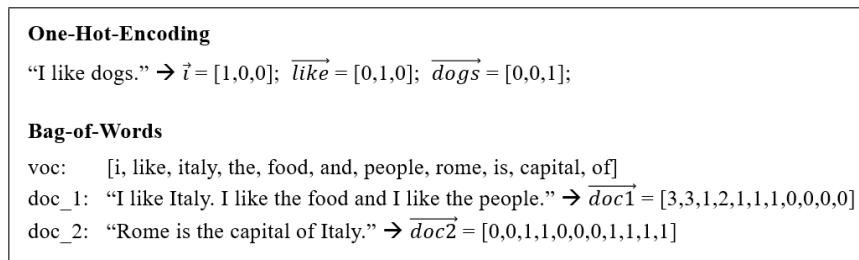


Figure 1: Example of One-Hot-Encoding and Bag-of-Words

words. A simple method to generate a vector representation of a document is *Bag-of-Words* (BoW)[7]. The term ”document” in the context of vector embeddings means text of variable length, ranging from single sentences to documents. BoW creates a representation of the occurrence of words in a document. A vocabulary of all unique words is created across all documents of a given text corpus. The frequency of each word occurring in the vocabulary is then counted per document. From this the document vector is created. Examples are given in Figure 1.

As simple as these methods are, they have many disadvantages. There are no relationships between One-Hot-Encoded vectors, because they all have the same cosine similarity. Thus, semantic and syntactic relationships between the words are not embedded. In addition, the dimensionality of the vectors increases by more different words, which can lead to memory and calculation problems. BoW on the other hand is a popular method to compare texts with each other by comparing their document vectors. Nevertheless, they ignore semantics and the order of the words, thus different sentences can have the same representation, if the same words are used. Other techniques are needed to overcome these problems and to make vector embeddings for textual data more efficient. That’s why we introduce Word2Vec, Doc2Vec and BERT. These methods use algorithms and models from the field of *Natural Language Processing* (NLP)[14] and are called *Word Embeddings*. NLP is a “sub-field of computer science, artificial intelligence and linguistics which aims at the understanding of natural language using computers”[1]. In the following, methods based on neural networks are presented, which provide new properties and possibilities. A neural network is a model that calculates predictions based on data.

3 Text Preprocessing

Preprocessing of a text corpus is one of the key components in vectorization of textual data. The quality of the vectors depends on this step. The most common text preprocessing methods are tokenization, filtering, lemmatization and stemming. In the following we will briefly describe them.

Tokenization is the task of breaking up a character string into pieces such as word parts, words or phrases. These pieces are called tokens. Depending on the task, punctuation marks are removed simultaneously.

Filtering is the process, where certain words are sorted out of the text corpus. A common filter is the removal of stop-words. These are words that appear frequently in the text without containing much content information (e.g. prepositions, conjunctions, etc.). Words occurring very rarely are also possibly of no significant relevance and can be removed from the corpus.

Lemmatization transforms each word into its basic form (lemma). Verbs are getting converted to their infinitive and nouns to their single form.

Stemming methods aim to obtain stem form (root) of derived words.[1]

4 Word2Vec

Word2Vec was developed in 2013 by a team of researchers led by Tomas Mikolov[16]. A distinction is made between two models, the *Continuous Bag-of-Words* (CBOW) and *Skip-gram*. Both are flat neural networks which take a text corpus as input and generate a multidimensional vector space, in which each word is represented as a feature vector. One feature of Word2Vec is the mapping of semantic and syntactic relationships between the words in the corpus to the word vectors. Such semantic connections can be identified by the context, where context corresponds to the words mentioned in the immediate vicinity of another word. The more often two words are in the same context, the higher is their semantic relationship or similarity[6]. A well-trained Word2Vec model places words with similar context closer together. So, word vectors representing these words will have similar numerical representations. Another feature of Word2Vec models is that they learn semantic and syntactic regularities independently, so-called analogies[16]. For example, the relationship between countries and their capital is mapped to the word vectors. "France is to Paris as Germany is to Berlin" (see Figure 2). Such word relationships can be mapped because arithmetic operations can be performed on Word2Vec vectors. The most famous example is: $\vec{v}ec("queen") \approx \vec{v}ec("king") - \vec{v}ec("man") + \vec{v}ec("woman")$. (Figure 3)

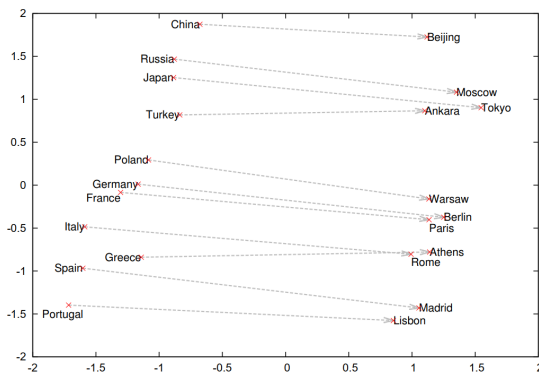


Figure 2: Country and capital vectors projected by Principal Component Analysis (PCA)[17]

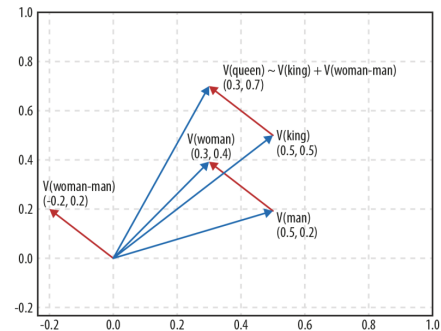


Figure 3: Arithmetic operations on Word2Vec vectors[18]

4.1 Continuous Bag-of-Words and Skip-gram model

As already mentioned, there are two different Word2Vec models for calculating word vectors. These differ in their architecture and thus in their approach to make predictions (see Figures 4 and 5). The Continuous Bag-of-Words model predicts the most likely word to a given context. The Skip-gram model, on the other hand, predicts the most likely context words to a given word. According to the paper of Mikolov et al.[16] the predictions' accuracy of the Skip-gram model achieves better results than the CBOW model, whereas the CBOW model requires a shorter duration for training. The accuracy of the CBOW model decreases, because the vectors of the context words get averaged to predict the target word.

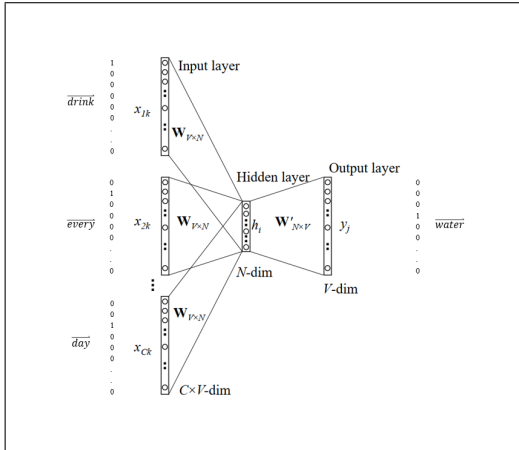


Figure 4: Continuous Bag-of-Words model[20]

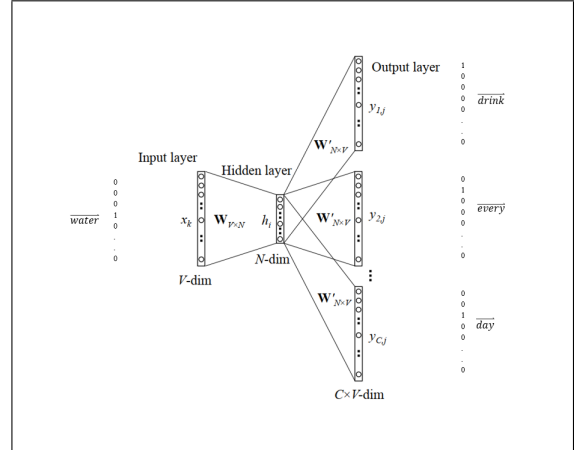


Figure 5: Skip-gram model[20]

4.2 Training and Functionality

For training the Word2Vec models various open source libraries are available, such as *Gensim* or *Tensorflow*. The following text corpus, consisting of two sentences, serves as a demonstration: "I enjoy playing TT. I like playing TT." (TT = table tennis)[2][20]. First, a text preprocessing should be applied, to get optimum results (see chapter Text Preprocessing). In this example a tokenization is enough.

$$\text{sentences}[0] = [i, \text{enjoy}, \text{playing}, \text{tt}] \quad \text{sentences}[1] = [i, \text{like}, \text{playing}, \text{tt}]$$

The resulting lists are passed to the model input layer. The model then creates a vocabulary and context-target word pairs (see Figure 6). The vocabulary contains each unique word as One-Hot-Encoded vector. The context-target word pairs consist of a list of context words and the corresponding target word. A window parameter determines how many neighboring words (context) should be considered before and after a word within a sentence. The hidden layer consists of two initial randomly weighted matrices \mathbf{W} and \mathbf{W}' (see Figures 4 and 5). The accuracy of the predictions depends on the values of the matrices. So, the training goal for both models is to adjust the matrices' weights so that the predictions are as accurate as possible. Therefore, all rows of the table with the context-target word pairs are used successively as input and a prediction is calculated. The CBoW model takes all context words and predicts the target word, while the Skip-gram model takes the target word as input and predicts the associated context words. Using the actual target word for CBoW or the actual context words for Skip-gram, the model evaluates the accuracy of its prediction after each run. A loss sum is calculated and the weights of the matrices are adjusted through Backpropagation. After the training is finished the matrix \mathbf{W} represents the Word2Vec word vectors, which is the model's special feature. Each row of the matrix corresponds to a word vector (see Figure 7). The matrix behaves like a reference column, where a simple multiplication with a word of the vocabulary outputs the corresponding word vector. For tools in the field of data analysis, only this matrix with the word vectors is of importance for further processing.

One-Hot-Encoding					
	i	enjoy	playing	tt	like
i	1	0	0	0	0
enjoy	0	1	0	0	0
playing	0	0	1	0	0
tt	0	0	0	1	0
like	0	0	0	0	1

Context			Target
enjoy	playing	tt	i
i	playing	tt	enjoy
i	enjoy	tt	playing
i	enjoy	playing	tt
like	playing	tt	i
i	playing	tt	like
i	like	tt	playing
i	like	playing	tt

Figure 6: Word2Vec vocabulary and context-target word pairs[2]

One hot encoding					
	I	enjoy	playing	TT	like
I	1	0	0	0	0
enjoy	0	1	0	0	0
playing	0	0	1	0	0
TT	0	0	0	1	0
like	0	0	0	0	1

Hidden layer		
3.38	-5.78	-0.98
1.78	3.19	3.63
-5.65	-1.66	-0.24
3.11	5.18	-3.17
1.66	3.34	3.76

word vector		
3.38	-5.78	-0.98
1.78	3.19	3.63
-5.65	-1.66	-0.24
3.11	5.18	-3.17
1.66	3.34	3.76

Figure 7: The words "enjoy" and "like" have a similar word vector representation, because of their similar semantical meaning[2]

4.3 Doc2Vec

Word2Vec offers the possibility to find similarities and relationships between single words, but in many applications documents should be compared with each other. Without a further processing of the word vectors, it is not possible to use Word2Vec on document level tasks. Hence, Tomas Mikolov and Quoc Le extended Continuous Bag-of-Words and Skip-gram model by a *Paragraph Vector* (document vector) and called the new algorithm Doc2Vec[12]. The name Paragraph Vector is to emphasize the fact that the method can be applied to variable-length pieces of texts, anything from a phrase or sentence to a large document. The algorithm creates unique vector representations by training two neural networks, which predict words in a document, the *Distributed Memory* (DM) and *Distributed Bag-of-Words* (DBOW) model. Like Word2Vec models DM and DBOW contain a matrix \mathbf{W} , representing the word vectors and in addition a matrix \mathbf{D} , representing the paragraph vectors. While paragraph vectors are unique among documents, the word vectors are shared among the text corpus.

Distributed Memory is an extension of the CBOW model and averages or concatenates the paragraph vector and word vectors to predict the next word in a context. The paragraph vector represents the missing information from the current context and can act as a memory of the documents' topic. Just like with Word2Vec the context size is determined by the parameter window. The paragraph vectors and word vectors are trained using stochastic gradient descent and the gradient is obtained via backpropagation. If using the model on a new document, an inference step is performed at prediction time to compute a new paragraph vector by fixing the word vectors and training the new paragraph vector until convergence. In summary, the algorithm itself has two key stages. First, training to get word vectors and paragraph vectors on already seen documents. Second, the inference stage to add new paragraph vectors in matrix \mathbf{D} and gradient descending on \mathbf{D} (see Figure 8).

Distributed Bag-of-Words is an extension of the Skip-gram model and predicts words randomly sampled from the document. DBOW is trained to predict the words in a small window. Opposed to Skip-gram model the input is not a single word but a paragraph vector (see Figure 9).

After training the matrix \mathbf{D} contains all document representations of the given text corpus, so any other extra operation is not necessary in order to obtain these vectors.

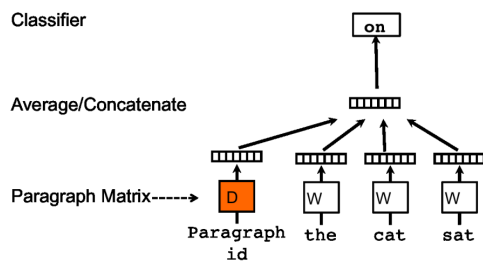


Figure 8: Distributed Memory model[12]

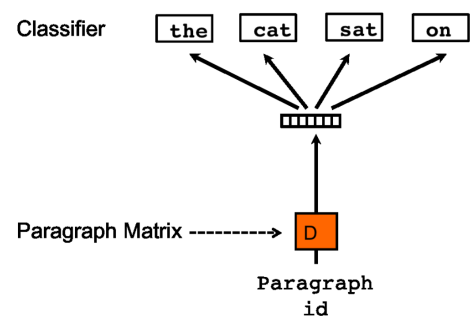


Figure 9: Distributed Bag-of-Words model[12]

5 BERT

In 2018, Devlin et al. (research team at Google AI) developed the new language model *Bidirectional Encoder Representations from Transformers* (BERT)[5]. Like Word2Vec, this is a vector embedding technique based on neural networks. The main difference is that BERT uses transformers. This is a new model architecture, especially designed for NLP tasks. The architecture allows the bidirectional processing of a text corpus[23]. BERT is a contextual language model. Unlike Word2Vec, which creates only one vector for each unique word, BERT creates multiple different vectors of a word depending on its meaning in the respective context. The bidirectional training achieves a deeper understanding of natural language context and flow[5]. The vectors are also placed in such a way, that words with similar meanings are close together. But BERT generates a separate vector for each meaning of a word. Thus, semantic clusters are formed. This is helpful for words with many different meanings. For example, the word "lie" can mean that someone tells the untruth, that someone is lying down or that a geographical indication is made. This is visualized in Figure 10.

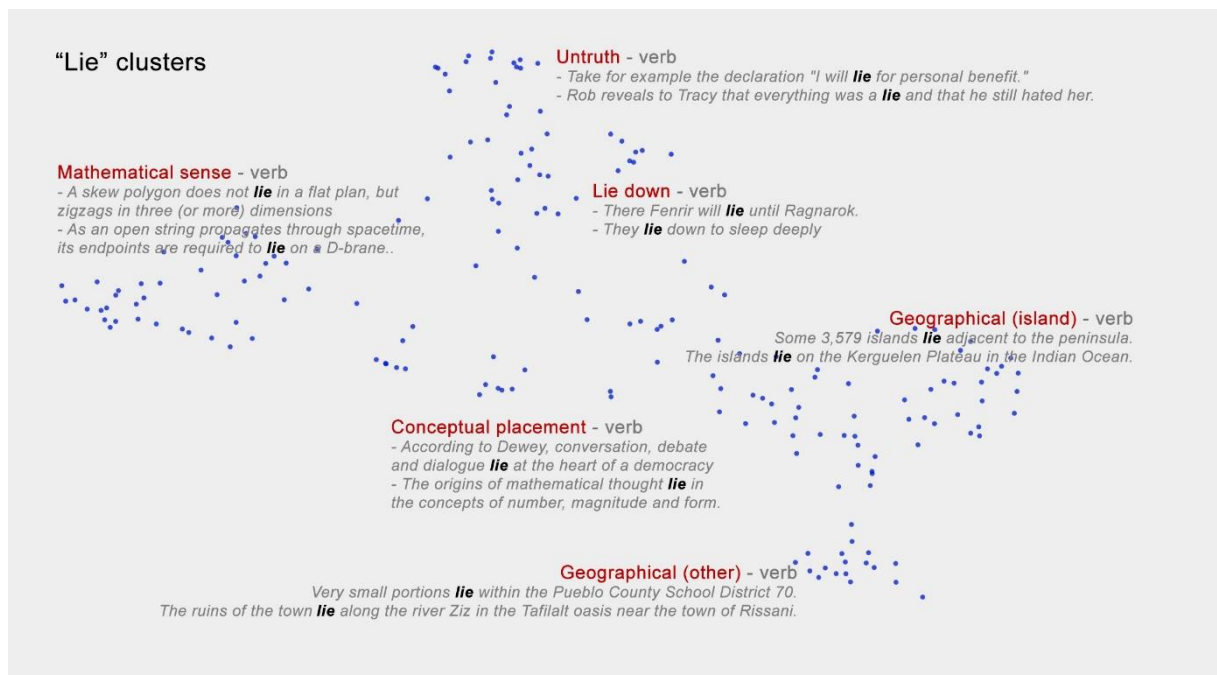


Figure 10: Vector clusters of the word "lie" in BERT[3]

5.1 Training

BERT embeds words and word parts in vectors. In this way, even complex words can be put together easily. BERT has a fixed vocabulary that varies accordingly to the model. In the paper of Devlin et al. the vocabulary consists of about 30,000 tokens[5]. These range from a single letter to whole words. Thus, for example, the word "playing" is divided into two parts "play" and "##ing" and embedded separately. In order to recognize that the two parts of the word belong together, the corresponding parts are marked with "##". This step is the tokenization of the preprocessing part.

For training, the input is represented as a token sequence. Each sequence represents a sentence or pairs of sentences, which belong together. The first token of a sequence is a special classification token ([CLS]). This is followed by the individual words of the sentences. The single records of a sequence are separated by a special token ([SEP]) (see Figure 11). In order to keep the information on which token belongs to which sentence, an embedding is created that contains exactly this information, the *Segment*

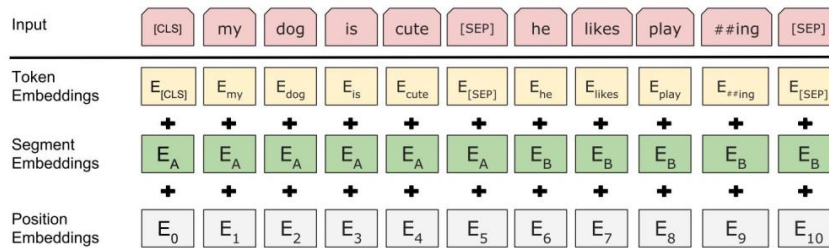


Figure 11: BERT embedding layers[5]

Embedding. This embedding is unique per sentence. The position of the token in the sequence is also shown in the *Position Embedding*. These embeddings are then combined to form a final embedding.

The training of the model takes place in two phases: First, it is pretrained with a large data set and then fine-tuned to the respective task with a significantly smaller data set.

The pretraining of the language model runs unsupervised and parallel based on two methods. The first method is the *Masked Language Model* (MLM). To train a bidirectional language model, a certain percentage of the tokens is masked. These masked tokens are then attempted to be predicted based on the context. 15% of all tokens in each sequence are randomly exchanged with the [MASK] token. The second method is the *Next Sentence Prediction* (NSP). In the training process the model receives a pair of sentences as input and learns to predict whether the second sentence was really the next sentence in the original document. Here 50% of the second sentences are original excerpts from the text corpus, while the other half of the sentences are randomly selected. Thus the model should predict only 50% of the sentences as original[5].

The reason for the good results in many different NLP tasks is, that the model is fine-tuned specifically to the respective task. This is done in a further supervised training step, using a labeled data set. Since the language model is already pretrained with a large data set and has therefore already an understanding of the language, only a small data set is necessary. A data set with only a few thousand entries is already enough.

Comparable to Word2Vec’s extension, Doc2Vec, it is possible to create document vectors. A commonly used method is to calculate the average of the generated sentence embeddings for each document. The result is an expressive document vector.

5.2 Different Models

However, BERT also has disadvantages. It is a very large model with many layers. This means that in non-computational-intensive systems the inference time, the time until a prediction is done, can be very high. In addition, BERT with its fixed vocabulary is very susceptible to terms of technical jargon (e.g. the medical field or words from the natural sciences). For these reasons there are many different models based on BERT.

TinyBERT[9], *ALBERT*[11] and *DistilBERT*[21] are models, that reduce the size of the model considerably without losing much of their accuracy. The authors of DistilBERT, for example, achieve this by using “knowledge distillation” to make the model 60% smaller and faster, while still 95% of the accuracy of BERT remains.

For special jargon there is, for example, *ClinicalBERT*[8] for the medical field or *BioBERT*[13] for the biomedical field.

There are also several models for different languages. *BETO* is the Spanish model, *CamemBERT*[15] the French model and *GermanBERT*[4] the German model.

6 Comparison of Word2Vec & BERT

Word2Vec provides flat neural network models consisting of an input layer, one hidden layer and an output layer, so the complexity of these models is relatively low. The dimensionality of Word2Vec vectors is not fixed and can be defined before training. In the paper of Mikolov et al.[16] vectors with 50 to 1000 dimensions were generated and it was found, that for better accuracy the vector dimensionality and the amount of training data must be increased together. However, in various NLP studies a dimensionality of 300 dimensions is commonly used for Word2Vec vectors and the Gensim library applies 100 dimensions as default size. BERT on the other hand consists of either 12 (= BERT_{BASE}) or 24 (= BERT_{LARGE}) transformer layers and is therefore a more complex model[5]. Due to the architecture of the BERT model the generated word vectors have a higher dimensionality with a fixed size of 768 or 1024 dimensions, depending on the number of transformer layers.

Both embedding methods can train models on a large text corpus. BERT for example is trained on the BooksCorpus (800M words) and English Wikipedia (2,500M words), resulting in a data set consisting of 3.3 Billion words. The training took the authors four days to complete[5].

As we are still at the beginning of our research, we could not yet perform benchmarking experiments between Word2Vec and BERT models, and we are not aware of any such publications. But it can be assumed, that under same conditions, both training and prediction time will last longer with BERT than with Word2Vec models. This is due to the greater complexity of the model architecture and vector dimensionality. Therefore, many different BERT models are pretrained and then fine-tuned for a specific task, using a smaller dataset to reduce training time.

In terms of model precision, it can be said that BERT outperforms Word2Vec in prediction accuracy and language understanding. One reason for this is that Word2Vec generates only one vector representation for each word, even if the same word has several different meanings. Whereas, BERT generates different vectors for each meaning of a word, based on the context in which the word is used. Another reason is that Word2Vec don't consider the word position, while BERT takes as input the position (index) of each word in the sentence before calculating the vector embedding. The last reason for better accuracy of BERT models is that Word2Vec generates vector embeddings only at word level. The number of generated vectors corresponds to the number of unique words in the text corpus. That means words encountered outside the vocabulary space are not supported by Word2Vec models. BERT, on the other hand, learns representations at a word and word parts level. Thus, a BERT model can generate vector representation of any arbitrary word and is not limited to the vocabulary space of the text corpus.

7 Usage of Vector Embeddings

Vector embeddings are increasingly used in Business Intelligence, especially in the field of Text Mining. Text Mining is the process of extracting high quality information from any kind of textual data[19]. Texts are analyzed according to recurring patterns, formulations and keywords. The term covers methods for text preprocessing, structuring and the application of known analysis algorithms from data mining. Analysis and Business Intelligence solutions use text mining for many different applications. Vector embeddings are mainly used for text classification, sentiment analysis, text clustering and text summarization. In the following we will briefly describe them.

Text Classification: The system determines to which predefined category a document or word should be assigned. Classification algorithms can be applied to word vectors or document vectors, such as the Support Vector Machine classifier. This forms a divider and assigns new words or documents to the

respective category. With BERT, classification algorithms do not have to be used because the model can classify its vectors itself using the classification token. Thus, for example, documents can be presorted or collected. Spam filters use such classifications to evaluate the content of an e-mail. E-mail redirection also uses this technique to forward messages from a group mailbox to the right person, based on the content.

Sentiment Analysis: The system records the subjective opinion or feeling of a person and recognizes, for example, whether it is positive, negative or neutral. Sentiment analysis is similar to Text Classification and is often used in the field of e-commerce. It helps companies to identify trends, patterns and opinions within different text sources. For example, product reviews can be automatically evaluated.

Text Clustering: The system independently divides a collection of documents or single words into topics or categories. Similar content gets summarized into groups or clusters by cluster analysis algorithms, which are applied to the word or document vectors, e. g. the k-means algorithm. This allows a fast information search or filtering. Search engines use text clustering to provide meaningful search results for the word or text passed. For example, search engines in online shops, where the customer receives products, which are similar in their name.

Text Summarization: The system independently summarizes the content of a document. It is also possible to combine several documents from a subject area. The system generates a topic word or a series of words that describe the content. This can be very useful if the existing text files are very large.[1]

8 Conclusion

The goal of this paper was to introduce the concept of vector embeddings for textual data. These are methods that create a vector representation of words and documents and allow data mining algorithms to analyze unstructured textual data. Vector embeddings enable great applications, especially in the Business Intelligence context. Three word embedding methods from the field of Natural Language Processing were presented, Word2Vec, BERT and an extension of Word2Vec for document embedding, called Doc2Vec. In addition to the word vectors Doc2Vec generates paragraph vectors, that are unique for each document. These embedding methods use neural networks, which take a text corpus as input and generate a multidimensional vector space, where all vectors are related to each other. The vectors get arranged in a way that they represent semantic and syntactic relationships between words and thus the natural language. Word2Vec and Doc2Vec are an older, but very popular, approach for embedding words and documents, while BERT was published recently in 2018. As we are still at the beginning of our research, we could not yet perform benchmarking experiments between the presented models, but from our research we can say that BERT achieves state-of-the-art accuracy in many different NLP tasks. Our future work will include benchmarking experiments between different word embedding techniques. Furthermore, we want to develop applications based on these models to further study their practical usage in the field of Business Intelligence.

Acknowledgement: "This work was supervised by *Prof. Dr. Frank-Michael Schleif*, from *University of Applied Sciences Würzburg-Schweinfurt*".

References

- [1] Mehdi Allahyari, Seyedamin Pouriye, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. <http://arxiv.org/pdf/1707.02919v2>, 2017.
- [2] V. Kishore Ayyadevara. *Word2vec*, pages 167–178. Apress, Berkeley, CA, 2018.

-
- [3] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of bert. <https://arxiv.org/pdf/1906.02715.pdf>, 2019.
- [4] deepset GmbH. deepset - open sourcing german bert. <https://deepset.ai/german-bert>, 2020.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. <http://arxiv.org/pdf/1810.04805v2>, 2018.
- [6] Rene Goetz, Alexander Piazza, and Freimut Bodendorf. Hybrider ansatz zur automatisierten themen-klassifizierung von produktrezensionen. *HMD Praxis der Wirtschaftsinformatik*, 56(5):1–15, 2019.
- [7] Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.
- [8] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. <https://arxiv.org/pdf/1904.05342>, 2019.
- [9] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. <https://arxiv.org/pdf/1909.10351>, 2019.
- [10] Hans-Georg Kemper, Walid Mehanna, and Carsten Unger. *Business Intelligence — Begriffsabgrenzung und Ordnungsrahmen*, pages 1–11. Vieweg+Teubner Verlag, Wiesbaden, 2004.
- [11] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. <https://arxiv.org/pdf/1909.11942>, 2019.
- [12] Quoc Le V and Tomas Mikolov. Distributed representations of sentences and documents. <http://arxiv.org/pdf/1405.4053v2>, 2014.
- [13] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. <https://arxiv.org/pdf/1901.08746>, 2019.
- [14] Elizabeth D Liddy. Natural language processing. 2001.
- [15] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamel Seddah, and Benoît Sagot. Camembert: a tasty french language model. <https://arxiv.org/pdf/1911.03894>, 2019.
- [16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. <http://arxiv.org/pdf/1301.3781v3>, 2013.
- [17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.
- [18] Douwe Osinga. *Deep Learning Kochbuch: Praxisrezepte für einen schnellen Einstieg*, pages 37–50. O’Reilly, Heidelberg, 1. auflage edition, 2019.
- [19] Foster Provost and Tom Fawcett. *Data Science für Unternehmen: Data Mining und datenanalytisches Denken praktisch anwenden*, pages 295–298. mitp Business. MITP, Frechen, 1. auflage edition, 2017.
- [20] Xin Rong. word2vec parameter learning explained. <https://arxiv.org/pdf/1411.2738>, 2014.
- [21] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. <http://arxiv.org/pdf/1910.01108v4/>, 2019.

- [22] Splunk Inc. The state of dark data: Industry leaders reveal the gap between ai’s potential and today’s data reality. <https://www.splunk.com/pdfs/dark-data/the-state-of-dark-data-report.pdf>, 2019.
- [23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. <https://arxiv.org/pdf/1706.03762>, 2017.

Eugen Becker
Faculty of Computer Science
and Business Information Systems
University of Applied Sciences Würzburg-Schweinfurt
Sanderheinrichsleitenweg 20, 97070 Würzburg
Germany
E-mail: *eugen.becker@student.fhws.de*

Daniel Wagner
Faculty of Computer Science
and Business Information Systems
University of Applied Sciences Würzburg-Schweinfurt
Sanderheinrichsleitenweg 20, 97070 Würzburg
Germany
E-mail: *daniel.wagner.2@student.fhws.de*

Tampering detection for in-vehicle systems

Roland Bolboacă

Abstract

Tampering denotes the procedure that changes the behavior of a process (e.g., automotive system, process control system) for particular advantages (e.g., financial, operational). Compared to cyber attacks, the scope of tampering is not to cause specific damages, but to alter the system's behavior in order for the owner to gain particular advantages. To the extent of our knowledge, this paper documents the first approach to detect tampering in automotive systems. The approach embraces a two-step methodology. At first, Principle component analysis (PCA) is applied to decrease the complexity of the data exploration and analysis procedure. Then, the Gaussian mixture model is applied to classify the data and to detect tampering attempts. Experimental results based on measurements extracted from the OBD II system of a KIA automobile driven by 10 drivers over a total period of 23 hours in Seoul, South Korea demonstrate the applicability of the developed approach in real-world scenarios.

1 Introduction

Anomaly detection, that is, the detection of deviations from a normal behavior learned apriori, is a well-established field of research. In actual fact, researchers have developed a wide range of practical instruments that embrace the recent advancements from the field of artificial intelligence. These are nowadays readily applicable in the detection of anomalous behavior caused by cyber attacks, as well as faults and disruptions.

To this end, a large body of research has been found to embrace anomaly detection algorithms for the detection of cyber attacks in various cases ranging from traditional Information and Communication Technology (ICT) systems [1, 2], to the latest technological paradigms such as intelligent vehicles [3], Internet of Things [4, 5], and Industry 4.0 [6]. Nevertheless, while cyber attack detection and mitigation received significant attention, we believe that an emerging direction requires a similar level of involvement. To this end, *tampering* denotes the procedure that changes the behavior of a process (e.g., automotive system, process control system) for particular advantages (e.g., financial, operational). Compared to cyber attacks, the scope of tampering is not to cause specific damages, but to alter the system's behavior in order for the owner to gain particular advantages. A fundamental distinction between tampering and cyber attacks, is that tampering occurs with the owner's consent. In the case of automotive systems we find a wide range of tampering-like behavior, namely car tuning, the replacement of sensors, and of Electronic Control Units (ECUs). A particular example is the heavy-load transport vehicle domain, where we find frequent tampering aimed at reporting false data regarding gas consumption in order to reduce the costs of after-treatment systems. However, this drastically raises the amount of NOx emissions in the public environment [7].

To the extent of our knowledge, this paper documents the first approach to detect tampering in automotive systems.

In order to address these issues, this paper documents, to the extent of our knowledge, the first approach to detect tampering in automotive systems. The approach embraces a two-step methodology.

At first, Principle component analysis (PCA) is applied to decrease the complexity of the data exploration and analysis procedure. Then, the Gaussian mixture model is applied to classify the data and to detect tampering attempts. Experimental results based on measurements extracted from the OBD II system of a KIA automobile driven by 10 drivers over a total period of 23 hours in Seoul, South Korea [8], demonstrate the proposed method's capability to detect anomalies caused by tampering attempts in automotive systems.

The remainder of this paper is structured as follows. Section 2 presents the state-of-the-art techniques in the field; and Section 3 describes the proposed approach. Section 4 showcases the experimental results used in the validation of the developed methodology. The paper concludes in Section 5.

2 Background and related studies

2.1 Architecture of automotive systems

During 1970 - 1980 car manufacturers started incorporating electronic devices as a mean to control and diagnose automotive systems. As the automotive industry evolved the performance and the reliability of the devices found in such systems increased significantly. Nowadays, most of the processes implemented in modern vehicles are controlled with the use of Electronic Control Units (ECU). An ECU is an embedded micro-controller having various functions inside the car. Most modern cars can incorporate a large number of ECUs with various functionalities from engine control, automatic transmission, ABS control, airbag control to climate control, electric window control, and even telemetry control. The networks on which the ECUs communicate with each other are numerous, CAN buses, LIN, FlexRay, etc.

The CAN protocol defines a bus communication standard, where each message sent by an ECU is broadcasted on the bus network. This denotes that each ECU connected to the CAN bus will receive all packets, and it can decide whether to accept or drop the packet. Usually, a vehicle encompasses multiple CAN buses, the data flow between them is ensured via gateways. One of the requirements needed for the connected ECUs is to be able to communicate via a common "language". This "language" is the On Board Diagnosis (OBD) system [8].

The OBD standard was initially released as a result of the need to reduce the vehicle emissions by monitoring the processes inside a vehicle and to assure the engine was operating at optimum efficiency. The latest version of OBD is OBD II, a standard introduced in 1996. The OBD II systems offer universal diagnosis and inspection methods to assure the vehicle is functioning within OEM standards. In a nutshell, the main role of the OBD II can be summarized as: (i) the reduction of fuel consumption by optimizing the engine's parameters; (ii) emission reduction; (iii) the reduction of the notification time, in case of faults, by constantly monitoring the vehicle's subsystems; and (iv) aid in the diagnosis of defective equipment. Furthermore, the OBD II standard also defines the connectors which need to be incorporated in the vehicle in order to export CAN data [8].

2.2 Related studies

In the scientific literature we find a wide variety of techniques documenting the development of CAN communication-based anomaly detection systems. Groza and Murvay [9] developed an approach that focuses on the number of CAN identifiers (CIDs), their periodicity as well as the entropy carried by the data-field associated to a particular CID.

Similarly to the work of Groza and Murvay, other researchers have acknowledged that anomaly detection algorithms need to be lightweight in order to be practically applicable to in-vehicle systems [10, 11].

In [12], Moore, *et al.* observed that CAN communications exhibit a certain level of regularity in terms of the timing of CAN frames. Based on the identified communication patterns, a network-based intrusion detection strategy was developed that measures the inter-signal wait times, and issues alerts in case communications deviate from the patterns learned a priori.

Moving towards the direction of more complex algorithms, we have the work of Narayanan, *et al.* [13]. The developed approach embraces Hidden Markov Models (HMM) to learn offline the communication

patterns between ECUs. In a similar direction we find the work of Theissler, *et al.* [14], where multivariate time series were recorded and analyzed from a moving (on-road) vehicle.

In contrast to the above-mentioned studies, this work presents a methodology aimed to detect tampering attempts on the data layer. An important observation is that in most of the cases tampering does not affect communications, but it mainly impacts the application layer, that is, the data transported by the underlying communication system. While most prior works targeted the underlying communication system, the undertaken methodologies may still be applicable to the detection of tampering. Therefore, this work demonstrates that priorly developed intelligent techniques may also find applications in the detection of tampering.

3 Developed approach

3.1 Overview

The proposed tampering detection method comprises two parts. Firstly, the dimensionality reduction, that aims to decrease the number of features of the data, while retaining a high percentage of the information. Secondly, the output from the dimensionality reduction algorithm is transferred to a probabilistic model-based clustering algorithm that outputs a collection of clusters, which are afterwards used in the tampering detection process.

3.2 Step 1: Dimensionality reduction

The approach used for dimensionality reduction is the Principle component analysis (PCA). PCA builds the covariance-variance matrices in order to provide an explanation of dependencies between variables. Furthermore, PCA provides this explanation via a reduced set of variables. For a given dataset of size $n \times p$, where n denotes the number of observations and p denotes the number of attributes, the PCA procedure outputs a new dataset of the same size, where each column holds a principal component. The largest possible variance is explained by the first component and in a decreasing order the subsequent components explain the remaining variance.

The first step in the PCA procedure consists of computing the covariance matrix. Let \mathbf{X} denote a matrix of m random variables such that $\mathbf{X} = [X_1 \ X_2 \ \dots \ X_m]$, where X_i is a column vector of n observations. Let $\hat{\mathbf{X}}$ denote the normalized matrix of \mathbf{X} . Then, the covariance matrix, Σ , defined as:

$$\Sigma = \frac{1}{n-1} \hat{\mathbf{X}}^T \hat{\mathbf{X}}. \quad (1)$$

Next, the PCA procedure computes the eigenvalues and the eigenvectors for Σ such that:

$$\Lambda = \mathbf{U}^T \Sigma \mathbf{U}, \quad (2)$$

where \mathbf{U} denotes a matrix of eigenvectors for the covariance matrix Σ . The dimensionality reduction entails keeping only the first l principal components, resulting in a new dataset Y of size $n \times l$.

3.3 Step 2: Data clustering and tampering detection

The data clustering utilizes the Gaussian mixture technique. These are centered on a combination of multiple multivariate normal density components, such that each observation is categorized to belong to a certain data cluster. Fundamentally, the clustering process entails the application of an Expectation Maximization (EM) algorithm to select the component that maximizes the posterior probability. This means that the algorithm uses the EM in order to determine the probability of measurements to fit into a particular cluster.

Cyber attacks may produce outliers with low fit values. This aspect can be useful in the detection of outlier observations as a density measure approach is used. Therefore, prior works have also used the Gaussian mixture model to detect anomalous behavior caused by cyber attacks [15]. In this work, as documented later, the same clustering approach will be used to detect tampering in automotive systems.

Field Name	Range	Unit
Fuel consumption	0-10000	mcc
Accelerator pedal value	0-100	%
Throttle position signal	0-100	%
Intake air pressure	0-255	kPA
Absolute throttle position	0-100	%
Engine speed	0-6000	rpm
Torque of friction	0-100	%
Engine coolant temperature	(-)40-(+)215	*C
Engine torque	0-100	%
Calculated load value	0-100	%
Maximum indicated engine torque	0-100	%
Wheel velocity front left-hand	0-511.75	km/h
Wheel velocity rear right-hand	0-511.75	km/h
Wheel velocity front right-hand	0-511.75	km/h
Wheel velocity rear left-hand	0-511.75	km/h
Torque converter turbine speed	0-16383.75	rpm
Vehicle speed	0-200	km/h

Table 1: Features used in the experiments.

Lastly, the detection process is divided in two parts. The first part involves clustering a large enough, tamper free dataset Y , so that the resulting k clusters can incorporate all the possible operating states the system can occupy. Subsequently, to classify a new data point p , the proposed method computes the Mahalanobis distance from p to each of the previously k computed clusters. The Mahalanobis distance (denoted by MD) from point p to a distribution D_k with mean μ_k and variance-covariance matrix S_k , is given by Equation 3. Here, the distance represents how far the point p is from the distribution D_k in terms of standard deviations. A threshold-based decision, is further used to classify p as normal or anomalous. In the same equation, $(p - \mu_k)^T$ denotes the transposition of the $(p - \mu_k)$ vector.

$$MD(p, D_k) = \sqrt{(p - \mu_k)S_k^{-1}(p - \mu_k)^T}. \quad (3)$$

4 Experimental results

This section describes the experiments conducted in order to test the proposed in-vehicle tampering detection method. A prototype for the developed approach was implemented using MATLAB.

This section is further divided into three parts. Firstly, a brief overview of the dataset used for the experiments is provided. Secondly, the process behind the creation of the anomalous datasets is described. Lastly, the evaluation of the proposed detection method is presented.

4.1 Dataset description

The dataset used in the experiments incorporates measurements extracted from the OBD II system of a KIA automobile driven by 10 drivers over a total period of 23 hours in Seoul, South Korea [8]. The original dataset consists of 94380 observations containing 51 features. For our experiments 9000 data points were randomly selected, containing 17 features. Table 1 provides a detailed description of the selected features. From the dataset containing 9000 entries, 7000 random points were used to create a tamper-free dataset, while the remaining 2000 entries were used to create the anomalous dataset.

4.2 Anomalous Datasets

As previously mentioned, from the original dataset, 17 features were selected for the experiments. To be able to test our proposed method with anomalous data ranging from 1 feature to all 17 features we ended up creating 17 new anomalous datasets. The anomalous values for each feature x_i were computed by estimating the mean (μ_{x_i}) and standard deviation values (σ_{x_i}). Afterwards, a new value in the range of $(\mu_{x_i}, \mu_{x_i} \pm \sigma_{x_i} * r_i)$ was generated, where r_i is a random real value in the range of $(3, 5]$. Each newly computed value was limited to the valid measurement intervals, as given in Table 1. Overall, each of the resulting 17 anomalous datasets contained 500 entries.

4.3 Method evaluation

As detailed in Section 3, the first step in the developed methodology is the dimensionality reduction. Accordingly, the dataset was normalized with respect to the limits imposed by each feature. To achieve the dimensionality reduction, the PCA algorithm was applied on the dataset containing the 17 features. Based on the output of the PCA algorithm, only the first 2 largest principal components were kept, the selected components explaining $> 90\%$ of the total variance. The dimensionality reduction was followed by an initial clustering of the newly obtained dataset, using GMM. The result has been visually illustrated in Figure 1.

One of the parameters needed for the clustering algorithm is the number of clusters k . The optimum value for k was computed using Bayesian information criterion (BIC); the best results were obtained with 3 clusters. Another parameter needed was the detection threshold, denoted by δ . Based on the statistical analysis and the initial clustering result, the value for δ was set at $\delta = 3$, denoting three standard deviations.

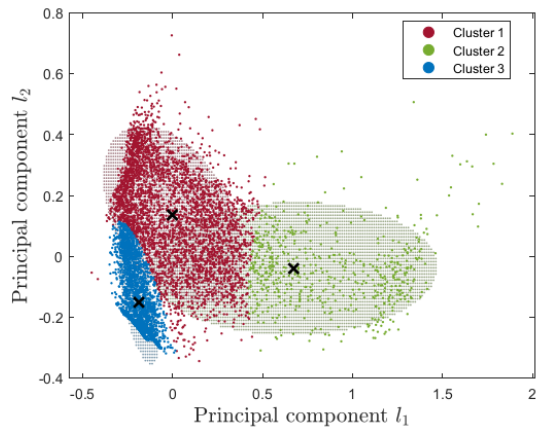


Figure 1: 2D illustration of the tamper-free dataset clustering results.

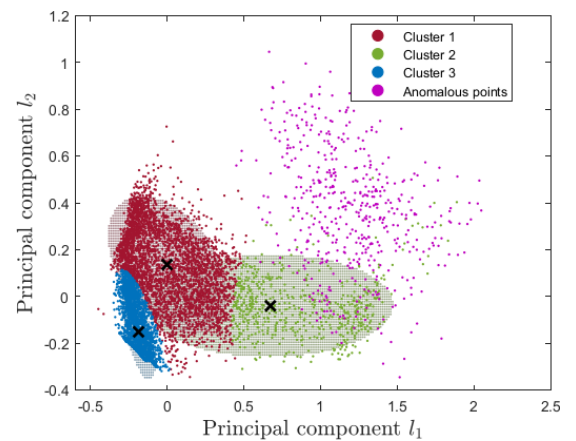


Figure 2: 2D illustration of the clustering results with 7 affected features.

Next, the proposed method was evaluated in terms of false negative rates for each of the 17 anomalous datasets. For each of the data points p the Mahalanobis distance $MD(p, C_k)$ to the clusters C_k was computed and compared to the δ threshold. A point was marked as anomalous if the resulting $MD(p, C_k)$ was greater than δ for all k clusters. The results of this analysis have been summarized in Figure 4.

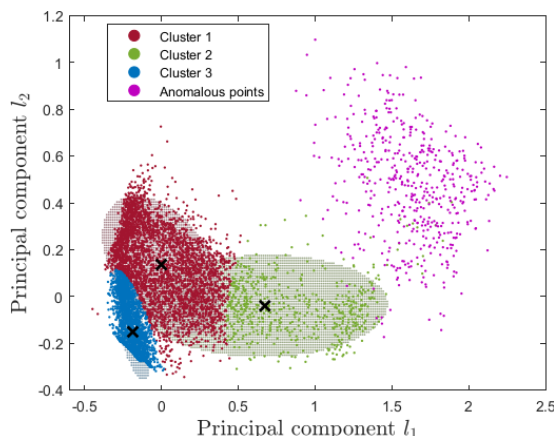


Figure 3: 2D illustration of the clustering results with 17 affected features.

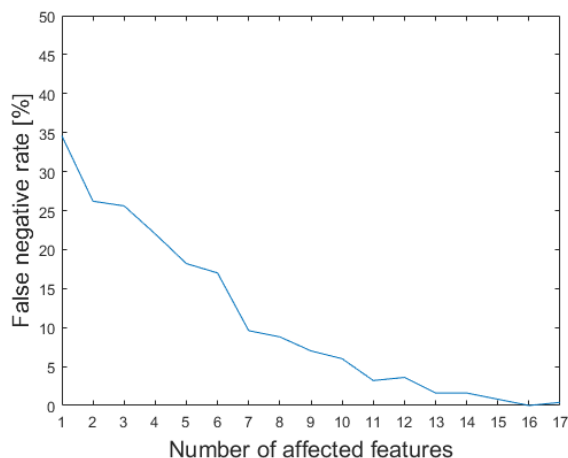


Figure 4: False Negative Rates.

As can be observed, the proposed method's false negative rates are higher for the anomalous datasets, where only a few features contain anomalous data. In contrast, the proposed algorithm shows promising results for a higher number of features with anomalous values. As an example, Figure 2 visualizes 7 features that contain anomalous data, and Figure 3 depicts a scenario where 17 features contain anomalous data. As can be seen in both figures, after applying PCA, the distances between the initial clusters and the anomalous points increases as we increase the number of anomalous features. Furthermore, a clearly visible result is that the anomalous values tend to form a separate cluster with the increase of the affected number of features.

5 Conclusions

We presented an approach for detecting tampering attempts in automotive systems. The proposed method describes a tampering detection technique based on PCA, Gaussian mixture model clustering and Mahalanobis distance. Experimental results based on a real-world dataset containing measurements extracted from the OBD II system of a KIA automobile driven by 10 drivers proved the applicability of the approach in realistic scenarios.

As interesting extensions we intend to improve the proposed tampering detection method by exploring the possibility of integrating other algorithms and we plan to apply this method to a real automotive environment.

Acknowledgement: This work was supervised by Assoc. prof. dr. habil. eng. *Béla Genge* and Assoc. prof. dr. eng. *Piroska Haller*, from "George Emil Palade" University of Medicine, Pharmacy, Sciences and Technology of Târgu Mureș, Romania..

References

- [1] Anderson Hiroshi Hamamoto and Luiz Fernando Carvalho and Lucas Dias Hiera Sampaio and Taufik Abrão and Mario Lemes Proença, Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic *Expert Systems with Applications*, 390 - 402, 2018.
- [2] Kardani-Moghaddam, Sara and Buyya, Rajkumar and Ramamohanarao, Kotagiri, Performance anomaly detection using isolation-trees in heterogeneous workloads of web applications in computing clouds *Concurrency and Computation: Practice and Experience*, e5306, 2019.
- [3] O. Avatefipour and A. S. Al-Sumaiti and A. M. El-Sherbeeney and E. M. Awwad and M. A. Elmeligy and M. A. Mohamed and H. Malik, An Intelligent Secured Framework for Cyberattack Detection in Electric Vehicles' CAN Bus Using Machine Learning *JIEEE Access*, 127580-127592, 2019.

- [4] C. Enăchescu and H. Sándor and B. Genge, A Multi-Model-based Approach to Detect Cyber Stealth Attacks in Industrial Internet of Things *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 1-6, 2019.
- [5] B. Genge and P. Haller and C. Enăchescu, Anomaly Detection in Aging Industrial Internet of Things *IEEE Access*, 74217-74230, 2019.
- [6] R. Bolboacă and B. Genge and P. Haller, Using Side-Channels to Detect Abnormal Behavior in Industrial Control Systems *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 435-441, 2019.
- [7] Pöhler, Denis and Adler, Tim and Krufczik, Christopher and Horbanski, Martin and Lampel, Johannes and Platt, Ulrich, Real Driving NOx Emissions of European Trucks and Detection of Manipulated Emission Systems *EGU General Assembly Conference Abstracts*, 13991, 2017.
- [8] Martinelli, Fabio and Mercaldo, Francesco and Orlando, Albina and Nardone, Vittoria and Santone, Antonella and Kumar, Arun, Human behavior characterization for driving style recognition in vehicle system *Computers & Electrical Engineering*, 2018.
- [9] Groza, Bogdan and Murvay, Pal-Stefan, Identity-Based Key Exchange on In-Vehicle Networks: CAN-FD & FlexRay *Sensors*, 4919, 2019.
- [10] Stabili, Dario and Marchetti, Mirco and Colajanni, Michele, Detecting attacks to internal vehicle networks through Hamming distance *2017 AEIT International Annual Conference*, 1-6, 2017.
- [11] Cho, Kyong-Tak and Shin, Kang G., Fingerprinting Electronic Control Units for Vehicle Intrusion Detection *Proceedings of the 25th USENIX Conference on Security Symposium*, 911–927, 2016.
- [12] Moore, Michael R. and Bridges, Robert A. and Combs, Frank L. and Starr, Michael S. and Prowell, Stacy J., Modeling Inter-Signal Arrival Times for Accurate Detection of CAN Bus Signal Injection Attacks: A Data-Driven Approach to in-Vehicle Intrusion Detection *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, 1-4, 2017.
- [13] Sandeep Nair Narayanan, Sudip Mittal, and Anupam Joshi, OBD SecureAlert: An Anomaly Detection System for Vehicles *IEEE Workshop on Smart Service Systems (SmartSys 2016)*, 2016.
- [14] Andreas Theissler, Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection *Knowledge-Based Systems*, 163 - 173, 2017.
- [15] I. Kiss and B. Genge and P. Haller, A clustering-based approach to detect cyber attacks in process control systems *A clustering-based approach to detect cyber attacks in process control systems*, 142-148, 2015.

Roland BOLBOACĂ
"George Emil Palade" University of Medicine, Pharmacy,
Sciences and Technology of Târgu Mureş.
Faculty of Engineering and Information Technology
Str N. Iorga no. 1, Târgu Mureş
ROMANIA
E-mail: roland.bolboaca@umfst.ro

Data classification and applications

Stela Caramihai, Marina-Larisa Indrean

Abstract

Data modelling and classification is of interest in any field, because any activity requires certain calculations. This paper presents a Matlab application that aims to model and classify data, in order to obtain results, which will be used as a starting point in the future activities. Modelling, in the context of this paper, is performed on a set of concentrations of chemicals taken from the soil. The purpose of modelling is to be able to classify data, depending on the results obtained from the transformations made. We want to classify the soil types in different groups / categories. Thus, the optimal combination of soil grouping is calculated, using an genetic algorithm and the K-means algorithm.

1 Introduction

The agricultural field has known an important development in the last years, existing at the moment a high request of systems with the help of which the productions can increase as high as possible.

Currently, there is a growing demand for development in terms of digitalization of agriculture, its representatives talking about the deficiency of software systems that can make their work easier, first of all starting from the basic element of this activity, namely the soil. We made this study because, during time, we met various problems that led to the desire to create a software system to solve the so-called crop rotation. [5]

2 Methodology

2.1 K-means algorithm applied on initial data

K-means clustering is a method of clustering based on „distances” between the clustered objects.

The „k” in K-means refers to the number of clusters we want to obtain at the end.

Having the „k” number we calculate the best „k” clusters by calculating the distances between the objects we want to cluster.

2.1.1 Hubert Index

The Hubert Index is a graphical method of determining the optimal number of clusters. In the graph of this method, we look for a point in the first part of the graph, which corresponds of a significant increase in value, found in the second part of the graph.

2.1.2 Silhouette

Silhouette is a method that measures the quality of clustering. It measures how well the data is divided into clusters.

2.2 Genetic algorithm for data classification

After applying K-means algorithm, with k equals to 2, on the data set, we want to use genetic algorithm to modify data in order to obtain the best solution possible.

GA is an algorithm based on natural reproduction in order to obtain the best generation possible.

Genetic algorithm requires multiple steps.

2.2.1 Fitness value calculation

Each chromosome must be comparable to the others. For this to be possible, we calculate fitness value for each one.

In this situation, the fitness value of a chromosome represents the distance between clusters. The bigger the distance is, the better the chromosome is.

2.2.2 Application of selection operators

The selection in genetic algorithm, is the phase when chromosomes are selected to be part of the next procedures.

2.2.3 Application of genetic operators

Genetic operators are crossover and mutation. Those two operators can change the structure of chromosomes.

3 Results and discussions

After performing K-means algorithm and our variant of Genetic Algorithm on the given data set of soils, we have been able to obtain the best solution possible for the chemicals clustering and that means the best form of grouped soils that we need.

3.1 K-means algorithm applied on initial data

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ID	Cd	Cu	Ni	Pb	Zn	Co	Cr	Ba	Fe	K	Mg	Mn
2	1	0.5	0.307692	1	0.8125	0.8125	1	0.921053	0.918919	0.46	0.305942	0.759558	0.208696
3	2	0.5	0.076923	1	0.96875	0.9375	1	0.947368	0.864865	0.425862	0.3083	0.684367	0.243478
4	3	0.5	0.192308	0.979167	0.96875	0.984375	1	0.947368	0.918919	0.566207	0.390356	0.774002	0.286957
5	4	0.5	0.230769	0.958333	1	1	1	0.947368	0.891892	0.597586	0.375619	0.816907	0.304348
6	5	0.5	0.192308	0.958333	0.96875	0.953125	1	0.421053	0.945946	0.643793	0.358406	0.892948	0.26087
7	6	0.5	0.153846	0.9375	0.9375	0.78125	1	0.684211	0.810811	0.724138	0.376326	0.651657	0.443478
8	7	0.5	0.153846	0.958333	0.953125	0.8125	0.973684	0.631579	0.675676	0.714138	0.422896	0.915888	0.478261
9	8	0.5	0.192308	0.979167	0.953125	0.890625	1	0.894737	0.891892	0.624828	0.406626	0.835599	0.356522
10	9	0.5	0.230769	0.958333	1	1	1	0.657895	0.837838	0.710345	0.411106	0.826253	0.382609
11	10	0.5	0.269231	0.979167	0.984375	1	1	0.973684	0.891892	0.582069	0.369135	0.93373	0.269565
12	11	0.5	0.269231	0.979167	1	1	1	0.921053	0.972973	0.62	0.433978	0.924809	0.252174
13	12	0.5	0.230769	0.979167	0.984375	0.953125	1	0.973684	0.891892	0.595517	0.431502	0.834325	0.252174
14	13	0.5	0.230769	0.916667	1	0.734375	0.973684	0.736842	0.864865	0.788966	0.435982	0.778675	0.426087
15	14	0.5	0.230769	0.895833	1	0.875	0.973684	0.736842	0.810811	0.943103	0.570856	0.976211	0.478261
16	15	0.5	0.153846	1	0.984375	0.75	1	0.894737	0.972973	0.498966	0.3083	0.787171	0.226087
17	16	0.5	0.230769	0.916667	1	0.921875	0.973684	0.789474	0.972973	0.871379	0.396487	0.990654	0.234783
18	17	0.5	0.230769	0.958333	0.984375	0.953125	1	0.868421	0.945946	0.641034	0.417237	0.87808	0.330435
19	18	0.5	0.230769	0.9375	1	0.875	0.973684	0.763158	0.891892	0.775517	1	0.933305	0.452174
20	19	0.5	0.153846	0.8125	1	0.359375	0.973684	0.342105	0.810811	1	0.560717	1	0.521739
21	20	1	0.230769	0.958333	0.984375	0.890625	1	0.763158	0.864865	0.672759	0.312426	0.805862	0.313043
22	21	0.5	0.192308	0.8125	0.984375	0.84375	0.973684	0.605263	0.810811	0.638276	0.629686	0.759983	0.652174
23	22	0.5	0.153846	0.9375	1	0.78125	0.973684	0.736842	0.864865	0.837241	0.425136	0.720901	0.452174
24	23	0.5	1	0.770833	0.984375	0.0625	0.973684	0.657895	0.594595	0.766207	0.428319	0.820731	0.913043
25	24	0.5	0.115385	0.9375	0.984375	0.96875	0.973684	0.894737	0.891892	0.832069	0.448125	0.774427	0.295652
26	25	0.5	0.192308	0.9375	1	0.828125	0.973684	0.842105	0.864865	0.704483	0.378448	0.712829	0.330435
27	26	0.5	0.230769	0.9375	1	0.84375	0.973684	0.710526	0.72973	0.925862	0.703726	0.759983	0.391304
28	27	0.5	0.153846	0.916667	0.984375	0.765625	0.973684	0.789474	0.675676	0.796552	0.400731	0.924384	0.321739
29	28	1	0.192308	0.895833	1	0.75	0.973684	0.684211	0.72973	0.822414	0.466871	0.886576	0.4
30	29	0.5	0.269231	1	0.984375	0.875	1	1	1	0.481379	0.269866	0.644435	0.147826
31	30	0.5	0.153846	0.958333	0.9375	0.921875	1	0.921053	0.918919	0.649655	0.446593	0.844945	0.269565
32	31	0.5	0.153846	-2	0.9375	0.828125	0.763158	0.289474	0.594595	0.838621	0.647607	0.816058	1
33	32	0.5	0.269231	0.958333	0.953125	0.953125	0.973684	0.894737	0.837838	0.657586	0.40356	0.675021	0.295652
34	33	0.5	0.269231	0.895833	1	0.875	0.973684	0.789474	0.837838	0.925862	0.75784	0.735769	0.478261
35	34	0.5	0.230769	0.9375	0.96875	0.75	0.973684	0.552632	0.864865	0.744138	0.375737	0.868734	0.4
36	35	0.5	0.269231	0.9375	0.96875	0.9375	0.973684	0.526316	0.918919	0.782069	0.496345	0.920561	0.426087

Fig. 1: Initial data set

3.1.1 Hubert test

Firstly, we want to establish the right number of clusters, using 2 different tests in R software [4], applied on the data from the previous picture (figure 1).

As can be seen in the next two pictures, this method determines a the number of 3 clusters that can be made, due to the values in the normalized data set. (figure 2, figure 3)

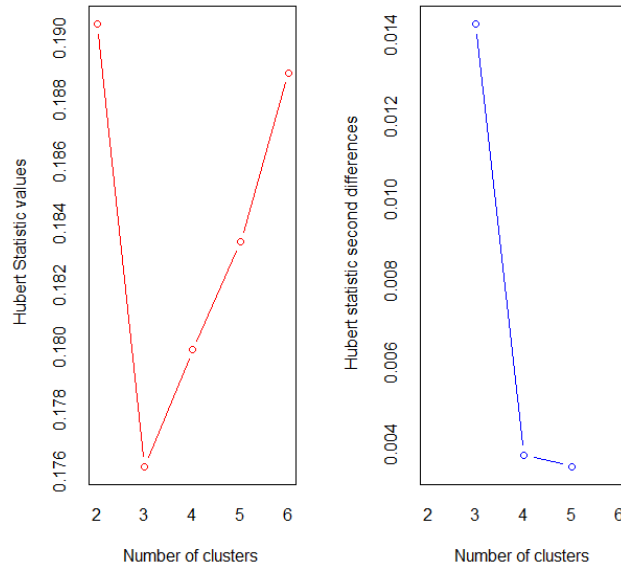


Fig. 2: Results after applying Hubert test on the initial data set

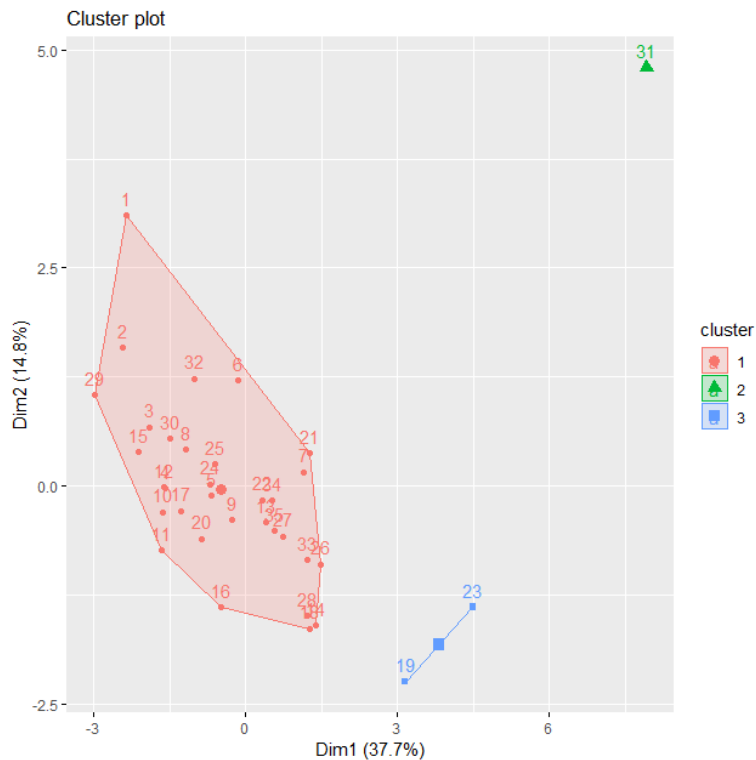


Fig. 3: Clustered data from the initial data set after applying Hubert test

By applying this method to the standard data set we obtained a value for the number of clusters equal to 2.

3.1.2 Silhouette

By applying this method to the standard data set we obtained a value for the number of clusters equal to 2. (figure 4)

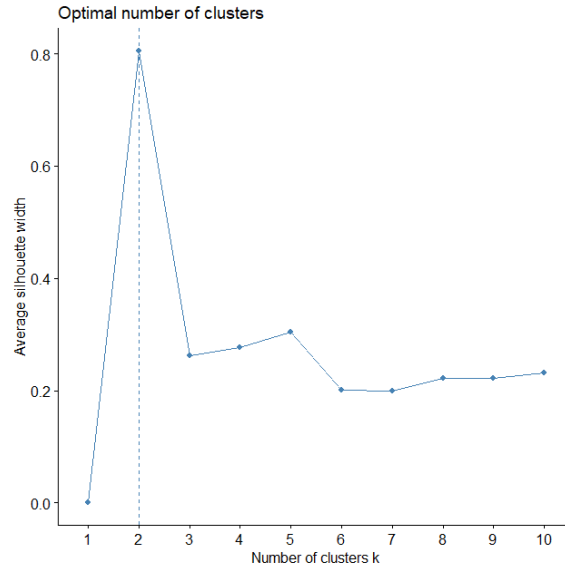


Fig. 4: Optimal number of clusters resulted after applying Silhouette method on the initial data set

Next, we will continue with the clusters number equal to 2.

3.2 Genetic algorithm for data classification [1,2]

After the fitness value calculation and application of selection operators, made at the very beginning of the algorithm, we tried different types of crossover as: crossover with one point, k-point crossover, discrete crossover and reduced surrogate crossover. Then we made 240 tests to find out which one of the crossover types gives us the best solution.

It has been shown that Reduced Surrogate Crossover method, with 0.60 crossover rate gives the best solutions, because this combination of fitness calculation type, selection type and crossover type led us to the biggest fitness value.

We were searching for the biggest fitness value, because the chromosome with the biggest fitness value has the biggest distances between the clusters and that means the best clustering/ differentiation of the groups.

In the end, we obtained a chromosome using the combination of parameters presented above. (figure 5)

```

Fitness Value:

resultFitness =

    0.2176

Best Chromosome:

resultChromosome =

Columns 1 through 12

    1     2     2     2     2     2     1     2     1     2     2     2

Columns 13 through 24

    2     2     2     2     2     1     2     1     2     2     1     1

Columns 25 through 35

    2     2     2     2     1     2     2     2     2     2     2

```

Fig. 5 Output data. Final result obtained after applying the algorithm; best chromosome and it's value

4 Program

The program has been realized using the Matlab [3] and R software.

In order to realize our study, below are shown the steps that have been followed:

- Taking the initial data set, represented by the chemicals from the soil, and normalize it;
- Applying K-means algorithm on the normalize data in order to obtain 2 clusters;
- Considering the K-means result, we wanted to step further and find other best solutions with genetic algorithm;
- Applying genetic algorithm;
- Generating random population and calculate the fitness value for every chromosome using the normalized data set;
- Applying select operator on certain chromosome;
- Applying crossover and mutation on population;
- Choosing from different types of crossover the one that gives the best final result;
- Testing for getting the best chromosome.

5 Conclusions

The present study proved that choosing different values for the rate of crossover can lead to different results, better or not. Better results have been obtained for the reduced surrogate crossover method, combined with 0.60 crossover rate.

Acknowledgements: This work was supervised by Dr. *Alina Bărbulescu* and Dr. *Cristina Serban*, from *Ovidius University of Constanta*.

References

- [1] N. Sujatha, Refinement of land cover classification of satellite images using GA based k-means clustering algorithm. *International Journal of Multidisciplinary Research and Development*, 2(4): 406–409, 2015.
- [2] Lect. Univ. Dr. C. Serban. *Calcul evolutiv. Algoritmi genetici*.
- [3] Introduction to matlab statistics assignment. <https://statisticsassignmentshelp.com/online-matlab-homework-help/>, 2020.
- [4] R software. https://www.siue.edu/~jpailde/Intro_to_R.html, 2020.
- [5] Soil health card boosts farm income up to rs 30,000/acre. <https://kashmirobservers.net/2020/02/18/soil-health-card-boosts-farm-income-up-to-rs-30000-acre/>, 2020.

Stela CARAMIHAI
Ovidius University of Constanta
Faculty of Mathematics and Computer Science
124, Mamaia Blvd. Constanta
ROMANIA
E-mail: caramihaistela@gmail.com

Marina – Larisa INDRECAN
Ovidius University of Constanta
Faculty of Mathematics and Computer Science
124, Mamaia Blvd. Constanta
ROMANIA
E-mail: maryna_larysa@yahoo.com

Proximity marketing, personalized offers and banking, all on mobile

Arina Ioana Cazacu

Abstract

A quick research about 2020's top 10 trending ideas in applications development includes the three key concepts in this thesis: Machine Learning, Internet Of Things and Beacon Technology [1]. The fundamental idea that initiates the view of this paper is the well known mobile app, Revolut [2]. The banking concept quickly progressed into a self scan software, in which a client enters a store, scans the products on his own, and checks out, all of these using the mobile application. But then the whole painting needs its colours: proximity marketing and personalized offers came up as ideas. The proximity marketing strategy is implemented using a Bluetooth Low Energy device, named Beacon. The recommendations regarding the offers are extracted from users' transactions using Association Rules Mining. That's how a user passing by a shop that has a beacon placed at the entry gets the privilege of being notified with the offers extracted from his own preferences. A nice approach here seemed to be getting the user to believe he is in control. Therefore, let him do the scanning on his own and greet his habits with discounts while he's taking a walk outside - this is the whole philosophy behind the original manner in which this paper manages to combine beacon detection and proximity marketing with recommendation systems. Overall, this project emphasizes the importance of psychological factors in a software experience. It consolidates user engagement through real time personalized notifications and through granting full control to its clients.

1 Introduction

It is speculated that in the future, jobs that can be fully automated will be lost, given that robots are much more reliable and financially advantageous alternatives than human employees. The intention of this paper is to propose a software product - a mobile application - that builds a bridge to this future that is getting closer and closer through the contributions of all programmers with a lateral dose of thinking.

The key words around which the whole article revolves are *self-checkout*, *personalized* and *beacon*. The initial prototype of the application that is to be described in this article, was a mobile banking application, similar to Revolut [2], meeting all the classical functionalities: top up, transfer, paying bills, checking the balance, friends, and more. As the research continued, the application structure ramified given several reasons and ideas, as follows.

Cash registers without employees became one of this decade's attractions. A 2014 NCR global study, conducted by the NPD Group, 90% of the customers identify themselves as users of *self-*

checkout [3]. The application proposed by this article adopts this strategy - the products from a store can be scanned using the phone when they are taken off the shelf before adding them to the shopping cart. The total price is calculated in no time and the payment is one click away.

The next step in the development phase was *personalizing* the user experience, as the second key word reveals. Data mining is the approach used to discover intriguing associations between customers' shopping transactions. These associations, if used properly, can bring the user experience to another level by an increasing degree of empathy. This paper uses the extracted associations to give recommendations by notifying every user with personalized offers, thus bringing benefits to both the customer and the store.

The third key word, *Beacon*, intervenes for connecting the dots between the smartphone and the shopping location (store). This small, very affordable device is at the heart of an entire industry called proximity marketing. Being strategically placed at the store's entrance, a beacon broadcasts Bluetooth signals that are detected by the application. A notification pops up on the users' screen letting him know that he is in the proximity of a store that provides him personalized offers.

How does the algorithm responsible for personalizing the user experience work and how is a Beacon detection system implemented are the basic concepts of this paper and they are presented in Section 2.

How these theoretical notions are combined, as well as the difficulties encountered in actually building a final software product, are described in Section 3. The first subsection outlines the concept of the entire application, by highlighting some of the use cases. Along the way, the paper gradually relates to technical details, describing the technologies approached for building the interface, the server and the database. The following subsections describe the coding process and the two types of users for which the application was designed: client and administrator.

The motivation behind these ideas is emphasizing what pleasing both sides of a business means - both the client and the manager -, their common pawn being money. On one hand, it increases store sales by notifying passers-by with personalized offers, thus attracting them to enter the store. On the other hand, by personalizing the experience, customers can buy their favourite products at a lower price.

On the whole, the purpose of this software is to greet customers' shopping habits with discounts. The uniqueness of the approach lies in the way beacon detection is transposed in a complex marketing strategy. The fact that the emphasis is on offering a more personalized experience, as much as the amount of control gained by the regular user, has a great psychological impact.

2 Theoretical Ground

2.1 Data Mining

2.1.1 Association Rules Mining

Association rules are a well-known technique in data mining, used to identify the relationships between records of a very large data set [4]. These relationships are a result based on probabilities and "if-then" statements. The algorithm is very suitable especially in the field of sales, using as input data the transactions of a customer. Predictions of a customer's behaviour can be obtained by analysing his shopping cart. Therefore, further notions will be explained by maintaining the marketing environment.



Fig. 1: Association Rule

As seen in Fig. 1, an association rule consists of two elements: an antecedent and a consequent. Each of them represents a set of products. The difference is that the antecedent can be randomly selected from the data set, whereas the consequent is the one that is in the same place as the antecedent in transactions. Therefore, the implication in this context refers to co-occurrence and not to causality. The reunion between antecedent and consequent forms an itemset - a set of products. All rules will be generated from itemsets, which are obtained by extracting all subsets of at least two elements from the set of products. However, if all possible rules were extracted from all identifiable subsets, the execution of the algorithm would be excessively long and only a few of the rules obtained would be truly relevant. For this reason, different metrics intervene here that help to choose the right items and rules: support, confidence and lift.

2.1.2 Metrics

Support:

How often do certain products appear in customer's transactions?

$$\text{Support}(\{X\} \rightarrow \{Y\}) = \frac{\text{transactions containing both } X \text{ and } Y}{\text{total number of transactions}}$$

Fig. 2: Support [4]

The formula in Fig. 2, gives the frequency of the itemset $\{X, Y\}$ in all registered transactions. If the value is too low, there is not enough information about the association and no relevant conclusions can be drawn from the rule.

Confidence:

How many times has the association been found to be true?

$$\text{Confidence}(\{X\} \rightarrow \{Y\}) = \frac{\text{transactions containing both } X \text{ and } Y}{\text{transactions containing } X}$$

Fig. 3: Confidence [4]

Confidence defines the probability of a consequent to appear in the shopping cart that already contains the antecedent, as seen in Fig. 3. The maximum value of this parameter is 1, and any rule that is as close to 1 as possible is called a high confidence rule.

Lift:

Does confidence live up to expectations?

$$\text{Lift}(\{X\} \rightarrow \{Y\}) = \frac{(\text{transactions containing both } X \text{ and } Y) / (\text{transactions containing } X)}{\text{Support}(Y)}$$

Fig. 4: Lift [4]

Lift calculates the frequency of the consequent, taking into account the conditional probability of occurrence of $\{Y\}$ having $\{X\}$, as mathematically described in Fig. 4. Even if the rule is a high confidence one, if the value of the lift will be less than 1, it means that the rule is not good enough. On the other hand, in cases where a $\{X\}$ leads to having a certain $\{Y\}$ in the shopping cart, the value of the lift is greater than 1 - thus a strong association is identified.

2.1.3 Apriori Algorithm

The Apriori algorithm is one of the most well-known techniques for extracting associations from massive data sets. It can be described in a few steps, as follows [5]:

- 1. Generate all frequent itemsets from an initial set.** At this step all partitions of one element from the initial set are generated. The *support* for each generated itemset is to be calculated, eliminating those that do not respect a minimum value.
- 2. Generate associations from frequent itemsets.** A rule is formed by binary dividing each itemset, consisting of at least two elements, into two nonzero subsets - so an antecedent and a consequent will result. This is where *confidence* comes in as a measure. From the set of candidate rules thus obtained, only those that have a higher *confidence* than the minimum value initially set will be chosen (analogous to step 1).

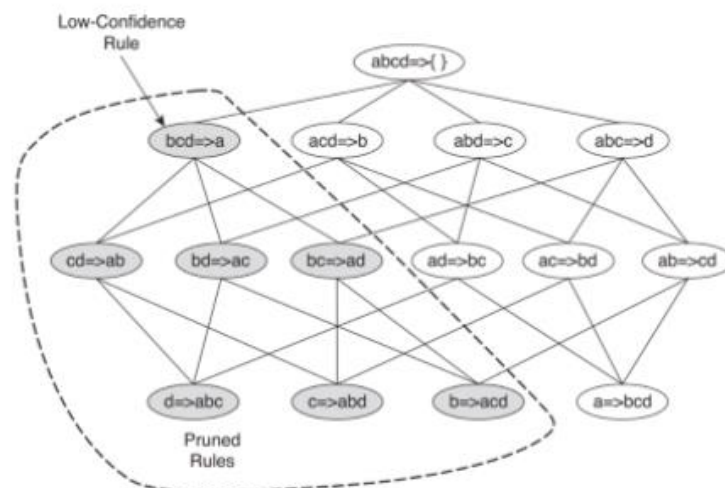


Fig. 5: Rules mining tree [5]

This whole algorithm consists in removing, from the beginning, the unnecessary tree nodes for a more efficient search of frequent itemsets, as Fig. 5 shows.

2.2 Proximity marketing

2.2.1 Overview

Proximity marketing is one of the most powerful business strategies today. Its essence is communication based on the real time geographical position of customers. Depending on the location, their devices receive notifications containing offers or advertising content [6].

Customer location is possible through Internet of Things technologies, such as: Wi-Fi, BLE - Bluetooth Low Energy, GPS, radio frequencies, NFC - Near Field Communication, RFID - Radio Frequency Identification. It is assumed that in order to be identified, the customer owns a smart device.

Notifying the clients at the right moment brings empathy between the business owners and their customers, thus reaching a higher psychological level in their contact with the store. There are many benefits to this approach for both parties. Merchants have advantages because they are aware of customer demands and needs, thus offering customized offers and discounts, and this attracts buyers to purchase their products. Customers are informed in real time, and the fact that the behaviour of previous transactions is taken into account, provides a personalized experience, facilitating the buying process by the power of suggestion. This is a "win-win" pattern.

2.2.2 What is a Beacon?

The Beacon (Fig. 6) is a *small Bluetooth Low Energy device that broadcasts with limited amounts of data through radio frequencies* [7].



Fig. 6: Beacon

This data is called "advertising packages" and is transmitted periodically at certain intervals of time - up to 10 signals per second. Thus, via IoT, signals are received by all devices that have Bluetooth enabled and are nearby (100 m maximum range). The more BLEs placed in a given area, the more accurate the position of a smart object becomes.

Figure 7 describes the way a Beacon interacts with a customer. The store advertises its promotional event with a Beacon placed on the sales aisle that will notify the passers-by accordingly. The process takes place as follows:

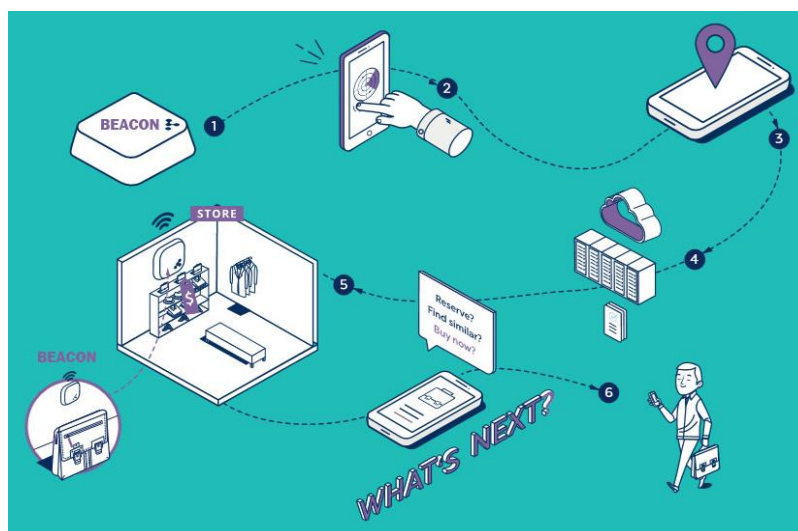


Fig. 7: Beacon - customer interaction [8]

1. The device continuously transmits radio signals.
2. The signal is detected by nearby smart objects via Bluetooth.
3. This signal informs the device that a Beacon is there by sending it its ID.
4. The smart device sends the ID to the cloud or to the application server, where it is verified which action is associated with it.

5. The notification corresponding to the store appears on the customer's screen (eg: "Book? Similar items? Buy now?"), and the answer remains at his discretion.

6. In the end, the customer is much more connected to what is happening in the store and up to date with the important offers he can take advantage of.

2.2.3 Properties and setup

These devices are very affordable and are quite easy to integrate into the project. A single Beacon costs an average of \$ 25, while with the purchase of 1000 Beacons, the price can drop to \$ 15-\$ 17 per device. Most Beacons have a 2 years battery lifetime, the power consumption is very low, but it depends on the manufacturer and the programmer's setup. More relevant details can be found in the "Beacon Setup Guide" published by kontakt.io [7].

There are 2 main BLE technologies: iBeacon and Eddystone. Both of them are Android and iOS compatible, but iBeacon is native to iOS only. Eddystone is designed by Google. In the following, the iBeacon will be described, as it is the one used in this application.

The identifiers of an iBeacon are called UUID (Universally Unique Identifier), Major and Minor (Fig. 8).

UUID	Major	Minor
f7826da6-4fa2-4e98-8024-bc5b71e0893e	100	1

Fig. 8 iBeacon identifiers [7]

The purpose of the UUID is to delimit a network of beacons - this avoids confusing those in the personal application with others active in the same area. Major and Minor are unsigned integers that take values between 1 and 65535. Major values are intended to identify a class or group of beacons. Minor values are the hallmark for all devices with the same Major.

There are two main Beacon configurations: the transmission power (Tx Power) and the Interval. They influence the way the Beacons network operates by affecting:

- signal radius (transmission power setup),
- signal stability (both settings, but especially the Interval),
- battery life (especially the Interval).

Tx Power

The transmission power determines how strong the Beacon signal is and is measured in dBm (decibel-milliwatts). It is scaled from 0 to 7, and the default is 3. It is true that the choice of a value as close as possible to 7 will make the transmission power reach maximum levels, and at the same time will ensure a longer range, but there is also a proportional energy consumption. Value assignment is possible from any beacon administration application.

Interval

The Interval determines how often the signal will be emitted and is measured in milliseconds - the default value is 350 ms. Intuitively, the lower the value, the shorter the battery life. Setting the Interval to more than 700 ms can cause major signal stability problems, while the value of 100 ms can lead to an unrecommended overload. For these reasons, the recommended setup is the default one, for both of the parameters.

2.2.4 Applicability

News about the use of Beacons continues to appear in every domain - businesses such as restaurants, banks, hotels, gyms, parks, museums, and more. Understanding this new marketing tool and getting informed about how it is used in successful campaigns around the globe is a great

advantage for managers. More projects are described in detail in the material created by *beaconstac*: “*Beacon Success Stories*” [9].

An example that differs in an original manner from the other Beacon projects is the *GeoTrail GO* application created for a *mall* in Denmark. The goal is to keep the kids entertained while their parents are shopping. The game consists of a treasure hunt based on virtual maps that lead to the accumulation of points. This application has provided a boost in the income of those who run businesses there.

The Department of Cardiology at Leiden University Medical Centre in the Netherlands (LUMC) has a platform for *treating patients with acute myocardial infarction* as quickly as possible, greatly improving their chances of survival. When the patient is brought by the ambulance to the emergency department, before leaving the vehicle, a bracelet with a BLE is attached to his hand. Throughout his route, there will be tablets for monitoring the bracelet, to indicate, in real time, how fast he is transferred from one location to another.

The *Adelaide Zoo*, as well as many other institutions (especially museums), use beacons to enhance the visitor experience by providing information about animals, exhibits and how to care for them. At each point of interest, the user is notified with practical and interesting data about species, nearby facilities and other educational tools.

One of the most famous music festivals in Romania, *Electric Castle*, highlighted the history of the location where it is organized, through a treasure hunt. In Banffy Castle, 40 beacons were placed, and users had to solve short riddles - through which they accumulated new knowledge about the castle - to further qualify or win prizes.

3 The proposed solution

3.1 Application concept

Overall, the app starts similar to the well-known banking application Revolut, with features for transfer, depositing money, friends, and more. A banking application is quite tender in terms of extension to other features. It can easily resemble a person transposed into the virtual environment, who has a sum of money that he can invest whenever and wherever he wants, just like in reality. The difference is that many applications have not been built from this perspective, the phrase “*anytime and anywhere*” being rather constrained to “*anytime and in this way*”. In order to eliminate these constraints, the second idea appears: *self pay*.

It is natural for a banking software with a self-pay functionality to involve a certain level of collaboration with at least one company. The store's database, with the prices and barcodes of the products, is needed, and the obligation of the application is to ensure confidentiality. Access to this information is not easy to obtain, because it demands confidence, and the risks will not be assumed by the managers if the benefits are not commensurate. This is how the third idea and essence of the application appears: *proximity marketing with personalized offers*.

Advertising methods are evolving, and the old marketing strategy - the delivery of promotional paper catalogues by mail - still persists, although the percentage of customers who are still interested in reading this kind of materials in 2020 is considerably small. Moreover, from an ecological point of view, they are an extremely harmful agent. In this way, the application proposes a new level of advertising: *notifications with customized advertising content triggered in the vicinity of the store*. The usefulness of the offers increases exponentially, and the degree of empathy when interacting with the user is one of the strongest psychological factors of a software product.

Basically, a customer registered in the application, passing by a partner store, will be notified on the spot with personalized offers. How is this possible without resource consumption or continuous location monitoring and other costly strategies? The answer is *Bluetooth*. Upon registration, the application delivers to the company a device called a beacon, which will be

placed at the entrance to the store, and which detects customers within a radius of up to 70 m via Bluetooth. This is one of the key concepts of the application, and is also the most interesting part in terms of implementation.

Therefore, after the application detects the beacon of a store there is the scenario in which the customer actually goes shopping there. He can scan all his products, directly from the application, before adding them to cart, and at the end he can choose one of the available offers. Thus, the discount is applied and the payment is also one app click away. The flow is shown in Fig. 9.



Fig. 9: Application flow

With all these possibilities, the application also provides a registration mode for the store administrator, who can see details about the battery life of the beacon and sales statistics. Moreover, he can change at any time the current list of promotional products that will reach the offers of customers.

3.2 Technologies

3.2.1 Front end

The technology used to implement the user friendly interface is React Native. The language used is TypeScript. The application is created in React Native CLI (Expo being the second possibility). The motivation for this choice comes from the fact that, being of open source type, some installed components required changes in their source code (from node modules) to fix bugs or various incompatibilities regarding the installation phase. In Expo, these areas are not accessible in this way.

3.2.2 Back end

The application server is designed with the ASP.NET Core MVC framework, written in C#. The MVC (Model-View-Controller) architectural template suggestively separates the application into three large groups of components. The Controller is responsible for receiving requests and replying with the corresponding answers, working with Models. The View is not present as part of the template, as everything related to viewing is exclusively built in React Native.

3.2.3 Databases

SQL Server with SSMS (SQL Server Management Studio) is used to store and organize data. As a structure, the SSMS project consists of two large hierarchies, one starting from the User entity (the customers) and the other from the Company entity (the store). Data about beacons are stored in the second hierarchy. Beacons owned by the application are previously registered in the Beacons table, and are marked with `taken = 1` when assigned to a company.

3.3 Types of users

Initially, the only type of user was the customer – a basic Revolut user. Subsequently, the idea of beacon detection and, implicitly, finding a way to manage the companies registered in this strategy came up. This is how the second category of users appeared: company administrators. This approach ensures that each company will have, firstly, an associated beacon that the application will know how to manage, and secondly, an account in which self-pay transactions will be recorded.

Regardless of the user, registration requires the deposit of a minimum amount, a protocol found in most banking applications. For companies, part of this amount is invested in purchasing a beacon - it is the application's responsibility to provide a properly configured beacon for each company recruited.

3.4 Beacon Detection

To implement beacon detection in React Native, it was necessary to install the *react-native-beacons-manager* library, designed by MacKentoch [10]. Other libraries have been found to be largely incompatible, deprecated or dysfunctional. Problems occurred in the library source code, therefore some changes for the code in the node modules were required. After installation and build, running the example available in the documentation caused the application to crash repeatedly. The first fix was to change the source code, followed by modifying the way the call is made (in *componentDidMount* method), as in Fig. 10. A listener is then added to the *DeviceEventEmitter* component for the "beaconsDidRange" event (note that this part is called at each mount, and the number of listeners must remain at 1). When the event occurs, a handle function is called that checks if the found beacon is registered in the application database and acts accordingly. Initially, the list of devices returned by the listener was empty - the reason was the omission of a very important step that was not mentioned in the documentation: adding permissions to the native android files. It is obvious that the detection of such devices involves Bluetooth activation and, implicitly, location services. The application needs permission to access these services. However, the following lines are added to *AndroidManifest.xml*:

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Next, in the detecting beacons method, code requesting for location permission and verification that Bluetooth is active is added - otherwise, the application starts it automatically, notifying the user.

```

Beacons.setBackgroundScanPeriod(1000 * 60);
Beacons.setBackgroundBetweenScanPeriod(2000 * 60);
Beacons.setForegroundScanPeriod(1000 * 10)
Beacons.detectIBeacons();

// Range beacons inside the region
Beacons
  .startRangingBeaconsInRegion('REGION1')
  .then(() => {
    console.log('Beacons ranging started successfully')
  })
  .catch(error => console.log(`Beacons ranging not started, error: ${error}`));

```

Fig. 10: Beacon detection code

After successfully detecting a beacon, there are four possible scenarios related to the application policy:

1. The beacon does not appear in the database.
2. It is in the database, but has not been detected until now.
3. It is in the database, but was detected the other day.
4. It is in the database, but was detected on the current day.

In cases 1 and 4, no behaviour is implemented. For cases 2 and 3, the above-mentioned handle function is responsible for triggering the notification. Also in these cases, the moment when the beacon was detected is recorded in the database. Thus, when new signals appear from the same device, a check will be made that the last detection will be recorded on a different day, in order to prevent the assault with notifications of the user.

3.5 Offers Extraction

This procedure is triggered or not when a beacon is detected, because the notification received offers the customer two alternatives: "not now" and "see offers". By clicking on "see offers", the application will open on a page for offers on which there will initially be a progress circle, while the server applies the Apriori algorithm to the user's transactions. Previous transactions for each client are stored in a csv.

First, it deals with the possible scenario of a "cold start": the minimum number of transactions that the user must have at a store to apply the algorithm of association rules has been set to 10. If a customer has less than 10 transactions, it will receive as an offer three pre-set promotional products for all customers (if the administrator has not yet set them from the application, three randomly selected products will be offered). Basically, the role of these promotional products is to take the place of general recommendations. This scenario can be seen in Fig. 11, where each offer contains a single product (from the promotional list).

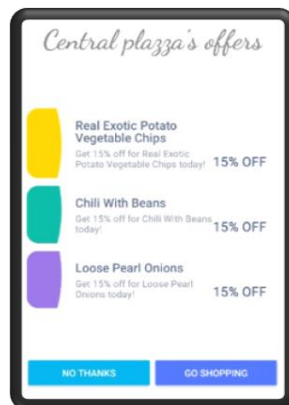


Fig. 11: Offers page

If the number of transactions exceeds the minimum threshold, Association Rules Mining applies for the extraction of three personalized offers. The algorithm is implemented by the server, in the DataController, DataService and DataRepository components, following the steps described in Section 2. In the DataRepository the data is extracted in memory from several csvs, and the minimum values set for the *support* and *confidence* parameters are 0.25 , respectively 0.5. The results obtained are sorted in a descending order by the third parameter, *lift*, and the first three strongest combinations consisting of a *consequent* and an *antecedent* (sets containing at least one product) are sent to the interface and displayed in the format: "Buy {*antecedent*} and get% off {*consequent*}". If, however, not enough rules have been found, more products from the promotional list are added, until the number of offers reaches 3.

When calculating the recall for a quality check, it is concluded that the designed algorithm extracts offers that have a relevance of over 97% for the customer, which indicates a flawless functioning.

3.6 Self Scan

This feature becomes available after viewing the offers and following the series of events and choices described above. It is assumed that if the user is interested in offers, he is in the vicinity of the store and there is a great possibility that he enters the store for shopping. Otherwise, he can press the "no, thanks" button.

The "go shopping" button navigates to a component that has a built-in barcode reader, as in Fig. 12. The customer can scan their own products in the store before adding them to cart. When scanning a product, a call is made to the server to map between the barcode and the product. After scanning the code, a dialog window with the details of the identified product appears on the screen, and the user can choose from there whether to add it to the cart or not. The total amount is displayed at the bottom of the page and is automatically recalculated. By clicking on one of the products added to the list, you can view the details or remove products from the list.

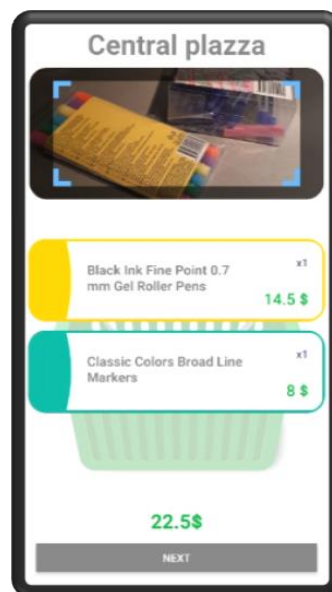


Fig. 12: Self Scan page

The "next" button leads to the completion of purchases (it is disabled if the current cart is empty). Next, the page with previously notified offers is displayed and the user has the possibility to choose one of the available ones. An offer is available if all the products in it are on the shopping cart, and it will be colored green, and otherwise gray. If no offer is compatible then a

suitable message is displayed. The customer is given the opportunity to navigate back to the Self Scan section to complete their list. When you select a compatible offer, the price displayed is updated.

When the "pay" button is clicked, several processes take place: checking if the user has that amount of money in the account, transferring it to the store administrator's account, registering the transaction in the database and updating the customer's savings. If this process has been completed successfully, the app navigates to homepage.

3.7 Beacon Configuration

It is the duty of the application administrators to configure the beacon, taking into account the information mentioned in Section 3, so that the only task of the store administrator remains its placement. There are several Android applications that are specially designed for this purpose, when browsing the app store. The one used in this project is called "*iBKS Config Tool*" [11].

At start-up, this requires permission to turn on Bluetooth to detect nearby devices. The scan starts and a list of devices is displayed. The device sought and used in the experimental evaluation of this application is an iBeacon from Accent Systems (Fig. 6). Clicking on iBKS105 (the detected beacon) will open a page with details about the device and tabs for different settings. There you can see that the battery life is estimated at 17 months, due to the configurations made. To change them, enter the iBeacon Service tab. In Fig. 13 the inputs from each parameter presented in Section 3 can be observed:

- The signal interval is at 950 ms, close to the maximum value, as it is aimed at conserving the battery in the implementation stage.
- TX Power is at -20, close to the lower limit, for similar reasons. The ideal value in this case, to cover a broadcast radius of up to 70 m, would be 4. In this case, the battery life decreases.
- The Minor and Major parameters did not require setup, as in the current application there is only one registered beacon, and there is no problem of differentiation or division into classes.

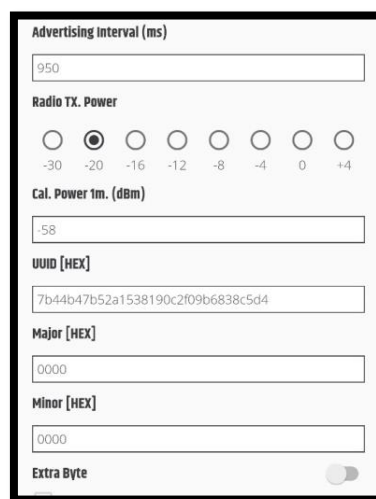


Fig. 13: Beacon Configuration [11]

3.8 Resources required

Providing the beacon is part of the benefits the application has to offer. It needs to be properly configured and registered in the database before being placed in the store. Thus, when creating a local store account, the specified address will be considered the destination of the beacon parcel,

and the mandatory registration amount will be invested in everything related to purchase, configuration and transport.

Assuming it is used in reality, the application requires the database of each registered store, containing the products mapped to barcodes and prices. For reasons of competition and security, access to such resources is difficult to obtain. Therefore, there are two approaches:

1. Designing a confidentiality agreement between the application and the company, which ensures that the data is secure. When creating an account, the administrator is notified of the terms and conditions of the contract, and is asked to confirm. Subsequently, access to resources will be established by contacting the administrator by email or by telephone.

2. Concluding a partnership with a willing company and developing the application exclusively for it.

In order to obtain conclusive results after applying the Association Rules Mining technique, a fairly large volume of data is needed. Generating the data may cause storing shopping transactions that lack intuitive logic because of the randomization. The data used for this software can be found on *kaggle.com* [12]. It was published in 2017, and the data was gathered using the Instacart application - designed for home delivery of online purchases, containing more than three million orders and over 200k customers.

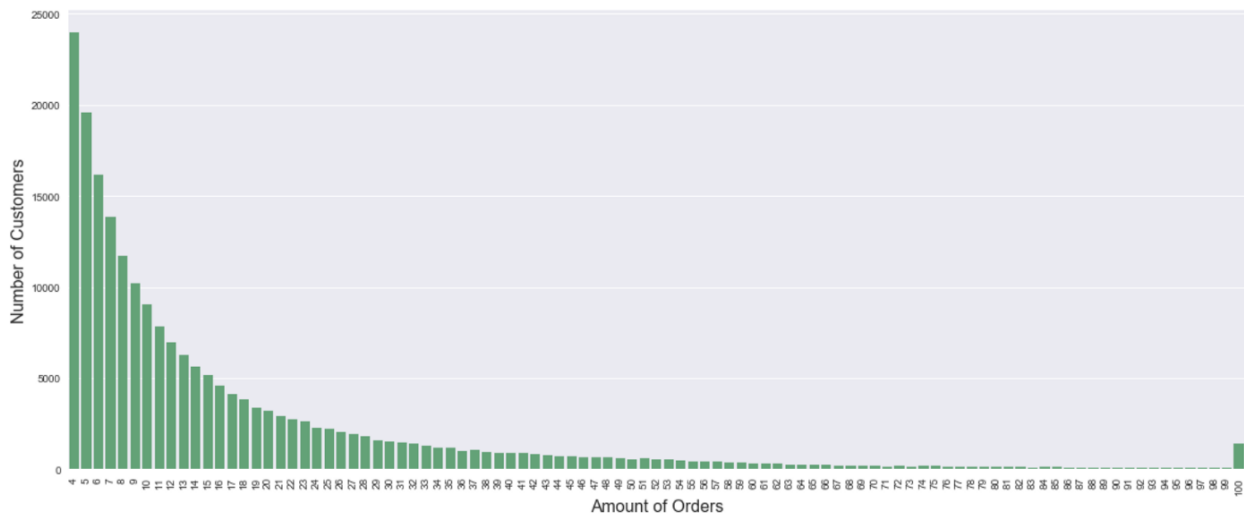


Fig. 14: Amount of orders distribution [12]

Each customer is registered with a number of orders between 4 and 100, and the distribution of the number of customers regarding the amount of orders is shown in Fig. 14. The minimum number of transactions that a customer must have in order for the rules to be extracted is set to 10 because the graph shows that 10 is a value of average frequency among users. If this value is not met for a user, they will be recommended three products from a promotional list that can be edited by the store administrator, but which is initially randomly initialized by the server.

For the experimental evaluation, the *recall* was computed (Fig. 15). This parameter determines the percentage of relevant results classified as correct by the algorithm.

$$\text{recall} = \frac{|\{\text{relevant results}\} \cap \{\text{actual output}\}|}{|\{\text{relevant results}\}|}$$

Fig.15: Recall [13]

For the current problem, 50 different users – with minimum 10 transactions registered - are randomly chosen for computing this parameter. For each user, the set of relevant results is the top 20 most frequent products in his transactions. Then, his offers are extracted (the actual output). The intersection from the numerator can be reformulated as follows: what percentage of products from the extracted offers are also listed in the top purchased by the customer? Thus, a list of all the results for each user is conceived, and at the end, the average value obtained for recall is: 0.971428. Therefore, it is concluded that the designed algorithm extracts offers that are more than 97% relevant for the customer, which indicates that the proposed approach is indeed suitable for the current problem.

4 Conclusion

Most of the time, the recommendations are personal, in the sense that they are extracted based on the user's preferences and thus represent a more one-way benefit. A slightly more lateral approach, however, can give it a two-way meaning by revealing a strong marketing strategy. Thus, it is not about notifying daily offers from a certain store, but rather about the real-time location based notification containing personalized offers for each user, based on previous personal purchases from the nearby store. Moreover, iBeacon technology emphasizes the moment when a potential customer is passing through the approximate area of the store, and the psychological impact of the promotions notified at the right time is huge.

The most profitable future improvement would be, by far, improving the context of use in which several administrators from different stores register in the application. A friendly registration process, which does not involve legislative details (such as confirmation of administration rights) and hard-to-obtain resources (store databases), but at the same time a validation, would make the application very accessible and used. The answer for all this would be the customers - they can confirm whether or not a person is a store administrator, they can enter the barcodes of the products in the application by themselves, without the need for the store database, etc. Basically, if the perspective were a little more oriented towards the real power of the users, their knowledge could be used in a very beneficial way for the application.

However, since humans are involved, trustfulness is an unpredictable issue to deal with. The stores might have to deal with theft attempts, therefore additional security measures should be implemented. Moreover, some users might find the whole self-checkout process difficult and might require assistance (faulty barcodes, age verification, etc).

The original view of this paper stays not only in the personalized promotions, but also in the fact that the app incorporates as well the banking features. The technologies used in this project are new, therefore the information gathering process was lacking examples and detailed documentations, and this was the most challenging phase of the project – how to encapsulate proximity marketing, banking and data mining, all at once, in one mobile application?

In Romania, Carrefour developed an application valid in all 28 supermarkets, offering customers information about products, services and special offers through beacons placed on each aisle. The difference is that the promotional products are not individually extracted for each customer, regarding his previous transactions, and the notification pops up on each aisle inside a store, but its purpose is not store detection. However, the number of users of the application increased by 600%, an impressive percentage for just seven months. Kaufland also had an application attempt similar to Carrefour's, which, in addition, includes products scanning, but the application is still non-functional [9].

Overall, the application greets customers' business habits with discounts while they follow the natural course of the day. The uniqueness of the approach lies in the way beacon detection is projected into a whole proximity marketing strategy. The fact that the emphasis is on offering a

more personalized experience, as well as the power of control that users have, has a great psychological impact on them.

Acknowledgement: This work was supervised by Professor *Florentin Bota*, from *Babes-Bolyai University, Cluj-Napoca*.

References

- [1] R. Mehul, “mindinventory.com,” [Online]. Available: <https://www.mindinventory.com/blog/mobile-app-development-trends-2020/>. [Accessed 2 June 2020].
- [2] “Revolut,” [Online]. Available: <https://www.revolut.com/en-RO>. [Accessed 20 January 2021].
- [3] NCR, [Online]. Available: https://www.ncr.co.jp/wp-content/uploads/files/solutions/self/fl/fl_wpa/RET_SCO_wp.pdf. [Accessed 10 February 2021].
- [4] A. Garg, “Association Rules,” [Online]. Available: <https://towardsdatascience.com/association-rules-2-aa9a77241654>. [Accessed 11 September 2020].
- [5] A. Garg, “A Complete Guide To Association Rules,” [Online]. Available: <https://towardsdatascience.com/complete-guide-to-association-rules-2-2-c92072b56c84>. [Accessed 11 September 2020].
- [6] H. B. N. Levesque, “Proximity Marketing as an Enabler of Mass Customization and Personalization in a Customer Service Experience,” in *8th World Conference on Mass Customization, Personalization and Co-Creation (MCPC)*, Montreal, 2015.
- [7] kontakt.io, “Beacon Setup Guide,” [Online]. Available: <https://kontakt.io/resources/beacon-setup-guide/>. [Accessed 20 June 2020].
- [8] Kontakt.io, “Proximity Notification Campaigns Reach More Customers,” [Online]. Available: https://www.xsights.com.au/wp-content/uploads/Proximity_Notification_Campaigns_Whitepaper.pdf. [Accessed 11 April 2020].
- [9] beaconstac, “Successful beacon campaigns across industries,” 2018. [Online]. Available: <https://www.beaconstac.com/ebook/successful-beacon-campaigns>. [Accessed 18 May 2020].
- [10] MacKentoch, “github.com,” [Online]. Available: <https://github.com/MacKentoch/react-native-beacons-manager>. [Accessed 20 June 2020].
- [11] Accent Systems, “iBKS Config Tool User Manual,” [Online]. Available: <https://accent-systems.com/support/knowledge/ibks-config-tool-user-manual/?v=f5b15f58caba>. [Accessed 13 June 2020].
- [12] kaggle.com, “Shopper Behaviour Analysis,” [Online]. Available: <https://towardsdatascience.com/shopper-behavior-analysis-de3ff6b696b8>. [Accessed 2 June 2020].
- [13] “wikipedia.org,” [Online]. Available: https://en.wikipedia.org/wiki/Precision_and_recall#Recall. [Accessed 2 Iunie 2020].

Arina Ioana CAZACU
Babes-Bolyai University
Computer Science
Mihail Kogălniceanu 1, Cluj-Napoca 400000
ROMANIA
E-mail: arina.cazacu@gmail.com

Educational Data Mining using Supervised Learning Techniques

Alexandru-Mihail Crăciun

Abstract

The main domain of this paper is Educational Data Mining (EDM). In this article we analyze a subdomain of the EDM, called Academic Performance Prediction. A good way to achieve a high level of quality of the educational system is to extract knowledge from the existing data, in order to be able to analyze the behaviors of the students, to see the most popular subjects, to detect the disadvantages of traditional learning methods or to predict the students' academic performance. It is clear that due to large volumes of data this "extractions" have to be made using some machine learning algorithms. This paper aims to make a detailed analysis of how predictions of the academic performances can be made using various machine learning techniques and especially using the Ensemble Learning based methods. This special type of learning has been present in numerous research studies and has proven to be highly effective in solving both classification and regression problems. An original aspect that we introduce consists in the idea of trying to predict students' academic results based on psychological information that we have about them. In order to do this we used a dataset received from the Leibniz Institute for Psychology Information. Moreover, the originality of the machine learning models that we made is ensured by the fact that they can be trained on any new data set, without first knowing its structure.

Keywords: data mining, education, machine learning, ensemble learning, random forest, academic performance prediction, supervised learning, decision tree

1 Introduction

Education has always been an important factor in the development of society, and its connection with other fields has always been a highly debated topic in the literature. The main objective of the educational institutions was to offer students an auspicious environment in which they can develop certain skills. For achieving a quality learning process, it is important to study in detail how students learn and assimilate the knowledge provided by the teachers and how they manage to apply it later. Thereby, methods of Computational Intelligence were used in the last years to realise studies regarding these topics and thus, the notion of Educational Data Mining was introduced in literature. Data Mining is the process of extracting new knowledge and identifying templates from large data sets, using methods that combine Machine Learning, statistics and database management systems. Due to the use of online educational platforms on a large scale, there is a much greater variety of data that can be processed to study the behavior of the students.

The present paper aims both to give an overview of Educational Data Mining, and to analyse supervised learning techniques that can predict academic performance, with a greater emphasis on models that use learning algorithms based on ensembles. A number of new approaches regarding not only the knowledge extraction methods used, but also the type of data used will be analysed.

It is known that the existence of a relevant dataset underlies any machine learning algorithm, thereby this was one of the first aspects we took into account. If most of the previous studies conducted in the prediction of academic performance used only data based on the students' previous performance and their behavior during the educational process, this paper aims to use psychological data, which reflects a high degree of originality.

Recent studies show that, in addition to IQ, another significant factor in the prediction of academic results is represented by the personality traits, with focus on the The Big Five (The big five personality traits): Neuroticism, Openness, Extraversion, Agreeableness and Conscientiousness. Thus, I intend to study the potential that the results of personality and intelligence tests applied to samples of students can have in achieving predictive models with satisfactory performance.

Considering that artificial intelligence has become an integral part of our lives, its use in the educational process was inevitable. More and more educational institutions, both in elementary and higher education, are being transformed by smart means, helping not only students to learn better and achieve their goals, but also the teachers to improve their style of teaching and develop themselves professionally and academically. When Artificial Intelligence manages to connect people based on what they want to learn and who they really want to become, a huge gain for society is achieved.

The first section of the article is a brief introduction regarding the way of how the artificial intelligence can be applied in the context of the educational field, followed by a section with a detailed presentation of the general Data Mining and of the Educational Data Mining domain. The next section is intended to present the current state of the art in research in the field of the prediction of the academic performances. The last part of the article describes my approach regarding this kind of predictions, the datasets that I used, the results obtained from the evaluation of the machine learning models as well as my proposed web application. At the end, I will provide the conclusions drawn from my study regarding the prediction of academic performance.

2 The use of the Artificial Intelligence in Education

One of the greatest challenges when it comes to education is the fact that people have different learning styles and skills, and a unique way of thinking. Students enter the educational system with a different baggage of knowledge and different skills. Therefore, the educational institutions and teachers must be able to easily adapt and meet the needs of each student.

While some students might have a more developed left brain and have many analytical thinking skills, others are more oriented towards creativity, art, literature and communication, being followers of 'right brain' thinking. In this situation, those who study and develop intelligent methods to use in education must elaborate easily adaptable systems that are able to bend to different personality types and ways of thinking.

In his article [9], Schmelzer argues that by 2024 more than 47% of learning management platforms are expected to be equipped with Artificial Intelligence features. As for teachers, they will be able to use augmented intelligence to create a variety of materials, designed to meet the needs of each student, instead of creating a single curriculum for all students.

Thus, the concept of Hyper-Personalization is introduced in the literature; a method that allows intelligent systems to adapt to each individual and to meet their needs. Using the power of machine learning, it will be possible to create a suitable profile for each student, based on skills, favourite styles of learning and previous experiences.

The personalized and on demand digital content is currently realised using the most modern techniques of computational intelligence, significantly changing the way things are conducted in education.

If in the past the sources of information in schools were limited to books or study guides, it is believed that in the near future the use of physical materials, on paper, will be gradually abandoned and the attention will be directed to the digital resources. Systems that use Artificial Intelligence offer students an attractive interface and take their feedback into account in real-time. Moreover, teachers can track the progress of each student and solve the problems they face more easily.

3 Introduction in Educational Data Mining

The increasing use of technology in the education system has led to the storage of very large amounts of student data, which makes it very important to use data mining to improve teaching, learning and evaluation processes. Manual data processing has become almost impossible and thus, in recent years the machine learning techniques are more and more used to obtain information from the collected and processed data.

Compared to other fields, the implementation of knowledge extraction techniques in the educational field has also a lot of additional requirements, because the pedagogical aspects of the student and of the system must be taken into account very carefully. Although Educational Data Mining is a relatively new field, there are already a significant number of contributions in the research area, which show its great potential.

The implementation of data mining techniques in the education systems for improving learning can be seen as a formative evaluation technique [8]. The formative evaluation is the evaluation of an educational program, while it is still being developed, in order to remedy possible errors. In other words, by examining the way students relate to the system, teachers can observe both the negative and the positive aspects and can gradually adapt their teaching and the content. [11]

The extraction of knowledge from academic data has become an intensely debated field in recent years, on the border between technology and psychology. The necessary data can be obtained from a very large number of students and can contain many psychological variables that data-mining algorithms can process to build machine learning models.

Various sources can provide data, such as:

- Traditional learning environments (face-to-face learning, in the classroom).
- Educational software
- Evaluating educational software
- Online courses
- Learning management platforms
- Test for students regarding the quality of the educational process
- The information provided by the teachers about the performance of their students

3.1 The stages of extracting knowledge from academic data

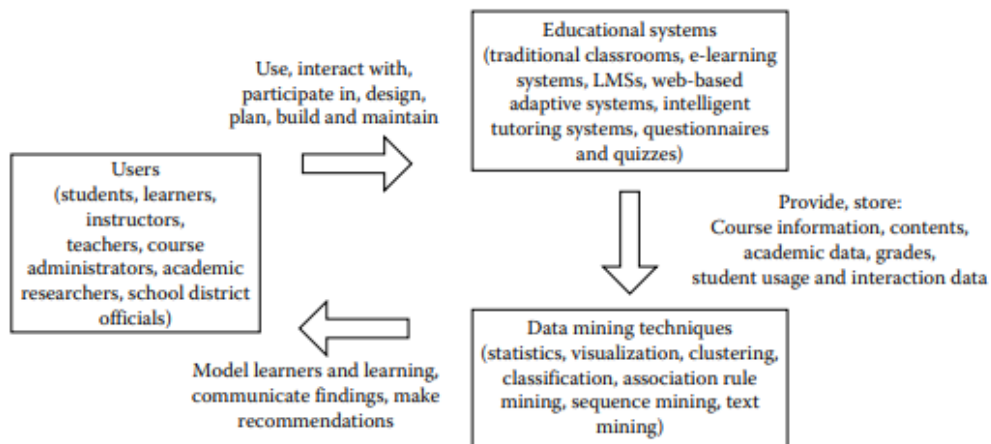


Fig. 1: Applying data mining to the design of the educational systems [8]

As the research in the field of Educational Data Mining has continued to develop, a multitude of data extraction techniques have been implemented in a wide variety of educational contexts. The main objective of each case was to translate the 'raw' data into relevant information about the learning mechanism, in order to find solutions for designing a quality educational environment [2]. Thus, extracting knowledge from academic data involves four main steps [6]:

- The first phase of the process (without taking preprocessing into account) is to discover relations in the data. It involves searching through an entire collection of educational data in order to find relevant relations between variables. Over the years, numerous algorithms have been used in the research of identifying such relations, including: regression, classification, sequential template extraction, factor analysis, social network analysis, clustering etc.
- Subsequently, the discovered relations must be validated to avoid overfitting. Statistically speaking, overfitting is, by its definition, the analysis that corresponds very closely or even exactly to a particular data set. Therefore, the model performed is not able to correctly predict future information, which would be totally different from that in the original data set.
- The already validated relations will be used to make predictions about future behaviours in the educational environment.
- The predictions made are ultimately used to support the decision processes. If the results obtained are those expected or are relevant to the experiment, the testing phase of the previously developed and trained model will be continued.

During the stages 3 and 4, the data and the results are visualized graphically most of the time, so that they can be evaluated by the human factor. The graphs mainly capture information about the performance of the model, based on different metrics that are already calculated (accuracy, recall, mean absolute error, mean squared error, etc.).

After a detailed analysis of the results, a retraining of the model will be considered, and the four stages will start over, taking into account a new configuration, which can be represented by the elimination of attributes, instances or the modification of the specific parameters of the model in question.

4 Related work – Students performance prediction

The students are the center of the educational process, and the studies have shown that a deep understanding of how they assimilate knowledge is required to provide a quality education for them. Thus, depending on their academic results, learning can be coordinated more easily, and teachers can better adapt to students' needs.

The prediction of students' academic performance has thus become an increasingly addressed topic in the literature and various models and learning techniques have been tried in order to extract the most relevant information.

A first issue that was debated was that of the types of data used in the prediction. If most studies took into account the average of the grades from the previous years and the grades of each subject, there were also studies that focused on students' activity in class or on their results at intelligence and personality tests.

After the analysis of the different data sets, the appropriate learning model and the corresponding type of learning, necessary to extract the information, had to be chosen. Thus, the following lines will present in comparison the research results in terms of supervised learning.

4.1 Approaches that use fundamental models of supervised learning

Although there have been studies on Educational Data Mining (EDM) since the 1990s (Romero and Ventura [7] covered most of the research projects in this field between 1995 and 2005), the prediction of academic results using supervised learning remains a highly debated topic nowadays, because we try to develop systems that are defined by the highest level of performance and accuracy possible.

One of the simplest proposals regarding supervised learning was to create a model for the students' performance, that would make a classification and divide them into 2 categories: passed and failed.

Even though this was an important step, the classification had problems with data distribution and the results were not as edifying as expected in order to contribute to a significant improvement of the educational process.

Subsequently, a much more complex classification was attempted, in which the students' grades could be organised in several possible categories. In the article [5], Dorina Kababchieva divided the results into 5 distinct categories: "excellent", "very good", "good", "average" and "bad". The study was based on the data of 10,330 students from a university in Bulgaria, each of them described by 20 parameters, which included information such as: date of birth, place of birth, place of residence, place of the last unit of study, the average grade from the last study cycle, the average grade of admission, the average grade of the current semester, the specialization, etc.

To perform the experiment, several supervised learning algorithms were considered, which had a high potential to lead to high-performance results. Classifiers from the WEKA library were used, as follows: an algorithm based on a decision tree (J48), two Bayes classifiers (Naive Bayes and BayesNet), a KNN algorithm and two rule learners (OneR and Jrip). Each classifier was applied for two different test options: cross validation (10 sets were used and the algorithm was applied 10 times. Each time 9 of the sets were used in the prediction and one set was used for testing) and division percentage (2/3 of the dataset was used for training and 1/3 for testing). After performing all the tests, it was concluded that the decision tree (J48) was the one that obtained the best results (with the highest total accuracy), followed closely by JRip and the KNN classifier. The Bayes classification was the least performing. However, all the tested classifiers had a global accuracy below 70%, so the predictions are not very accurate and the error rate is high.

Over the years, there have been numerous similar studies, which have also provided better results on various other data sets. An example is the study [10], which combined students' previous results with psychological data and factors of their living environment. On this new data set, artificial neural

networks were also used, which provided significantly better results than the other models used (Vector Support Machines, KNN, Naive Bayes).

This study also managed to emphasise the importance of certain attributes when such a prediction is made and to highlight the significant influence of the main attributes that come into contact with the neural networks. Using various attributes in turn, it was noticed that the grade obtained by students at the final exam (97% accuracy) had the greatest influence in the prediction of this study, whereas the psychological factors (accuracy of 69%) had the least significant one.

4.2 Approaches that use Ensemble Learning

Even though there are models that provide high-performance results and have a high accuracy, the idea of using Ensemble Learning to predict students' academic performances naturally brought up in discussion. Ensemble-based learning essentially requires that several models be combined in order to achieve a more efficient model.

In research, the concepts of Bagging and Boosting have proven to provide very good results when applied to decision trees. In this case, the dataset will be divided into several subsets, and each subset will be used to train its own decision trees. This is how the so-called Random Forest is made, the result of which will be represented by the average of the predictions from each tree in its composition. However, a less approached technique that seems to offer high performance and accuracy is Stacking. As a subcategory of ensemble-based learning, this method proposes a combination of several totally different models and types of learning, in order to achieve a much stronger model. Studies have shown that using such a complex model, accuracy can be improved by up to 30% compared to using a simple model. [4]

Such an approach in predicting academic performance is discussed in the article [1] at the University of the West of Scotland, which proposes a hybrid model using three distinct data sets and three different learning algorithms (ANN, Decision Tree, MSV).

The purpose of such a Stacking Ensemble Learning approach was to create a Meta classifier, designed to combine the previously successful performance of other independent models, to minimize error and to improve accuracy.

As expected, the results obtained showed a high accuracy of 81.67%.

In Fig 2 it is represented a diagram that shows how a new Stacking model is made:

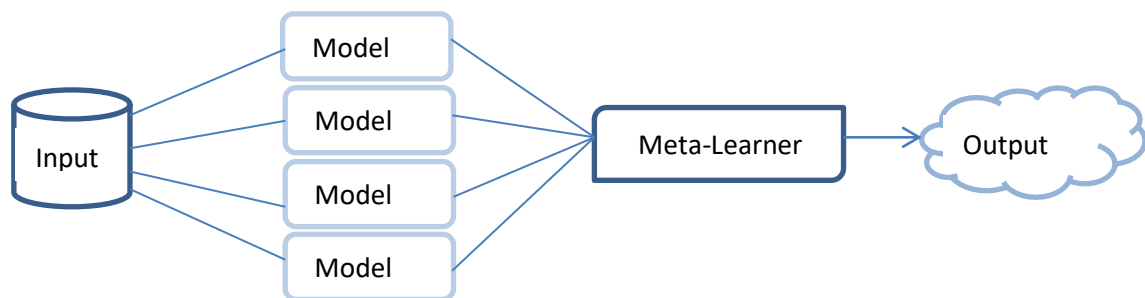


Fig. 2: Stacking model diagram

5 Personal approach

The whole idea of the experiments and the application was based on two data sets called "Personality and Intelligence Interact in the Prediction of Academic Achievement" [12], provided upon request by the Leibniz Institute of Psychology. Based on these, we developed several different classification and regression models that included ensemble-based learning algorithms. The most performant

classification and regression models were later used in a web application, client-server type, meant to help both educational trainers and researchers in pedagogy or DataMining.

One of the main purposes of the experiments was to find out if the psychological and cognitive information is sufficient to predict a student's academic results, or if other types of information are needed.

5.1 Datasets

The two datasets that we mentioned above were obtained from the study [3] conducted by Bergold and Steinmayr, in which an intelligence test and a personality test were applied on two samples of 243 and 421 students of 11th grade. The idea of the study was to demonstrate the impact that psychological factors have on student performance and the usefulness of these factors in predicting the academic outcomes.

Regarding the two datasets, students had to complete the Intelligence-Structure-Test 2000 R and a personality test based on the Big Five model, which captures the five major personality traits: Neuroticism, Extraversion, Openness to the new, Agreeableness and Conscientiousness.

The differences between the two data sets are represented both by the number of students who were involved and by the personality test that was different in the two cases. Moreover, if in the case of the first set of data 35 personality factors were taken into account, in the case of the second only 5 factors (features) were used. The result of the intelligence test was a common feature in both situations, and the value which was intended to be predicted was represented by the Grade Point Average (in the German scoring system). This metric can have values between 1 and 6, where 1 is an excellent result and 6 is an unsatisfactory result. The prediction could also provide results greater than 6, if there are missing values or the data are irrelevant.

Even though the two datasets are intended for regression, after preprocessing we managed to make a division into classes, in order to find out if they can be used in classification algorithms.

After that, in order to perform a comparative analysis and see what are the average results in the field of academic performance prediction, we also used a data set from the Kaggle platform, called "Students' Academic Performance Dataset". This educational data set was collected from the learning management system called Kalboard 360. A system like this provides users access to a plethora of educational resources from any device that is connected to the internet. [13]

The dataset consists of 480 student records and a number of 16 features. The features are classified into three main categories: Academic characteristics such as educational status or class level, Demographic characteristics, such as gender and nationality and behavioral characteristics, such as the number of responses given during classes or parental satisfaction. The data were collected during one academic year, over two semesters: 245 records from the first semester and 235 from the second semester.

As an output, the students are classified into three categories according to their final grade:

- Low-Level: values between 0 and 69
- Middle-Level: values between 70 and 89
- High-Level: values between 90 and 100

5.2 Regression

The first step in construction of the model is to calculate the correlation coefficient for each feature in the dataset. For this, we used some statistical specific methods to find the Spearman and Pearson correlation coefficients and we identified the following values of the degree of importance for each property in the data set:

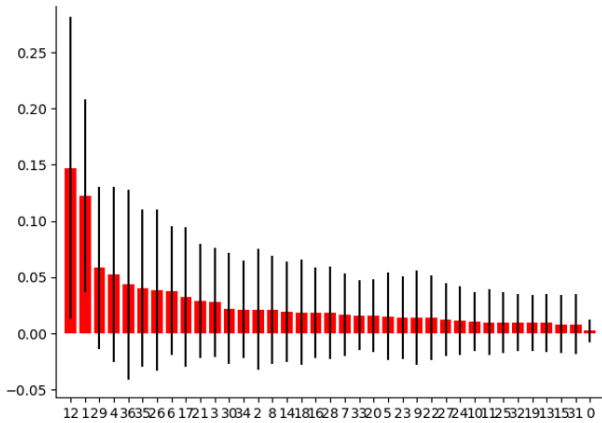


Fig. 3: Correlation coefficient values for each feature from the first dataset

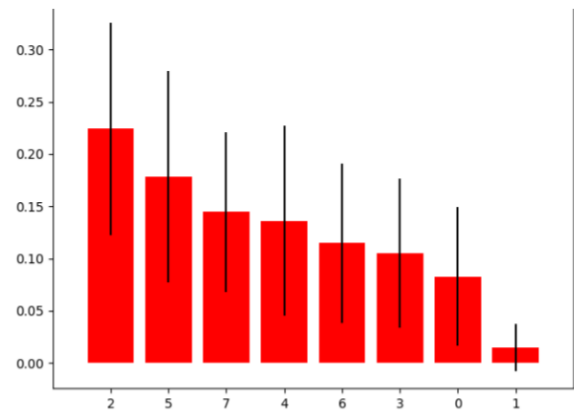


Fig. 4: Correlation coefficients values for each feature from the second dataset

Analyzing the two graphs we can observe that in both situations the correlation coefficients have quite low values, so the next step is to discover if by combining all the features we can get satisfactory results.

In order to identify the most performant model, we used three ensemble learning based algorithms from the scikit-learn library (RandomForestRegressor, XGBoost, GradientBoostingRegressor) on the first two datasets that we have.

We successively performed several tests, in which we varied the number of estimators, as well as the way in which the data were divided into test data and training data. The average results obtained can be seen in the tables below:

	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R2Score
RandomForestRegressor	0.77	0.96	0.98	-1.59
XGBoost	0.87	1.8	1.34	-3.86
GradientBoostingRegressor	0.81	1.28	1.13	-2.46

Table 1: The performances obtained after applying the three learning algorithms on the DS 1

	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R2Score
RandomForestRegressor	0.65	1.14	1.06	0.02
XGBoost	0.74	1.30	1.14	-0.11
GradientBoostingRegressor	0.76	1.33	1.15	-0.13

Table 2: The performances obtained after applying the three learning algorithms on the second dataset

After we analyze the two tables, we can notice that the model that obtained the best results was the one based on the RandomForestRegressor algorithm and the second data set, which has only 7 attributes (features). Having a mean absolute error of 0.65, it was noticed that in many cases the real values are very close to the predicted ones, the results being thus satisfactory. This value was obtained after 50 successive runs, in which the training and test data were randomly selected.

Performing a comparative analysis of the two datasets, it can be seen that regardless of the learning algorithm that was used, the performances were significantly better in the case of the dataset with more existing instances, but with fewer properties (features). A removal of the features with a low correlation coefficient was subsequently tried, but the results did not improve significantly.

5.3 Classification

Using the dataset that previously provided better results in the case of regression and which contains a larger number of instances, we divided the values of GradePointAverage into equal intervals, thus identifying a number of 6 possible classes, which follow the distribution from the Figure 5.

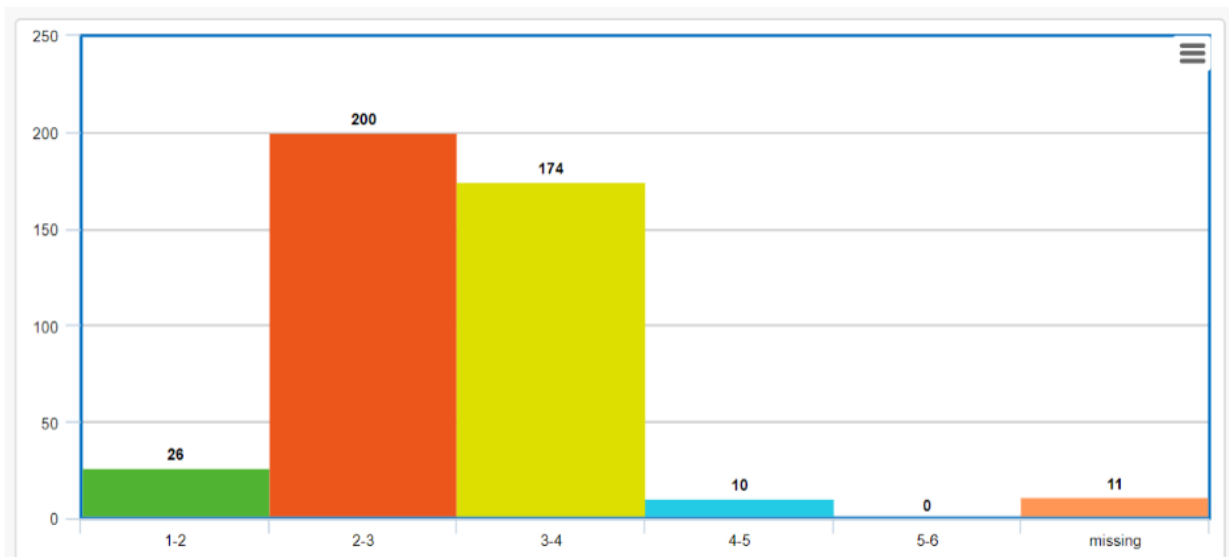


Fig. 5: Distribution of the values in the identified classes

Based on this division, we subsequently transformed the data set from one specific to regression into one dedicated to classification and we used the RandomForestClassifier algorithm to achieve a new learning model. In parallel, the MLPClassifier was used on the same data set, and a comparative analysis of the results that were obtained can be seen in Table 3.

	Accuracy	Recall	Precision
RandomForestClassifier	0.54	0.54	0.48
MLPClassifier	0.41	0.41	0.38

Table 3: The performances obtained by the two learning algorithms on the dataset corresponding to the classification

Carrying out an analysis of the performances obtained by the two classification algorithms, it is observed that in this case neither of them managed to obtain satisfactory performances. Although the basic idea was to use models based on decision tree sets, in this case a compromise was made due to their low performance and a simple learning algorithm was used to identify the possible potential of the dataset.

However, if we analyze the table in detail we will observe that the current dataset is not able to provide relevant information for the realization of a satisfactory classification, regardless of the type of the learning algorithm that was used.

After that, the third dataset presented in the previous section was used and in order to maintain the consistency we developed a learning model using the same RandomForestClassifier. The performance was as below:

- Accuracy: 0.78
- Precision: 0.79
- Recall: 0.78

It can be observed that a model that has in its composition the third dataset, obtained from the Kaggle platform, and the RandomForestClassifier algorithm, is able to provide satisfactory results and achieve a qualitative classification with a satisfactory accuracy.

5.4 The proposed web application

In order to be able to correctly identify the problem and the main purpose of the application, it is important to consider from the beginning who are the beneficiaries (users) of *AcadPredict*. Thus, it is intended for both teachers and educational trainers, as well as researchers who carry out specialized works in the field of pedagogy and in the field of computational intelligence. The main purpose of the application is to make predictions of the grades that students can obtain, as well as a possible classification of them according to certain performance criteria.

The originality of the *AcadPredict* application is represented by the fact that the user can use both a model for classification and one for prediction already trained, on which he can test new instances and perform some performance analyzes, but he also has the possibility to upload completely new datasets and to train their own models, that will be saved for later.

In order to do this, in the beginning the user needs to log in the application based on his username and password . This authentication allows the application to remember for each user the last trained model as well as the results obtained by it. Later, after authentication, the users will be able to choose which functionality they want to use next: Classification or Prediction. For both, at the first use of the application the user will have at his disposal a Random Forest model already trained on a

corresponding data set. Later, it can be changed and the model will be automatically retrained. For each of the two basic functions mentioned above, the user will see in a table the results obtained for all instances in the test dataset. This table can be sorted according to its existing attributes and also provides a quick way to search for values. Moreover, the user can see and analyze the performance obtained by the models. For classification, accuracy, precision and recall will be displayed, and for the prediction, three different error calculation metrics will be displayed: Mean Absolute Error, Mean Squared Error and Root Mean Squared Error. After that, the user has the opportunity to test his new instances. Depending on the dataset that was used, all fields that need to be filled in will be generated automatically. In addition, for both classification and regression, it will be possible to view performance graphs of the algorithms used on the current dataset. Last but not least, the models used in the application do not require a retraining every time the application is started, because they remain saved in memory, and the user will be able to access each time the last model he trained, analyzed and used.


To sum up, within the AcadPredict application, we created two machine learning models, which are based on the RandomForestRegressor and RandomForestClassifier algorithms from the scikit-learn library. The previous section also explained in detail the reasons that led to the choice of these algorithms, as well as other algorithms that were proposed for use. The main quality of the created models is that they were parameterized, being very versatile and able to successfully use any kind of datasets, without first knowing the data form or properties (features).

Back

Show 10 entries Search:

Gender	Nationality	PlaceofBirth	StagelD	GradelD	SectionID	Topic	Se
F	Iraq	Iraq	lowerlevel	G-02	B	Arabic	F
F	Iraq	Iraq	lowerlevel	G-02	B	Arabic	S
F	Jordan	Jordan	MiddleSchool	G-07	A	Biology	F
F	Jordan	Jordan	MiddleSchool	G-07	A	Biology	S
F	Iraq	Iraq	MiddleSchool	G-07	A	Biology	F
F	Iraq	Iraq	MiddleSchool	G-07	A	Biology	S
F	Syria	Syria	MiddleSchool	G-07	A	Biology	F
F	Syria	Syria	MiddleSchool	G-07	A	Biology	S
F	Jordan	Jordan	MiddleSchool	G-07	B	Biology	F
F	Jordan	Jordan	MiddleSchool	G-07	B	Biology	S

Showing 1 to 10 of 96 entries Previous 1 2 3 4 5 ... 10 Next



Classification
Metrics

0.78
Accuracy

0.79
Precision

0.78
Recall

Choose a new dataset and train your new model

Upload CSV file...
 No file chosen

Fig. 6 The main page for the classification

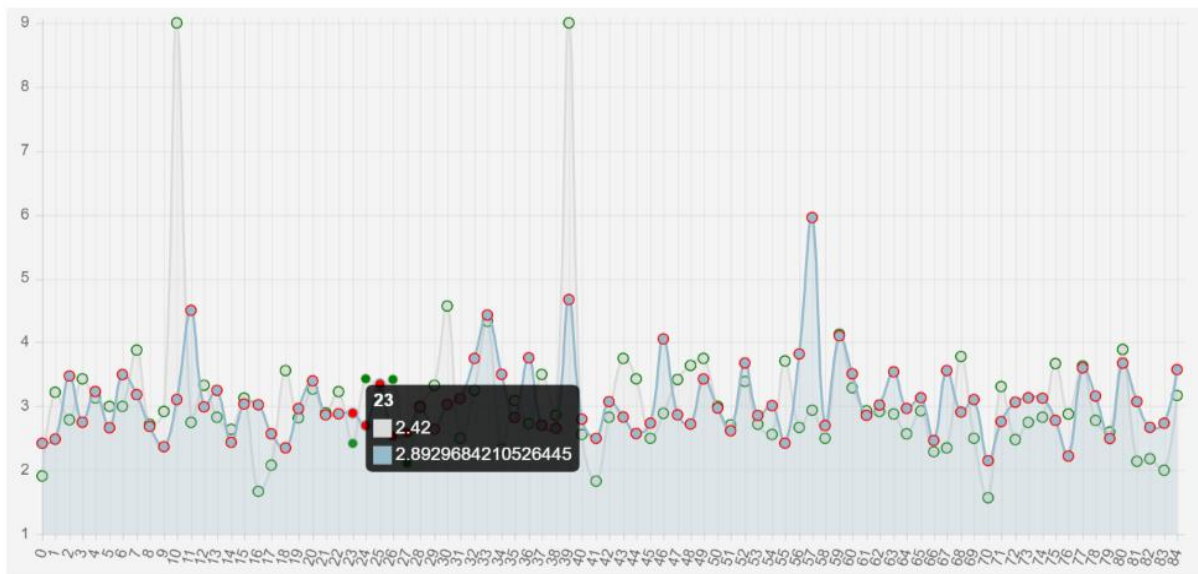


Fig. 7: Graph representing the real values compared to the predicted ones

6 Conclusion

Although there have been studies on the prediction of the academic performances for a long time, the usage of machine learning techniques in this field remains a topic that still has much to offer. It is demonstrated that these predictions lead to a significant improvement in the quality of the educational process, and unfortunately for the moment only a few types of models have been addressed in the literature, mostly based on supervised learning. Of these, the most efficient were those based on decision trees and neural networks, but it is clear that other possible variants should not be excluded. Moreover, recent studies have shown that a combination of several models, especially the use of the Ensemble Learning, can lead to a significant increase in the accuracy and performance of the algorithms.

As with all machine learning applications, a major problem is the datasets needed to train and test the models. If most papers used academic data, represented by previous performances, the current paper aims to take a different approach, trying to find out if psychological and environmental factors can be used in predicting academic results.

After realizing several experiments, we concluded that these factors are not sufficient, but that used together with other data about students' previous academic performance and their behavior throughout the educational process, can lead to the development of some high-performance models, both classification as well as regression, in order to provide satisfactory results and to open new research directions in the domain.

I believe that this paper managed to capture a large part of the concepts applied in the prediction of academic performances and at the same time it was an introduction to the supervised learning based on ensembles.

When we talk about future developments on the current application, I think that an useful feature would be to introduce directly in the application the personality and the intelligence tests. Thus, a new type of user would appear in the AcadPredict application, the student, who would have access only to the tests section, in order to complete them, and the results would be collected automatically. In this way, teachers or researchers would no longer have to manually enter data for testing new instances, as

they are generated dynamically. This streamlines the testing process, and the focus may be on the results and their analysis.

Another good idea for future development would be to try other types of learning to predict academic performances, thus being able to use unsupervised learning and reinforcement learning.

Finally, I strongly believe that the extraction of knowledge from academic data remains an area with great potential in the future, which may gradually lead to a reform of the current educational system. Education must be seen as a basic element of the society in which we live and must always adapt to the increasingly rapid technological evolution. Only in a quality educational environment can characters be formed, and students can develop, acquire the necessary skills and learn the true values.

References

- [1] Adejo, O.W., Connolly, T., *Predicting student academic performance using multi-model heterogeneous ensemble approach*, Journal of Applied Research in Higher Education, Vol.10, 2017
- [2] Baker, R., *Data Mining for Education*, International Encyclopedia of Education (3rd edition), vol. 7, pp. 112-118. Oxford, 2010
- [3] Bergold, S., Steinmayr, R., *Personality and Intelligence Interact in the Prediction of Academic Achievement*, Journal of Intelligence 6(2):27, 2018
- [4] Finlay, S., *Multiple classifier architectures and their application to credit risk assessment*, European Journal of Operational Research, Nr. 210(2), 2011
- [5] Kabakchieva, D., *Predicting Student Performance by Using Data Mining Methods for Classification*, CYBERNETICS AND INFORMATION TECHNOLOGIES, Vol.12, Nr.1, Sofia 2013
- [6] Romero, C., Ventura S., *Educational Data Mining: A Review of the State-of-the-Art*, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews. 40(6), 601-618, 2010
- [7] Romero, C., Ventura, S., *Educational Data Mining: A Survey from 1995 to 2005*, Expert Systems with Applications, Vol. 33, 2007, 135-146.
- [8] Romero, C., Ventura, S., Pechenizkiy M, Baker R., *Handbook of Educational Data Mining*, CRC Press, 2011
- [9] Schmelzer, R, *AI Applications In Education*, July 2019, <https://www.forbes.com/sites/cognitiveworld/2019/07/12/ai-applications-in-education/> (accessed 14 May 2020)
- [10] Shahiri, A.M., Husain W, Rashid N.A., *A Review on Predicting Student's Performance using Data Mining Techniques*, The Third Information Systems International Conference 2015, <https://doi.org/10.1016/j.procs.2015.12.157>
- [11] Triantafillou, E., Pomportsis, A., and Demetriadis, S. *The design and the formative evaluation of an adaptive educational system based on cognitive styles*. Computers & Education, 2003
- [12] <https://www.psychdata.de/index.php?main=search&sub=browse&id=srra08pe02&lang=eng>
- [13] <https://www.kaggle.com/aljarah/xAPI-Edu-Data>

Alexandru-Mihail CRĂCIUN
 Babes-Bolyai University
 Faculty of Mathematics and Computer Science
 Mihail Kogalniceanu nr.1, 400084, Cluj-Napoca,
 ROMANIA
 E-mail: alex.mihail.craciun@gmail.com

DICOM image segmentation

Valentin-Gabriel Crăciun, Matei-Florin Graură

Abstract

Organ segmentation is one of the most complex tasks for current image preprocessing techniques, due to some characteristics of DICOM medical images. Therefore, the training of neural networks that can accurately detect tumors or malignant soft tissue, is determined by the quality of the segmentation performed on input images. Due to the differences in CT/MRI machines calibrations and manufacturers, DICOM images have a wide range of Hounsfield units, resulting in machine learning models having difficulties in differentiating between organ tissue from the rest of the pixel map. This paper presents a semi-automated image segmentation approach, based on both the original DICOM image and the image on which a doctor performed a manual segmentation using a DICOM viewer. Having an approximation of the organ area and not relying only on the Hounsfield units chart, it reduces the risk of error when choosing an entry point on the translated DICOM image. This approach combines clustering models with a technique of pixel flooding, by adding depth to the pixel map, resulting in an easier and more accurate representation of the real organ. This workflow has been tested against ideal segmentations of a particular organ, achieving on average similarity coefficients of over 90%. The approach is beneficial both for generating quality training inputs and for augmenting doctors segmentations by refining cut or clipped edges.

1 Introduction

Segmentation of images is an important part of image processing and is used in medical applications such as malignant tissue detection and in monitoring the evolution of abnormalities. Segmentation refers to the process of partitioning an image into multiple segments that can be used or interpreted on their own i.e the liver in an CT scan. The main motivation behind medical image segmentation is to achieve computer aided detection that improves diagnostics and prevention. Considering that many abnormalities can be detected early the importance of image segmentation becomes clear.

There are many methods used to achieve a clean segmentation, such as fuzzy clustering [4] for initial processing, level set techniques used for object border tracking and variations between region growing based approaches. These methods on their own do not result in consistently accurate segmentations, due to differences in Hounsfield units present in input images. A combination of clustering and morphological transformations lead to improved results. Also, an important aspect to consider is the different characteristics of each organ, so in order to achieve a segmentation that performs well in different scenarios, there is a need to employ techniques that can establish seed values as close as possible to the true Hounsfield units of that particular organ.

This paper is organized as follows. Section 2 describes the techniques and how they were used to create a semi-automated approach for DICOM images segmentation. In Section 3, results from the experimentation process are shown and the last section presents conclusions.

2 The proposed method

In this section we present the main steps of our proposed segmentation method for obtaining more accurate images of organs from DICOM images. Neural Networks and other machine learning (ML) techniques are used to detect tumors or malignant soft tissue from DICOM images. Due to the differences in CT/MRI machines calibrations and manufacturers, DICOM images have a wide range of Hounsfield units, resulting in ML models having difficulties in differentiating between organ tissue from the rest of the pixel map. Organ segmentation is a medical image preprocessing operation used to that mitigates this issue. The quality of the organ segmentation strongly influences the accuracy of neural networks predictions.

It has been found that applying established image segmentation algorithms [2-3] on their own will not suffice because of:

- the nature of DICOM images and their Hounsfield units mapping
- the domain of these algorithms which need to be tuned or modified for the specific task being performed.

We propose a semi-automated image segmentation approach based on both the original DICOM image and the image on which a doctor performed a manual segmentation using a DICOM viewer. We use the approximation of the organ area given by a doctor and do not rely only on the Hounsfield units chart. Thus, we reduce the risk of error when choosing an entry point on the translated DICOM image.

Our approach combines clustering models with a technique of pixel flooding, by adding depth to the pixel map, resulting in an easier and more accurate representation of the real organ. This workflow has been tested against ideal segmentations of a particular organ, achieving on average similarity coefficients of over 90%. The approach is beneficial both for generating quality training inputs and for augmenting doctors segmentations by refining cut or clipped edges.

2.1 DICOM preprocessing

Digital Imaging and Communications in Medicine (DICOM) is a standard for the management of medical images which stores related data in a certain format. It is used for transmitting medical images such that different devices can display a CT scan without the need of any special equipment. The measurement unit in CT scan is the Hounsfield unit (HU), which measures radiodensity. These units have to be extracted from pixel values.

In this article, Python has been used to load the CT scans and then convert pixels to Hounsfield units. DICOM files contain metadata about pixels, such as how long one pixel is in a real representation and its direction. In order to extract this data and then sample each slice accordingly to size, the pydicom package [6] has been used. Scans differ one from another, leading to inaccurate results when processing the converted image. To over-

come this issue, after loading the scan slices, missing pixel metadata can be inferred using the existing information.

The code presented in Figure 1, shows an example of loading and processing CT scans which have missing metadata corresponding to the Z axis of pixels and how it can be inferred. The formula used to extract Hounsfield units from pixels is the following:

$HU = PV \cdot Slope + Intercept$:

- Hu = Hounsfield unit corresponding to pixel value
- PV = pixel value in grayscale image
- Intercept, the rescale intercept provided as metadata
- Slope, the rescale slope provided as metadata

```
def pixels_hu(scan):
    slices = sorted(scan, key=lambda s: s.SliceLocation)
    thickness = np.abs(
        slices[0].ImagePositionPatient[2] - slices[1].ImagePositionPatient[2]
    )

    for s in slices:
        s.SliceThickness = thickness

    image = np.stack([s.pixel_array for s in slices])
    image = image.astype(np.int16)
    image[image == -2000] = 0

    intercept = slices[0].RescaleIntercept
    slope = slices[0].RescaleSlope

    image = slope * image.astype(np.float64)
    image = image.astype(np.int16)}

    image += np.int16(intercept)
    image = np.array(image, dtype=np.int16)-

    return image
```

Figure 1: Code Example - Loading and processing CT scans which have missing metadata

2.2 Thresholding

We applied filters to the input data to reduce the noise on the image because we need to standardize the data. The filter that evens out the data as well as possible is a blur filter with a Gaussian kernel. After that we extract the points in the middle of the doctor's segmentation in order to have a landmark related to the area that we need to segment and also to know what values are in it.

A part of the implementation used to standardize the data is presented in Figure 2.

```
def eliminate_noise(data):
    ksize = 9
    kernel = cv2.getGaussianKernel(ksize, 0.3 * ((ksize - 1) * 0.5 - 1) + 0.8)
    data_preprocessed = cv2.filter2D(data, -1, kernel)
    data_preprocessed = cv2.blur(data_preprocessed, (const.kernel_size, const.kernel_size))
    return data_preprocessed
```

Figure 2: Code Example - Data standardization

2.3 Clustering

The main scope of clustering analysis is to divide a given set of image data or objects into a clusters, which represents subdivisions or a subgroup. The partition should have two properties: homogeneity and heterogeneity. The homogeneous clusters consist of similar data whereas the heterogeneous clusters contain non-similar data.

We used K-means++ algorithm for data clusterization. K-means++ is an algorithm for choosing the initial values for the k-means clustering algorithm. The k-means problem is to find cluster centers that minimize the intra-class variance, i.e. the sum of squared distances from each data point being clustered to it's cluster center (the center that is closest to it). The core of K-means++ clustering algorithm is presented in Figure 3.

```
for clusters_number in range(2, 15):
    k = KMeans(n_clusters=clusters_number, n_init=1, n_jobs=-1).fit(data_input_as_array)
    for index, centroid in enumerate(k.cluster_centers_):
        if fabs(doctor_segmentation_values_avg - centroid) < distance_between_centroid_average:
            distance_between_centroid_average = fabs(doctor_segmentation_values_avg - centroid)
            cluster_containing_organ = index
            winner_label = k.labels_
    return [winner_label, cluster_containing_organ]
```

Figure 3: Code Example -. K-means++ clustering algorithm

Given a multitude of observations (a_1, a_2, \dots, a_m), where each observation is a d -dimensional real vector, k -means clustering aims to partition the m observations into $k \leq m$ sets $S = \{S_1, S_2, \dots, S_k\}$ so as to minimize the within-cluster sum of squares. After choosing the set that has the most points that coincide with the manually made segmentation, we notice that not only the region of interest is found. The cluster we get contains both our organ and other areas which have very similar values.

Based on the observation from the first clustering, we decided that we need one more clustering of this data based on the density of the remaining points after the first clustering, to get only the portion of interest. For this purpose, we applied the DBSCAN (Density-based spatial clustering of applications with noise) algorithm, presented in Figure 4. Given a set of points in some space, the algorithm groups together the points that are closely packed together, marking as outliers the points that lie alone in low-density regions (whose nearest neighbors are too far away).

```
db_scan = DBSCAN(eps=0.1, min_samples=100, n_jobs=-1).fit(scaled_indices)
labels = db_scan.labels_
label_containing_point_index = -1
result = np.ndarray([const.input_size, const.input_size])
```

```
for index in range(0, len(labels)):
    if Point(indices[index][0], indices[index][1]) == doctor_segmentation_middle_point:
        label_containing_point_index = labels[index]

for index in range(0, len(labels)):
    if labels[index] == label_containing_point_index:
        result[indices[index][0]][indices[index][1]] = 1
```

Figure 4: Code Example - DBSCAN algorithm

After applying the DBSCAN, we obtained the organ with small adjacent areas that are not part of it, but have similar values. For example, in Figure 5, we can see that a liver has been segmented, but it contains small portions, on the edges, which have very similar Hounsfield values, but are still not part of the organ.



Figure 5: Image of liver after applying DBSCAN

2.4 Watershead

Watershed is an algorithm used in image processing for separating different objects in an image, using a basin flooding technique. The algorithm requires user or model defined markers in order to be able to create a topographic map of the pixel values. After establishing these markers, the next step is flooding the basins of the elevation created until different markers meet on the watershed lines. For accurate results markers need to be chosen as local minima of the image, from which the basins are then flooded. In [1] the importance and process of thresholding have been explained in detail. Without thresholding the classic approach produces over segmented results due to noise or Hounsfield units irregularities. To address this, besides specifying local minima markers, valley points are

chosen as accurately as possible in order for pixels that are not morphologically similar to be excluded, by labelling the foreground area of the organ being segmented. Watershed segmentation steps are as follows [5]:

1. Compute an image whose dark regions are the objects that are being segmented.
2. Compute foreground markers for the pixels within each of the objects.
3. Compute background markers for the pixels that are not part of the objects being segmented.
4. Identify the local minima at the location where foreground and background markers meet.
5. Apply the watershed transform

In Figure 6 is given a visual representation of how image topography is treated in the watershed algorithm

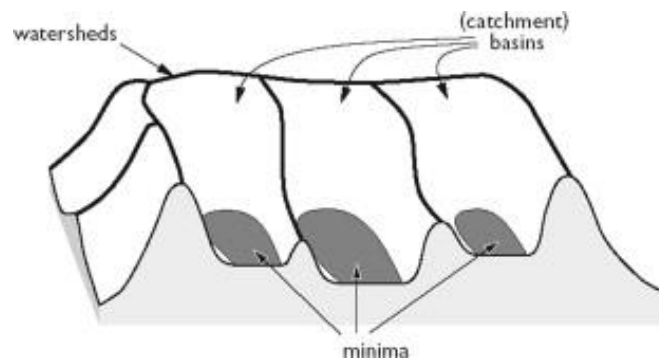


Figure 6: Watershed algorithm visualization [3]

3 Conclusion

In this paper, we combined different techniques used to improve the results of medical image segmentation and proposed a more efficient semi-automated segmentation approach. The main focus of this approach is to achieve a lightweight implementation which can be used both for diagnosis and neural networks training. Our method is based on the observation that using a dataset containing DICOM images manually segmented and using a combination of clustering [1] and watershed [2] techniques can improve the accuracy of the segmentation and produce quality training material for automatic segmentation. As future work, the implementation of a neural network is desirable as the approach described in this paper would be greatly enhanced.

Acknowledgement: We thank Prof. Dana Simian for useful indications and for supervising this article.

References

- [1] D. Arthur, S. Vassilvitskii, *k-means++: The Advantages of Careful Seeding*, SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007
- [2] K. Lalitha, R. Amrutha, M. Stafford, M. Shivakumar Asst Professor, *Implementation of Watershed Segmentation*, International Journal of Advanced Research in Computer and Communication Engineering, 2017
- [3] B. Preim, Ch. Botha, *Image analysis for Medical Visualization*, Visual Computing for Medicine (Second Edition), 2014
- [4] J. Umamaheswari, G. Radhamani, *An Optimal Approach for DICOM Image Segmentation Based on Fuzzy Techniques*, Springer, Berlin, Heidelberg, 2012. https://doi.org/10.1007/978-3-642-30157-5_79
- [5] OpenCv, <https://opencv.org/>
- [6] Pydicom, <https://pydicom.github.io>

Valentin-Gabriel CRACIUN
"Lucian Blaga" University of Sibiu
Informatics
Strada Doctor Ion Rațiu 5-7, Sibiu 550012, România
ROMANIA
E-mail: valentin.craciun99@gmail.com

Matei-Florin GRAURA
"Lucian Blaga" University of Sibiu
Informatics
Strada Doctor Ion Rațiu 5-7, Sibiu 550012, România
ROMANIA
E-mail: grauramatei@gmail.com

Approaches for reducing the number of intersections between the components of software architectures

Ligia – Izabela Crăciunescu

Abstract

Sorting of EA Diagrams is an application which aims to bring a **substantial improvement** of project SW architectures, by **automatically reducing the number of intersections between architectural components connections** and **rearranging architectural diagrams** so that there can exist a **logical and systematic organization** in structuring related components and interactions between these. In order to achieve all these goals, the **Sorting of EA Diagrams** application was developed as an extension of the high performance and complexity tool, **Enterprise Architect**. This tool offers **the possibility of extension** through the **add-in feature**, which allows the programmer to improve the user interface by adding new menus, submenus and other controls to perform a variety of functions and create new features that are not available in the original product. Using this powerful expansion feature provided, **Sorting of EA Diagrams** application adds to **Enterprise Architect**, a menu called "Sorting of diagrams", which makes the transition to the "Sort" button, through which all necessary changes are made to achieve these objectives previous mentioned.

Keywords: SW architectures, SW components, architectural diagrams, overlappings, Sorting of EA diagrams, Enterprise Architect, add-in feature

1 Introduction

Over time, science evolves, and adapting at changes in any field becomes vital. Due to the fact that software systems have reached a high complexity which requires an organization and planification as efficient as possible, the IT industry from recent years has popularized the need and the importance of software architectures [1].

Because architecture is an essential part of any software development process, it is necessary to understand its purpose and value. Software development often fails, especially when development teams deal with very large and very complex systems. Analyzing the main reasons for these project errors, it was found that most of the problems were raised by the software architecture of the project [1].

We can define a software architecture as being the concept of the highest level of a system, which involves the organization or structuring of significant components, the import and export relations between them and the interfaces through which they interact. These components can be composed in their turn from a succession of other components and smaller interfaces [2].

Due to the difficulties caused by the manual modeling of high complexity architectures, certain tools have been developed for automated generation in Enterprise Architect. However, the problem with automatic generation is that sometimes the result is chaotic and almost impossible to control.

In order to obtain an accuracy in analyzing the software architectures essential in the development of highly complex projects, as well as in understanding the functionalities and dependencies between the components of the architectures, I developed the Sorting of EA Diagrams application.

2 Theoretical aspects

2.1 Software Architectures

According to [5], a solid architectural vision is a key aspect in the triumph of a software project. The software architecture of a system focuses on the design and implementation of high-level software structures. This is the result of embedding a significant number of architectural elements, in certain well-chosen forms to meet the major requirements of the system's functionality and performance, as well as non-functional ones. Software architecture usually represents the connection between requirements and code implementation. (Fig. 1).

Although there are many definitions of software architecture, at the heart of them all is that the architecture of a software system involves the definition of architectural elements (Fig. 2), called components, along with certain properties of them, to meet system requirements. The components communicate with each other through interfaces, and the interfaces are connected using connectors [6].

A component is an executable software unit with defined interfaces and individual identities that can be modified. This, similar to a class, can be instantiated and offers a complex functionality [7].

The connector connects the required interface of one component to the interface provided by another component; this allows one component to provide the services that another component requires [8].

A port is a specific construction of a component. The use of ports can specify the behaviors or services that the component provides, as well as the behaviors or services that a component requires. Ports can specify inputs and outputs because they can operate bidirectionally [6].

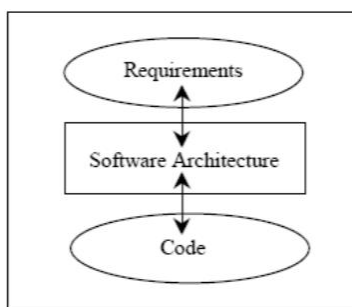


Fig. 1: The role of SW Architecture

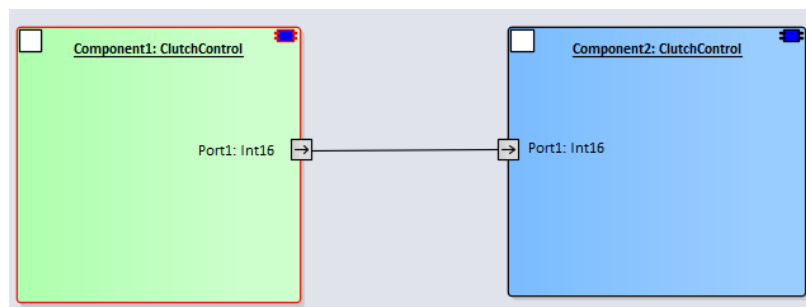


Fig. 2: The defining elements of a SW Architecture

According to [9], the system development architecture focuses on organizing the real software modules of the software development process. The software is divided into smaller components, which can be developed by one or more developers. Thus, the development architecture is represented by diagrams of modules and subsystems, which show import and export realities. The complete development architecture can only be described when all elements of the software have been identified.

Within the *Drivetrain* department of *Vitesco Technologies Engineering Romania SRL*, the software development architectures for various projects are made through the Enterprise Architect (EA) which is a licensed tool. Fig. 3 illustrates an architectural diagram of a software component within a project.

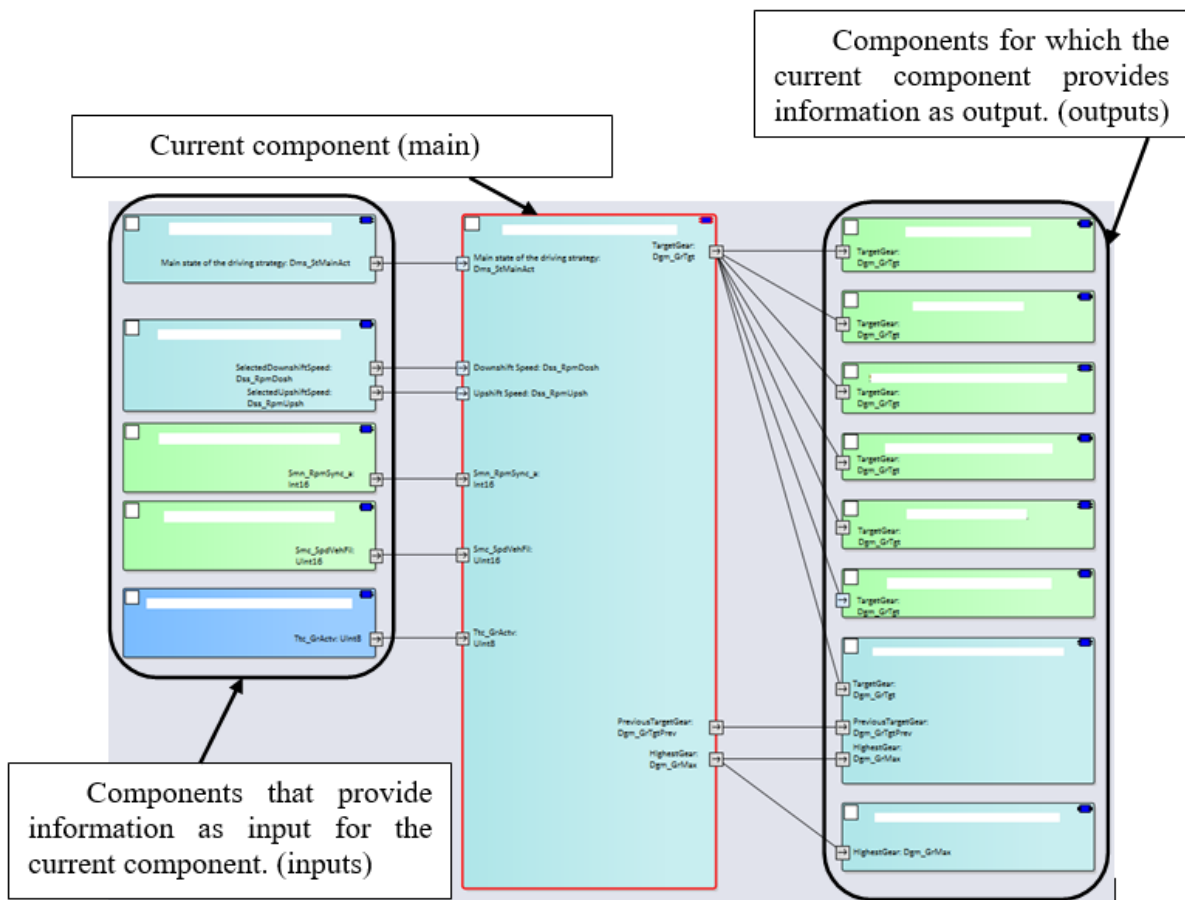


Fig. 3: Architectural diagram of a SW component from a project

3 Application design

3.1 Phases of development of reduction approaches

Consider the following diagram from a software architecture of a project, which contains a main component with 7 ports and 4 external components connected according to fig. 4.

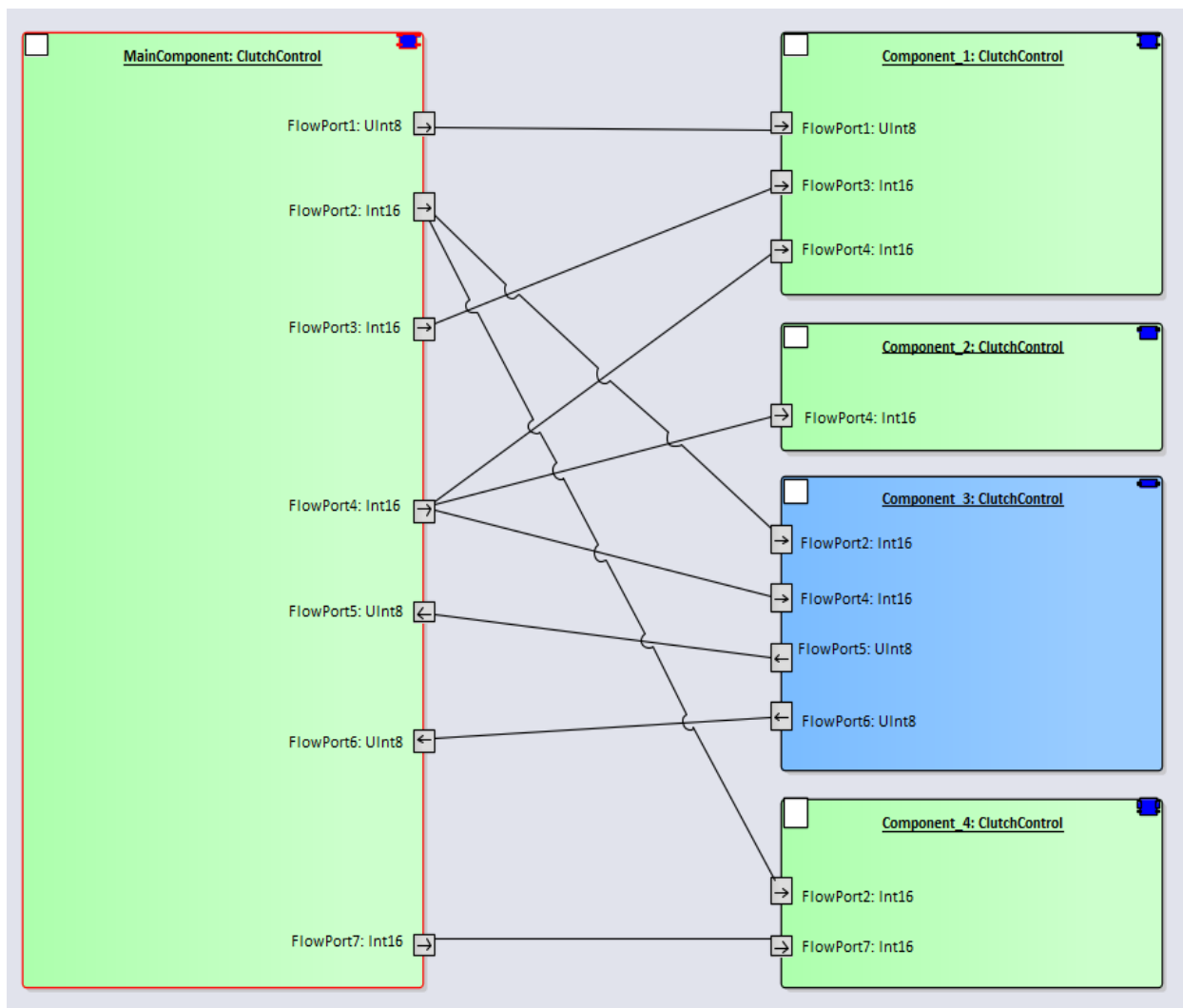


Fig. 4: Example of an architectural diagram of a SW project

Note: Initially there are 9 intersections of connections between the components of the diagram.

To improve the visual appearance of software architectures, four major steps have been developed in order to reduce the number of intersections between component connections.

Observation: Next, the four steps developed will be defined and exemplified on the diagram given in fig. 5.

3.1.1 Phase I. Formation of the initial matrix

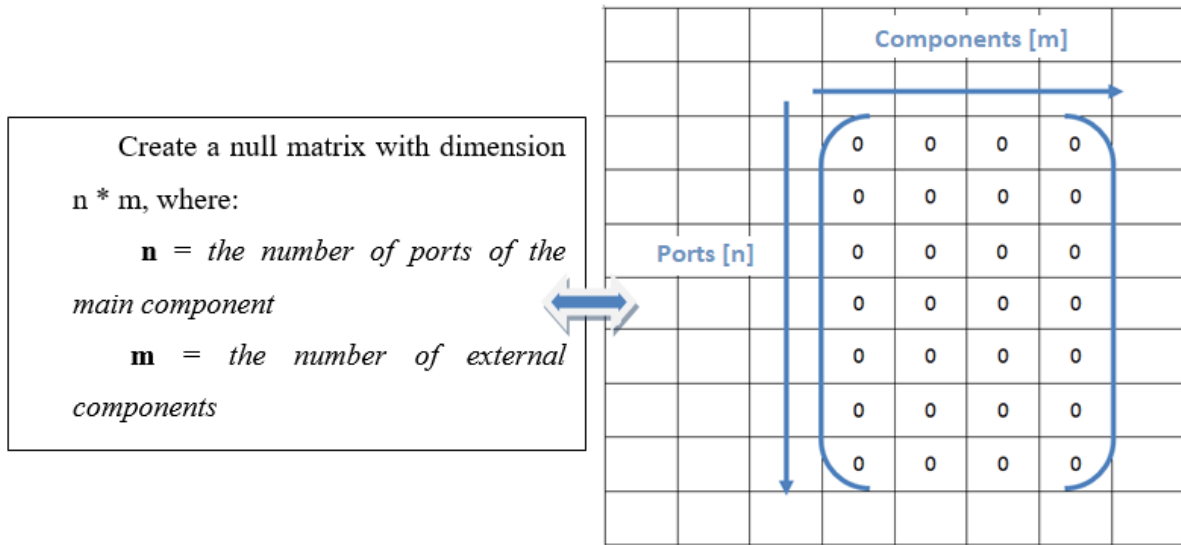


Fig. 5: Null initial matrix

Example:

For the architectural diagram initially given, the following are considered:

P1 = Flowport1

P2 = Flowport2

P3 = Flowport3

P4 = Flowport4 **n = 7** **and**

P5 = Flowport5

P6 = Flowport6

P7 = Flowport7

C1 = Component_1

C2 = Component_2

C3 = Component_3 **m = 4.**

C4 = Component_4

Identify the connections between each port of the main component and the external components of the diagram and fill in the array with the value "1" where connections were found, respectively with the value "0" where there are no connections. Fig. 6 illustrates how to complete the initial matrix for the architectural diagram given as an example.

Example:**P1** has connection to **C1**.**P2** has connections to **C3** and **C4**.**P3** has connection to **C1**.**P4** has connections to **C1**, **C2** and **C3**.**P5** has connection to **C3**.**P6** has connection to **C3**.**P7** has connection to **C4**.

		C1	C2	C3	C4	
	P1	1	0	0	0	
	P2	0	0	1	1	
	P3	1	0	0	0	
	P4	1	1	1	0	
	P5	0	0	1	0	
	P6	0	0	1	0	
	P7	0	0	0	1	

Fig. 6: How to fill the initial matrix

3.1.2 Phase II. Identification of the first component

For each component, the total number of connections between the other components and each port connected to the current component is calculated.

Example:

		C1	C2	C3	C4	
	P1	1	0	0	0	
	P2	0	0	1	1	
	P3	1	0	0	0	
	P4	1	1	1	0	
	P5	0	0	1	0	
	P6	0	0	1	0	
	P7	0	0	0	1	

According to fig. 7, component C1 has the following connection ports:

P1 -> 0 connections with other components**P3** -> 0 connections with other components**P4** -> 2 connections with other components

=> **total = 0 + 0 + 2 = 2 connections**

Fig. 7: Identifying connections for component C1

		C1	C2	C3	C4
P1	1	0	0	0	0
P2	0	0	1	1	
P3	1	0	0	0	
P4	1	1	1	0	
P5	0	0	1	0	
P6	0	0	1	0	
P7	0	0	0	1	

According to fig. 8, component C2 has the following connection port:



P4 -> 2 connections with other components

=> **total = 2 connections**

Fig. 8: Identifying connections for component C2

		C1	C2	C3	C4
P1	1	0	0	0	0
P2	0	0	1	1	
P3	1	0	0	0	
P4	1	1	1	0	
P5	0	0	1	0	
P6	0	0	1	0	
P7	0	0	0	1	

According to fig. 9, component C3 has the following connection ports:



P2 -> 1 connections with other components

P4 -> 2 connections with other components

P5 -> 0 connections with other components

P6 -> 0 connections with other components

=> **total = 1 + 2 + 0 + 0 = 3 connections**

Fig. 9: Identifying connections for component C3

		C1	C2	C3	C4
P1	1	0	0	0	0
P2	0	0	1	1	
P3	1	0	0	0	
P4	1	1	1	0	
P5	0	0	1	0	
P6	0	0	1	0	
P7	0	0	0	1	

According to fig. 10, component C4 has the following connection ports:



P2 -> 1 connections with other components

P7 -> 0 connections with other components

=> **total = 1 + 0 = 1 connection**

Fig. 10: Identifying connections for component C4

Choose the component with the lowest number of connections as the first component and move to the appropriate position.

Example:

In our case, the minimum number of connections obtained is 1, corresponding to component C4. Therefore, it will occupy the first position in the matrix (fig. 11).

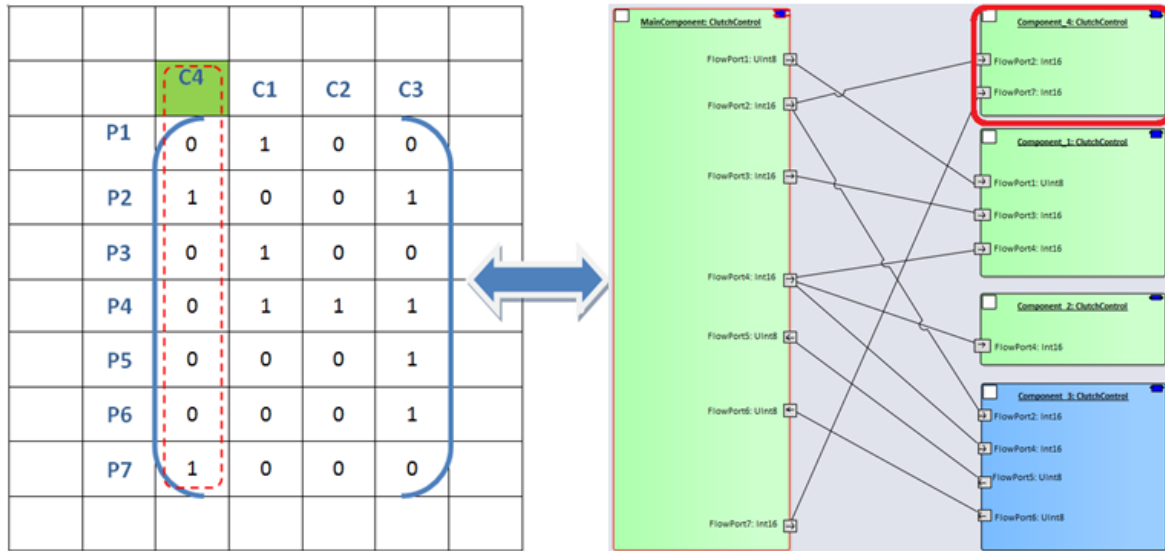


Fig. 11: Setting the C4 component to the first position

If there are several components that have the same minimum number of connections, a new method is applied for those components, called the Final Decision. This method involves calculating the decimal values of the components in question, by converting the binary code corresponding to them, taken from the matrix on the column corresponding to each required component.

After obtaining the decimal values, the component with the highest value will be chosen as the first component.

If the decimal values obtained from the application of the Final Decision will be equal, then the first of these components will be chosen according to the initial order.

3.1.3 Phase III. Port ordering

This step involves rearranging all ports that have connections to the previously set component so that they are grouped and ordered accordingly.

For each port connected to the component set in the previous step, the number of connections to the other components is calculated.

Rearrange the respective ports in ascending order of the number of connections obtained.

Example:

The component previously set in this method is C4, whose connecting ports, P2 and P7, are grouped and ordered according to fig. 12.

	C4	C1	C2	C3				C4	C1	C2	C3
P1	0	1	0	0				P7	1	0	0
P2	1	0	0	1	1 connection			P2	1	0	1
P3	0	1	0	0				P1	0	1	0
P4	0	1	1	1		$P7 < P2$		P3	0	1	0
P5	0	0	0	1				P4	0	1	1
P6	0	0	0	1				P5	0	0	1
P7	1	0	0	0	0 connections			P6	0	0	1

Fig. 12: Grouping and ordering the ports of the component C4

The result obtained on the architecture after the application of this method is presented in fig. 13.

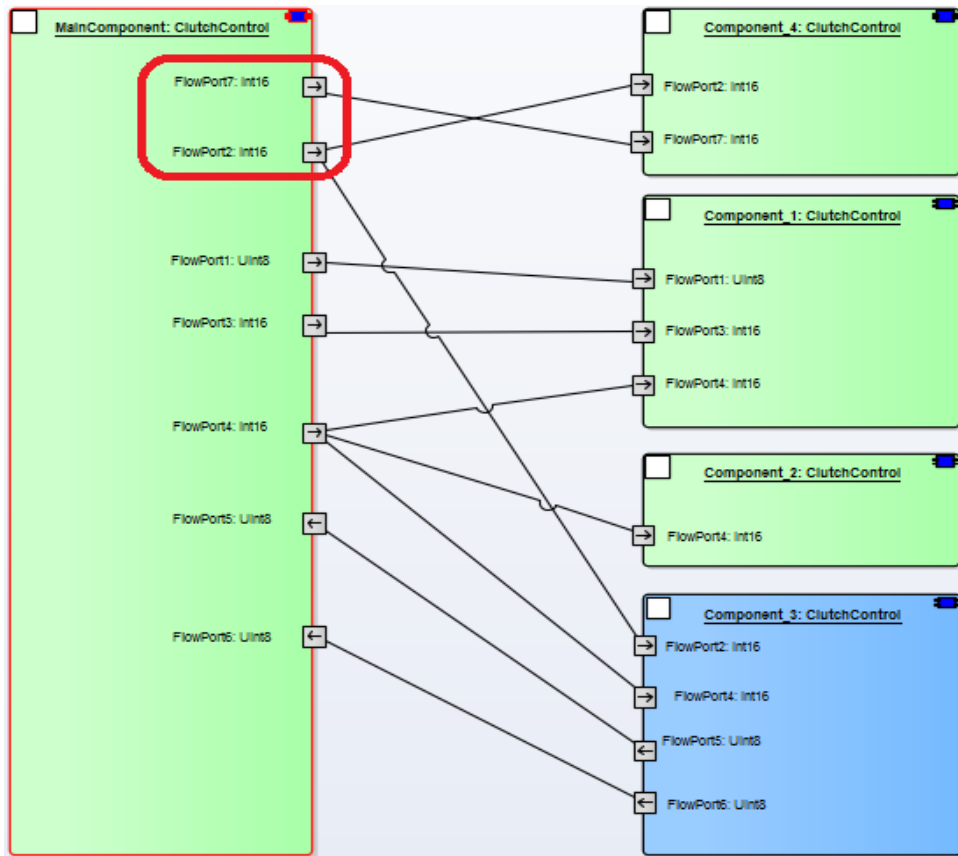


Fig. 13: The result obtained on the architecture

3.1.4 Phase IV. Ordering the components

This phase aims to determine the next component by identifying the connections that the current port has with the other components that have not yet been set.

In our case, the aim is to group all the components that are connected to the current port.

Example:

For the current port P2, only one connection to C3 was found, so it will occupy the second position in the matrix, according to fig. 14.

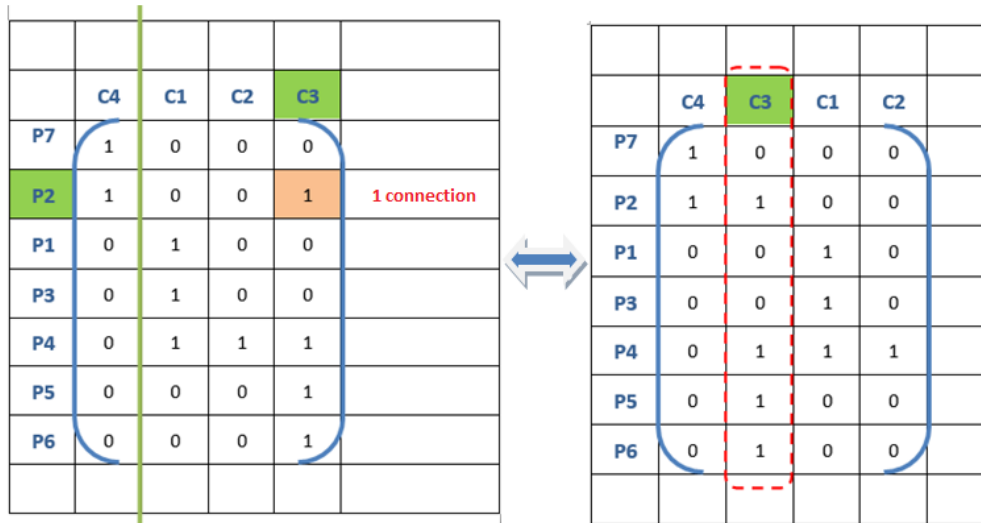


Fig. 14: Component C3 will occupy the second position

The result obtained on the architecture following the setting of component C3 is illustrated in fig. 15.

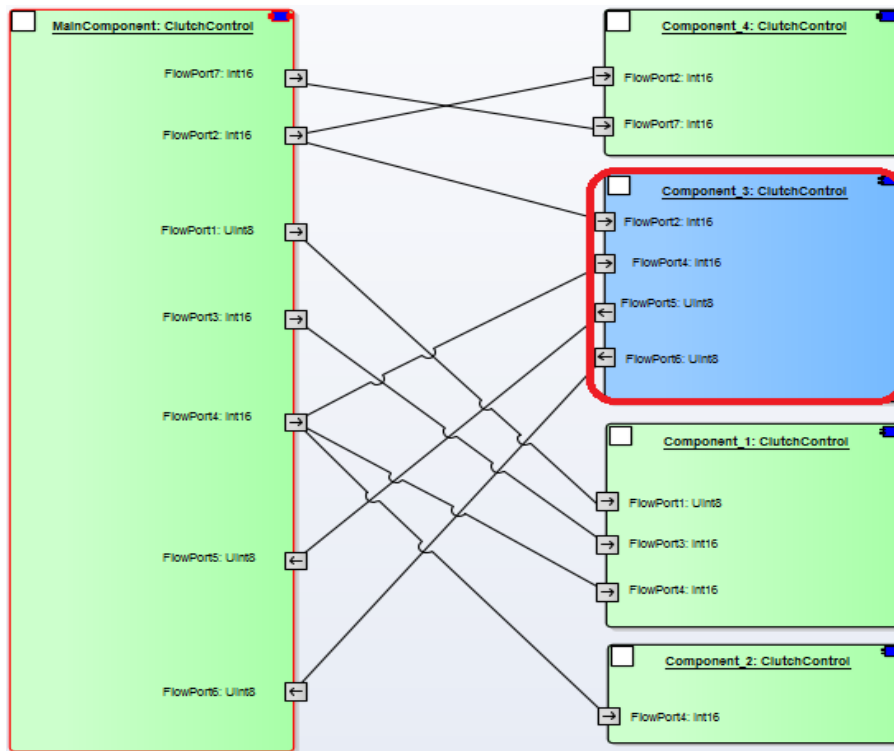


Fig. 15: The result obtained on the architecture following the setting of component C3

After performing the four major steps defined above, the steps III and IV will be repeated until all components and all related ports have been rearranged accordingly.

If neither phase III nor phase IV can be applied, the whole process of reducing the number of intersections on a partial matrix is resumed, excluding ports that are type 1 to 1.

A type 1 to 1 port means a port connected to a component that does not contain ports with connections to the other components.

The result obtained after the appropriate and integral application of the presented method is illustrated in fig. 16.

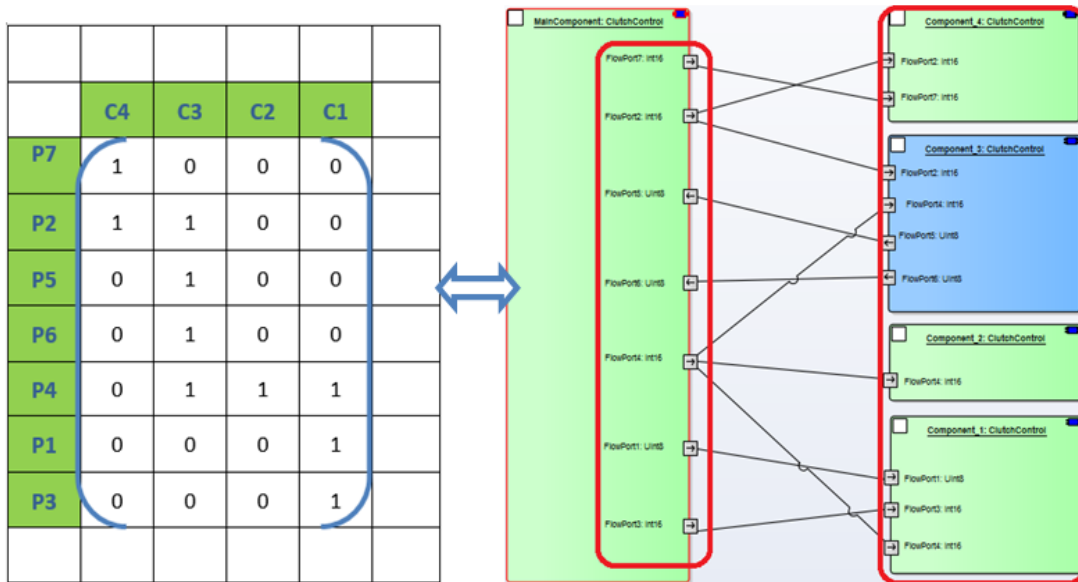


Fig. 16: Result obtained after application of the presented method

However, in order to complete the entire procedure of rearranging the elements of the software architecture, the ports inside the external components (fig. 17, fig. 18 and fig. 19) will be ordered based on the analysis of the ports in the main component, performed in the previous phase.

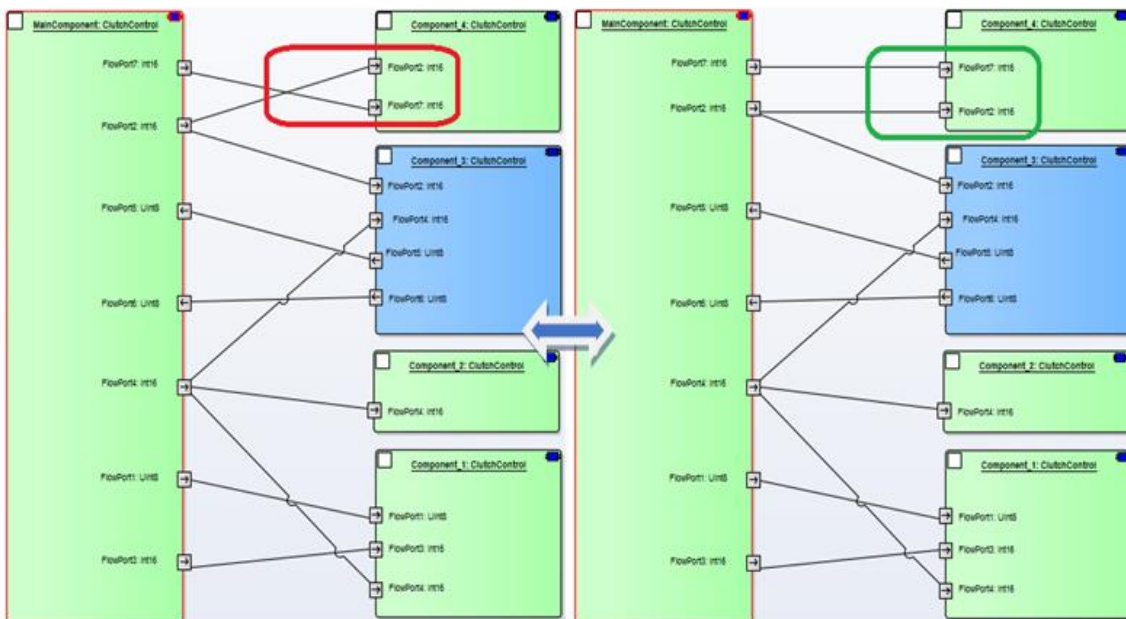


Fig. 17: Rearranging the ports of component C4

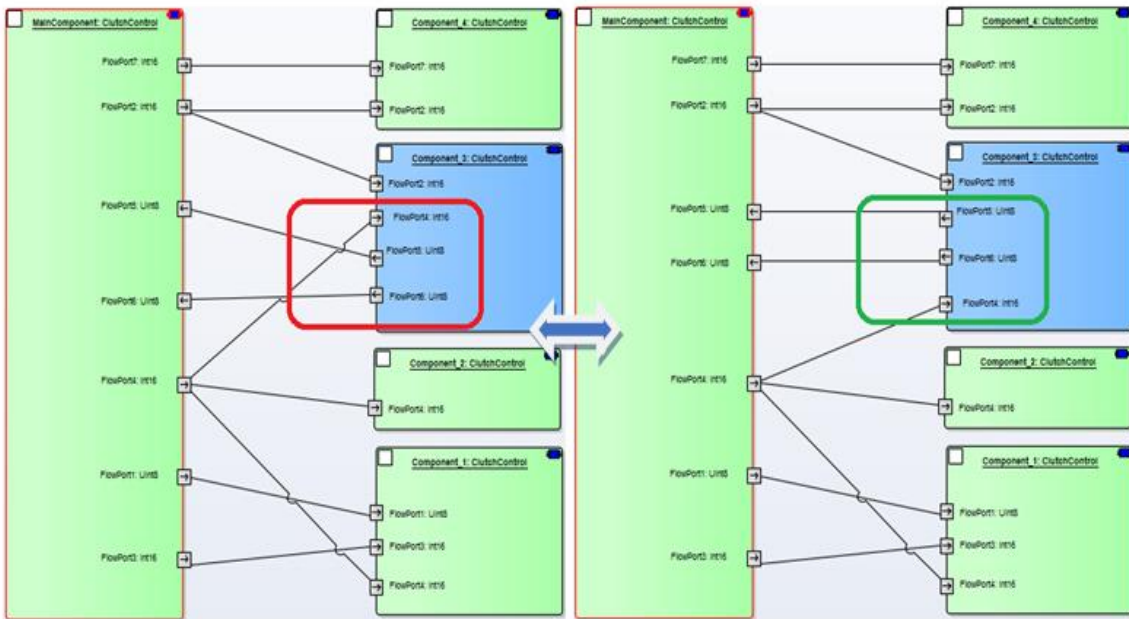


Fig. 18: Rearranging the ports of component C3

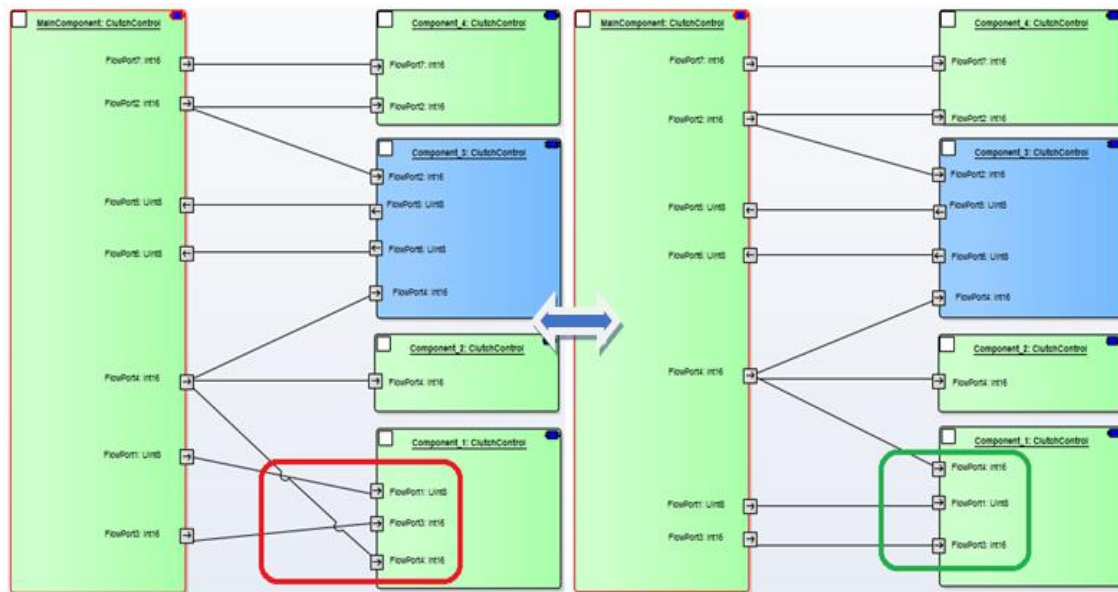


Fig. 19: Rearranging the ports of component C1

The final result, obtained by applying the procedures for reducing the number of intersections between the connections of the components, are illustrated below:



Fig. 20: Initial architectural diagram

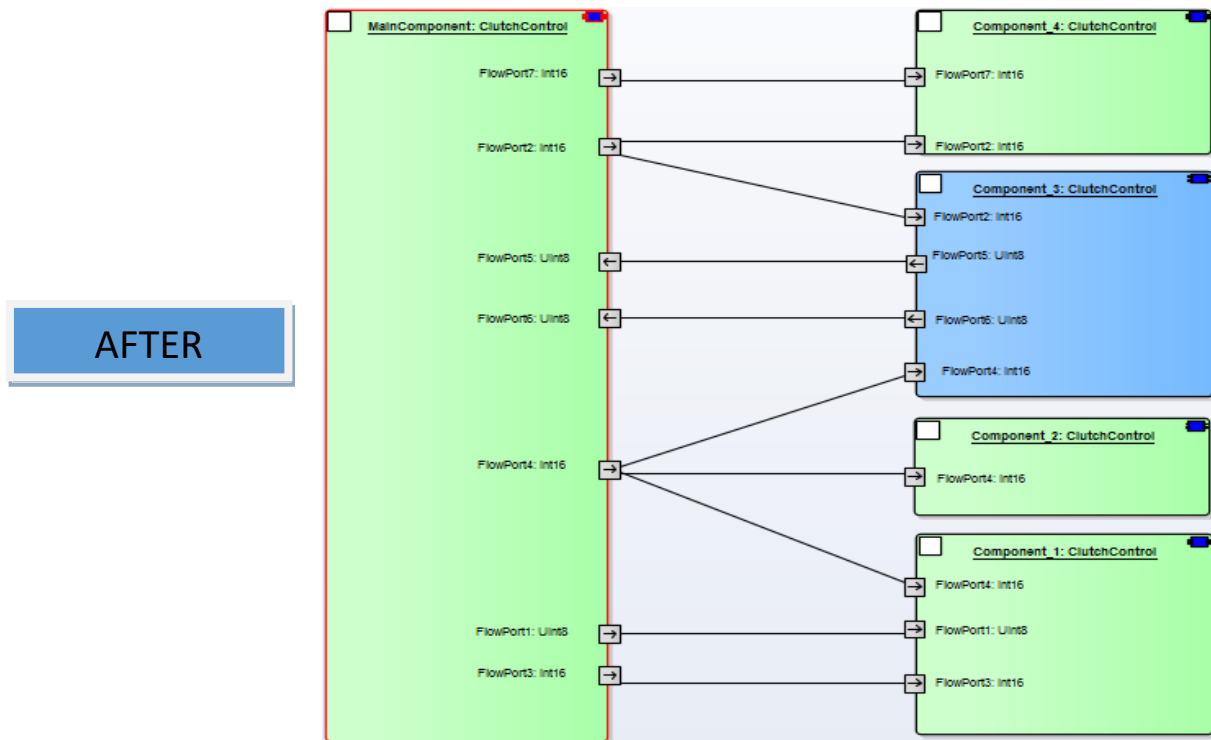


Fig. 21: Initial architectural diagram

4 The using of the application

Proper use of the Sorting of EA Diagrams application involves the following steps:

- Install the *InstalleASortingAddin* application (fig. 21), according to the instructions in the user manual attached to the paper.

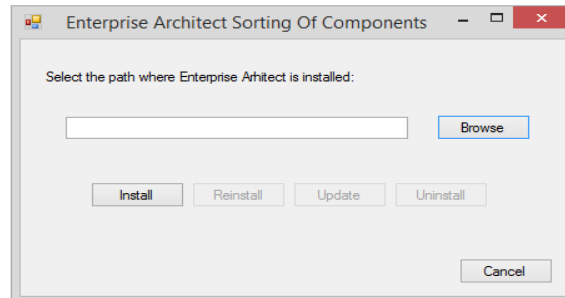


Fig. 22: Install the InstalleASortingAddin application

- Open the desired project architecture using Enterprise Architect, select the diagram or diagrams that need rearrangement, and run the Sorting of EA Diagrams application, added as an extension of EA, according to fig. 22. More significant details about performing this step correctly can be found in user manual attached to the paper.

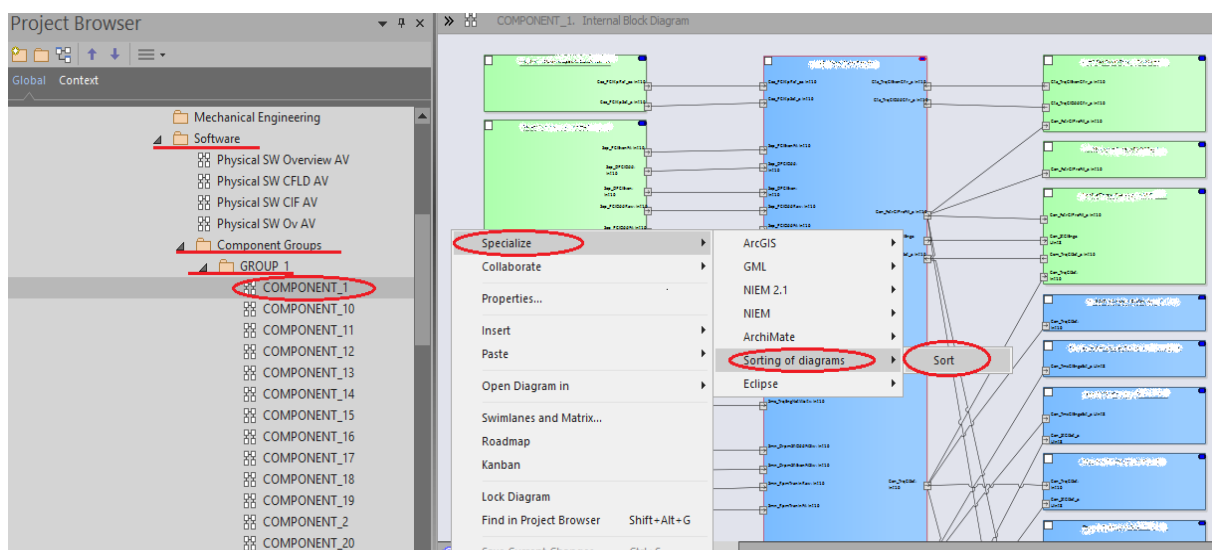
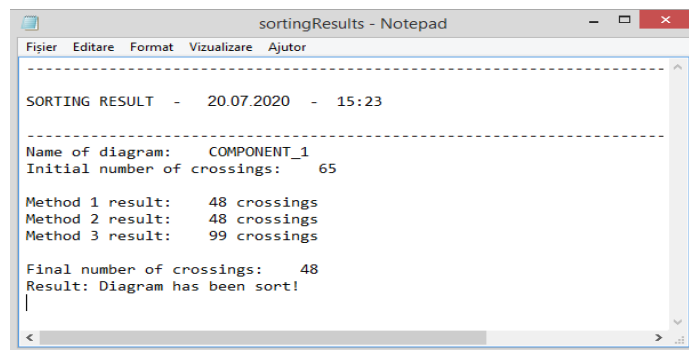


Fig. 23: View Sorting Of EA Diagrams application as EA extension

- The last step involves analyzing the results obtained, found in the file "sortingResults" (fig. 23), which will open automatically after closing the rearrangement process.



```

-----
SORTING RESULT - 20.07.2020 - 15:23
-----
Name of diagram: COMPONENT_1
Initial number of crossings: 65

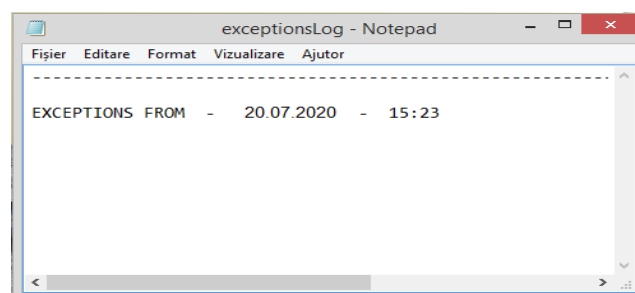
Method 1 result: 48 crossings
Method 2 result: 48 crossings
Method 3 result: 99 crossings

Final number of crossings: 48
Result: Diagram has been sort!

```

Fig. 24: The file with the results obtained

The application also generates a file for capturing exceptions thrown during running, called "exceptionsLog" (fig. 24), which will be automatically saved on the Desktop.



```

-----
EXCEPTIONS FROM - 20.07.2020 - 15:23

```

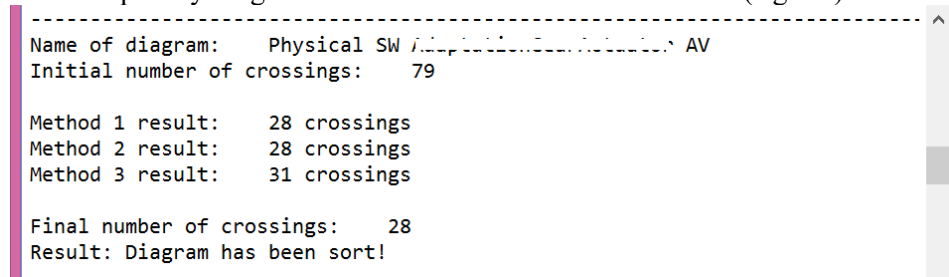
Fig. 25: The file with the exceptions that occurred during run

5 Results

After testing the Sorting of EA Diagrams application on a considerable number of architectural diagrams related to the various software architectures of the projects within the *Drivetrain* department of *Vitesco Technologies Engineering Romania SRL*, it was found that it works successfully in 99.24% of cases.

Carrying out a classification of the architectural diagrams, related to the projects within the department, according to their complexity and the number of intersections of the existing connections between components, it was found that Sorting of EA Diagrams can be applied for:

- Very low complexity diagrams: between 5 and 100 intersections (fig. 25)



```

-----
Name of diagram: Physical SW Adaptation... AV
Initial number of crossings: 79

Method 1 result: 28 crossings
Method 2 result: 28 crossings
Method 3 result: 31 crossings

Final number of crossings: 28
Result: Diagram has been sort!
-----

```

Fig. 26: Example of an architectural diagram of very low complexity

- Low complexity diagrams: between 100 and 500 intersections (fig. 26)

```

Name of diagram: Physical SW Communication Control AV
Initial number of crossings: 482

Method 1 result: 170 crossings
Method 2 result: 170 crossings
Method 3 result: 122 crossings

Final number of crossings: 122
Result: Diagram has been sort!

```

Fig. 27: Example of low complexity architectural diagram

- Medium complexity diagrams: between 500 and 1000 intersections (fig. 27)

```

Name of diagram: Physical SW Signal Management Control AV
Initial number of crossings: 543

Method 1 result: 339 crossings
Method 2 result: 339 crossings
Method 3 result: 669 crossings

Final number of crossings: 339
Result: Diagram has been sort!

```

Fig. 28: Example of architectural diagram of medium complexity

- High complexity diagrams: between 1000 and 8000 intersections (fig. 28)

```

Name of diagram: Physical SW Signal Management Control AV
Initial number of crossings: 7232

Method 1 result: 5623 crossings
Method 2 result: 5619 crossings
Method 3 result: 5105 crossings

Final number of crossings: 5105
Result: Diagram has been sort!

```

Fig. 29: Example of a highly complex architectural diagram

- Very high complexity diagrams: over 15000 intersections (fig. 29)

```

Name of diagram: Physical SW Signal Management Control AV
Initial number of crossings: 23821

Method 1 result: 15023 crossings
Method 2 result: 14534 crossings
Method 3 result: 14677 crossings

Final number of crossings: 14534
Result: Diagram has been sort!

```

Fig. 30: Example of an architectural diagram of very high complexity

However, there was a minimum number of diagrams that could not be improved because the initial number of intersections between the connections of their components could not be reduced. Fig. 30 shows the results of a diagram that could not be rearranged in a better version than the initial one.

```

-----
Name of diagram:   Physical SW ..... 1 AV
Initial number of crossings:   15

Method 1 result:   15 crossings
Method 2 result:   15 crossings
Method 3 result:   15 crossings

Final number of crossings:   15
Result: Number of crossings couldn't be minimized!
-----

```

Fig. 31: The results of a diagram that could not be improved

6 Related work

Regarding the comparison with other similar software products, no other general utility applications were found to meet the objectives of the Drivetrain department of Continental Powertrain Engineering SRL. It is possible, however, that there are such internal applications and specific to the needs of other companies.

7 Conclusion

As the complexity of software systems increases remarkably, being built from many components, the organization of the general system is a real problem. Because of this, the need to develop software architectures that allow a more efficient structuring and planning of the system, has become increasingly popular in the IT industry.

Due to the considerable amount of time and effort in manually creating highly complex software project architectures, within the *Drivetrain* department of *Vitesco Technologies Engineering Romania SRL*, a tool was developed for their automatic generation using Enterprise Architect. However, the problem with automatic generation is that sometimes the result is chaotic and almost impossible to control.

The purpose of this application is to bring a substantial improvement in the organization and structuring of the components of the software architectures of the projects, so that they can be analyzed and understood correctly. To achieve these objectives, it was necessary to find a way to automatically reduce the number of intersections between the connections of the architectural components and to rearrange the architectural diagrams so that there is a systematic logic in structuring the related components and their interactions.

After a long period of research and analysis of the various architectural diagrams of the projects within the department, it was found that due to their high complexity it is almost impossible to cover all existing cases by a single method of reduction. Finally, I managed to find a way to reduce the number of intersections between component connections in 99.24% of the analyzed diagrams. To achieve this percentage I developed and implemented four major reduction steps.

Also, the application rearranges the architectural diagrams in Enterprise Architect according to this method that obtained the best reduction, meaning the smallest number of intersections of the connections between the elements of the diagram.

One direction of developing the Sorting of EA Diagrams application could be to discover, develop and implement a new method to cover the minimum percentage of 0.76% architectural diagrams, for which a better version than the initial one could not be obtained, by this method developed so far.

Acknowledgement: This work benefits from funds given by *Vitesco Technologies Engineering Romania SRL*.

References

- [1] Veres L. Arhitectura software (I). [Internet]. 2017 Available from: <https://www.todaysoftmag.ro/article/1257/arhitectura-software-i>.
- [2] Systems S. Enterprise Architect Add-In Model. 1st ed. 2015-2016.
- [3] Schach SR. Object-Oriented and Classical Software Engineering. 8th ed. The McGrawHill Companies, Inc; 2011.
- [4] Farrell J. *Microsoft® Visual C#® 2010 An Introduction To Object-Oriented Programming*. 4th ed. Boston: Course Technology, Cengage Learning; 2011.
- [5] Perry DE, Wolf AL. Foundations for the Study of Software Architecture. Software Engineering Notes. 1992 oct.
- [6] Steinpichler D, Kargl H. Project Management with UML and Enterprise Architect. Vienna: SparxSystems Software GmbH.
- [7] Systems S. UML 2 Tutorial - Component Diagram. [Internet]. Available from: <https://sparxsystems.com/resources/tutorials/uml2/component-diagram.html>.
- [8] Kruchten P. Architectural Blueprints—The "4+1" View. IEEE Software; 1995.
- [9] McCarthy R, Halawi L. Foundations of EA - Historical Perspectives, 2001.

Ligia – Izabela CRĂCIUNESCU
Politehnica University Timișoara
Faculty of Automation and Computers
Timișoara
ROMANIA
E-mail: craciunescu.ligia@yahoo.com

Cashout

Alexandru Dancau, Paul-Robert Ceolca

Abstract

The purpose of this project is to empower each and every individual to become a better version of themselves. Nowadays, education has become a crucial element on the path to success. What lacks in many educational systems is the Financial Education, so this is where we come in. We designed and developed an application named „Cashout”, to help users improve their financial management skills. We want to develop the sense of self-awareness regarding expenses and financial decisions and we do this by challenging the user to manage a certain amount of money per month, proposed by us or set by the customer themselves. The application down expenses, challenge the user to give a rating on every acquisition, and generate useful reports. Moreover, the higher the amount of money is saved, the more recommendations our customer receives.

1 Introduction

This article responds to the need for a better way to manage the expense-making process. Nowadays every individual's life has become very busy, each having their own target. Everyday life has gained a lot of speed, and so have our actions. Most of the time people tend to make choices by instinct, not necessarily thinking them through. This might lead to wrong life decisions, wrong financial decisions, and the chain keeps going. By offering people a tool for tracking down their expenses based on their real-life importance, things have the potential to change. The “Savings” concept might sometimes be misunderstood. Saving often means choosing to buy one product which is not vital at a lower price, from a lower quality brand, whereas in reality it should refer to the concept of Making a Smart Financial Decision by not buying at all one unnecessary product.

Some people do track down their expenses by writing them down in a notebook, or somewhere else but their own mobile device/laptop. This is an old-fashioned manner, however it might still be viable in some cases. It proves to be ineffective, time consuming and wasteful over the long term. tracking your expenses should be acquired using modern technology.

There are a couple of applications which offer the possibility to manage your budget, a couple of such applications are: Expensify, Monefy, Mobills. All these applications offer the possibility to add expenses, to link them to your bank account. However, Expensify does not offer the feature to set a budget, but it does offer a report based on your expenses. Moreover, Expensify lacks a high-level security login system, only requiring email verification. In addition, Monefy also has the potential to be broken, because the application does not require any email verification or password. From our point of view, Mobills application is the most secure in comparison to its competitors.

Moreover, our application was built from scratch with the concept of interacting with the user in mind, and giving him the possibility to rate his expenses, and also manage his personal data used to register initially.

We first wanted to develop the Desktop Application to have an example for the Mobile Application. The rest of the paper is organized as follows. In Section 2 we present the Application Design.

2 Application design

Firstly, our application is running on a MySQL Database, which we have designed from scratch. The application stores user's data into the database, and retrieves it according to the option chosen by the user. Secondly, the interface was created using the Java Swing Interface Builder found in Eclipse IDE. Regarding the appearance of the application, we chose a light theme for background. The interface is designed to be as simple as possible. The different buttons open up multiple pop-up windows to simplify the application.

To conclude, in terms of application design we could say that it runs on an MySQL database, in terms of colours used, we chose them so as not to disturb the user while using the application.

2.1 Database Design

The Figure1 represents our application's database, which consists of 8 tables.

1. mainusers table - stores the information used when following the registration process.
2. expenses table - stores all the acquisition entered by user.
3. finance table - stores the information regarding the current and past budgets.
4. currgoals table - stores the information regarding the current saving goals, if any.
5. pastgoals table - stores the information regarding the past saving goals.
6. savings table - stores the information regarding the savings of the user.
7. session table - stores the information about every new login.
8. reasons table - stores the reason which the user mentioned when deleting their account.

<input type="checkbox"/>	currgoals	☆	Browse	Structure	Search	Insert	Empty	Drop
<input type="checkbox"/>	expenses	☆	Browse	Structure	Search	Insert	Empty	Drop
<input type="checkbox"/>	finance	☆	Browse	Structure	Search	Insert	Empty	Drop
<input type="checkbox"/>	mainusers	☆	Browse	Structure	Search	Insert	Empty	Drop
<input type="checkbox"/>	pastgoals	☆	Browse	Structure	Search	Insert	Empty	Drop
<input type="checkbox"/>	reasons	☆	Browse	Structure	Search	Insert	Empty	Drop
<input type="checkbox"/>	savings	☆	Browse	Structure	Search	Insert	Empty	Drop
<input type="checkbox"/>	session	☆	Browse	Structure	Search	Insert	Empty	Drop

Figure 1: Database Structure

These tables store all the information about the user and everything that has happened in the application. It is stored all the data used in the registration process, financial data, which means the budget, amount to save, current available amount, all the acquisitions, and all data about every particular acquisition. The only independent table is called "reasons", and is not related in any way to the user itself. Data can only be inserted inside when the user is about to delete their account and all the relevant information about them. The Foreign Key across the tables is derived from the PID field in the mainusers table, which contains a unique 4-digit number for every user generated from the application during the registration process. We chose this type of structure for the database in order to have data stored as accurately as possible.

2.2 Application Functionalities

Our application has the following functionalities.

1. Login with email.
2. Password reset with a code received on email.
3. Delete account.
4. Setting a budget.

5. Setting a savings goal.
6. Extending the budget.
7. Adding acquisitions and sorting them by category.
8. Rating every acquisition.
9. Accessing finished acquisitions by day, month, or from the beginning.
10. Live feedback consisting of the average of current session acquisitions.

The first one, and the most important feature of our application is the login system which can be done with email and password. If the password got forgotten or lost can be reset just with email, where you get a code, and with that code you can enter the new password. If you don't like, or you want to get rid of your account, you can easily delete your account by entering your email and the unique code received by email. Talking about the internal functionalities of application, we implemented many facilities like setting a monthly budget. On this budget you can add how much do you want to save or what expenses you have done. Another internal important feature is the budget extension, with this you can add more money to your monthly budget. Talking about the expenses or acquisition you have done lately, you can see them ordered by day, month or from the first time you joined our application. Moreover you can rate your expenses, giving them grades at importance and the importance of you expenses will be displayed on the main screen.

The user has to login in the application with his email and password. If the user forgets his password, he has to enter his email and a validation code will be send to the address. If the user no longer wishes to use the application, there is the possibility of deleting your account. The user has the possibility to set a savings goal derived from the total budget. The user has the possibility to set a monthly budget. If something unexpected has happened, or the limit has been reached, the user can choose to increase the budget. The user has the possibility to add expenses and pick a certain category for every instance. The user has the possibility to view his expenses from the current day, the current month or since first registering. Every acquisition can be rated with a mark ranging from 0 – 10. The homescreen is constantly updated and the user can see his spending performance.

2.3 Technologies

We have chosen the Java programming language, combined with MySQL programming language to manage all the data coming in. The IDE which has been used is Eclipse.

For connecting the .java file to the database, there have been primarily used prepared statements, this being done mostly for security reasons. These statements were used for:

- inserting data into the database.
- extracting data from the database.

Figure 2, presents an example of such statements. The goal of the statements from Figure 2 is to validate the login of one particular user by extracting from the mainusers table the username where the emails and passwords match in a variable. If the variable is empty, no user has been found and the login process fails. The setString method is used for replacing the question marks with the input values.

```
logst = conn.prepareStatement("SELECT uname FROM mainusers WHERE email = ? AND password = ?");
logst.setString(1, String.valueOf(emailF_2.getText()));
```

Figure 2: Login example statement

2.4 Security

During development of the application, security has been a priority. Therefore, we have come up with a way of encrypting data. We have used two different encryption algorithms: SHA512 and MD5.

The steps for assuring the security of our application are:

1. The user enters their password.
2. His password is converted to an MD5 string.

3. The MD5 string is reversed. (eg: passConverted -> detrevnoCsapp).
4. The reversed password is then converted into an SHA512 string;
5. The password is then processed depending on the user action

In the figure 3 is the code which is able to do the encryption process.

```

encrypter enpass = new encrypter();
proceed enpass1 = new proceed(String.valueOf(pass1F.getPassword()));
String enPass = enpass1.convert();

```

Figure 3: Login example statement

3 Application interface

When opening the application for the first time, you will be sent to the welcome screen, which has multiple options for users. He can either login or sign-up. The default screen is the sign-in screen- Figure 4 illustrates the login screen.

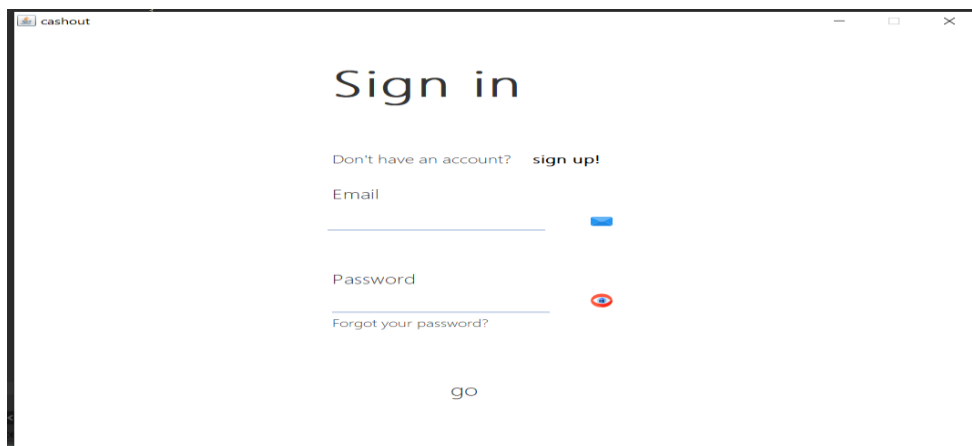


Figure 4: Login screen

In the Sign in screen the user has the possibility to use his credentials in order to log in.

The credentials for logging in consist of:

1. User's email.
2. User's password.

If the user press the “sign up” button and does not have an account, he will be sent to the sign-up panel. If the user has forgot-his password, he has to press the “Forgot your password?” button.

Figure 5 presents the forgotten password screen.

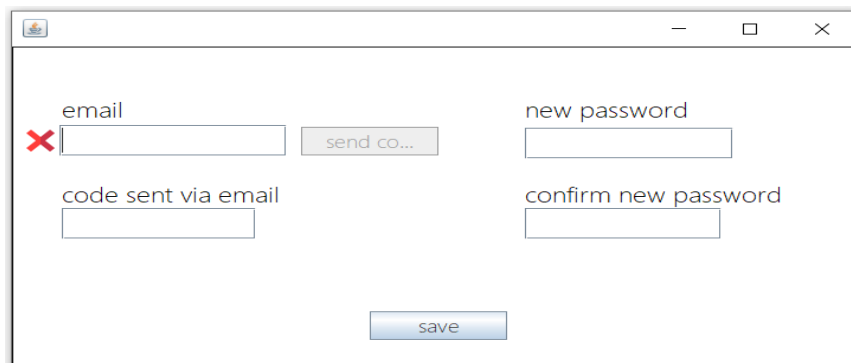


Figure 5: Forgotten password menu

One key point to take into account is the fact that Cashout requires validation via email with a security code. This is being done by generating a random 5-digit number and passing it to the email sender, along with uploading it encrypted in the same manner to the database.

After logging in, assuming that all input data is correct, the user will be able to see the home screen. This one consists of:

1. Left menu:
 - the Cashout logo.
 - the Add money button, allowing the user to set their monthly budget.
 - the History button, which opens up the History panel containing all the past expenses initiated and finished.
 - the Add Expense button, which opens up the panel for adding a new expense.
 - the settings button, which opens up the Settings panel for the user.

2. Content panel, which includes:
 - the budget available
 - the number of remaining days until the end of the month
 - one active search bar, which the user can use to find a particular expense
 - the Add new expense button, which opens up the Adding Expense panel
 - the current performance of the user. If it has not initiated any expense, the current performance is set by default 0.

Following the process of everyday life, Cashout is structured on multiple sessions. A session is represented by a month. Therefore, the year is structured in 12 sessions.

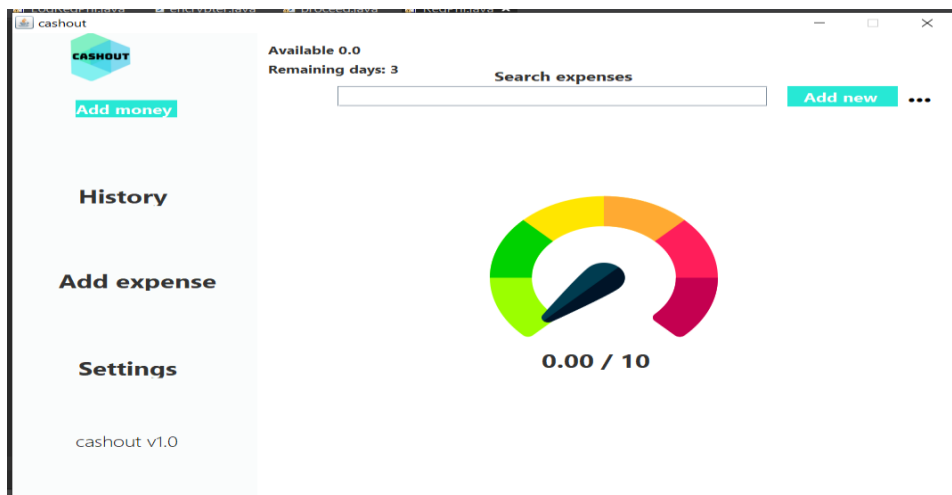


Figure 6: Home screen with the “Add money” button

The user is being constantly updated with the current number of days left in the session he is running on. Adding balance is also fairly easy. Everything the user has to do is click the “add money” button, which is visible in top-left side of Figure 6.

When the user hits the Add New button, a pop-up window will be presented. The Figure 7 presents this window.

Going back to the adding an expense process, everything the user has to do is press the Add Expense button located in the content panel, top right side. This opens up the Adding Expense panel.

available: 4322 safe ammount: 5678

item name: bread

item price: 12 0.2... %

item type: food

importance rating: 8

current performance: 0 / 10

go

Figure 7: Adding expense menu

In the Adding Expense window, presented in Figure 7, the user can find the available amount, the “safe” amount, meaning how much the user is willing to save, and the expense details consisting of:

1. The item name
2. The item price, along with the percent of the available amount it represents.
3. The item type, being:
 - food
 - non-food
 - combined
 - other (if the user is not willing to specify)
 - manually introduced

After the user hits the go button, all the data are processed and updated by the server in the backend. The panel closes, and the user is shown again the main panel, with the updated performance rating and budget.

The home screen data will be updated once the acquisition is confirmed.

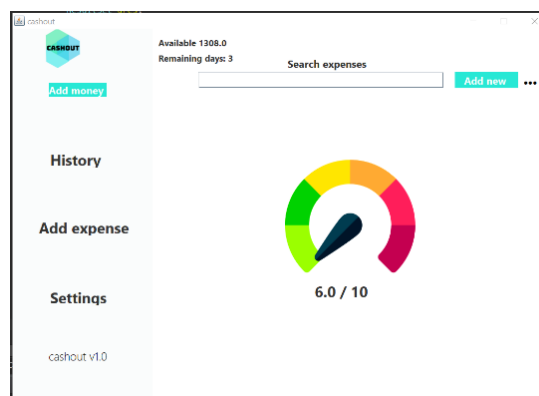


Figure 8: Updated home screen

The current performance is being constantly updated by making an average based on expenses initiated in the current session.

When the current session has ended, the user is presented an ending screen, and will be automatically redirected to the next session. Figure 9 presents the session ending screen.

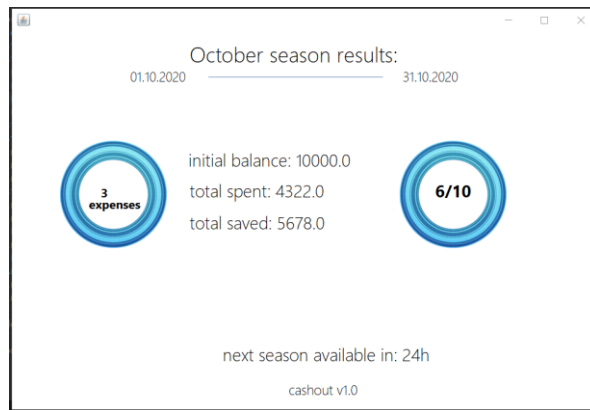


Figure 9: Session ending screen

4 Technical Details

Multiple Java classes have been developed for making it all possible. There are two main components:

1. The application itself, built using the Swing Builder found in Eclipse IDE.
2. The data, which is extracted from the MySQL database.

We have designed 37 Java classes. Inheritance relationships are used with JComponents. We have also used the composition and association relationships. For example, one such class represents the Account Deletion Frame. Here, the class inherits the JFrame component, and associates several other jComponents, such as: JPanel, JPasswordField.

There are two steps which make possible the communication between user and the application: rendering the application and filling in all the relevant data. We have multiple Java objects, each representing a smaller component, for example a JPanel. First of all, the component is being rendered, and then the content is set to the specified attribute from the database. For example: the application would render one JTextField called unameField, and the content would be set using the method .setText(username). This is the process which the application repeats for every single component, such as a button, field, or table.

5 Future Developments and Conclusions

The app Cashout, presented in this article implements a few useful functionalities for allowing the users to manage their money, such as: rating acquisitions, viewing past acquisitions, receiving real time performance based on the rating given, which might encourage the user to make better financial decisions. Our application is simple, free and user-friendly.

In what regards future development of our application, we intend to add an Automatic Rating Algorithm for giving every individual expense a proper “grade”. We are currently working on the Recommended Investments algorithms, which is going to launch the user investment ideas as he manages to save more and more money. Another direction of development is oriented to the implementation of a fully functional mobile application linked to the same cloud-hosted database as the Desktop application. This means migrating the database to the cloud of removing the localhost component.

The market always welcomes more and more services, and Cashout is willing to become such a service. There is a lot of potential. Our application will be extended in more areas of interest, for example time management, an daily planner based on user’s schedule, which could also be linked to the user’s Google Calendar.

Acknowledgement. We thank Prof. Dr. Dana Simian for useful suggestions and for supervising this article.

References

- [1] Linda Yueh, *What Would the Great Economists Do?: How Twelve Brilliant Minds Would Solve Today's Biggest Problems*, Picador, 2018.
- [2] MD5 algorithm in java, <https://www.geeksforgeeks.org/md5-hash-in-java/T>, accessed 01/10/2020.
- [3] MD5 algorithm in java with example, <https://howtodoinjava.com/java/java-security/how-to-generate-secure-password-hash-md5-sha-pbkdf2-bcrypt-examples/>, accessed 01/10/2020.
- [4] Java user authentication with google engine, <https://cloud.google.com/appengine/docs/standard/java/users?csw=1>, accessed 14/09/2020.
- [5] Expense tracking applications, <https://www.businessnewsdaily.com/6233-expense-tracking-apps-smartphone.html>, accessed 18/09/2020.
- [6] Expense tracking applications available on google play, <https://play.google.com/store/search?q=expense%20tracker&c=apps>, accessed 23/09/2020.

Alexandru DANCAU
Lucian Blaga University of Sibiu, Faculty of Science
Computer Science Department
Doctor Ion Rațiu Street 5-7, Sibiu 550012
ROMANIA
E-mail: alexandru.dancau@ulbsibiu.ro

Paul-Robert CEOLCA
Lucian Blaga University of Sibiu, Faculty of
Science
Computer Science Department
Doctor Ion Rațiu Street 5-7, Sibiu 550012
ROMANIA
E-mail: paul.ceolca@ulbsibiu.ro

3D Head Reconstruction via Volumetric Regression

Andreea Dogaru

Abstract

This paper describes a novel method for automatic 3D head model reconstruction from just a few images featuring the human subject's head. In contrast to alternative techniques, which require numerous images captured under specific controlled conditions for a single head model, this approach can reconstruct accurate representations even from a single image. The proposed method can be used on unconstrained portrait images, in both single, and multi-view setups. Firstly, a Convolutional Neural Network with an encoder-decoder architecture regresses a coarse volumetric representation for each input image. Afterwards, a facial landmark oriented fusion process combines the raw volumes into a fused 3D volume of the entire head. Finally, the surface of the 3D head model is extracted from the fused volume.

1 Introduction

3D reconstruction is a problem of great interest to researchers in the field of computer vision. The need for fast and autonomous generation of high-quality three-dimensional models of the human head is felt in many areas like animation design, hair styling, and virtual reality avatar creation [10]. Obtaining a three-dimensional digital object is usually done through the process of 3D modeling of the object's surface by a specialized artist. The complexity of the process and the difficulty of capturing accurate details in representations prevent the widespread use of 3D modeling in the reproduction of real objects. As an alternative, the reconstruction is performed using 3D scanning methods that, depending on their technology, require long processing time, tedious capture procedures, or high costs.

Continuous advances in deep learning proved effective in uncovering new solutions to long-standing problems. 3D reconstruction is one of the numerous computer vision tasks that benefit from these discoveries. The proposed solution is based on a Convolutional Neural Network, which learns in a supervised fashion a mapping from portrait focused images to corresponding 3D head volumes. The chosen representation of the head geometry facilitates the adaptation of existent deep learning architectures, which prove to also be a good match for this task. A significant part of this work is the fusion process. From different viewing points, distinct parts of the head are visible. Considering that the reconstruction quality of the visible parts is higher than that of the hidden parts, the multi-view fusion process merges multiple coarse representations into a comprehensive one. In summary, the contributions of this work are: showing that the 3D head geometry can be reconstructed through volumetric regression, developing a process that combines the geometries of multiple partial reconstructions, along with a method for coloring the vertices of the full 3D head model from multiple source images.

The remaining of the paper is organised as follows. Section 2 presents an overview of related work. In section 3 is described the deep learning-based method used for single-view 3D head reconstruction. Section 4 introduces the fusion process which enables multi-view reconstruction. In section 5 experimental results are presented. Finally, conclusions and ideas for future developments are outlined in section 6.

2 Related Work

Numerous works approach certain aspects of the targeted topic, the reconstruction of facial 3D models has remarkable results, but the problem of reconstructing the entire head is insufficiently addressed. This section presents an overview of relevant related works in 3D face and head reconstruction. Also, because the proposed method employs a facial landmark detection algorithm for the alignment step in the multi-view fusion process, as described in subsection 4.2, some approaches of this task are also briefly reviewed.

2.1 3D Face Reconstruction

Due to the fundamental role played by the human face in numerous applications, 3D face reconstruction has been intensively researched, with multiple solutions being developed. The main categories into which these can be divided are reconstructions that use 3D scanners, methods that employ 3D Morphable Models (3DMMs), and approaches that are based on deep learning. Since the introduction of 3DMM representation by Blanz and Vetter [3], methods based on such a model have been frequently used in making three-dimensional monocular reconstructions of the human face. A standard 3DMM is obtained through principal component analysis from a paired example set of 2D face images and associated 3D face models. A known challenge of 3DMMs is obtaining the right balance between the low-dimensional space of the parametric model and the level of detail the model is capable of representing [8].

Directly regressing the facial geometry for an input image can be achieved through recent deep learning methods. Feng et al. introduced for this problem the UV position maps [9]. This representation illustrates the entire 3D facial structure in parametrized UV coordinates. The proposed solution adopts a Convolutional Neural Network to learn a mapping from facial images to the associated UV position maps. Another approach, which does not require parametrized 3D facial models, is the one presented by Aaron et al. in [13]. The method works for arbitrary positions and facial expressions and uses a CNN to regress a volumetric representation of the complete 3D facial geometry from a single 2D image.

2.2 3D Head Reconstruction

3D head reconstruction is a much more challenging task than 3D face reconstruction as it needs to additionally represent the subject's hair, ears, and neck. Recent progress made in statistical modeling led to the creation of a complete 3DMM of a human head [22, 23] that models the shape of the head, the facial features, and the ears. This 3DMM achieves realistic results for single image reconstruction, but lacks hair representation and is still limited in geometry to the space of the composing 3DMMs. Other notable methods that employ 3DMMs for 3D head reconstruction are [6, 11, 12, 18, 26]. These approaches require as input either a video of the subject [12, 18], a set of sparsely captured images in specific positions [6], RGBD data from a special camera [26], or a single image [11]. Apart from the 3DMM used to represent the facial features, [6, 18] use an additional one to represent the hair. In contrast, the proposed method consistently regresses the entire geometry of the head, including, in addition to the region of the face, representations of hair, neck, and ears.

2.3 Facial Landmarks Detection

Facial landmarks detection refers to the task of automatically identifying a set of key points that mark the main components of the human face and its contour. The information carried by these points is fundamental for many face analysis applications. The difficulty of this problem comes from the variety present in face centered images, in terms of different facial features, expressions or self-occlusions and induced by the image taking conditions i.e. illumination. The detected facial landmarks are either 2D, when they are located in the image plane, or 3D, when each point's depth is also estimated. The performance of the methods that detect 2D facial markers is considerably reduced for images that present occlusions. A robust localisation in this case requires 3D information which can be achieved either by fitting a 3DMM [9, 15, 29] or by using an artificial neural network to infer the third coordinate of the 2D landmarks [4, 5, 27].

3 Single-view 3D head reconstruction

Motivated by the favorable results for monocular face reconstruction [13], the selected approach for single-view 3D head reconstruction task is volumetric regression. The principal characteristic of this method is the ability to directly regress the head geometry, bypassing the construction and fitting procedures of a 3DMM.

3.1 Dataset

Because the chosen deep learning-based method requires 3D supervision in the training phase, the dataset used drastically impacts the robustness of the solution. As one of the end goals of the present work is the use on unconstrained images, the dataset needs to provide a large amount of variation both at the subject level (age, gender, pose), as well as in terms of image taking conditions (background, illumination, etc.). Unfortunately, a paired dataset that contains such images and associated 3D head models is not publicly available, and cannot be directly obtained through existing methods. Instead, in this work, a synthetic dataset is used. I generated this dataset through rendering, starting from a generous subset of 110 high-quality 3D head models¹. For each one of these models, approximately 120 images were rendered, varying imaging factors like illumination, camera position relative to the model, and background. Some illustrative images are presented in Figure 1.

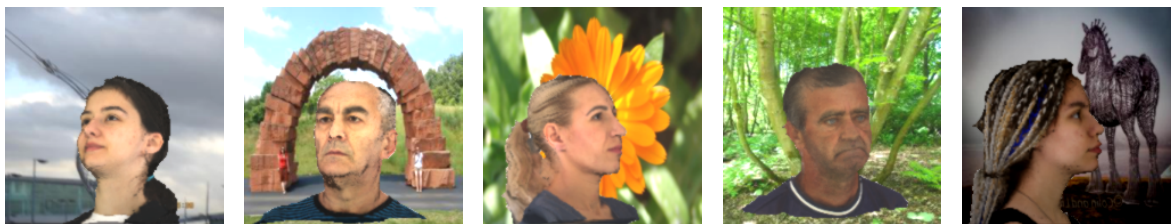


Figure 1: Example images extracted from the synthetic dataset

The 3D models were transformed according to the voxelization process described in [13], to obtain 3D volumes aligned with the corresponding images. A volume is composed of elements named voxels and represents a discretization of a 3D mesh. If a voxel is enclosed by the surface of the mesh, it has a value of 1 and 0 otherwise. The same dimensions of $192 \times 192 \times 200$ for the volumes and $192 \times 192 \times 3$ for the images were used. The resulted data was split into train and test subsets: the paired images and volumes from 100 subjects form the training data, while the remaining pairs are used to evaluate the accuracy of the method.

3.2 Architecture

Considering the volumetric data representation, in concordance to the previous works of Aaron et al. [13, 14], the task of 3D geometry reconstruction is simplified to a 3D binary volume segmentation problem. A Convolutional Neural Network is being used to learn the mapping from an input image illustrating a human head to its corresponding voxelized 3D model.

Since the head geometry is represented as a three-dimensional voxel grid, existent deep learning architectures designed for 2D image analysis can be easily extended. A direct transformation of the architectures for processing 3D voxel tensors involves replacing the convolutional 2D operations with their 3D equivalents, which significantly increases the computational burden. Because the network's output can be understood as a discrete set of slices that segment the 3D space, the networks are adapted for volumetric regression by increasing the number of filters in the convolutional layers.

In this context, I explored several encoder-decoder CNN architectures that proved good results in image analysis tasks. The first one is VRN [13], based on Stacked Hourglass Networks [20] introduced

¹The high-quality 3D models used are the property of Xperi Corporation and were kindly provided for this work.

for the task of human pose estimation. This network consists of two stacked “hourglass” modules and uses skip-connections and residual learning. From the family of medical image segmentation networks, are employed the original U-Net [24], and its later enhanced versions, R2U-Net [1] that swaps the convolutional blocks with recurrent residual convolutional units, Att U-Net [21] which adds attention gates at each level in the decoder and UNet++ [28] that improves the encoder-decoder connection with a series of nested dense convolutional blocks.

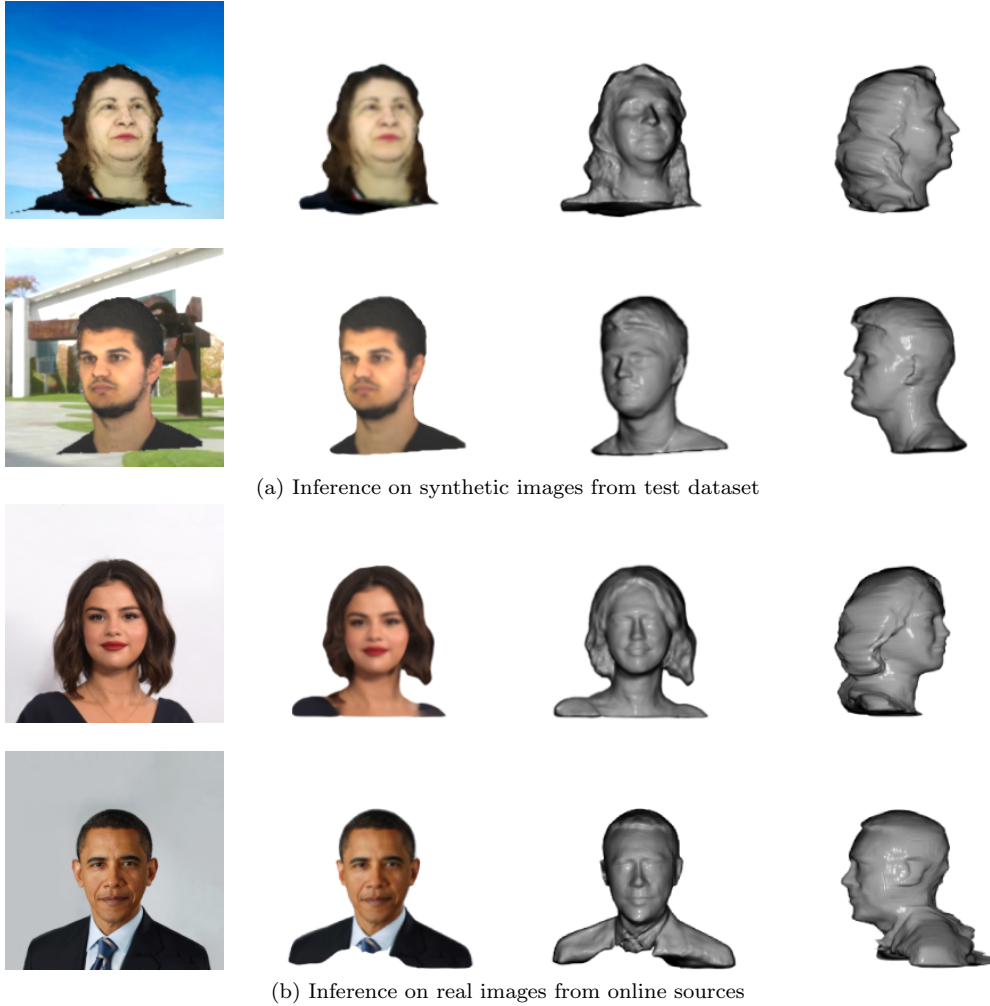


Figure 2: Illustrative single-view 3D head reconstructions. For each subject are illustrated in order: the input image, the reconstructed 3D head model with color and without color from two different viewpoints.

3.3 Training details

The network receives as input an RGB image and directly regresses the corresponding 3D voxel grid. All the architectures were trained in a supervised manner, having the objective of minimizing the binary cross entropy loss function:

$$\mathcal{L} = -\frac{1}{N} \sum_{w,h,d} \left[V^{whd} \log \hat{V}^{whd} + (1 - V^{whd}) \log (1 - \hat{V}^{whd}) \right] \quad (1)$$

where N represents the number of voxels in the ground truth volume, \hat{V}^{whd} is a voxel from the output 3D grid and $V^{whd} \in \{0, 1\}$ is the corresponding ground truth voxel. Reconstructing the entire geometry of the human head from a single image poses difficulty because a considerable part of the object to be

reconstructed is not visible. The accurate reconstruction of this part is an ill-posed problem and not the main goal of the network. Therefore, the loss function is adjusted accordingly:

$$\mathcal{L}_{full} = \lambda \cdot \mathcal{L}_{visible} + \mathcal{L}_{hidden} \quad (2)$$

where λ is a hyperparameter that balances the weight of the error over the visible voxels to that over the hidden voxels; $\mathcal{L}_{visible}$ and \mathcal{L}_{hidden} are computed with binary cross entropy loss function as defined in equation 1 on the corresponding parts of the volume.

The architectures were trained for 60 epochs using an Adam optimizer with a learning rate of 0.0002 and $(\beta_1, \beta_2) = (0.5, 0.999)$. The cubic storage capacity imposed by the volumetric approach limits the batch size to 4 on an NVIDIA RTX 2080 Ti GPU.

4 Multi-view Fusion Process

The complexity of the human head geometry makes it impossible to entirely and accurately reconstruct a representative 3D model from just a single RGB image. This section presents the multi-view fusion process that is introduced to reconstruct complete 3D head models from multiple images.

4.1 Workflow

After following the training process described in subsection 3.3, the network is able to generate, for an input image, a coarse 3D voxel grid representation of the 3D head. As previously stated, a single image is not a sufficient input for accurately reconstructing a complete representation of the head, several viewing points being required. When multiple images are available, the single-view reconstructions are merged together to achieve a global representation of the head.

For an input image set, $\mathcal{I} = \{I_k | k = 1, \dots, n\}$, the following steps describe the multi-view reconstruction process:

1. generate a set of corresponding head volumes $\mathcal{V} = \{\widehat{V}_k | k = 1, \dots, n\}$, where \widehat{V}_k is being directly regressed by a pretrained network from the image I_k
2. merge partial representations into a global one which aggregates individual reconstructed features
3. extract the iso-surface triangular mesh of the resulted volumetric 3D data
4. color the vertices of the complete 3D head model.

4.2 Volume fusion

Because the network’s output is aligned with the input image, combining multiple such volumes requires their alignment to the same reference position. Given that the images illustrate a subject’s head, the visibility of a portion of the face, sufficient for a facial key points detector, may be assumed. To this end, I employed an off-the-shelf 3D facial landmarks detector, 3DFAN [5], built on the Stacked Hourglass Network [20] architecture.

For each input image, I_k , a set of three-dimensional facial key points, $\mathcal{P}_k = \{P_i \in \mathbb{R}^3 | i = 1, \dots, 68\}$, is detected². One of them, P_{ref} , is considered as having a reference position, and for the others, the T_k optimal alignment transformation is computed. T_k represents the transformation that minimizes the sum of the distances between corresponding points in the sets \mathcal{P}_k and P_{ref} . T_k is determined by composing the following transformations: scaling \mathcal{P}_k according to the inter-ocular distance of P_{ref} , optimal rotation computed with Kabsch algorithm [16], and the translation of the centroid of \mathcal{P}_k into the centroid of the reference point set P_{ref} .

Each transformation, T_k , is applied to the corresponding volume. An additional step to further improve this approximate alignment is made. From each head volume, the outer iso-surface³ is extracted.

²Most facial landmarks detection algorithms identify a set of 68 key points; less common amounts of regressed points are 21 and 194 [25]

³Iso-surface extraction is described in subsection 4.3

By applying Iterative Closest Point [2] registration algorithm to the resulted meshes, a second transformation is identified for each partial representation. Through these transformations, the global alignment is enhanced, taking into consideration the entire head geometry, in addition to the facial landmarks.

Once the volumes are aligned, they can be combined by averaging the values of the voxels in corresponding positions in each volume. Let \tilde{V}^{whd} denote a voxel in the fused volume, its value is computed as follows:

$$\tilde{V}^{whd} = \frac{1}{n} \sum_{k=1}^n \hat{V}_k^{whd} \quad (3)$$

4.3 Mesh extraction

The final mesh of the head is obtained by extracting the iso-surface from the resulted fused volume. An iso-surface is a triangular mesh that represents the wrapping surface of all voxels with values larger than a specified threshold named iso-value. For this step, the improved Marching Cubes algorithm [17] is employed.

For visualisation purposes, the color for the vertices of the final 3D head model can be retrieved from the input images. When a single image is available, the color of the vertices is obtained by interpolating the colors of the neighboring pixels for their projected coordinates. Obviously, the colors obtained for the reconstructed parts of the head that are not visible in the image are not representative, and for them, an arbitrary color is assigned.

In the case of multi-view reconstruction, the alignment with the original images is lost in the combination process, so the vertices colors cannot be retrieved through direct projection. Instead, the colors are obtained starting from the initial output volumes. Firstly, their iso-surfaces are extracted, then the hidden parts of the meshes are removed using ray-casting. The remaining points are colored from the associated images as described above. All the points are aligned with the previously identified transformations for the volumes and combined into a global point cloud. The color of each vertex in the final head model is obtained by interpolating the colors of the neighboring points in the global colored point cloud.

5 Results

To measure the reconstruction quality of the different architectures explored for the task of volumetric regression, three established metrics are considered: voxel Intersection-over-Union [7], normalised Chamfer-L2 distance and Normal-Consistency score that were presented in the supplementary material of [19]. The Chamfer-L2 distance is normalised by the diagonal of the minimum bounding box of the ground truth 3D model. The first metric is evaluated between the network output and the voxelized ground truth model, while the other two target the reconstructed surfaces from the output volumes. Numeric results are reported in Table 1. Several visual results are illustrated in Figure 2 for single-view reconstruction and in Figure 3 for multi-view reconstruction.

Article Title	IoU \uparrow	Chamfer-L2 \downarrow	Normal-Consistency \uparrow
VRN	0.835	0.011	0.832
U-Net	0.832	0.011	0.856
Att U-Net	0.832	0.011	0.854
R2U-Net	0.834	0.011	0.839
UNet++	0.834	0.011	0.859

Table 1: Single view 3D head reconstruction results on the test split. The threshold used for evaluating Intersection-over-Union [7] is 0.5. The iso-value used for extracting the surfaces is 0.5. The Chamfer-L2 distance and the Normal-Consistency score [19] were computed on a set of 100000 sampled points from the surface of the 3D models. For IoU and Normal-Consistency higher is better, for Chamfer-L2 lower is better.

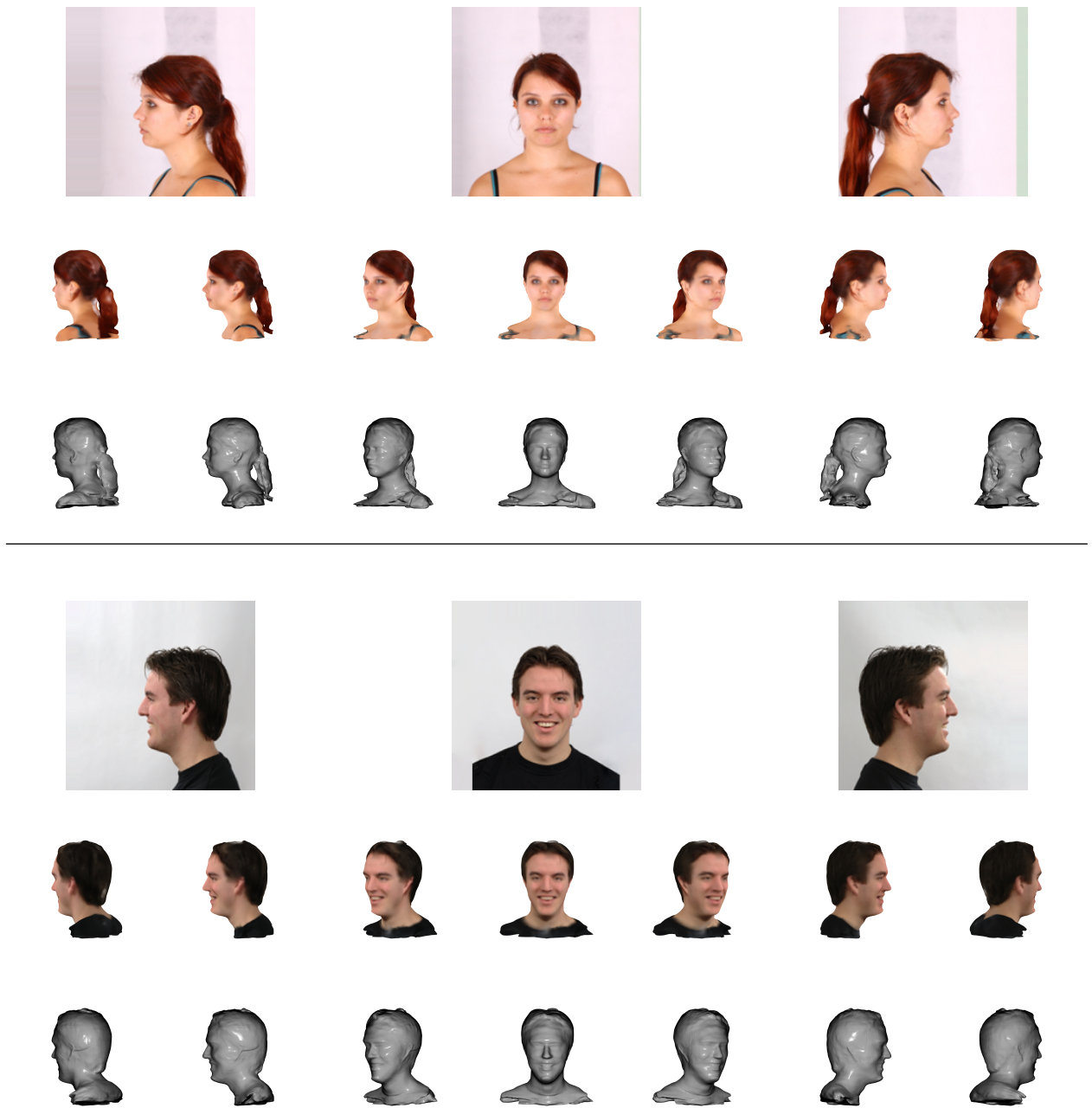


Figure 3: Some multi-view reconstruction examples. For each subject, the first row contains the input images, and the other rows illustrate the reconstructed 3D head model, with and without color, from different viewpoints.

6 Conclusions and future work

In this work, a solution for both single-view and multi-view 3D head reconstruction was presented. Different from alternative approaches, the proposed method is not focused on a shape model fitting process and directly reconstructs head geometry through volumetric regression. The novel facial landmark oriented fusion process, which combines several coarse volumes into a global representation, enables multi-view head reconstruction. The proposed solution shows promising results and has a high extensibility level. Some future development directions include enhancing the texture recovering process and improving the geometry detail representation for more realistic looking 3D models.

Acknowledgement: This work was realised under the supervision of Associate Professor Lucian Mircea Sasu, PhD at Faculty of Mathematics and Computer Science, Transilvania University of Brasov, and benefited of great help from Xperi Corporation which provided both hardware and logistic support.

References

- [1] Md. Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha, and Vijayan K. Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *ArXiv*, abs/1802.06955, 2018.
- [2] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 2 1992.
- [3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [4] Adrian Bulat and Georgios Tzimiropoulos. Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge. *Computer Vision – ECCV 2016 Workshops*, page 616–624, 2016.
- [5] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.
- [6] Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Trans. Graph.*, 35:126:1–126:12, 2016.
- [7] Christopher Bongsoo Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. *ArXiv*, abs/1604.00449, 2016.
- [8] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhöfer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3d morphable face models - past, present and future. *ArXiv*, abs/1909.01815, 2019.
- [9] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. *Lecture Notes in Computer Science*, page 557–574, 2018.
- [10] Huayun he, Guiqing Li, Zehao Ye, Aihua Mao, Chuhua Xian, and Yongwei Nie. Data-driven 3d human head reconstruction. *Computers & Graphics*, 80, 04 2019.
- [11] Huayun He, Guiqing Li, Zehao Ye, Aihua Mao, Chuhua Xian, and Yongwei Nie. Data-driven 3d human head reconstruction. *Comput. Graph.*, 80:85–96, 2019.

-
- [12] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Trans. Graph.*, 34:45:1–45:14, 2015.
- [13] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. *International Conference on Computer Vision*, 2017.
- [14] Aaron S. Jackson, Chris Manafas, and Georgios Tzimiropoulos. 3d human body reconstruction from a single image via volumetric regression. In *ECCV Workshops*, 2018.
- [15] Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Pose-invariant face alignment with a single cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3219–3228, 2017.
- [16] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976.
- [17] Thomas Lewiner, Hélio Lopes, Antônio Wilson Vieira, and Geovan Tavares. Efficient implementation of marching cubes’ cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003.
- [18] Shu Liang, Xiufeng Huang, Xianyu Meng, Kunyao Chen, Linda G. Shapiro, and Ira Kemelmacher-Shlizerman. Video to fully automatic 3d hair model. *ACM Trans. Graph.*, 37(6), December 2018.
- [19] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465, 2018.
- [20] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. *Lecture Notes in Computer Science*, page 483–499, 2016.
- [21] Ozan Oktay, Jo Schlemper, Loïc Le Folgoc, Matthew C. H. Lee, Mattias P. Heinrich, Kazunari Misawa, Kensaku Mori, Steven G. McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas. *ArXiv*, abs/1804.03999, 2018.
- [22] Stylianos Ploumpis, Evangelos Ververas, Eimear O’ Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick Pears, William A. P. Smith, Baris Gecer, and Stefanos Zafeiriou. Towards a complete 3d morphable model of the human head. *ArXiv*, abs/1911.08008, 2019.
- [23] Stylianos Ploumpis, Haoyang Wang, Nick Pears, William AP Smith, and Stefanos Zafeiriou. Combining 3d morphable models: A large scale face-and-head model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10934–10943, 2019.
- [24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.
- [25] Y. Wu and Q. Ji. Facial landmark detection: A literature survey. *International Journal of Computer Vision*, 127:115–142, 2018.
- [26] Yu Yu, Kenneth Alberto Funes Mora, and Jean-Marc Odobez. Headfusion: 360° head pose tracking combining 3d morphable model and 3d reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 40 11:2653–2667, 2018.
- [27] Ruiqi Zhao, Yan Wang, Carlos F. Benitez-Quiroz, Yaojie Liu, and Aleix M. Martínez. Fast and precise face alignment and 3d shape reconstruction from a single 2d image. In *ECCV Workshops*, 2016.
- [28] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *DLMIA/ML-CDS@MICCAI*, 2018.

- [29] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 146–155, 2016.

Andreea DOGARU
Transilvania University of Brasov
Faculty of Mathematics and Computer
Science
Iuliu Maniu 50, 500091 Brasov
ROMANIA
E-mail: *andreea.dogaru.d@gmail.com*

Garbage Collector

Răzvan Gheorghe Filea

Abstract

This paper describes the implementation of a cross-platform Garbage Collection algorithm for C++ and its improvements regarding the release of the dynamically allocated memory. The algorithm is developed in the form of a code library for C++17, which can be used by any program to identify and release unused memory areas. The library is very easy to use, having an intuitive API.

1 Introduction

The program is implemented as a cross-platform code library, using the CMake Build System, and was developed and tested in Visual Studio Community 2019 for Windows, Clang for Linux, and Android Studio for Android.

The very large number of computing systems implemented and appeared on the market thanks to Gordon Moore, who predicted since the 60s the doubling every 18 months of the number of transistors in electronic chips and especially processors, led to the need to implement applications that must run efficiently on several types of platforms (cross-platform) characterized by ISA (Industry Standard Architecture) of different processors and different operating systems [1].

With the explosion of mobile devices, the need for cross-platform applications has become mandatory for developers not to lose the consumer market. Users of mobile applications now expect companies whose services consume them to already have an application that provides services in the mobile version not only Desktop, although for developers this is not always simple, easy, or cheap [2].

This paper is structured as follows. Section 2 presents the main issue of memory in C++ programs, the need to implement a Garbage Collector in C++, the choice of appropriate algorithms, and the importance of implementing it in a multithreaded manner to increase the efficiency of the application. Section 3 describes the proposed solution insisting on the implementation of the dynamically allocated memory release algorithm and the classes used. Section 4 illustrates the test results obtained and the optimizations applied. Finally, section 5 highlights the main conclusions, the contributions made, and proposes some directions for further development of the application.

2 Garbage Collector in C++. Necessity. Algorithms. Benefits

2.1. Creating a Garbage Collector in C++

There are many ways to implement a Garbage Collection algorithm, especially in such a rich and extensible language as C++. An unsophisticated but limited solution is to create a base class, which would be inherited by all the classes that want the ability to use the Garbage Collection [3]. However, this solution is, too restrictive to be satisfactory.

A much more adequate solution would be one in which the Garbage Collector is usable by any class. To provide such a solution, the Garbage Collector must fulfill the following conditions:

1. Accompany the manual memory management method in C++.
2. Not have any impact on already existing codebases.
3. Work with all types of data, such as user-defined data as well as primitives or those provided by the C++ standard library.
4. Be easy to use and straightforward to use.

2.2. The main problem with memory in C++

The main challenge faced when implementing a Garbage Collection algorithm is knowing when an area of memory is no longer used. To better understand the problem, the sequence of code below is considered. There are two dynamically allocated integer objects. The first one contains the value 10 and the pointer to its value is saved in the variable `p`. Then another object with the value 20 is allocated and its address is stored in the variable `p` as well, overwriting the first one, before its memory can be freed. At this point, the memory address of the object `int{10}` is unknown to the program and can no longer be freed.

```
int *p;  
p = new int{ 10 };  
p = new int{ 20 }; // This creates a memory leak  
delete p;
```

So how does the Garbage Collector know when no object contains a pointer to `int{10}`'s memory? The precise way to answer this question is determined by the Garbage Collection algorithm used.

2.3. Choosing an algorithm type

As stated before, the Garbage Collection algorithm type chosen to be implemented determines the memory management model. The issue of Garbage Collection has for a long time presented a difficult yet interesting problem for which there are many different solutions, because of this, a multitude of such algorithms already exist. Some of the most used ones are Mark and Sweep [4], Stop and Copy [5], and Reference Counting [6]. Before choosing one, these three algorithms were reviewed.

a. Mark and Sweep

The Mark and Sweep algorithm is, as the name suggests, made up of two phases. In the Mark phase, all objects that can be reached directly or indirectly have their state set as marked. In the second phase, all allocated objects are scanned and all objects that have not been marked are destroyed and their memory freed.

This algorithm has two main advantages. First of all, the problem of cyclic references is easily solved. Secondly, it adds virtually no running costs other than the collection itself.

But it also has two disadvantages. First, it takes a long time for the Garbage Collector to run and during that time the normal program execution is normally suspended. Thus, the Garbage Collector may create unacceptable running conditions for performance-oriented programs. Second, although the concepts of marking and scanning are simple, they can be difficult to implement effectively.

b. Stop and Copy

The main idea of the Stop and Copy algorithm is that by separating the memory Heap into two, it can remove any fragmentation. One half is designated as active, where all allocations take place, and the other half is the inactive one. During the Garbage Collection process takes place, all objects are copied into the inactive half of the Heap. Then the role of the two memory spaces is reversed.

As stated, the main advantage of this algorithm is that it eliminates memory fragmentation, allowing for the execution of allocations in constant time. But its main disadvantage is that only half of the available memory Heap is usable.

c. Reference Counting

In the Reference Counting approach, each dynamically allocated object has a counter associated with it. When a new memory reference to the object is created, this counter is incremented and it is decremented when one of the references is deleted. And when the reference number drops to zero, the object is considered unused and its memory can be freed.

The main advantage of Reference Counting is its flexibility and simplicity - it is easy to understand and implement. Besides, it does not place any restrictions on the organization, meaning that the reference counter can be independent of the memory location of its associated object. The numbering of references may add overhead to each indicator operation, but the collection phase has a relatively low cost.

The main disadvantage is that the cyclic references prevent the release of memory that is not otherwise used. Cyclic references occur when two different objects retain a reference to each other, directly or indirectly. In the situation of this algorithm, the reference number of any object cannot decrease to zero. Several solutions have been designed for the cyclic reference problem, but they increase the collection costs and add a lot of complexity.

2.4. Multithreading

Multithreading is another important consideration when implementing a Garbage Collection algorithm. Whether it should be single-threaded or multithreaded fundamentally affects the architecture of the algorithm.

The principal advantage of using a multithreaded algorithm is performance. For example, the Garbage Collector can be collected asynchronously when the processor is not being used. But it has the disadvantage that the program sometimes has to be stopped while the collection is taking place. On top of that, it can also lead to increased complexity.

The multithreading approach is used in the described project to achieve increased efficiency. The main focus of this multithreading approach is the delegation of the object destruction and deallocation of memory on another thread. Efficient and thread-safe data structures have also been built to reduce program downtime as much as possible while this delegation happens.

3 Proposed solution

3.1. Algorithm implementation

To implement a Garbage Collector using the Reference Counting algorithm, it is necessary to track the number of objects that have a reference to a dynamically allocated memory location. The problem is that C++ does not have any language features that would allow an object to know when another object references it. Fortunately, there is a simple solution: a new type of reference can be created that aids the management of memory. The described method is also used by the algorithm implemented in this paper.

For the new type to facilitate memory management and make it as straightforward to use as possible, it has to accomplish several things. First of all, it must maintain the number of references to dynamically allocated active objects. Furthermore, it must free up the objects whose reference number falls to zero. In addition to that, the new type should behave similarly to a C++ pointer. For example, all dereferencing operations, such as `*` and `->`, are accepted.

Moreover, this solution is a convenient way of satisfying the constraint that the original C++ dynamic allocation system must not be affected in any way. When the use of the Garbage Collector is desired, the custom reference types are to be used. Instead, when Garbage Collection is undesired, C++ pointers are available. Consequently, both types can be seamlessly used in the same codebase.

3.2. Algorithm class structure

The Garbage Collector contains five main classes, all being located in the „gc” namespace: `gc::destroyer`, `gc::page`, `gc::ref`, `gc::ref_array`, and `gc::internal`.

`gc::destroyer` contains a void pointer (`void*`) to the object itself and another one to the allocated memory (can be null or the same as the object pointer). It also contains a function pointer to a function that destroys the object. Because this class must be able to be stored in arrays with other **`gc::destroyer`** objects that contain pointers to different data types, this class cannot be transformed into a template, thus only storing void pointers, not templated pointers.

`gc::page` is a queue like container but optimized for storing bulks of **`gc::destroyer`** objects efficiently.

`gc::ref` is the most important class of the library, at least from the user’s perspective. This is a class template that will allocate memory and construct objects. Besides the pointer to the object, it also stores a pointer to the reference counter of the object and when the object is no longer used, it delegates the destruction to the Garbage Collector. To make the API easy and straightforward to work with, it overrides the `*` and `->` pointer operators besides having a `get()` function for retrieving the pointer to the managed object.

`gc::ref_array` is a dynamically allocated array that is freed once it is no longer referenced. It represents a specialization of **`gc::ref`** for arrays.

`gc::internal` is a static class with most of its functions being private as they represent the majority of the Garbage Collector code, which should not be accessed by the user directly.

The relations between these classes can also be observed in the diagram in Figure 1.

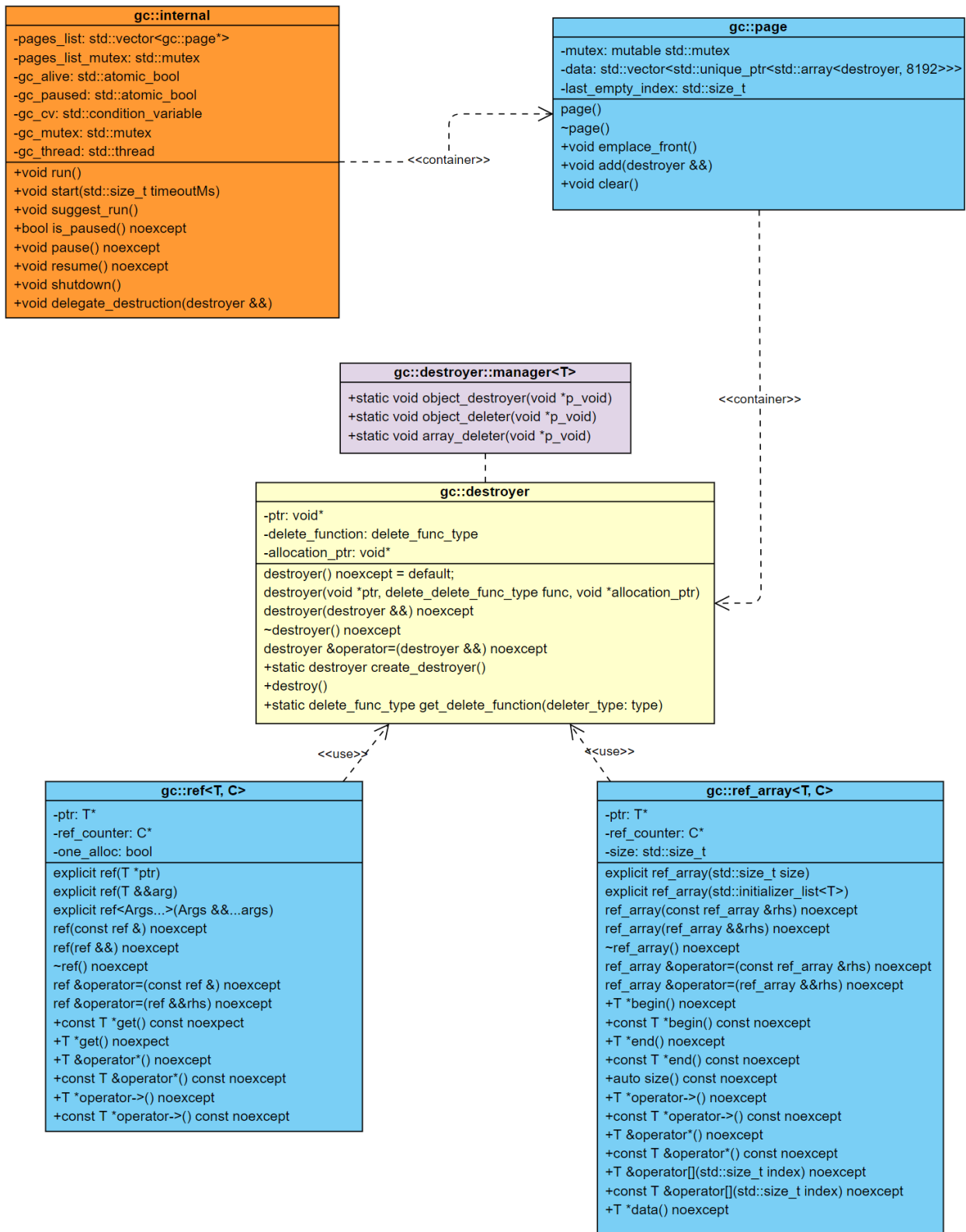


Figure 1: Classes structure on the algorithm

3.3. Implementation details

One of the biggest problems from the beginning was creating an efficient container for any type of object without this class being a template class, because without knowing the type of the class when at compile-time, C++, unlike some other programming languages, cannot create or destroy an object. Specifically, in this case, the Garbage Collector must know how to destroy a certain object. For this reason, a very specific solution was found.

Firstly, the object is allocated and constructed by a templated `gc::ref` class. Once the object is no longer being used the destruction is being delegated to the Garbage Collector, this is done by constructing a `gc::destroyer` with the memory and object pointer.

While `gc::destroyer` is not a template, thus does not know how to destroy the object directly, a private templated class named `manager` was built which has 3 functions: `object_destroyer()`, `object_deleter()`, and `array_deleter()`. A `gc::destroyer` object is constructed using a static template function named `create_destructor()` which properly initializes it.

When creating or copying a `gc::ref`, the reference count is incremented, and when it is destroyed, the counter is decremented. This way, in the end, the number of references always remains 0 and the class knows that it can delegate the memory to the Garbage Collector to be freed up later in a separate thread. Below is the code of the `gc::ref` destructor:

```

~ref() noexcept
{
// if the reference counter is not null (object has not been destroyed already)
// decrement the counter and delegate the destruction to the garbage collector
// if the counter has reached 0
    if (_ref_count && --(*_ref_count) == 0)
    {
        delegate_destruction(destroyer::create_destroyer<T>(_ptr, nullptr, destroyer::type::DELETER));
        _ptr = nullptr;
        _ref_count = nullptr;
    }
}

```

The most important functions in the `gc::internal` class are `start()`, `shutdown()` and `delegate_destruction()`, but it also contains other useful functions for the user like: `suggest_run()`, `pause()`, and `resume()`. Along with these, it contains global variables such as the thread on which the Garbage Collector runs, the list with pointers to all the `gc::page` objects, and other ones like the mutex and condition variable used to start the Garbage Collector up when needed. Below is the code for the `start()` function of the internal class:

```

void start(const std::size_t timeoutMs)
{
    gc_alive = true;
    gc_paused = false;

    gc_thread = std::thread([timeoutMs]
    {
        while (gc_alive)
        {
            // Set up the Condition Variable
            std::unique_lock gc_lock{ gc_mutex };
            gc_cv.wait_for(gc_lock, std::chrono::milliseconds(timeoutMs), [] { return !gc_paused; });

            if (!gc_alive)
                break;
        }
    });
}

```

```

// Run the GC
std::lock_guard pages_lock{ pages_list_mutex };
if (pages_list.empty())
    return;
++gc_count;
for (auto *ptr : pages_list)
    ptr->clear();
    }
});
}

```

4 Tests results and implemented optimizations

To test the performance of the algorithm, a test was designed as extreme as possible to highlight any kind of bottleneck that could occur in an application that requires increased performance. In this test, 128 threads are created, each of them creating 2^{16} `gc::ref` objects containing a test structure that would manually allocate an int on construction and deallocate it on destruction. Half of those `gc::ref` objects are stored in a `std::vector` until the allocation of all objects is completed, to create both short-lived objects and objects that exist for a longer time. The debugging tools in Visual Studio were used to obtain a memory graph and the CPU Usage of the algorithm.

An older version of this paper had a completely different implementation of the library. One of the key differences is that in the older version of this paper `gc::page` objects were shared between threads to save up memory. This idea though, proved, even after repeated attempts at optimizing it, that the amount of CPU time lost on acquiring locks for a page and blocking all other threads to ensure thread-safety was too inefficient for the small amounts of memory it was supposed to save. In the newer implementation, to achieve far better performance and not waste CPU time on waiting for locks in a multithreaded data structure, it has been decided that each thread created to have its `gc::page` object using the `thread_local` keyword. While this new implementation might result in slightly higher memory usage, it has proven itself as a far more performant implementation, even before applying any optimizations.

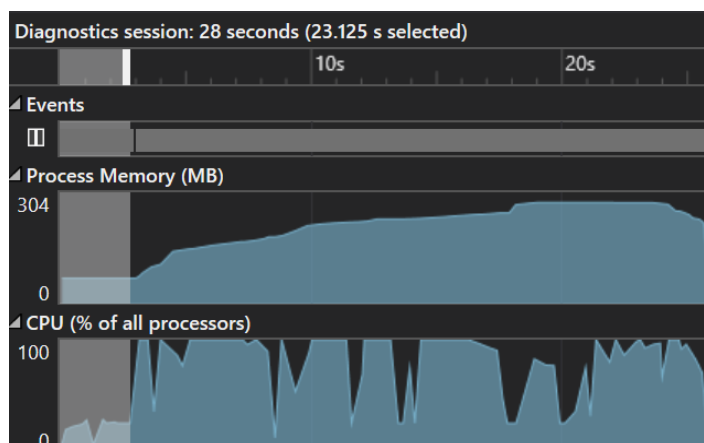


Figure 2: Performance results of the original implementation of this library

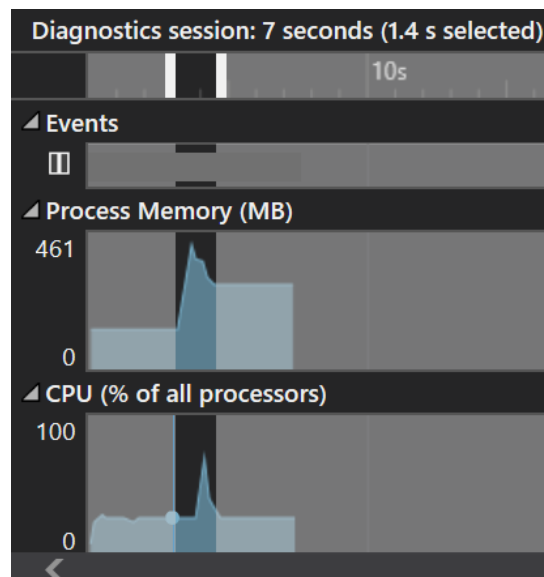


Figure 3: The initial test results from the new implementation

To reduce the memory used by Garbage Collector, the first change made was to alter the type of the variable that contains the reference number from `std::atomic_uint64_t` to `std::atomic_uint8_t`, following an observation made in the article "Getting Reference Counting Back in the Ring" [7]. This article shows that approximately 95% of all objects have a maximum of three references, and on average 99% of objects have a maximum of six references or less. That should mean that for 99% of the objects created the maximum of 255 references (uint8) is more than enough. To offer more flexibility and address those less than 1% that may need more a higher value of possible references, the atomic variable type is now templated in `gc::ref` and `gc::ref_array` classes but is defaulted to `std::atomic_uint8_t` for convenience.

Another optimization applied is avoiding making a second allocation for the reference counter object. This optimization is applied whenever the `gc::ref` constructor is used to construct the object (not when passing a pointer to an already constructed object). The implementation is exemplified below:

```
template <typename... Args>
explicit ref(Args && ...args) : _one_alloc(true)
{ // Allocate the total memory needed for the reference counter and the object itself
  void *memory = operator new(sizeof(std::atomic_uint8_t) + sizeof(T));
  // Construct the reference counter in the first part of the allocation
  _ref_count = static_cast<counter *>(memory);
  new (_ref_count) std::atomic_uint8_t{ 1 };

  // Move the current pointer past the location of the reference counter
  _ptr = reinterpret_cast<T *>(_ref_count + 1);
  // And use that pointer as the construction location for the object
  new(_ptr) T(std::forward<Args>(args)...);
}
```

This optimization also changes the code in the `gc::ref` destructor:

```
~ref() noexcept
{
  if (_ref_count && dec_ref() == 0)
  {
```



```

// Delegate the destruction to the garbage collector
void *allocation_ptr = nullptr;
destroyer::type type;

if (_one_alloc)
{ // one allocation was made so we specify the pointer to the memory which will be deleted after
the object has been destroyed
    type = destroyer::type::DESTROYER;
    allocation_ptr = reinterpret_cast<void *>(_ref_count);
}
else
    type = destroyer::type::DELETER;

delegate_destruction(destroyer::create_destroyer<T>(_ptr, type, allocation_ptr));

_ptr = nullptr;
_ref_count = nullptr;
}
}

```

After all optimizations, the execution time of the test has decreased to only 1.2 seconds, while the memory usage stayed largely the same, as can be observed in *Figure 4*.

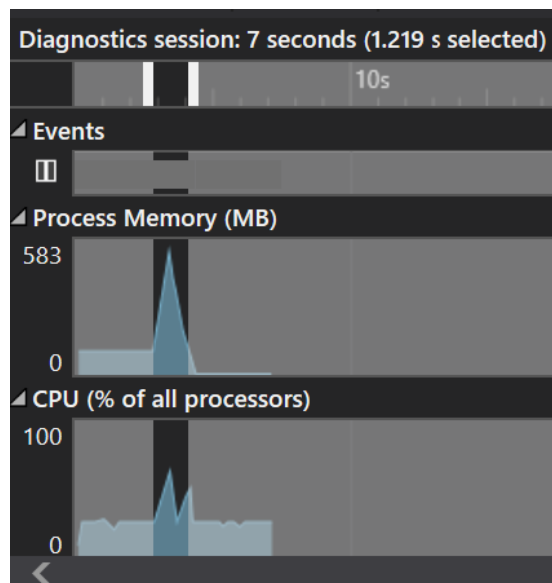


Figure 4: The final test results

5 Conclusions and future developments

The subject of the Garbage Collector is a relatively old and complicated problem. There is no single perfect way to implement one. Each Collector algorithm has its advantages and disadvantages, as discussed in this article. Furthermore, each of the implementations varies greatly depending mainly on the desired features. This paper analyzes the existing C++ memory management model together with several Garbage Collection algorithms and describes the implementation of a library that uses the Reference Counting algorithm.

The Garbage Collection cross-platform algorithm implemented and described in this article is open source and offered as source code under the MIT License and can be found at <https://github.com/TheLuckyCoder/ReferenceCountingGC>. The size of the library is quite small and is usable on any modern system, it may also run on embedded systems but it is unlikely to since it uses multiple execution threads. The implementation can undoubtedly be further enhanced and optimized. It has also been suggested to offer the library as a compiled binary, especially as a Dynamic Link Library (DLL) that could be shared between multiple processes using it. Implementing a cyclic reference detection algorithm has also been proposed, but it will most likely have a too high impact on the performance of the Garbage Collection.

Acknowledgement: This work was supervised by Professor Delilah Florea, from Samuel von Brukenthal National College, Sibiu, Romania.

References

- [1] G. E. Moore, "Cramming more components onto integrated circuits, Reprinted from *Electronics*, volume 38, number 8, April 19, 1965, pp.114 ff.", in IEEE Solid-State Circuits Society Newsletter, vol. 11, no. 3, pp. 33-35, Sept. 2006.
- [2] Andrade, Paulo Roberto Martins de & Frota, Otavio & Silva, Fátima & Albuquerque, Adriano & Silveira, Robson. (2015). *Cross Platform App: A Comparative Study*. Journal of Computer Science and Technology. 10.5121/ijcsit.2015.7104.
- [3] *Garbage collection, the explanation of the concept* - [https://en.wikipedia.org/wiki/Garbage_collection_\(computer_science\)](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science))
- [4] *Mark-and-Sweep: Garbage Collection Algorithm* - <https://www.geeksforgeeks.org/mark-and-sweep-garbage-collection-algorithm/>
- [5] *The Stop and Copy Garbage Collection technique* - <https://wiki.c2.com/?StopAndCopy>
- [6] *Automatic Garbage Collection, the explanation of multiple types of algorithms* - <https://caml.inria.fr/pub/docs/oreilly-book/html/book-ora086.html>
- [7] Rifat Shahriyar & Stephen M. Blackburn & Daniel Frampton, "Getting Reference Counting Back in the Ring" - <https://rifatshahriyar.github.io/files/others/rc-ismm-2012.pdf>

Răzvan Gheorghe FILEA
 Samuel von Brukenthal National College
 Piața Albert Huet 5, Sibiu
 ROMANIA
 E-mail: razvan.filea@gmail.com

Improving automatic meter reading using data generated from unpaired image-to-image translation

Andreas Filinger

Abstract

The automatic reading of water readers or electric readers is a challenging problem that may be solved using deep neural networks (DNNs). Unfortunately, the training of DNNs requires large sets of labeled data that are expensive to obtain. We alleviate this problem by generating artificial training data using unpaired image-to-image translation based on CycleGANs, ie a special form of Generative Adversarial Networks. We have implemented and evaluated a pipeline for generating artificial meter images using a medium sized dataset (6000 images) of water meter images. Our results confirm that this methodology is indeed effective, especially on small initial datasets.

1 Introduction

One core problem to be solved in any neural network project is the acquisition of sufficient training examples. This can prove difficult if the collection process in the particular domain is too expensive or time intensive for large datasets or the collected data does not reflect every class equally.

Especially the latter is true for the problem of reading values of meters from an image, or Automatic Meter Reading: If one were to collect a dataset of photographs depicting meters, the number of times each individual digit appears will differ vastly. To alleviate these problems we propose to supplement existing training data with generated data using unpaired image-to-image translation. This has already been done on the related problem of automatic license plate reading [1]. Inspired by the succesful generation of synthetic licence plate images and improvement of recognition accuracy in that paper, our goal is to replicate those results on the problem of meter reading. In addition to the application domain, our work also differs in scale: While they started with a total training size of about 250.000 images, used 4500 images for training CycleGAN and generated 200.000 artificial training samples, the dataset we use contains 6000 images of watermeters (see chapter 3) and we generate a number of samples in the same order of magnitude (4000) while training CycleGAN using 512 images. In other words, our dataset is about 2 % as big, 11 % as many images to train CycleGAN are used and 2 % as many images are generated in hopes of increasing recognition performance, illustrating the ambitiousness of our work.

Our work supports future projects in the field of automatic meter reading. One such possible project might be derived from what German law requires: Until 2032 every German household must be equipped with an electric meter that is considered "smart", which means it must be able to retrieve the value of the meter and transmit it to the measuring point operator via the internet. Replacing old meters with smart meters would be costly to consumers, thus alternative retrofitting solutions, which extract the value from the existing meter without a need to replace it, become attractive. For example, it might be possible to read the value by taking an image with a mounted camera or even a smartphone.

In particular we aim for the following contributions: I) Implementation of a pipeline which generates realistic meter images from arbitrary meter values (strings of digits). II) Evaluation of the pipeline on the SCUT-WMN-dataset [6] (a collection of water meter images) including implementation of their model for reading water meter images.

The rest of the paper is structured as follows: First some related work dealing with image generation as data augmentation is discussed in chapter 2, then chapter 3 presents the dataset our work builds upon. Next chapter 4 shows the Metrics that were used for evaluation, while chapter 5 describes our methodology in detail and chapter 6 explains how the implementation of CycleGAN we used differs from the original model. Chapter 7 deals with the experiments which were conducted as evaluation, and a conclusion is drawn in chapter 8.

2 Related Work

Data Augmentation, the practice of extending the available training data, has been common practice for some time. Simple transformations/alterations of the already available data such as flipping or cropping are both easy to implement and effective, rendering them a staple in Machine Learning.

A more novel approach is to use other Neural Networks to generate additional data. Generative Adversarial Networks (GANs) are a popular choice for generating new images which look close to a set of images used for training. Yi et al. [2] reviewed a variety of works generating additional training images using GANs for the medical domain, where collection of real data is further made difficult by privacy concerns. As an example, Frid-Adar et al. [3] successfully generated artificial images for a model classifying liver lesions.

One disadvantage of using GANs for generating training data is the fact that it is difficult to control the specific content of the generated images. For this reason another class of generative Networks becomes attractive for this problem: Image-to-image translation. As the name implies, these models allow us to translate images from one domain to another. For example, one might train a model to translate images of landscapes in summer to landscapes in winter. Here, the input to the model is an image which is then altered by the model to look close to images from another domain. Generally these models can be differentiated as being either paired or unpaired: Paired translation tends to give better looking results, but requires paired training data (eg. a training sample consists of an image of a landscape in summer and an image of the same exact landscape in winter), which greatly reduces the scope of application. An example for paired image-to-image translation is Pix2Pix [4]. Unpaired translation on the other hand is not bound by this restriction and is therefore applicable to a greater area of problems. CycleGAN [5] is an example for unpaired image-to-image translation, and it has already been used to generate artificial training examples: Wang et al. [1] improved the accuracy of their model for reading numbers from license plate images by using CycleGAN to generate additional data. They achieved this by first creating synthetic images containing arbitrary plate numbers using scripts, and then training CycleGAN to translate these images to the domain of real license plate images.

In this work this approach is adopted and applied to the related problem of meter reading, on a dataset from a smaller scale.

3 Dataset

There are some datasets for Automatic Meter Reading (AMR) publicly available, among those we searched for a collection of analog meter images and decided on the SCUT-WMN-Dataset [6], a collection of 6000 images of watermeters. Compared to [1], where the number of original training images was close to 250.000 and the number of generated images was 200.000, this dataset with its 4000 images used for training is rather small. Also, as can be seen from Table 1, the distribution of individual digits differs greatly. The authors further proposed a model for recognizing digits from their dataset called FCSRN ("Fully Convolutional Sequence Recognition Network"). How well this model performs on their dataset was measured using the metric "Character Accuracy Rate" (AR), which equals the ratio of correctly predicted digits to the total number of digits. An additional advantage of the WMN-Dataset is the fact that it features centered images of meters, saving us the step in the OCR-Pipeline of detecting the meter.

The authors of the dataset collected two kinds of images: There are 1000 "easy" images and 5000 "difficult" images, which are taken under less favourable conditions (obtrusions, blurriness, ...). The

latter are further divided into 4000 images used for training and 1000 images used for testing. Figure 3 shows examples of both easy images as well as difficult images.

Future works might repeat the experiments of this paper on other datasets. For example, Laroca et al. [10] collected 2000 uncentered images of various types of electric meters and published them as the UFPR-AMR-dataset. Also there is the Gas-Meter Reading Dataset [11] consisting of 153 images of gas meters. In contrast to above works which feature analog meters, Karthick et al. [12] collected 169 images of 7-segment energy meters (eg with digital display) as the YUVA EB Dataset.

character	0	1	2	3	4	5	6	7	8	9
number	6612	3234	1100	1214	517	916	754	1037	1538	920
character	10	11	12	13	14	15	16	17	18	19
number	157	210	215	340	165	184	202	221	245	219

Table 1: Distribution of digits in the SCUT-WMN-Dataset (adapted from [6]). Numbers 10-19 refer to "mid-state digits", which are half way between two values: For example, number 14 refers to a meter digit where both digits 4 and 5 are seen.

4 Metrics

4.1 Character Recognition Accuracy

As mentioned in the previous section, the proposed model from [6] was used to evaluate the results. More specifically, we implemented their model according to the descriptions in the paper (without the proposed additional Augmented Loss). The performance of the model was measured using the metric "Character Accuracy Rate" (AR), which [6] defined as follows:

$$AR = 1 - (D_e + S_e + I_e)/N_t$$

where D_e , S_e and I_e are the total numbers of deletion errors, substitution errors and insertion errors respectively and N_t is the total number of digits in the test set. To evaluate our implementation we trained and tested it 10 times on the same data the authors used. As can be seen in Table 4 our best score of 0.9694 comes close to what was reported in the paper (0.9698). Generated images are then evaluated by integrating them into the training of the FCSRN model, by first using generated images to pretrain the model, and then fine tuning using the original training data. More detailed descriptions of the experiments are the subject of chapter 7.

4.2 Image quality

To measure the quality of generated images the Frechet Inception Distance (FID) [8] was used, a metric for evaluating Generative Networks. This metric calculates the distance of generated and real images by comparing the activations of Inception Net (a model for classifying images) on the penultimate layer. From these activations mean and covariance are calculated. Then the Frechet Distance, a metric which can be used to calculate the distance between two probability distributions, is used to derive the Frechet Inception Distance using the mean-covariance pairs from both image sets. Given mean vector and covariance matrix (\mathbf{m}, \mathbf{C}) from activations using generated images and $(\mathbf{m}_w, \mathbf{C}_w)$ using real images, Heusel et al. [8] propose the following formula for calculating the FID:

$$d^2((\mathbf{m}, \mathbf{C}), (\mathbf{m}_w, \mathbf{C}_w)) = \|\mathbf{m} - \mathbf{m}_w\|_2^2 + Tr(\mathbf{C} + \mathbf{C}_w - 2(\mathbf{C}\mathbf{C}_w)^{\frac{1}{2}})$$

where TR represents the trace of a matrix, or sum of values on the main diagonal. The authors also published their Tensorflow implementation on Github [9] which we used for our experiments.

5 Methodology

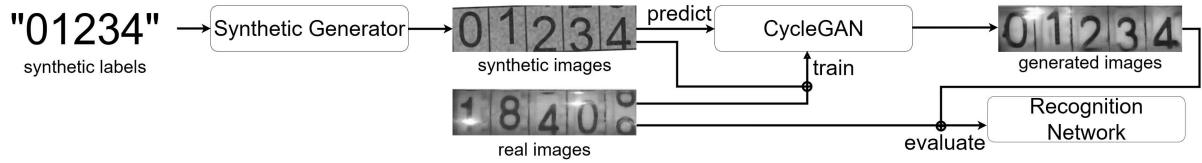


Figure 1: The full Pipeline. First, the Synthetic Generator creates synthetic images containing specific labels (digit strings). CycleGAN is trained to translate between synthetic and real images. The performance of the recognition network is increased by including the generated images into its training.

We adopt the approach of [1] and implement a pipeline to transform arbitrary labels (strings of digits in our case) to realistic meter images close to the dataset which is used. Then the generated data is integrated into the training of FCSRN by using them for pretraining and then fine tuning with the original training images. Images are generated from random labels to countervail the unbalanced distribution in the original data. For all experiments we generate 4000 images for pretraining, just as many images as there are for training on real difficult images.

The pipeline (see Figure 1) consists of the following three steps: First, from a string of digits a synthetic image is created neither using neural networks nor being realistic. Then this synthetic image is translated to a realistic image close to the dataset using CycleGAN. Finally, generated images are evaluated by integrating them into the training of FCSRN.

5.1 Generation of synthetic images

Synthetic images depicting a specific meter value are created by stitching together images of individual digits using OpenCV. Digit images were taken from the Char74k-dataset [7], a dataset for OCR which among others contains images of individual digits and characters in a diverse selection of fonts. To these initial synthetic images a number of transformations was applied to bring them closer to real images without using neural networks yet. These include:

- adding random noise to each pixelvalue
- adding a border around the image and a black line between digits
- rotating by a small random angle and cropping the result a little

These transformations made the synthetic images closer to real easy images, as can be seen in Table 2.

Imagesets	FID-value
synth-simple, real-easy	269
synth, real-easy	161
gen-easy, real-easy	63
gen-diff 1-step, real-diff	75
gen-diff 2-steps, real-diff	70

Table 2: Distances between sets of generated and sets of real images according to FID using 2048 images each. "synth-simple" are synthetic images without the transformations described in chapter 5.1. A small value indicates that the sets are close to each other.

5.2 Generation of realistic easy images

In order to establish a baseline, we started with generating easy images. Thus a CycleGAN model was trained on synthetic and easy images (details on the implementation of CycleGAN are covered in chapter 6). Table 2 shows the distance between a set of real easy images and different sets of generated images in terms of the FID.

5.3 Generation of realistic difficult images

Intuitively, generating easy images is simpler than difficult images as those additionally feature a variety of distortions and obtrusions. Thus two approaches to generate difficult images were examined: First CycleGAN was trained to translate synthetic images to difficult images directly (1 step). Then a CycleGAN model was trained to generate real difficult images by first training on real easy images and then fine tuning using generated images. This way difficult images are generated in 2 steps: One model translates synthetic images to easy images, the other translates easy images to difficult ones.

We find that the 2-step method produces images which are slightly closer to real difficult images (see Table 2), while the 1-step method has the advantage of not relying on real easy images.

6 Implementation Details

Based on [1], we want to improve the recognition rate by making use of CycleGANs. Our available hardware and time imposed limits on how long we were able to train CycleGAN. There are 2 measures which were thus employed to mitigate this limitation:

1 There are used 512 images per domain for training CycleGAN compared to the maximum of 1000 available easy images

2 The original architecture of CycleGAN was downsized

The original CycleGAN-Paper [5] described their architecture using the following notation:

Let $c7s1-k$ denote a 7×7 Convolution-InstanceNorm- ReLU layer with k filters and stride 1. dk denotes a 3×3 Convolution-InstanceNorm-ReLU layer with k filters and stride 2. Reflection padding was used to reduce artifacts. Rk denotes a residual block that contains two 3×3 convolutional layers with the same number of filters on both layer. uk denotes a 3×3 fractional-strided-Convolution-InstanceNorm-ReLU layer with k filters and stride 0.5. The [generator] network with 6 residual blocks consists of:
 $c7s1-64, d128, d256, R256, R256, R256, R256, R256, R256, u128, u64, c7s1-3$ [5]

The generator was downscaled as follows:

$c7s1-32, d64, d128, R128, R128, R128, R128, R128, R128, u64, u32, c7s1-3$

In summary the sizes of the channels in each intermediate layer were halved.

Similarly, the original discriminator is as follows:

Let Ck denote a 4×4 Convolution-InstanceNorm-LeakyReLU layer with k filters and stride 2. ... The discriminator architecture is: $C64-C128-C256-C512$ [5]

Our implementation removed one layer from the discriminator:

$C64-C128-C512$

Using the new model the training was about 3 times faster while the FID-values were similar to those using the original model. Additionally, we now were able to train more epochs, resulting in better FID-values. We hypothesize that the original cyclegan, which achieved good results on various domains such as landscapes, animals and paintings, is too complex for the domain of meter images. It may even be possible to achieve better results by downsizing the model further. Further experimentation with the model architecture is left to future work.

As for tools and environment, the machine learning framework Tensorflow [14] in Python was used. The GPU used to conduct experiments is NVIDIA GeForce GTX TM1050 Ti (Mobile). For the CycleGAN implementation, we started with the official example implementation from Tensorflow [15], and adapted it according to the PyTorch implementation from the authors of CycleGAN [16]. For the the discriminator, the discriminator from the Pix2Pix example of Tensorflow [17] was used.

7 Evaluation

The pipeline is evaluated in terms of the AR achieved by the recognition network FCSRN if trained on real images only (baseline) and if trained additionally on generated images. First, FCSRN is trained using real data and AR is calculated. This value comprises the baseline we want to improve using generated images. Next, FCSRN is trained afresh by first pretraining only with generated images, then fine tuning with the same real data as used for the baseline. As the WMN-dataset is split into easy and difficult images, this process is repeated on both these image sets. Also we examine how well the pipeline performs on a reduced dataset by evaluating the pipeline on a smaller subset of real difficult images.

7.1 Evaluation on easy images

First the generation of easy images is evaluated. To that end the 1000 available real easy images were randomly split into 500 images for training (easy-train) and 500 for testing (easy-test). The baseline was produced by training FCSRN with easy-train and calculating AR on easy-test. Next, CycleGAN was trained with easy-train and 4000 images were generated. FCSRN was then pretrained on these artificial images, and fine tuned on easy-train. Each training run of FCSRN was conducted 5 times, and for every run the best AR was noted. From these best scores the average, maximum and minimum are shown on Table 3. Using generated images resulted in a significant improvement in both average and minimum AR, and a small improvement in Maximum AR.

Method	Average	Maximum	Minimum
Baseline 500 easy	0.9346	0.9620	0.9112
Baseline 500 easy + 4000 generated easy	0.9653	0.9688	0.9604

Table 3: FCSRN Character Accuracy using 500 real easy images for training and remaining 500 easy images for testing

7.2 Evaluation on difficult images

Next the effectiveness of both generating easy and difficult images for recognition of real difficult images was evaluated. As mentioned previously the authors of FCSRN used 4000 real difficult images to train the recognition network and 1000 real difficult images to test it. For the baseline, we used the same data to train and test FCSRN as they used and repeated the training 10 times. As can be seen from Table 4, this yielded a best AR of 0.9694, which comes close to the value reported by the authors of 0.9696. Pretraining was done using both generated easy images as in the previous section as well as using generated difficult images. Also, the generation of difficult images employing either the 1-step or 2-step approach described in section 5.3 was tested. In each case, 512 real images were used to train CycleGAN and the training of FCSRN was repeated 5 times. As can be seen in Table 4, both generated easy and difficult images improved the average and minimum best AR by a similar degree, while the maximum remained close to the baseline with generated easy images and went down a little with generated difficult images using the 2-step approach. The best results were achieved using generated difficult images from the 1-step method, where both average and minimum AR were highest among all tests and maximum was the same as in the baseline.

7.3 Evaluation on reduced dataset

The effectiveness of generating difficult images was evaluated on a smaller image set. Of the initial 4000 difficult images for training, 500 images were sampled randomly (diff-train-subset). Next, a baseline was obtained by training FCSRN on diff-train-subset and testing with the 1000 difficult images for testing. As usual, this was repeated 5 times. Table 5 shows that the best AR-value was 0.7672. Then FCSRN was pretrained with generated images and fine tuned with diff-train-subset, 5 times for each experiment.

Method	Average	Maximum	Minimum
Baseline 4000 difficult	0.9617	0.9694	0.9522
Baseline 4000 difficult + 4000 generated easy	0.9638	0.9690	0.9616
Baseline 4000 difficult + 4000 generated difficult (1 step)	0.9665	0.9694	0.9632
Baseline 4000 difficult + 4000 generated difficult (2 steps)	0.9649	0.9662	0.9618

Table 4: FCSRN Character Accuracy using original data only (Baseline) and pretraining with generated images

In particular, 4000 difficult images were generated once with the 1-step and once with the 2-step method. For training CycleGAN, only diff-train-subset was used. Also pretraining with real easy images was tested. Table 5 shows that pretraining using real easy images or generated difficult images from 2 steps both improved the maximum best AR from 0.7672 to above 0.91. Additionally, in a scenario where only difficult images are available pretraining using generated images from the 1-step method (which means no easy images were used) results in similar improvements.

Method	Average	Maximum	Minimum
Baseline 500 difficult	0.6964	0.7672	0.5552
Baseline 500 difficult + 1000 real easy	0.8955	0.9106	0.8694
Baseline 500 difficult + 4000 generated difficult (1 step)	0.9002	0.9136	0.8904
Baseline 500 difficult + 4000 generated difficult (2 steps)	0.8878	0.9184	0.8564

Table 5: FCSRN Character Accuracy using only a subset of 500 from 4000 original images for training (Baseline) and pretraining with generated images

8 Conclusion

Inspired by the work of Wang et al. [1] we have shown that supplementing training data with images generated by unpaired image-to-image translation is indeed effective for the domain of Automatic Meter Reading. In contrast to their work we operated on a vastly smaller scale, adding 4000 generated meter-images to a dataset of the same size while only utilising 512 images for training CycleGAN. We found that this methodology is especially effective when an insufficient number of initial training samples is available. It is also usable on images which were taken under unfavorable conditions.

Future works might repeat these experiments using translation models other than CycleGAN, as newer models attempt to improve various aspects of translation, for example the diversity of the output. Also, datasets other than SCUT-WMN might be examined in the future.

Acknowledgement: I would like to thank Prof. Andreas Siebert from the University of Applied Sciences Landshut for supervising and supporting this work. Also special thanks to Nikolai Körber for the original idea and always providing helpful feedback and suggestions.



Figure 2: Examples of synthetic, generated easy and generated difficult images.



Figure 3: Examples of real easy and difficult images from the WMN-Dataset.

References

- [1] Wang, Xinlong et al., Adversarial Generation of Training Examples for Vehicle License Plate Recognition. *ArXiv abs/1707.03124*, n. pag, 2017.
- [2] Yi, Xin et al., Generative Adversarial Network in Medical Imaging: A Review. *Medical image analysis* 58, 101552, 2019.
- [3] Frid-Adar, Maayan et al., GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification. *ArXiv abs/1803.01229*, n. pag, 2018.
- [4] Isola, Phillip et al., Image-to-Image Translation with Conditional Adversarial Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5967-5976, 2017.
- [5] Zhu, Jun-Yan et al., Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2242-2251, 2017.
- [6] Yang, F. et al., Fully Convolutional Sequence Recognition Network for Water Meter Number Reading. *IEEE Access* 7, 11679-11687, 2019.
- [7] de Campos et al., Character Recognition in Natural Images. *VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications*, 273-280, 2009.
- [8] Heusel, Martin et al., GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium. *ArXiv abs/1706.08500*, n. pag, 2017.
- [9] Code accompanying the paper 'GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium'. <https://github.com/bioinf-jku/TTUR> accessed: 01.09.20
- [10] Laroca, R. et al., Convolutional Neural Networks for Automatic Meter Reading. *Journal of Electronic Imaging*, vol. 28, 013023, 2019.
- [11] Nodari, Angelo and Ignazio Gallo., A Multi-Neural Network Approach to Image Detection and Segmentation of Gas Meter Counter. *Proceedings of the Conference on Machine Vision Applications (MVA2011)*, n. pag, 2011.
- [12] Karthick, K. et al., Text detection and recognition in raw image dataset of seven segment digital energy meter display. *Energy Reports* 5, 842-852, 2019.
- [13] Zhang, R. et al., The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 586-595, 2018.
- [14] Tensorflow. <https://www.tensorflow.org/>. accessed 22-November-2020.
- [15] Tensorflow CycleGAN example. <https://www.tensorflow.org/tutorials/generative/cyclegan>. accessed 22-November-2020.
- [16] Official CycleGAN PyTorch implementation. <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>. accessed 22-November-2020.
- [17] Tensorflow Pix2Pix implementation. https://github.com/tensorflow/examples/blob/master/tensorflow_examples/models/pix2pix. accessed 22-November-2020.

Andreas FILINGER
 University of Applied Sciences Landshut
 Faculty of Computer Science
 Am Lurzenhof 1, 84036 Landshut
 GERMANY
 E-mail: andreas.flinger@gmail.com

Towards an Industrial Recommendation System for Quality Improvement: Comparison of Python and C++ Implementations in an Edge- and Cloud-Computing Environment

Alexander M. Frühwald, Steffen Kastner, Anna-Maria Schmitt, Simon Haas,
Leonhard Hösch, Lars Fichtel and Christian Bachmeir

Abstract

This paper discusses the comparison of an exemplary industry 4.0 recommendation system using image recognition algorithms. Foundation of our work is an industry driven use-case: manual assembly of a retaining ring in planetary gears. We apply state-of-the-art machine learning methods, such as principal component analysis for feature extraction and a random decision forest for classification were used to detect incorrect assemblies and generate insights for the recommendation system. We implement the exemplary recommendation system in Python and C++, provision and deploy both implementations in a newly developed state-of-the-art automated building and deployment platform. Finally we evaluate both implementations in an Edge-Computing and also in a Cloud-Computing environment.

1 Introduction

Machine Learning, Pattern Recognition and Industry 4.0 currently enjoy a high level of attention and are investigated whether and how these technologies could be used to generate new insights in production processes, and be used as recommendation-systems which then could be used to improve efficiency [3, 2]. We focus on a specific use-case, in close cooperation with an Industry-Partner:

Automation of quality controls in one specific manual assembly step, with the goal to reduce failures of the overall assembled product through a recommendation system. Specific manual assembly step: a planet gear is fixed on a bolt by a retaining ring.

Following this use-case, we cover and implement the end-to-end analysis process in this work:

- Raw-data generation in our experimental setup:
Our proposed system takes a picture of the specific monitored assembly step.
- Preprocessing of raw-data:
Our proposed system preprocesses the picture of the specific monitored assembly step, i.e. reduces unnecessary information and prepares the data for analysis

- Using an example algorithm and supervised machine learning approach to generate insights in our recommendation system:
Our proposed system analyzes the preprocessed picture, and gives feedback on whether one specific quality target was achieved or not. Hereby we use pattern recognition on pictures.

In our work we do not focus on the actual algorithm used for deriving insights. We acknowledge that choosing or even developing the algorithm is of major importance in order to deliver the expected quality improvements. However this is not the focus of our work in this paper: We just use an example algorithm, in order to be able to implement the full end-to-end analysis process as described above. The focus of this paper then lies in the comparison of different implementations of the same algorithm and a modified processing chain:

- Comparison of (example) Python and C++ based implementations
- Comparison of Edge and Cloud based implementations

First an automated software building and deployment platform is created, in order to be able to run our comparisons in a repeatable and reliable environment. In addition this platform is also derived from current state-of-the-art in Cloud and Edge Computing, hence our setup is very close to an actual implementation on a manufacturing shopfloor. Using this platform, performance differences are investigated between implementations using Python and C++ (of the same end-to-end analysis process). In addition also both implementations are tested in an Edge- or Fog-Computing setup[8] locally, and in a cloud environment (Amazon Web Services).

Our end-to-end analysis process is described in Sections 2, 3, and 4: In Section 2 we describe our use-case in detail, and outline our experimental setup of generating raw-data in the manual assembly process. In Section 3 the preprocessing of the images is presented. Finally, in Section 4 the example algorithm and the supervised machine learning approach of the recommendation system is presented. The core contribution of this work is presented in Section 5, and Section 6: In Section 5, we present the automated building and deployment platform. In Section 6, we present our toolchain, the measurements taken in our implementation as well as the evaluation of our measurements. Finally we end a summary and outlook on future work.

2 Raw Data Generation in an Experimental Setup and Use-Case Description

The primary focus of this work is the detection of incorrectly assembled or missing retaining rings. In order to detect these errors, it is necessary to divide the images into 'rejected' and 'correct'. In the process of generating image data for subsequent image recognition, various influencing factors such as lighting conditions, reproducibility, etc. should be taken into account. This process is not essential for the goal of this paper and for this reason will not be discussed in detail. The errors which can occur during the assembly and how the image acquisition was performed will be explained in the following.

2.1 Use-Case: Definition of rejected and correct Work Pieces

Errors that occur during the assembly of a retaining ring are described in this section and are shown as examples in Figure 1. The main cause of errors is the employee, who performs the manual operation and the pliers available for this work step. The retaining rings must be stretched slightly with the pliers to fit into the groove of the bolt. If the retaining ring is

overstretched, it cannot return to its original shape. In exceptional cases, it may happen that no retaining ring is fitted due to a distraction or carelessness of the responsible employee. Further possible errors include:

- Retaining ring not correctly engaged in groove
- Ridge at the groove for the retaining ring
- Chips in the groove for the retaining ring
- Retaining ring is placed diagonally on the bolt, one part in the groove, one part on the bolt

These problems result in two error groups that can be detected by image recognition. On one hand the absence of a retaining ring and on the other hand the exceeding of the deviation of tolerances (see Figure 1). These tolerances can be measured as the distance between the ends of the retaining ring.



Figure 1: Representation of a correctly (left) and incorrectly (right) mounted retaining ring on a planetary gear.

The classification of work pieces in the categories 'rejected' and 'correct' should be executed as fast as possible to increase the amount of manufactured components. For this reason, the technologies mentioned above are compared in detail.

2.2 Raw Data: Capturing the Pictures

In the current design of the assembly line, no space was planned for an additional quality assurance step. For this reason, a fixture had to be developed for the initial data acquisition, which made it possible to obtain multiple images with minimal manual effort while still under the same conditions and from a defined position. It was important to ensure that the person assembling the piece was not hindered in his work. For this purpose, a prototype device was created using aluminum profiles, a cable tie and a commercially available smartphone. The integrated camera had a resolution of 12 megapixels (4032×3024 px). Section 7 describes in detail which knowledge can be derived from this simple test setup for later quality control using image recognition. In order to guarantee a high degree of similarity with repeated recordings of images, the fixture must be attached to the tool carrier in a defined position and at a fixed angle. For this purpose, the fixture has been screwed to an existing device and thus had a fixed position on the tool carrier. Figure 2a shows the tool carrier and the fixture used in the experimental setup like they were used in the assembly line. On the basis of this test setup, 104 images of a assembled gearbox were created, which were then labelled by hand into 50 reject and 54 correct images. The images recorded in this experiment are comparable to Figure 2b. The graphic shows two correctly installed retaining rings (bottom left and top) and one incorrectly installed (bottom right). It is important to find and identify this using various methods. The steps which are necessary for the preprocessing of the images are explained in Section 3.

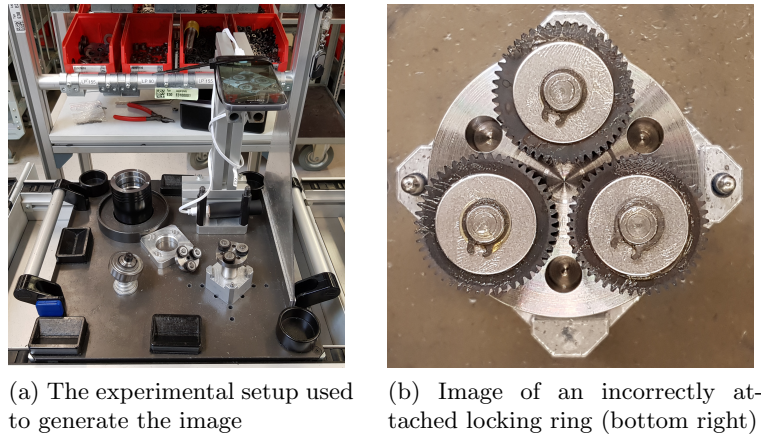


Figure 2: Presentation of the experimental setup (a) and an image of an incorrectly attached locking ring (b).

3 Preprocessing of the raw Input Images

Since a classification with high quality is only possible to a limited extent with the images at hand, a 'divide & conquer' approach known from computer science is used. In order to achieve a good classification, unnecessary information such as the work surface should be removed during preprocessing, but relevant information such as the retaining rings should be retained. This was achieved using computer vision with OpenCV, which is described below. The resulting cut images were again manually divided into the categories 'rejected' (107) and 'correct' (216). The number of images has increased because there have been several retaining rings on one input image. Later on, a classifier will be trained with these images. During preprocessing the input image is first converted to greyscale and then scaled down to the working size of 1/2 of the original size. Further, the working image is scaled again to 1/10 of the working size for a temporary circle search (see Figure 3a). Additionally, a Gaussian filter is used for noise reduction. These two operations make circle detection using OpenCV faster and more robust.

The wanted circles are the black gears on the planetary gears with the retaining rings. A tolerance area is added to the detected circles, because the circle search on the temporarily downscaled image causes a certain inaccuracy when the selected region is scaled up to the original working image (see Figure 3b).

There may be lubricant on the gearwheel, which can lead to problems with later classification. To remove lubricant from the image, a dilatation is performed. The remaining noise can be eliminated by a gray value dilation without a removal of the retaining ring to be classified (see [7]).

In the next step, another circle detection is performed on the previously cut circles with the tolerance to eliminate the inaccuracies of the first circle detection on the downscaled image (see Figure 3c). These circles are cut out again and scaled to a standardized size of 256×256 pixels (Figure 3d). Furthermore, a circle area in the center and the edge of the image is blackened (see Figure 4). The black areas were chosen in such a way that the other types of retaining rings (see Figure 2b) are not covered.

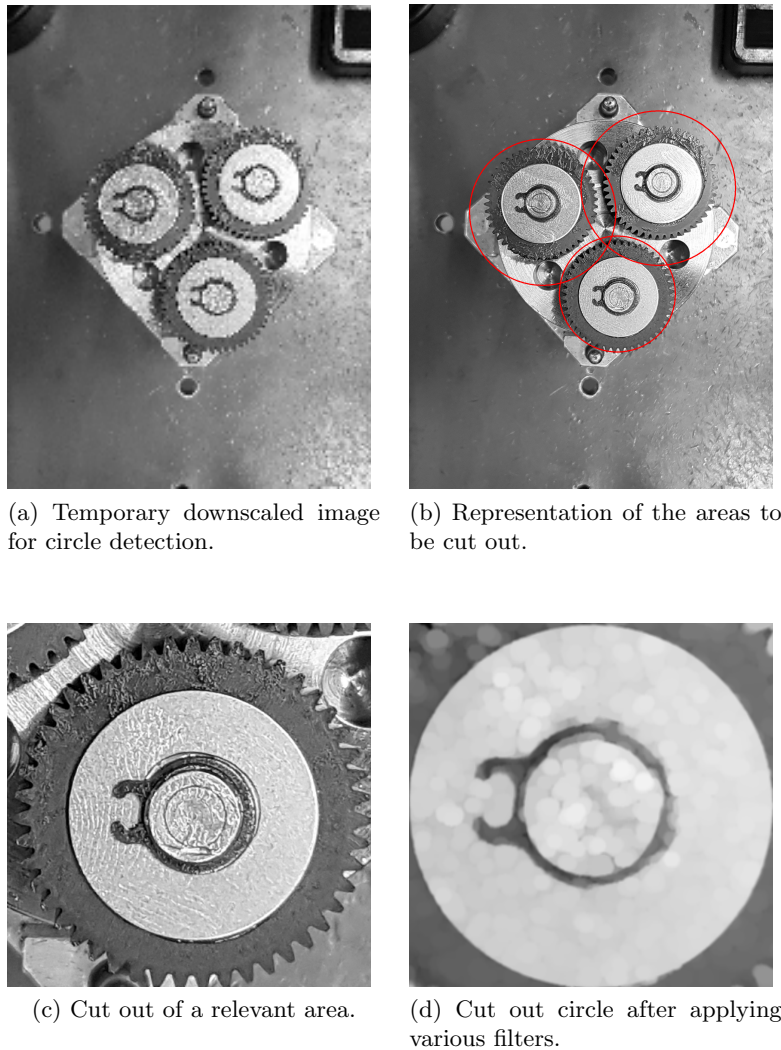


Figure 3: Display of individual images after some selected preprocessing steps.

4 Example Algorithm and Supervised Machine Learning Approach

The training images are already classified as 'rejected' and 'correct', which is why a supervised learning approach can be chosen. The input images contain a number of useless information e.g. the working surface and other parts of the gearbox. For this reason, the individual retaining rings are cut out in the preprocessing stage and the classified separately.

4.1 Feature Generation

After preprocessing, features are then extracted from the images. This is done using a principal axis transformation (PCA). For this purpose, the training images are vectorized in order to calculate a transformation matrix with this data. This matrix maximizes the mean square distance of the image features in the feature space.

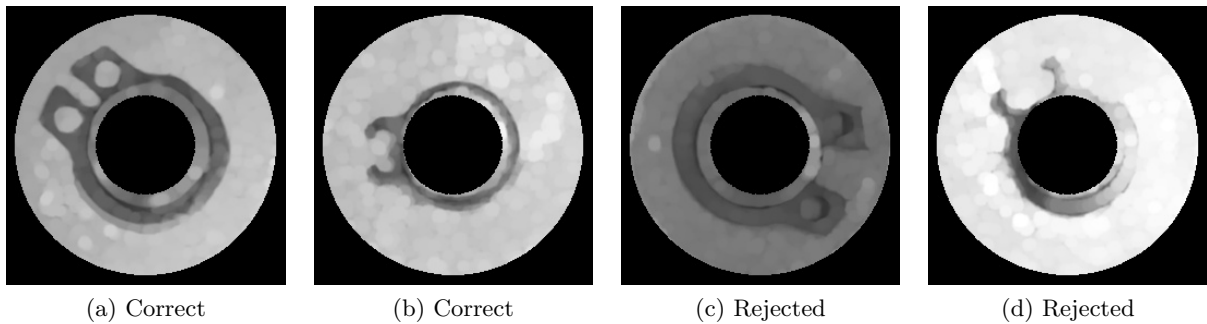


Figure 4: Cut out retaining rings with blackened sections after preprocessing.

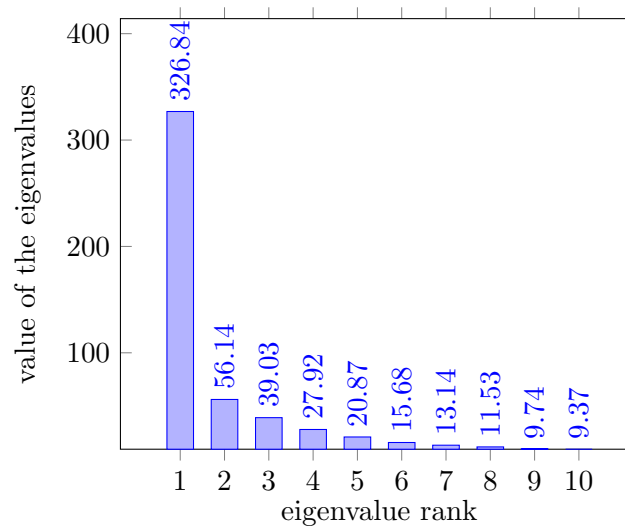


Figure 5: Plot of the 10 biggest eigenvalues of the PCA.

With the help of the PCA, an additional dimension reduction to six relevant features is performed, which are represented by the eigenvalues of the covariance matrix. This way about 90% of the information is preserved (see Figure 5, see [7]). This is especially important because in an image without feature reduction each pixel represents at least one dimension. This high number of dimensions would make mathematical operations on these features difficult and slow.

4.2 Training of a Classifier

With the six-dimensional features of the training images from Section 4.1 a random forest classifier was trained on two classes ('rejected' and 'correct') (see [6]). The test images delivered better and more robust classification results compared to the decision-tree-classification and the kNN-classifier in a cross validation procedure (see [7]). Depending on the amount of training and validation, these range from 88% to 92%.

The problem with this is that the classifier generates misclassifications on both sides. This means that the classifier, which classifies the test images correctly with 92%, has classified 20 test images as 'correct' (TP: True Positive) and 5 as 'rejected' (FN: False Negative). Furthermore, he also classified 79 of faulty ones as 'rejected' (TN: True Negative) and 5 as 'correct' (FP: False Positive). These values can be visualized in a confusion matrix (cf. [9], Table 1).

		Prediction	
		Correct	Rejected
Reality	Correct	20	5
	Rejected	5	79

Table 1: Truth matrix of the classification

$$P(\text{correct}|\text{as Correct classified}) = \frac{TP}{TP + FP} \quad (1)$$

Equation (1) can be used to derive how likely it is that a retaining ring classified as 'correct' is really correct. This is $P(\text{Correct}|\text{as Correct classified}) = 0.8$. This value is also called the positive predictive value (PPV).

$$P(\text{Rejected}|\text{as Rejected classified}) = \frac{TN}{TN + FN} \quad (2)$$

Analogously, Equation (2) can be used to calculate the probabilities, which indicate the proportion of those correctly classified as scrap to the total number of those classified as scrap. This is $P(\text{Rejected}|\text{as Rejected classified}) = 0.94$. The probability that a part classified as defective is actually scrap is very high.

5 Automated Building and Deployment Platform

The following sections show the tools that were used to implement the prototype. Since several people were involved in the development, the developed program code was stored in a version control, where each module had its own repository. This made it possible to develop the different sub-projects at the same time and to use them more easily in later steps.

5.1 Automated Build System

In order to make the process of prototyping as quick and simple as possible a modern continuous deployment approach has been chosen. This means that an installed tool, the build tool, is permanently waiting for changes in the Git repositories. As soon as a change occurs, the build tool receives it and creates a new Docker container for the corresponding project. A Docker container can be executed on target systems with just one command (cf. [5]) and thus enables the fast deployment of software.

The used build tool is TeamCity from JetBrains [4]. It was configured in such a way that it automatically creates the corresponding Docker containers and then publishes them to the public Docker Hub [1]. Figure 6 shows an overview of the built settings for the individual containers.

5.2 Details of the Build Procedure

As already described, TeamCity reacts to changes in the version control and starts a complete workflow process. Figure 7 shows this as an example.

First, the containers for which a change has been recognized are rebuilt and published in the Docker Hub (cf. [5]). Afterwards, the new containers are automatically updated and restarted on the specified endpoints. In this case, the endpoints are an AWS cloud instance and a server in the intranet that could represent a fog network. Finally, test runs are carried out on the four

FHWS Fog Computing

Builds and pushes docker containers required for the Fog-Computing-Course at the FHWS.

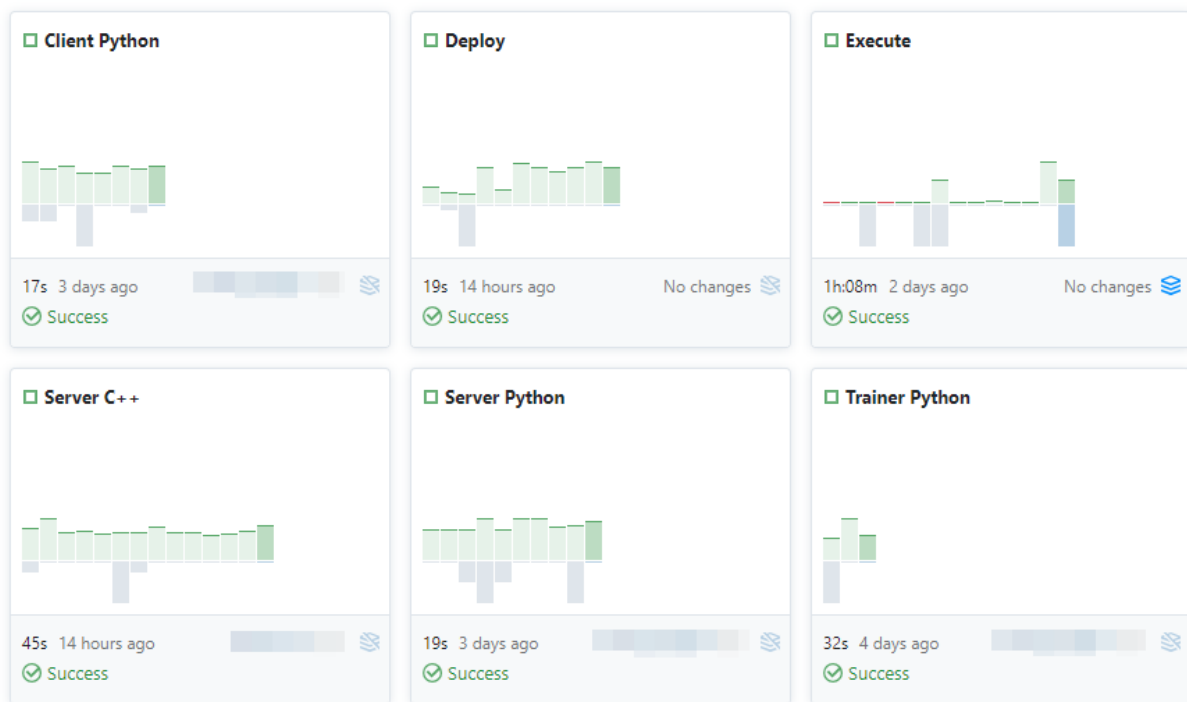


Figure 6: Representation of the build configurations in the TeamCity build tool.

server instances. These tests measure the runtime of the individual steps and save them to be utilized in the evaluation. It is important to note that a step in this process is only restarted if a change has actually been discovered for it or for previous steps. Thus, the classifier is only retrained when new images have been added to the corresponding repository.

5.3 Relationship between the individual Docker Containers

The container, which was called 'Trainer Python' in Figure 6, downloads all existing and classified images from a repository, processes them and uses them to carry out a principal component analysis (PCA). A random forest classifier (cf. [6]) is trained with the resulting features, see Section 4.1 and Section 4.2. These are then saved in a standardized XML format and made available for other containers.

The construction of the 'Server C++' and 'Server Python' containers integrates the XML files created in the 'Trainer Python' container, so that they can always fall back on the latest version of the PCA and the classifier. The 'Server C++' container uses a so-called multi-staged Docker build, since otherwise the required libraries would increase the size of the built Docker container. Afterwards, only the executable file and the required libraries are transferred to a standardized Ubuntu container. As soon as one of these containers is started, a server internally opens a TCP socket on which it waits for incoming connections and listens for incoming images. Any client can now connect, send an image and receive the result of the classification and the internally measured time spans of the individual steps in the process.

To test this on a large scale, the 'Client Python' container, also known as the container 'Test-Client', is created. It contains a Python script that sends images to a server one after the

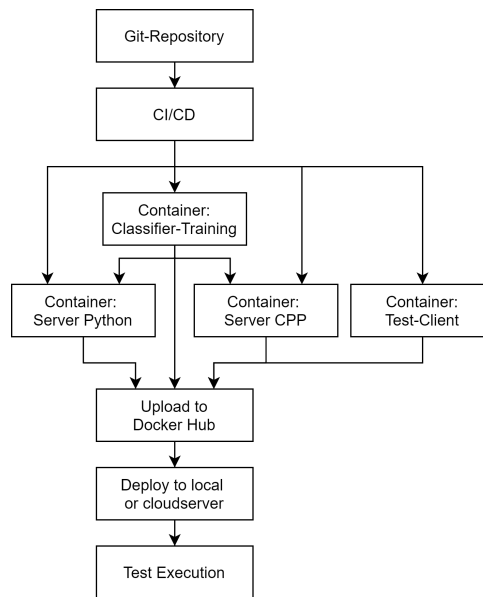


Figure 7: Complete build process for this project in TeamCity.

other and saves the results. It also offers some parameters in the form of environment variables, such as the address of the server or the number of repetitions, which indicates how often an image should be sent to the server. The fully built containers can be downloaded to all test servers from the publicly accessible Docker Hub, which means that tests can be carried out in parallel.

6 Measurements in the Toolchain and Evaluation of Results

In this section, we describe the the structure of the toolchain which is implementing the exemplary end-to-end analysis process of the recommendation system. We implement a live system for performance monitoring and measuring. Then we measure and display the runtime of the individual steps of the process both in Python and C++ implemenations and in Edge-Computing and Cloud-Computing environments.

6.1 Toolchain Structure aligned to an exemplary end-to-end Analysis Process

In Figure 8 we present our toolchain, meaning the sequentially followed steps of our exemplary recommendation system. All referenced steps are executed sequentially in a toolchain. The result of one step is the input of the next step:

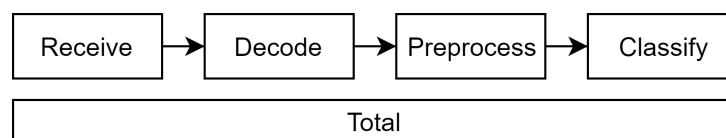


Figure 8: Break-down of the toolchain for time measurement.

Generate First the raw-data is generated, i.e. a picture of the whole planet gear is taken by a camera as described in Section 2.2, and sent to our implemented analysis system.

Receive Time between the successful server client connection and the complete receipt of the picture.

Decode After receiving the picture until after decoding the picture into a grey-scale picture.

Preprocess The retaining rings are extracted from the overall picture of the planetary gear, and prepared for the analysis by the methods detailed in Section 3.

Classify as referenced in Section 4 through the PCA, the extracted parts are reduced to few features. Then the Random Forest Classifier classifies these features into 'correct' and 'rejected'.

Recommendation Based on classification results the system issues recommendations.

Total Total accumulated time of the sequential steps Receive, Decode, Preprocess, and Classify: i.e. time elapsed between the successful server client connection and the completed classification of all pictures.

The runtimes for the different steps in the toolchain are timed separately. In the following we measure and present these in a summary.

6.2 Runtime Measurements of the Toolchain

For the performance measurements of the toolchain already described in Section 6.1, the camera and the sending of the picture were replaced by a test-client. The reason for this was that the same test sample and environment could be used for the local and for the cloud solution.

The test client sends 20 times 104 test pictures for a complete test run. These pictures have been taken in advance and were saved as JPEG on the hard drive. For every single Test, the picture is loaded, then the test client opens a connection to the classification software and sends the picture. The software runs the preprocessing and then classifies the pictures and returns the results back to the test client.

These time measurements have been conducted with the same docker containers for both the local-to-local and the local-to-cloud condition. This guarantees that the used code was identical in both conditions. Only the network architecture, the data transfer via the network and the device the code was run on differed. Figure 9 illustrates the performance measurements from this chapter. It shows the average runtime of all test runs for each step.

6.3 Evaluation: Comparison of Python and C++ Implementation

In the following the measured values collected in Section 6.2 are evaluated, with respect to comparing the execution of C++ and Python. Next, we will compare the runtimes of cloud and intranet. Here, we will take a closer look on the latency to the cloud server.

For receiving the pictures, Python is 8 milliseconds slower than C++. The reason for this could be that Python is interpreted, while C++ is compiled. Python would read the input in a loop in units of 1 kilobyte while C++ is able to natively read all of the data at once. This disadvantage of Python is only relevant on the local scale, because this operation is CPU limited.

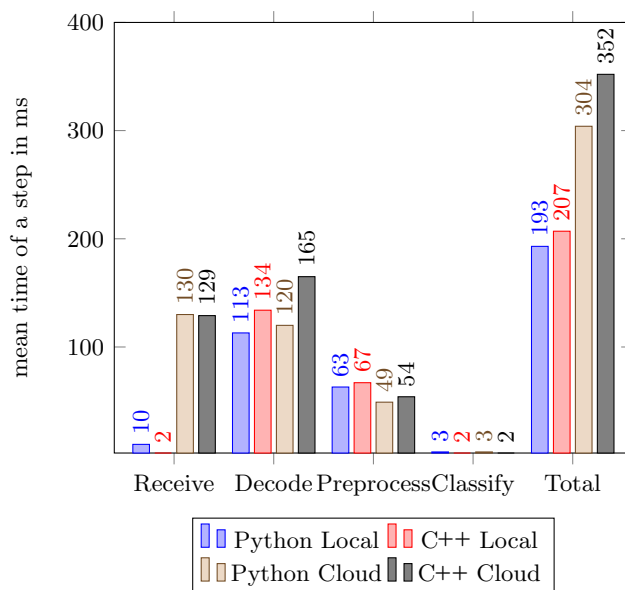


Figure 9: Plot of the mean run times of the single steps.

When transmitting the data via network, the limit is the bandwidth of the network connection instead. In a low bandwidth network, the difference between C++ and Python vanishes.

The preprocessing and decoding stages are completed faster on Python than on C++, despite both using the same OpenCV interface and Python uses the same C++ library like the C++ implementation. The difference in the preprocessing stage is negligible, as it is only 4 milliseconds. For the classification stage, the difference in runtime is nearly identical. The number was rounded to milliseconds for technical reasons. By this operation, a rounding error of $\pm 0.5ms$ is introduced. Therefore we can't make a definite statement comparing Python and C++ in this stage. The difference is in the range of the possible rounding error. Due to the better performance during the preprocessing and decoding stages, the implementation in Python is 14 milliseconds faster than the implementation in C++.

6.4 Evaluation: Comparison of Edge-Computing and Cloud-Computing based Implementations

In this section, the runtimes in the cloud are compared to the local runtimes. Receiving the pictures takes the same amount of time, both on the cloud and locally. On the cloud, the Python runtimes are faster than the C++ runtimes. Compared to a local execution, the difference is noticeable bigger on the cloud. While decoding in the cloud takes longer in the mean, preprocessing is faster. Classification times are the same in the cloud and on the local setup. The Python implementation in the cloud is 50 milliseconds faster than the C++ implementation in the cloud. However, it is also around 100 milliseconds slower than the local setup. The C++ implementation is 145 milliseconds slower than the C++ local setup.

7 Conclusions and Further Work

In this paper we evaluated the performance of Python and C++ implementations for image recognition in a local and a cloud environment. We presented a real world example, for which

we executed an image preprocessing step before the images were used in a random forest classifier. The classifier is able to tell the difference between a correct and incorrect assembled piece.

To automate the tests we set up an automated build system, which builds docker containers and afterwards automatically runs tests on a local and cloud environment. The results are then reported back to the user.

From the runtime differences that were found, it is possible to conclude that the cloud setup is not recommended for time critical applications. Transferring the image to the cloud takes almost the same time as the actual decoding of the picture. For the current implementation, running the software locally in the same network or even on the device taking the pictures is the better alternative. But in this real world example the worker will most likely not be able to benefit from the time saved, as the average human reaction time is not fast enough to react to the information in the small timescale. We identified the following open research points:

- **Image Generation** As already mentioned in Section 2, especially image recognition has a great potential for optimization. A system ready for serial production may lead to a significant improvement. The goal is to produce repeatable images with consistent quality. Ideal for this goal would be a firmly anchored darkroom with separate lighting and a defined stop. This would prevent problems such as changing daylight due to clouds and sunshine, shadows cast by objects or changing the image section. An additional closer look at the camera resolution shows that a lower resolution than the one used (4032×4024 px) is sufficient. It can save a large amount of data and thus further improve the performance of the whole system.
- **Adapting the Processing Chain** As can be seen in Figure 9, a large amount of runtime is lost while decoding the images. An additional part of the time the CPU spends on preprocessing and thus searching and cutting the section with retaining rings. Both operations can be performed with the GPU module built into OpenCV. With this, an existing graphics card can be used for decoding and the computationally and time-consuming search for circles. This would probably reduce the overall time needed quite drastically.

Furthermore, with a small change in the local implementation, it would be possible to access the camera directly using OpenCV. That way, it would be possible to bypass the decoding of the camera images and directly access the raw data of the camera. The result would be a complete elimination of the runtime used for decoding. This is also conceivable if sufficient bandwidth is available in a local network and a so-called fog environment (an on-premises cloud, see [8]) provides the necessary computing power, for example in the form of graphics cards. The fog performance should then be similar to the cloud performance, only in the 'receive' point of the data it should have a much shorter runtime, since the data traffic in the internal network usually has lower latency and higher bandwidth.

- **Selection of the underlying Architecture** In order to save costs and device size for the end devices responsible for image acquisition, the classification software can be outsourced to a fog network if the resulting delays in the classification are acceptable. This means, the local devices requires less computing power and only has to be equipped with a camera and one network module. If several local devices are used, it is even more effective. Within the fog-network, it is possible to run several different types of classification applications. With this solution, a change of the product can take place without much effort at the local devices. A new flashing or programming of the software on the local devices is no longer used.

- **Classifier** Section 4.2 showed that 94% of the retaining rings classified as rejected actually have errors. While only 80% of the as correct labeled rings are truly without errors. Depending on the use-case, it would be more useful if the values were switched. In quality assurance it is important that all incorrectly assembled components are also classified as faulty and less important if correct components are marked as faulty for safety reasons. With the approach used in this paper, a more economical production is achieved because more parts are labeled as correct.

Acknowledgment: The authors would like to thank the company *WITTENSTEIN SE* for providing the material for the fixture and the planetary gears.

References

- [1] Docker Hub, <https://hub.docker.com>.
- [2] Google Trends - Industry 4.0, <https://trends.google.de/trends/explore?date=all&q=Industry%204.0>.
- [3] Google Trends - Machine Learning, <https://trends.google.de/trends/explore?date=all&q=machine%20learning>.
- [4] JetBrains Teamcity, <https://www.jetbrains.com/teamcity/>.
- [5] S. Bhat. *Practical Docker with Python: Build, Release and Distribute Your Python App with Docker*. Apress, Berkely, CA, USA, 1st edition, 2018. ISBN 1484237838, 9781484237830.
- [6] L. Breiman. *Random Forests*, volume 45. Kluwer Academic Publishers, Norwell, MA, USA, Oct. 2001. doi: 10.1023/A:1010933404324. <https://doi.org/10.1023/A:1010933404324>.
- [7] H. Niemann. *Klassifikation von Mustern*. Springer Berlin Heidelberg, 2013. ISBN 9783642475177. <https://books.google.de/books?id=paykBgAAQBAJ>.
- [8] A. M. Rahmani, P. Liljeberg, J.-S. Preden, and A. Jantsch. *Fog Computing in the Internet of Things - Intelligence at the Edge*. 04 2017. ISBN 978-3-319-57638-1. doi: 10.1007/978-3-319-57639-8.
- [9] S. Raschka. *Python Machine Learning*. Packt Publishing, 2015. ISBN 1783555130, 9781783555130.

Alexander M. FRÜHWALD
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
GERMANY
E-mail: alexander.fruehwald@fhws.de

Steffen KASTNER
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
GERMANY
E-mail: steffen.kastner@student.fhws.de

Anna-Maria SCHMITT
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
GERMANY
E-mail: anna-maria.schmitt.1@student.fhws.de

Simon HAAS
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
GERMANY
E-mail: *simon.haas*
@student.fhws.de

Leonhard HÖSCH
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
GERMANY
E-mail: *leonhard.hoesch*
@fhws.de

Lars FICHTEL
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
GERMANY
E-mail: *lars.fichtel@fhws.de*

Christian BACHMEIR
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
GERMANY
E-mail: *christian.bachmeir*
@fhws.de

Bayesian networks with applicability in sonoluminescence

Bogdan-George Gros

Abstract

This project was born out of my curiosity of finding whether Bayesian networks, used in many fields and fulfilling different tasks could be used in explaining undocumented or poorly understood physical phenomenon. One of such phenomenon is sonoluminescence which in essence is the emission of short bursts of light from imploding bubbles in a liquid when excited by sound and can be observed in nature being used by Pistol shrimp to hunt prey. Many theories exist that try to explain this phenomenon and although one was accepted the exact definition still remains unclear. As such I decided that instead of a man trying to explain it, training a Bayesian network to best replicate a phenomenon and in turn try to explain it would be much more efficient. Although this seems like a revolutionary idea, a handful of such AI exist to this day studying diseases and their cures, the properties of the universe and many more but these AI require computing power that not many possess, that is why I decided to build a relatively light AI that can be used by many and can be scalable, the more power it gets the faster and more precise the response. I predict this type of AI will help research in many fields as we continue to stretch the limits of what we know about the world around us.

1 Related works

In the past years AI have been the hottest trend gaining ground in all forms of fields from physics, and medicine to philosophy and art, such technologies have integrated so well in our lives that we don't even notice them. As reminded above, AI trying to obtain information about certain topics in science is nothing new, many powerful AI are running non-stop to try to explain various phenomenon, and although our own Bayesian network seems nothing new, a light AI using mostly the same technology as various web sites for recommending products can show surprising results. As of recent years governments all over the globe have launched numerous research initiatives for artificial intelligence and deep learning. It is foreseeable that in the coming decades, billions of dollars will flow into this field. What new things can AI bring to physics then? As it turns out, a lot. The techniques may not be new as the so called deep learning dates as far as the early 2000 but the sheer computational power can bring a lot in a variety of domains. This has also enabled them to explore entirely new research directions and as more and more people begin to notice the benefits that AI can bring not only to research it is to be expected that we will see AI appearing and being integrated in every niche.

1.1 Sonoluminescence

Sonoluminescence is the phenomenon that can occur when a sufficient intense sound induces the quick collapsing of a cavity inside a liquid. This cavity can be created through cavitation or get

the form from a pre-existing bubble. In a laboratory the phenomenon can be stabilised such that only a single bubble will periodically expand and collapse over and over again, a short burst of light being emitted each time it collapses. For this feat to succeed, an acoustic wave is set up inside a liquid such that the bubble will stay at a pressure anti-node of the standing wave. The container in which the bubble is contained influence frequencies of resonance.

As I reminded in the abstract such a phenomenon is present in nature in the form of the hunting mechanism of the Pistol shrimp which uses the massive energy from cavitation to stun its prey. The exact mechanism of sonoluminescence remains unknown to this day hence why it caught my interest so much. Many hypotheses exist that include: "hotspot, bremsstrahlung radiation, collision-induced radiation and corona discharges, nonclassical light, proton tunneling, electrodynamic jets and fractoluminescent jets". [1,2]

2 Resources

2.1 Bayes' theorem

The main pillar of this project is Bayes' theorem which is used for prediction and learning. In essence Bayes' theorem in probability theory describes the probability of an event, this probability being based on prior knowledge of conditions that might be related to the event. An example of this would be in economics based on a company's history of transactions, profits, fiscal value, domain of work, history of similar companies and many more can be predicted a company's chance to go bankrupt, to expand and so on.

Bayes' theorem is stated as following:

$$P(A/B) = (P(B/A)P(A))/P(B) \quad (1)$$

The formula expresses the following relationship: the probability of the H hypothesis in light of the E data, corresponds to the probability of the E data assuming the H hypothesis, multiplied by the current probability of the H hypothesis, and divided by the probability of the E data.[3]

2.1.1 Bayesian Inference

Bayesian inference is a statistical inference method that uses Bayes' theorem to update the subjective probability of hypotheses based on existing probability and new data. Bayesian inference is used in many fields, such as research, engineering, medicine, philosophy, etc.

Basically Inference is crucial step in my opinion in every machine learning application reliant on data. One of the methods in which Bayes' theorem is used to update the probability is statistical inference. As more evidence or information becomes available the accuracy improving. In dynamic analysis of a sequence of data and in cases in which data is missing Bayesian updating is especially crucial.[4,5]

2.2 Infer.net

Infer.NET is a framework used for probabilistic programming and Bayesian inference. It can be used to solve problems ranging from recommendation and classification to clustering. It is published as open source on GitHub and is a great asset for programmers that are looking for a head start in probabilistic programming as it has all the needed features for any type of Bayesian networks.

Infer.net and many other such open source kits are used in many day to day applications which we don't even notice, that is why I decided to test if such AI could do what the most powerful computers the world has to offer do day to day 24/7. As for everything beyond the groundwork offered by Infer.net I did everything by my own volition, implementing features and modelling

the AI based on research on how those should work and nothing more. The final result may not be perfect but I take pride in the effort put in constructing it.[6,7]

2.3 Docker

Docker is an open software used to implement so-called "containerization" which is basically virtualization at the operating system level.

Docker is defined as a tool that allows you to place the program and its dependencies (libraries, configuration files, local databases, etc.) in a lightweight, portable, virtual container that can be run on almost any server running Linux. Containers and contents operate independently of each other and do not know about their existence. However, they can communicate with each other through well-defined information exchange channels. Using Docker was a great help in portability of the app as the app itself was just a container on a machine I could access the information from wherever I was, in turn granting real time monitoring and optimization of the program.

The process of dockerizing an application is simpler than you would expect, a single docker file incorporating the projects dependencies has to be made and using it a container containing the application can be created.[11]

2.4 React admin

React is mainly a frontend Framework for building admin applications running in the browser, on top of REST/GraphQL APIs, using ES6, React and Material Design. Open sourced and maintained by marmelab. A great piece of software with which you can create web sites that integrate your programs, with which I owe as much in portability as Docker as I was able to monitor everything that happened by just opening the browser, not being forced to have an app in my task bar or locally, being much more convenient and more reliable.

The main design part of this application was realised in React being heavily focused around functionality more than style. Instead of cramming my data into an excel spreadsheet and developing from there all data is displayed in neat graphs that can be accessed from anywhere, not only that but the main data of each experiment and their progress can be seen in real time, making data extrapolation and visualisation that much easier only the critical information being displayed, using graphs and precise numbers but I do have to give credit to React for its sleek default style, although many would see style overshadowing as a downside for myself is a big help.[12]

2.5 SQL Server Management Studio

SQL Server Management Studio is an integrated environment for managing all components included in Microsoft SQL Server. It allows building queries and scripts, includes both a script editor and graphic tools. Perfect for storing the data for the machine learning algorithm to learn and by each trial to accumulate, offering portable, and scalable databases of various sizes and all purposes.

Any sort of data storing service would have worked but I chose SSMS because of its great integration in C Sharp and familiarity. Ultimately this data can be transferred if need be into a MongoDB database but for my intents and purposes SSMS works great.

3 Proposed Solutions

3.1 Implementation-design

The design of this application is heavily focused around functionality more than style, only the critical information being displayed, using graphs and precise numbers but I do have to give credit

to React for its sleek default style, although many would see style overshadowing as a downside for myself is a big help.

For implementation the foundation I used was offered by Infer.net from there on all I had to do was build my own models, and train them based on data, and as straightforward this may seem as tedious the work of putting it into practice was.

As for what my project brings new to the table:

Infer.net and many other such open source kits are used in many day to day applications which we don't even notice, that is why I decided to test if such AI could do what the most powerful computers the world has to offer do day to day 24/7. As for everything beyond the groundwork offered by Infer.net I did everything by my own volition, implementing features and modelling the AI based on research on how those should work and nothing more. The final result may not be perfect but I take pride in the effort put in constructing it.

As for future previsions, the most logical thing to do is to hook up my application to a more powerful machine and see how it fares to the monsters of the industry, changes to the database host to MongoDB could also be an option and many more other features could be added. Also practical implementation is a crucial step in the development of the app which I hope can be soon finished.

In terms of usefulness science will always be happy with more data, and understanding the phenomenon around us is paramount to our development as a species, that being said I foresee a bright future for applications such as this.

4 Source Code

This project was written in C# in the Visual Studio IDE, one of the best of its kind offering many features, like real time debugging, git integration, NuGet packages and many more, overall the best, most solid free IDE existent on the market at the moment.

The credit going to Infer.net again for laying the groundwork for my app and allowing me to build from there without staying and lamenting of formula and their implementation in code.

Thanks to SQL NuGet packages in C# hooking my app to my needed database was as easy as it gets, with just a few lines of code, in the future I might decide to switch to MongoDB services for their better portability. Dockerising my app was equally as simple, with just a few files and some knowledge needed to put my app in a container.

The most tedious task out of all was building the model for my Bayesian network, needing to figure out how each variable influences the others and the result was extremely laborious and time consuming to say the least.

First things first using Infer.net we need two things a model of the phenomenon we try to study and an inference engine offered by Infer.net. Building the model depending on the complexity of the concept we try to study will most likely be the most labour intensive amongst all.

First the inference engine, this will help us to fill any missing data

```
public InferenceEngine Engine = new InferenceEngine();
```

Now for the model, the model has many variables and connections between them but it mainly boils down to:

```
ProbInductancePrior = Variable.New<Dirichlet>().Named("ProbInductancePrior");  
ProbInductance = Variable<Vector>.Random(ProbInductancePrior).Named("ProbInductance");  
ProbInductance.SetValueRange(C);
```

The ProbInductance variable based in previous experiences and

```
Inductance = Variable.Array<double>(N).Named("Inductance");
Inductance[N] = Variable.Discrete(ProbInductance).ForEach(N);
```

The actual Inductance variable that the network will be using for this instance of the experiment. I chose inductance because it's the easiest to show as the inductance is only influenced by length and diameter of the coil, number of turns of wire and its thickness and the magnetic permeability of the core of the coil, all of which but the magnetic permeability are at my discretion to decide and as such it can only influence other variables, not be influenced. With this the network can, based on previous instances, decide the inductance. The inductance can be changed in real time by the network by retracting or pushing in a metal rod in the centre of the coil, thus changing the magnetic permeability of the centre of the coil and in consequence the inductance.

Next we instantiate a new model object

```
SonoluminescenceBayesModel model = new SonoluminescenceBayesModel ();
```

And now we make the model learn the parameters of previous instances

```
model.LearnParameters(sample);
```

Sample is a list containing all of the previous trials results and the LearnParameters function calls for a class with the same name from Infer.net that basically decides the parameters values to whichever it decides to bring the best results and have the maximum chance of the phenomenon producing.

The crucial data is displayed on a react admin page which includes real time graphs, and the variables values for each trial.[12]

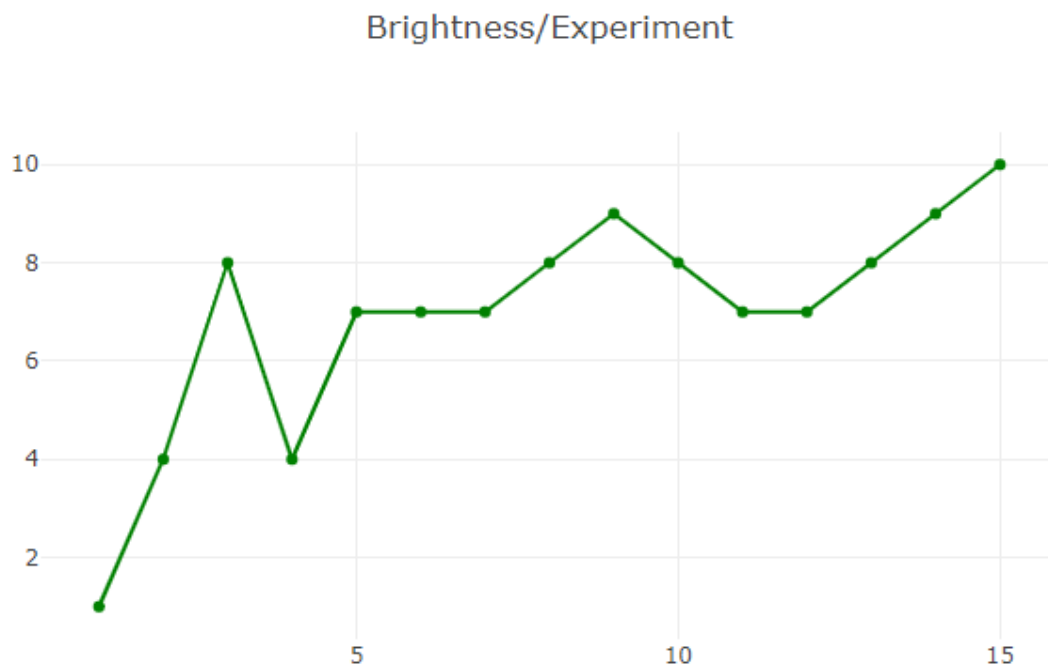


Figure 1: React Graph showing the evolution of bubble brightness (x axis) per experiment number (y axis)

5 Conducting the Experiment

First by measuring the transducer directly we can figure out what its actual capacitance is and then build our circuit around that, since we know approximately what frequency we want, in my case about 27 kilohertz. And now we have a measure of the capacitance, about seven or eight nano Farads. Now all that is needed is to put that into an LC circuit calculator to tell how much inductance is needed to make the circuit resonate in this case it was about four millihenry but that is mostly just informative in the end after tuning I found that I only actually needed about three millihenry. This number can then be taken into an inductance calculator and figure out approximately what size coil will be needed. The inductance of a coil is set by a few different values the first is the length and diameter of the coil, the next is the number of turns of wire and its thickness and the last is something called the magnetic permeability of the core of the coil. Basically how amicable the core is to letting magnetic fields pass through it, the permeability of air for example is approximately 1 whereas the permeability of ferrite or iron can be 300 or higher. Rather than trying to make the perfect coil all we have to do is get it below the value we need when assuming there's nothing in the core except air then we can slide in a piece of steel iron or ferrite to slowly raise the inductance until we hit the value we need. I ended up using 28 gauge wire and wrapped about 200 turns of wire around an empty spool. Then just check the inductance occasionally with a meter. When I hit the value I wanted and could insert a steel rod to get that four millihenry value it was ready.

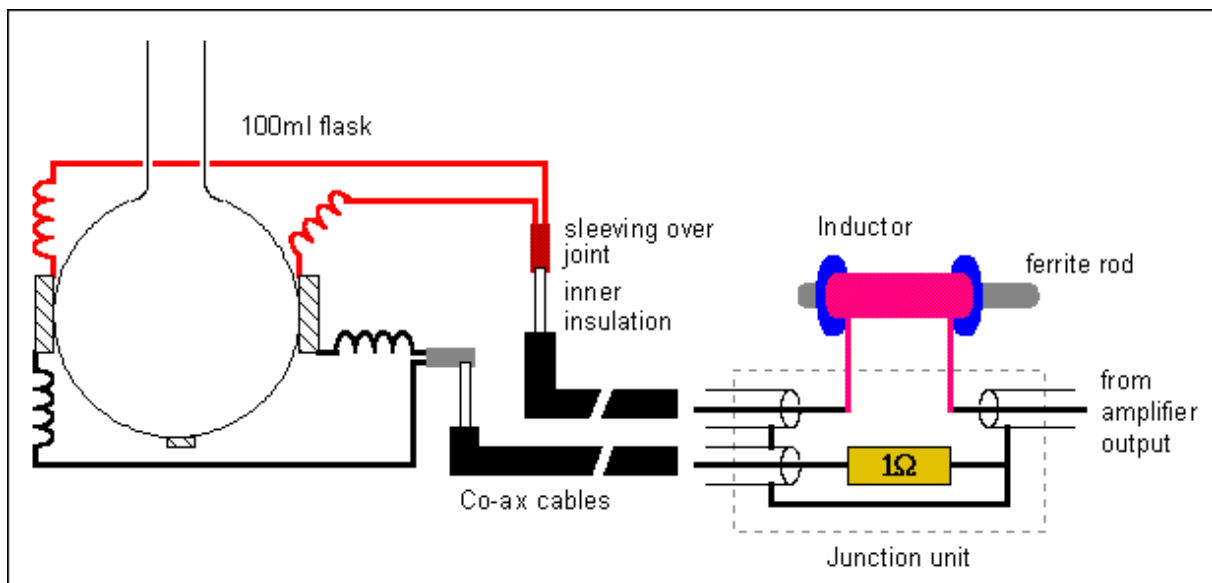


Figure 2: Circuit schema [13]

The frequency generator is fed into the input of the amplifier the output of the amplifier is connected as coax cable and the central line is connected to the inductor we just made. The output of the inductor is connected to another piece of coax which can then be connected to our transducer connect the outer shield's of both pieces of coax together and make sure the connections to everything are solid and there's no shorts. Before we actually connect the transducer though we need to mount it to the flask. For the flask I used a basic water glass to which on the side was glued the transducer and held in place until it cured. Once the transducer is glued I soldered a wire to each of the two electrode pads carefully and then short the two wires to dissipate any charge that built up from the transducer deforming from heat.

The last thing we need to do is degas our water then tip the flask and pour some water in down the side careful not to make any bubbles by accident. If bubbles happen to appear gently tip the flask back and forth to get rid of them fill it so the flask is full up to the neck but not further. Now we start tuning set the oscilloscope to 1 volt per division and the frequency to around 25 to 27 kilohertz. The power on the amplifier should be about half way up now. Watching the oscilloscope, adjust the frequency until the signal is seen and the scope grow as big as it's going to get. There may be several frequencies that do this but check them all and figure out which one gets the highest amplitude on the scope then once you've found the resonance point slide the metal rod in and out of the inductor until again the amplitude is as high as it's gonna get if the signal on the scope goes off scale turn the amp down a bit. If rather than getting a nice clean sine wave weird bumpy peaks appear or the trace isn't particularly clean there's still gas in the water and it needs to be degassed better.

Another test to see if things are working properly is by gently squeezing the sides of the flask if it massively changes the amplitude on the scope then we're near the resonance point now adjust the amplifier until it reads 3 volts peak-to-peak. If this doesn't happen either the frequency is wrong or your inductor isn't at the right value or the water is in degassed properly. When everything is working properly it should be possible to control the amplitude shown on the scope by just throttling the power on the amplifier and should be able to raise the amplitude until it goes all the way off scale. If this is possible it means everything is tuned properly.

Adjusting it back down to 3 volts and it is ready for the bubble. Using a clean pipette or eyedropper suck up a small amount of liquid and let it drop onto the water. It helps to have a light shining through the side of the flask to make it possible to see when this works. If everything is done properly the bubble should immediately seen get trapped mid water. If it shoots off to the side the frequency might be wrong and another frequency has to be found that gives a bigger peak. If the bubble disappears instantly the power is too high and the amp has to be turned down. With the bubble trapped, slowly turn up the power as it approaches about 4 volts will be a point where the bubbles suddenly vanishes. Turning the power down a hair and making a new bubble, if it gets trapped we can to take the picture or at least look at it in the dark.[13]

6 Conclusions

AI can be tedious work, especially when combined with physics and competing with world class technology and minds. But with this project I feel like I moved forward, I learned to persevere and I made something that people would be proud about. In the future I am sure that AI would be the norm even more than today, as I learned during this project AI can give surprising results, especially with fluctuations of sensors, background noise and many other disturbances.

A great lesson to be learned especially from Bayesian Inference is to always update your existing information with new one but never completely change it.

6.1 Speculations

To close out lets look at what happens when the liquid is changed and use something other than pure water. Some books claim that if we replace the water entirely and use concentrated 96 % sulfuric acid the glow is supposed to be 3200 times brighter. It's something worth thinking about but obviously exercise with extreme caution. In theory if it can be done the temperature at the moment of peak compression is much much higher than in water some estimates put it at around 75,000 degrees also it is worth mentioning that anything other than pure water will resonate at a slightly different frequency so re-tuning will be needed on everything to that end. Even just the temperature of the solution also changes the resonance so a re-tuning of the whole system every run to make sure it's all working properly and the colder the liquid is the brighter the glow is. In theory another thing worth trying is dissolving salts in the water and seeing how that

changes the bubbles glow. The one of these that caught my eye was sodium chloride but I don't think I can get a concentrated enough solution to really see any effect. One of the papers shows the glow gets a distinct orange from sodium emission so that's something that could be worth trying if this can be replicated, since the glowing bubble should release UV light a paper suggested that trying to add some fluorescent dye to see if it makes the surrounding water glow.

Up until now we've only talked about single bubble sonoluminescence and normally putting in more power just pops the bubble and ruins the effect but if more and more power is added eventually the result should be get multi-bubble sonoluminescence. Using a lot more ultrasound power it is possible to make multiple bubbles glow though each bubble will be much dimmer than the single bubble we used here. This is also running the risk of exploding the flask if not careful, normally this is done with a ram type ultrasonic probe.

Acknowledgement: This work was supervised by Professor *Crina Anina Bejan*, PhD from "*Aurel Vlaicu*" University of Arad.

References

- [1] Farley J.; Hough S. *Single Bubble Sonoluminescence*.
- [2] Lohse, D.; Schmitz, B.; Versluis, M. *Snapping shrimp make flashing bubbles* 2001.
- [3] Plato.stanford.edu., *Bayes' Theorem (Stanford Encyclopedia of Philosophy)*, 2014-01-05.
- [4] Choudhuri, Nidhan; Ghosal, Subhashis; Roy, Anindya. *Bayesian Methods for Function Estimation. Handbook of Statistics*. 2005-01-01.
- [5] Yu, Angela. *Introduction to Bayesian Decision Theory*, 2013-02-28.
- [6] Daniel, Roy. *Probabilistic Programming*. 2015.
- [7] Ghahramani, Z. *Probabilistic machine learning and artificial intelligence*. 2015.
- [8] Fienberg; Stephen E. *When did Bayesian Inference Become 'Bayesian'?* 2006.
- [9] Joyce; James; Zalta; Edward, N. *Bayes' Theorem*. 2019.
- [10] Koller, D.; Friedman, N. *Probabilistic Graphical Models*. 2014-04-27.
- [11] Barbier, Julien. *It's Here: Docker 1.0* 2019.
- [12] Krill, Paul. *React: Making faster, smoother UIs for data-driven Web apps* 2014.
- [13] <http://www.techmind.org/sl/>

Bogdan-George GROS
"Aurel Vlaicu" University of Arad
Applied Informatics
Str. Elena Drăgoi, nr. 2,
ROMANIA
E-mail: gros.bogdan@yahoo.com

Automation of the Examination Timetabling

Janik Hemrich, Stella Konieczek, Justin Seegets

Abstract

This paper deals with the implementation of an automated examination schedule for the University of Applied Sciences Würzburg-Schweinfurt. Due to a high number of students and a limited number of resources, a large amount of planning was required for each examination phase. In order to reduce the workload and to take advantage of other benefits of an automated plan creation, the general possibility of an automated examination plan creation should be investigated, and a separate solution should be programmed. By researching various sources, approaches and existing solutions for the planning problem were researched and evaluated. The general and specific requirements for an examination plan of the university were identified and an individual solution was programmed in Java. The result of the work was an overview of possible approaches to the planning problems of scheduling in the context of a university. The implementation of the own solution and the mode of operation of the used algorithms were analysed. A comprehensive list of the university's requirements was created and the implementation of these in the programmed software was evaluated.

1 Introduction

The Faculty of Informatics and Business Informatics at the University of Applied Sciences Würzburg-Schweinfurt accommodates an increasing number of students. Each of the approximately 1,000 students of the Faculty of Informatics and Business Informatics are scheduled to take six exams per semester. Resulting in about 4,500 individual exams written each semester. On the other hand, there are limited resources: 14 exam rooms, 40 lecturers and a time frame of only two weeks. Due to the rise in student numbers, among other things, there has been a potential for improvement in the preparation of the examination schedule in recent years. This potential should be used to achieve an increase in the quality of studies at the university within the framework of project work.

The creation of an examination timetable was previously done manually by administrative staff. This process includes rules that enable a general solution but leave details and special cases partly unnoticed. An example of these special cases is the participation in an exam of the first semester by a student of the seventh semester. This leads to inconvenience for teachers or students. One inconvenience for students is the tight timing of exams. An increase is a simultaneous participation of several exams of the same student, which makes it impossible to participate in all exams. Considering the teacher/examiner, frequent travel from their home to the university is an inconvenience that should be avoided. A further disadvantage of manual processing is the amount of work involved, which is shouldered by the research assistant.

The present thesis focuses on the automated generation of the examination schedule for the Faculty of Computer Science and Business Information Systems. The requirements for a customized solution are explained by the dean of the faculty and the administrative staff. Since solutions for comparable problems already exist, they should be considered and evaluated. The goal of the software is to increase the satisfaction of the teaching staff/examiner and the students and to reduce the workload for plan creation.

By researching various sources, the approach to the solution of the planning problem is determined. In addition, existing solutions to the planning problem are collected, which are also evaluated for their usefulness in creating solutions. The demanded requirements for the automated examination planner are evaluated and checked for their feasibility. An own solution is implemented with the programming language Java and evaluated on its achievement of objectives.

The work is structured in 6 sections. Section 2 analyses the current status of automated plan generation and deals in detail with plan generation in schools and universities. Section 3 explains the requirements of the university for the planning problem. Section 4 describes the solution developed by the project team. Section 5 compares the requirements with the developed solution and gives information about the result. The concluding section 6 summarizes the work.

2 Related Work

Although this issue has been examined before by other institutions and project groups, we feel as though our approach is unique. These other institutions have drawn their conclusions using different resources and examining different conditions. We will now describe our approach and how it stands in relation to existing literature.

For the automation of the exam timetabling to be implemented properly, the conditions that determine the examination plan must be reproduced by the software. The requirements of the project specified that the resulting software must calculate an examination timetable from existing data. The process of calculating the precise plan is often solved with metaheuristics, which serves to find an optimal solution to an optimization problem. Since the time spent on the project encompassed 900 hours in total, an independent implementation of an algorithm was not considered. „Renowned timetabling applications lack state-of-the-art timetabling solving methods”,[1] which is why simpler planner applications were excluded, as they often do not offer the possibility to use more complex algorithms like metaheuristics.

Notably, an out-of-the-box solution that would meet all requirements given by the faculty could not be found. Based on this, applications and frameworks which allow for individualization, such as Gurobi, CPLEX, UniTime, Google OR-Tools and OptaPlanner were taken into further consideration.

In addition, there was no budget available for the purchase of a software to aid with project work. Of the above mentioned software UniTime, Google OR-Tools and OptaPlanner can be modified under Open Source licenses. An implementation of our proposed software with Google-OR tools was excluded, due to the high complexity of the software and its documentation. UniTime, in this regard, would have proven to be technically appropriate. Although there were simple possibilities for adaptation and expansion, these did not meet the requirements of the project.

Comparable research by Müller and Rudová shows a successful use and extension of UniTime for Masaryk University. While its focus is based on course timetabling, there are several similarities between their curriculum model and the exam setting in this work. [2]

Finally, OptaPlanner offers extensive documentation and an immense number of extension possibilities. Among other things, it provides an array of different methods to find a solution to Constraint Satisfaction Problems (CSP). However, constraint satisfaction solvers are often

infeasible on their own for examination timetabling.[3] It is for this reason that they are used in combination with Heuristics.[4]

OptaPlanner also supports the use of Exhaustive Search algorithms for solving, but the use of them was ruled out due to their lack in scalability. Construction Heuristics and Metaheuristics are the other options provided by OptaPlanner. While Metaheuristics, such as Hill Climbing, Tabu Search, Simulated Annealing etc., are often used in practice, a combination with Construction Heuristic is recommended.[5] Since Metaheuristics require an existing solution as a base, their sole use was ruled out. An initial examination timetable is generated using Construction Heuristics and an improved solution is found with Metaheuristics. OptaPlanner provides the implementation of this approach in combination with CSPs.

The use of Metaheuristic based timetabling solutions has been heavily studied. Many of these studies concentrate on what are known as so-called “Local Search” approaches.[3] An example of a study that, like ours, aimed to create an exam schedule is that by White and Xie. They evaluated the use of a specific Local Search approach, Tabu Search.[6] The Tabu Search algorithm aims to overcome local optima which prevent further optimisation. Another algorithm which can overcome local optima is Simulated Annealing. Burke et al researched the performance and implementation of the Great Deluge algorithm. This algorithm is similar to Simulated Annealing and Hill-Climbing. Burke et al. also benchmarked the Great Deluge against both.[7] Their results for the Great Deluge were placed among the highest within an International Timetabling Competition, a tribute to this approach’s efficiency.

However, benchmarking different algorithms is not the focus of the project. Deciding upon the most efficient combination of specific algorithms is done individual to the task.[8] In the case of this project the Construction Heuristic First Fit and the Metaheuristic Late Acceptance Hill Climbing showed the best results.

While there is plenty research in the topic of timetabling, our approach offers a new perspective using OptaPlanner. The constraint solver offers flexibility and extensibility, desirable traits required by the faculty for our implementation. We describe how we address these traits in the following sections.

3 Relevance of the Problem

3.1 Overview

The manual preparation of the examination schedule by a member of the university staff has some disadvantages in terms of time and quality. One time-problem is that the employee is busy for about two working weeks to create an examination schedule. An automated creation finds a high-quality solution after only a few minutes and would therefore lead to considerable savings in working time per semester. Qualitative problems arise from the fact that a person can never have a complete overview of which student is enrolled in which exams. The creator of the examination schedule can only orientate himself on the planned semesters and ensure that the individual examinations scheduled for the respective semesters do not overlap. If a student takes an exam from another planning semester or has to retake an exam from a previous semester, there is no guarantee that there will be no overlap. Since the creator, in contrast to an automated solution, cannot try out and evaluate thousands of different combinations, there is no score for the result that indicates the quality of the created examination plan. With an automated solution, the requirements of individual students and teaching staff can also be better met. The iteratively produced result is evaluated by an overall score, which shows how close the result is to an optimal solution.

3.2 Detailed Problem Description

A valid examination schedule must meet certain conditions. A good examination schedule has additional conditions that should be met as often as possible. In the following, some requirements are described which result from the general conditions of an examination phase at a university and which must be implemented. Subsequently, further requirements are described which do not necessarily have to be fulfilled, but which further increase the quality of the examination plan.

First, it must be guaranteed that examinations involving at least one common student are not written at the same time. Furthermore, it must be ensured that the number of students in an exam does not exceed the capacity of the allocated room. In addition, there are other relevant conditions that must be considered individually for each university.

The following requirements must be considered additionally at the University of Applied Sciences Würzburg-Schweinfurt abbreviated as FHWS. Some lecturers are only at the university a few days a week. The examinations of these lecturers may therefore only take place if the respective lecturer is present. The FHWS has several locations in Würzburg where exams of the Faculty of Computer Science and Business Information Systems can take place. If two examinations take place at different locations for a student, it must be ensured that the student has enough time to change locations between the two examinations. Furthermore, examinations that are identical under different study programs must be written at the same time. For example, the exam "Programming I" is taken by both the Computer Science and the Business Information Systems study programs.

If these conditions are met, the next step is to optimize the requirements that increase the quality of the examination plan. For example, large exams should preferably be written at the beginning of the exam phase so that the respective lecturer has enough time to correct them. Finally, as many students as possible should have one or two days off between exams so that they can better prepare for each exam.

From the quantity and complexity of the requirements a computer program can considerably facilitate and optimize the process of creating the examination schedule, since it is not only oriented towards student groups and lecturers but can also respond to individual students. The total score provides information on how often the above-mentioned conditions cannot be met. This makes it possible to choose a plan that is optimized for the largest number of students.

4 Implementation

4.1 Explanation of the Application

The essential task in solving this problem is to arrange the individual exams in time and space in such a way that as few as possible of the conditions mentioned in the previous section are violated. Since the number of combinations among time, space, and exams is very large, not all possible combinations can be tried. Problems of this kind are called Constraint Satisfaction Problems (CSP). In a scenario like this one does not try to find the best of all possible solutions, but to achieve an optimized solution in an acceptable time.

In order to implement this, the requirements for the examination plan must be formalized in constraints. A constraint always describes a state in the examination plan that is either fulfilled or not. Each constraint has a numerical value, which is also called score. The score is the weight of the constraint and thus indicates how important it is to consider this constraint, compared to another one. If a state formalized in a constraint occurs, a penalty is applied to the current state of the examination schedule. The penalty is based on the score of the constraint and affects the overall score of the examination schedule. The overall score provides information about the quality of the solution.

During the initialization of the solution, all exams are distributed spatially and temporally on the exam schedule using a Construction Heuristic. This ensures that a certain initial solution is available, which is further optimized with Metaheuristic. In the section Algorithms the functionality is explained in more detail. After the "Construction Heuristic" has generated an initial solution, all constraints are checked and a penalty equal to the constraint score is applied to the total solution score each time a constraint is violated. Since the first solution is generated very quickly and the fulfilment of the conditions is not the main focus here, the overall score is not very good at the beginning. Using spatial and temporal shifting, distributions are now sought that violate fewer constraints and thus improve the overall score of the examination plan.

Regarding the constraints, the program distinguishes between constraints that must be fulfilled and constraints that should be fulfilled often as possible. Conditions that must be fulfilled are referred to as hard constraints. A solution is only fully valid if it does not violate any hard constraints. Conditions that should be fulfilled are called soft constraints. These conditions are intended to increase the quality of the solution. However, they are only taken into account if no hard constraint is violated anymore. The total score is therefore divided into a "Hard Score" and a "Soft Score". As mentioned above, the importance of the individual constraints is indicated by their respective score values.

Exam Timetable Solver

Generate the optimal schedule for your lecturers and students.

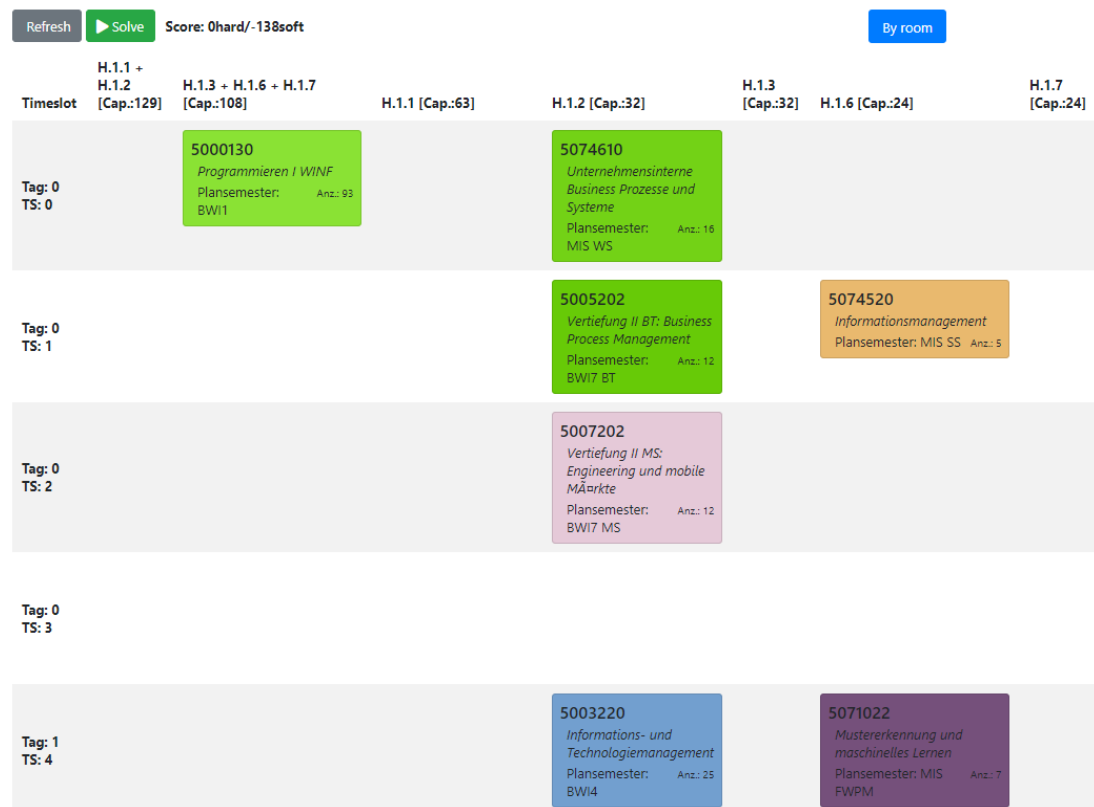


Fig. 1: Graphical Interface

Figure 1 shows a small part of the graphical interface. The columns in the figure represent the rooms and depicts their capacity "Cap." in brackets e.g. a combined room such as "H.1.1 + H.1.2" with the capacity of 129 students. The rows show the day "Tag" on which an exam is written and the timeslot "TS". In this example a day consists of four timeslots. The exams are assigned to the

cells and contain information on the exam number, exam title, semester for which the exam is planned “Plansemester” and the amount of students taking the exam “Anz”.

In the upper left corner the score can be seen. At the moment the assignment does not violate any hard constraints. The soft constraints are broken with a total score of 138 points.

4.2 Implementation and Libraries

The program for the automated creation of the examination plan is implemented in Java, running on Java 8 using Quarkus. We chose Quarkus over Spring since Quarkus is a relatively new framework and we wanted to test the capabilities of this new framework. The data required by the program for the solution is stored in a relational H2 in-memory database. OptaPlanner is used as the constraint solving engine. OptaPlanner accesses the data in the database via Hibernate and creates an examination plan that is continuously optimized. The current best solution is temporarily stored in the database. This allows OptaPlanner to temporarily lower the overall solution score to overcome certain plateaus and potentially arrive at even better solutions. Since all extensions necessary for the implementation of our solution were already provided, there was no need for as to implement any own extensions. The result is provided by Quarkus and RESTEasy via a REST End Point in JSON format as depicted in Figure 2. With the help of Bootstrap and JQuery the solution is presented visually in the form of a table, where the rows contain the timeslots and the columns indicate the rooms.

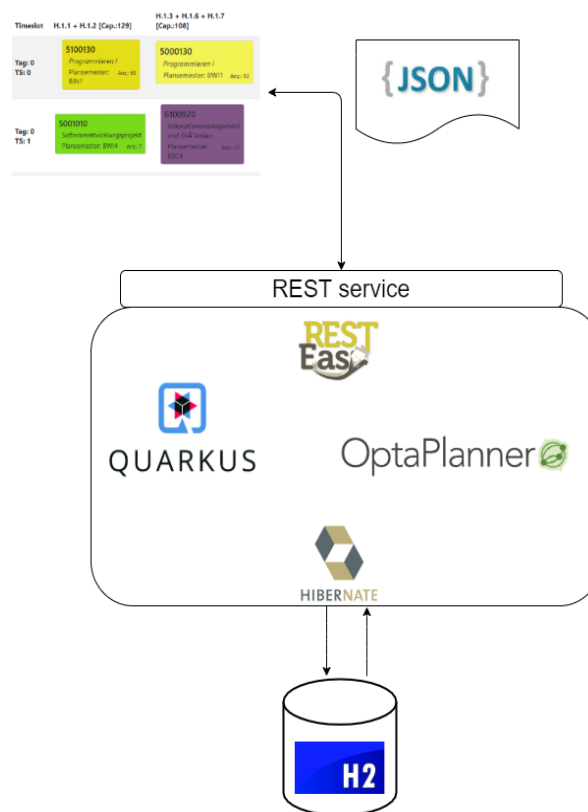


Fig. 2: Software Components

4.3 Algorithms

For the solution of the Constraint Satisfaction Problem different concepts and their algorithms are used in different phases of the solution creation. These algorithms are described in the following.

The concept of "local search" is used for the iteratively optimizing solution of the examination plan. This metaheuristic procedure requires an initial solution and based on this solution, searches for a better one. OptaPlanner supports some local search algorithms, such as Hill Climbing, Taboo Search and Simulated Annealing.[5]

Since the local search needs a first solution, a Construction Heuristic is used before applying a local search algorithm. The goal of Construction Heuristic is to create a quick solution. The validity is put behind. The algorithm used here is First Fit. This simple algorithm iterates through all exams in no specific order and places the exams in the best fitting place in the exam schedule. Exams that have already been set are not moved but are taken into account. The algorithm automatically schedules when all exams have been assigned.[9] This first solution is then passed on to the metaheuristic, local search algorithm. The "Local Search" works step by step. It selects the best of a number of possible moves.

The algorithm used is the Late Acceptance Hill Climbing algorithm. This behaves similar to the classic Hill Climbing algorithm. The Hill Climbing algorithm evaluates all possible moves and chooses the move that leads to the best solution. If there are several "Best Moves" one of them is chosen randomly. But since this algorithm only considers moves that improve the result, there is a high risk of getting stuck in a so-called "local optimum".[10]

This is illustrated by the following example. The deepest place in a mountain range should be found. The algorithm starts at a mountain and searches for deeper places step by step and moves in the direction of the steepest gradient. If the algorithm has reached a valley, it will get stuck there, because it goes uphill on all sides. However, it could be that behind a mountain or hill there is a deeper valley - a better solution. This second valley would never be reached by the classic "hill climbing" algorithm, because the result would have to be worsened to improve it further. To counter this, the classical algorithm can be optimized.

Unlike the Hill Climbing algorithm the Late Acceptance Hill Climbing algorithm accepts a move if it results in an overall score that is at least as good or better than the Late Score. The Late Score is the last score of a list with a fixed length. If a move gets accepted, the score of this move is added at the beginning of this list and the last score gets popped out of the list. Following this procedure, a local optimum can be overcome and the algorithm has a higher chance to produce a better overall result.[11]

5 Software-Evaluation

In order to be able to evaluate the achievement of objectives, the fulfilment of the set requirements is analysed. The fulfilled requirements are explained by means of the programmed constraints. Requirements that could not be fulfilled are clarified afterwards. First the implementation of the requirements in the form of hard constraints is explained, followed by the soft constraints.

A particularly relevant, difficult-to-meet requirement in manual planning is the timing of a student's exams. This requirement was implemented with the constraint `studentConflict`. The constraint compares two exams that take place at the same time. The sets of exams, which contain the student's matriculation numbers, are compared with another method `studentInTwoExams`. This method returns a boolean, which is true if one student number in both sets matches and thus means the constraint is violated.

The requirement to assign lecturers only to matching timeslots is ensured with the constraint `lecturerTimeslotAvailable`. The constraint uses the method `availableAtTimeslot` to check the availability of the lecturer of an exam. The method checks the currently planned timeslot with the available timeslots of the lecturer and returns a Boolean. If the constraint is fulfilled, true is

returned. If the requirement is not fulfilled, the return value is false and a penalty on the total score is applied.

The distribution of the rooms needs several constraints because the requirements for the appropriate distribution are also extensive and include several special cases.

First of all, the compliance with the capacity must be mentioned. To prevent that the capacity of an exam is larger than the capacity of a room, the constraint `roomCapacityConflict` was implemented. The constraint makes a simple comparison of the number of students in the exam and the capacity of the currently planned room. If the number of students exceeds the room capacity, the constraint is violated.

The request of a single exam at the same time in a room is handled by the constraint `roomConflict`. The constraint compares two exams that take place at the same time in the same room. If this happens, the constraint is violated, and a penalty is given.

Certain exams must be written in certain room types. The implementation of this requirement is achieved with the `roomTypeConflict`. The constraint checks if the room type of the exam is zero and if the room type of the exam matches the type of the room. If these conditions are not met, a penalty is given. Besides the binding definition of a room type, the wishes of the lecturers must be fulfilled. These required rooms are usually the laboratories of the lecturers, which are not available for any other exam.

To distribute the rooms correctly, the two constraints `roomPreferredConflict` and `roomNotAllowed` are required. With the constraint `roomPreferredConflict` the preferred room of an exam is compared with the name of the selected room. If the values match, a reward is given that reduces the total score. The second constraint `roomNotAllowed` is needed to prevent the room from being occupied by an exam of other lecturers. A reward is given if the preferred room of the considered exam is zero and the type of the selected room is "special".

To run an exam with a large number of students, room combinations are required. The rooms of the room combination can also be used individually, therefore the constraint `roomCombinedConflict` is needed. This constraint compares two exams that are planned at the same time. If one of the two combinations contains the single room of the other exam, a penalty is given. The simultaneous occupancy of two room combinations is possible with the constraint `roomCombinationConflict`. The constraint therefore checks whether the two planned rooms are combined. Then, the method `isCombination` is called. The sets of the room combination are passed to this method; they contain the individual rooms. The method runs through both sets with `foreach` loops. If there are different room combinations, but one room is contained in both combinations, `true` is returned. Based on these values a penalty is given.

The constraint coincidence code is used to cope with the requirement to hold identical exams of different study programs at the same time. It compares two exams and transfers them to the `coincidenceCodeHelper` method. The method returns a `Boolean`. If both exams have the same coincidence code, but the exams are scheduled at different times, `true` is returned. With this return value the constraint assigns a penalty because the requirement is violated. The request to hold full-day events is handled with the constraint `fullTimeExam`. Since a student can take additional exams on this day, only the attention of the lecturer is important. The constraint therefore compares two exams that are held by the same lecturer. If one of the two exams are saved as a full day course and scheduled on the same day of the other exam, a penalty is given.

A requirement that increases the quality of an examination schedule for students is handled with the `multipleExamsPerDay` constraint. When planning two exams, the constraint checks if their students overlap and if they are scheduled to take the exam on the same day. This case should be avoided, which is why a penalty is given.

Hard constraint	Short description
studentConflict	Penalty if several exams take place at the same time with the same students.
lecturerTimeslotAvailable	Allocation of a penalty for multiple scheduling of a lecturer for the same timeslot.
roomConflict	Allocation of a penalty if several exams take place in the same room.
roomCombinedConflict	Allocation of a penalty if individual rooms and the combination of these or individual rooms are occupied.
roomCombinationConflict	Allocation of a penalty if room combinations with the same rooms are occupied by the same timeslot.
roomCapacityConflict	Allocation of a penalty if the number of students in an exam is greater than the capacity of the planned room.
roomTypeConflict	Allocation of a penalty if the planned room does not correspond to the desired room type.
roomPreferredConflict	Award of a reward if the planned room matches the desired room type.
roomNotAllowed	Allocation of a penalty if no preferred room for the exam exists, but a special room is occupied.
multipleExamsPerDay	Allocation of a penalty for multiple exams on the same day with the same student.
fullTimeExam	Allocation of a penalty if, in addition to an all-day event, a further exam of the same lecturer is to take place.
coincidenceCode	Penalty if exams to be written in the same timeslot are scheduled in different timeslots.

Table 1: List of hard constraints

Besides the described hard constraints, soft constraints are required as well.

The constraints for room planning are extended by the constraint `roomTooBig`. This constraint compares the number of participants in an exam with the capacity of the planned room. If the difference exceeds a previously defined value, the requirement is violated, and a penalty is assigned. A special feature of this constraint is the variable amount of the penalty. With the `examRoomCapacityDifferenceHelper` method, a floor division of the room capacity minus the number of participants with 3 is performed. This should put the weighting of the constraint into perspective.

The constraint `frontLoad` fulfils the requirement to write large exams at the beginning of the examination phase in order to give the lecturers more time for correction. Values are set for the number of participants and the last timeslot for the exam. The constraint then only compares

whether the values are exceeded when planning an exam. If both values are exceeded, the requirement is violated, and a penalty is given. This constraint also has a variable punishment. The method `frontLoadHelper` is passed the selected exam. The return value is calculated with the selected timeslot minus a previously determined integer. The later a large exam is planned, the greater the penalty.

In order to pay less attention to certain timeslots in the planning, the constraint `periodPenalty` was implemented. The constraint checks if a penalty is defined for the planned timeslot. If a penalty is stored, a penalty is added to the total score.

The pause between the exams, for example to change the location, is ensured with the constraint `pauseBetweenExams`. `pauseBetweenExams` passes the sets of students of both exams and the scheduled timeslots to the method `pauseBetweenExamsHelper`. The method checks with the method `studentInTwoExams` mentioned before whether students want to take part in both exams. If the second exam is placed on a timeslot before or after the timeslot of the first exam, `true` is returned and a penalty is given.

The requirement of a break between students' exams is fulfilled with the constraints `oneDayPause` and `twoDaysPause`. Both constraints compare two exams and pass them with the number of break days to the method `xDayPauseBetweenExam`. If `FWPM` is stored for one of the two exams in the planned semester, the requirement is fulfilled and `false` is returned. If both exams of the same semester are on days within the specified time span, `true` is returned and a penalty is applied.

Soft Constraint	Short description
<code>roomTooBig</code>	Allocation of a penalty if there is a large difference between room capacity and number of students in the exam.
<code>frontLoad</code>	Allocation of a penalty if large exams take place on a late day of the exam phase.
<code>periodPenalty</code>	Allocation of a penalty if exams are planned at an unwanted time.
<code>pauseBetweenExams</code>	Allocation of a penalty if no break between exams with the same students is planned.
<code>oneDayPause</code>	Allocation of a penalty if no day break between exams of the same semester is planned.
<code>twoDayPause</code>	Allocation of a penalty if no two-day break between examinations of the same semester is planned.

Table 2: List of soft constraints

In order to demonstrate the efficiency of the solution generation, three runs were performed with the data of the winter semester 2019. The execution time of each run was varied. The data was limited to the exams and students of the bachelor Business Information Systems. This data covers 856 students, who write 54 exams in 12 rooms, including 2 room combinations. The students and exams are in about 5100 relationships.

Table 3 shows the individual exams with the respective duration and the unresolved constraints. Due to the output by `OptaPlanner` only a part of the constraints can be fulfilled. The hard constraints result from the fulfillment of the requirement to occupy a desired space. The total

score of the Hard Constraints is only worsened because of the violation of the room size. However, since this is related to the preferred room, it is negligible. This can be seen in the output of a positive number to the hard constraints. A large number of soft constraint violations are caused by the constraints `roomTooBig` and `periodPenalty`. `roomTooBig` can cause problems if many exams with more participants must be planned. In the current scenario this does not lead to problems. Avoiding certain timeslots is desirable, but not necessary. Therefore, the constraint `periodPenalty` does not have any serious effects.

The table also shows that a solution of the hard constraints in the current scope is possible very quickly, but that the soft constraints do not decrease significantly. After one minute there are 128 points, after 10 minutes 102 points.

Passage	Duration	Violations
1	1 minute	Hard constraints: 9 (fully fulfilled) Soft constraints: - 128 (still to be solved) Indictments: -Programmieren II: - 14 soft constraint because of <code>frontLoad</code> - 5 soft constraint because of <code>periodPenalty</code> -Grundlagen der Wirtschaftsinformatik: - 10 soft constraint because of <code>oneDayPause</code> ...
2	5 minutes	Hard constraints: 9 (fully fulfilled) Soft constraints: - 117 (still to be solved) Indictments: - Programmieren II: - 14 soft constraint because of <code>frontLoad</code> - Mathematik I: - 10 soft constraint because of <code>oneDayPause</code> - 9 soft constraint because of <code>roomTooBig</code> ...
3	10 minutes	Hard constraints: 9 (fully fulfilled) Soft constraints: - 107 (still to be solved) Indictments: -IT-Projektmanagement: - 10 soft constraint because of <code>roomTooBig</code> - 5 soft constraint because of <code>periodPenalty</code> -Innovationsmanagement und Unternehmensgründung: -14 soft constraint because of <code>roomTooBig</code> ...

Table 3: Evaluation of the planning runs

Regarding the termination, we chose a time based termination criterion. This works best for this problem since we can say for sure, that some of the soft constraints will never be fully satisfied. The `roomToBig` constraint for example is implemented in a way that it will never reach zero for all cases. Therefore an acceptable solution for our faculty is a solution which violates no hard constraints and minimizes the amount of soft constraints violated. After running the planner

with altered datasets we found out that after about 45 minutes a solution like this is met and after 90 minutes the planner stops finding significantly better solutions. It rather finds different solutions which violate roughly the same amount of soft constraints.

6 Conclusion

For the automatic generation of an examination schedule for the Faculty of Computer Science and Business Informatics at the FHWS, an application was successfully developed within the scope of the project assignment. The application is based on the constraint solver OptaPlanner. The implementation included data management, the integration of constraints, the integration of OptaPlanner and the Quarkus framework for the calculation of the examination plan as well as the evaluation of the solution and its output. It creates the possibility to save resources in the context of exam administration, ensures easy adaptation of the software to new circumstances and provides a basis for follow-up projects.

Essential for the development was the knowledge of OptaPlanner, Hibernate and Quarkus. Documentation is available for these tools. However, Quarkus is a young framework, which is why the knowledge platforms are still under construction and the scope of the sources is not yet as extensive as for the Spring framework, for example. There are only a few scientific papers dealing with OptaPlanner. Therefore, a project-specific documentation of the third-party software used can accelerate a later adaptation.

During the development of the application, release management was recognized as a possible problem. If libraries are further developed in new versions, compatibility is no longer guaranteed.

A detailed benchmarking would have been useful before the implementation of this project. During the project work, the metaheuristics provided by OptaPlanner were not thoroughly compared in their efficiency. Although tests with different algorithm configurations have been performed, a detailed benchmarking, in which different data sets and constraint configurations are tested, is still missing.

Other research in the field of test scheduling focuses largely on the mathematical links of efficiency, as mentioned in section 2. In these studies, the algorithms used and their performance are examined in detail. While this knowledge can be useful for the further development of the software created, it provides very abstract knowledge. In this paper a simple approach to automate the exam scheduling of a faculty with respect to flexibility and extensibility using state-of-the-art tools is demonstrated. The project team therefore invested a significant amount of effort into finding the most suitable frameworks, OptaPlanner and Quarkus, and implementing the specific requirements of the faculty.

Projects that extend the automated creation of the examination schedule could be the connection of the student portal or the development of a registration platform of available resources. These steps could bundle the information processing and reduce the remaining work steps to a minimum. An improvement of the developed software could be the increase of user-friendliness. Since the focus of the work was different, there is a potential for improvement in the graphical user interface. The modification directly via the user interface would be a considerable increase in quality to be able to make manual changes to the test plan after the calculation.

A related project could be the automation of the lecture plans. As soon as the application for automating the examination plan is mature, it could be built on what has already been developed.

Acknowledgement: This work was supervised by Prof. Dr. Peter Braun, dean of the Faculty of Computer Science and Business Information Systems at the University of Applied Sciences Würzburg-Schweinfurt and Laura Vogel, member of the Faculty of Computer Science and Business Information Systems at the University of Applied Sciences Würzburg-Schweinfurt.

References

- [1] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans et al., “Practices in timetabling in higher education institutions: a systematic review,” *Annals of Operations Research*, vol. 275, no. 1, pp. 145–160, 2019.
- [2] T. Müller and H. Rudová, “Real-life curriculum-based timetabling with elective courses and course sections,” *Annals of Operations Research*, vol. 239, no. 1, pp. 153–170, 2016.
- [3] R. Qu, E. K. Burke, B. McCollum et al., “A survey of search methodologies and automated system development for examination timetabling,” *Journal of Scheduling*, vol. 12, no. 1, pp. 55–89, 2009.
- [4] M. W. Carter and G. Laporte, “Recent developments in practical examination timetabling,” in *Practice and Theory of Automated Timetabling: First International Conference, Edinburgh, UK, August 29 - September 1, 1995. Selected Papers*, E. Burke and P. Ross, Eds., pp. 1–21, Springer Berlin Heidelberg; Imprint: Springer, Berlin, Heidelberg, 1996.
- [5] “Local Search: Overview,” 8/17/2020, https://docs.optaplanner.org/7.42.0.Final/optaplanner-docs/html_single/index.html#localSearch.
- [6] G. M. White, B. S. Xie, and S. Zonjic, “Using tabu search with longer-term memory and relaxation to create examination timetables,” *European Journal of Operational Research*, vol. 153, no. 1, pp. 80–91, 2004.
- [7] E. Burke, Y. Bykov, J. Newall et al., “A time-predefined approach to course timetabling,” *Yugoslav Journal of Operations Research*, vol. 13, no. 2, pp. 139–151, 2003.
- [8] S. PETROVIC, Y. YANG, and M. DROR, “Case-based selection of initialisation heuristics for metaheuristic examination timetabling,” *Expert Systems with Applications*, vol. 33, no. 3, pp. 772–785, 2007.
- [9] “Construction Heuristics: First Fit,” 8/17/2020, https://docs.optaplanner.org/7.42.0.Final/optaplanner-docs/html_single/index.html#firstFit.
- [10] “Local Search: Hill climbing (simple local search),” 8/17/2020, https://docs.optaplanner.org/7.42.0.Final/optaplanner-docs/html_single/index.html#hillClimbing.
- [11] “Local Search: Late acceptance,” 8/17/2020, https://docs.optaplanner.org/7.42.0.Final/optaplanner-docs/html_single/index.html#lateAcceptance.

Janik HEMRICH
University of Applied Sciences
Würzburg-Schweinfurt
Faculty of Computer Science and
Business Information Systems
Sanderheinrichsleitenweg 20
97074 Würzburg
GERMANY
E-mail:
janik.hemrich@student.fhws.de

Stella KONIECZEK
University of Applied Sciences
Würzburg-Schweinfurt
Faculty of Computer Science and
Business Information Systems
Sanderheinrichsleitenweg 20
97074 Würzburg
GERMANY
E-mail:
stella.konieczek@student.fhws.de

Justin SEEGETS
University of Applied Sciences
Würzburg-Schweinfurt
Faculty of Computer Science and
Business Information Systems
Sanderheinrichsleitenweg 20
97074 Würzburg
GERMANY
E-mail:
justin.seegets@student.fhws.de

Towards a comprehensive attack framework against commercial and private UAV

Leonhard Hösch, Max Arndt, Lars Fichtel, Alexander M. Frühwald,
Vitaliy Schreibmann, Helena Schmiedl, Andreas Schütz and Christian Bachmeir

Abstract

Both commercial and private unmanned aerial vehicles (UAV) like drones became popular and widespread over the last years. Seen from an attackers' perspective, this means a continuously growing number of possible targets (UAV) that could be used to generate incidents and damage on purpose. In our work we encounter numerous security-flaws in commercially available UAV-solutions. Consequently, numerous attacks against UAV have been described, making it obvious that commercial and private drones can be hacked. In some cases, no significant effort is required.

In order to improve the security-posture of commercial and private UAV, we propose a comprehensive two-step-approach. First, we describe a comprehensive framework to develop attacks against commercial and private UAV and validate the framework with a detailed attack-vector towards DJI's DT7 remote control. We propose that taking the view of an attacker using our comprehensive framework and identifying vulnerabilities is the basis to come up with a comprehensive security architecture for commercial and private UAV in a second step.

In this paper we focus on the first step. We examine the security architecture and formulate a replay attack against the market leader DJI's DT7 remote control with the help of the proposed framework.

1 Introduction

While aerial drones have been around for a while, private and commercial use of drones has increased during the last years [1]. After some incidents at airports caused by private drones, public awareness for dangers of drones did increase [2]. Several airports had to close down temporarily as drones were spotted in the corresponding air space. Examples of this are at Gatwick airport in 2018 [3] and at Frankfurt airport in 2020 [4]. Also, there have been crashes and near crashes between UAV like drones and manned aircraft. A helicopter was reported to have crashed because of an UAV [5], and a small A100 passenger plane disintegrated a drone near the arrival airport [6].

Especially the impact of safety and security issues connected to UAVs gained attention over these events. As a consequence, several analyzes of drone communication systems have been published, and weaknesses were identified: We cite e.g. the Spektrum drones [7], the DJI Parrot (Lightbridge protocol) and the DJI Parrot Bepop 2 (WiFi) [8]. In this paper, we propose a

universal attack framework against commercial and private UAV, and validate the framework with the examination of the security of the DT7 Remote controller.

The DJI DT7 is the most basic and probably widespread used model that DJI offers [9] and, to our knowledge, has not been analyzed yet. It communicates with a DJI DR16 Receiver [9] on the drone that is connected to the drone’s flight controller via DBUS or Futaba SBUS, which is based on the CAN-Bus protocol [10]. In order to be able to get to an understanding of the security features of the DJI DT7, we followed our proposed framework.

In section 2, we present our comprehensive framework to run structured attacks against commercial and private UAV. Then we apply and validate the proposed framework practically in section 3 with the presentation of the structured finding of a vulnerability & development of a an attack against DJI DT7. In section 4, we discuss results of our found structured attack. Finally, we conclude in section 5 with a summary and outlook.

2 Proposed Framework to Develop and Run Structured Attacks against Commercial and Private UAV

To understand how to attack a drone, we first take a look at the drone from the eyes of criminals, cybercriminals, persons suffering from surveillance through drones and anyone else that could have an interest in avoiding, removing, confusing or taking over a drone. This helps to identify

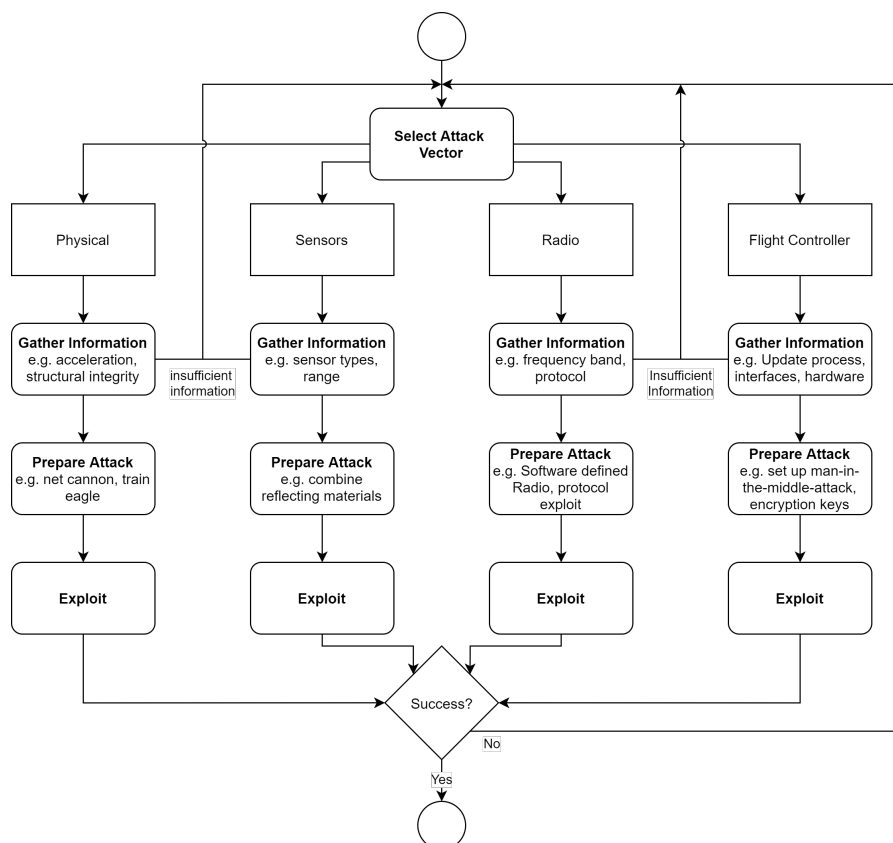


Figure 1: Comprehensive attack framework overview.

categories of attacks and to derive a comprehensive attack framework against commercial and

private UAV, as shown in Figure 1. We propose a model with four steps (Setting Attack Goals, Selecting Attack Vector, Gathering Information, and Attack Preparation) as we outline in the following:

2.1 Setting the Attack Goal

Before we can start to apply our framework, we have to decide on an attack goal that defines the success criteria of a future, planned attack. These goals can vary widely and depend highly on the attack intent. Typically, the following questions need to be answered when defining the goal:

- Do we need full control over the drone, or can we reach our goal by other means?
- Does the drone need to be able to continue flying?
- Is crashing uncontrollably desired?
- Do we need to stay hidden (e.g. against a surveillance drone)?

Some attack vectors can be limited by our attack goal. Crashing a drone might not be an option in highly populated areas or e.g. a demonstration. After the attack goal was decided on, the next step is to select an attack vector for our attack.

2.2 Selecting an Attack Vector

In the framework presented in figure 1, we propose four possible attack vector categories against commercial and private UAV:

- A **physical attack** would be anything that targets the structural integrity like a crossbow, guns but also less damaging methods of bringing drones down like netguns or drones that carry a net and catch other drones with it. Some of these solutions were already patented [19, 20]. Also, drone defense measures like growing dense trees or putting up nets around an area would be considered a physical attack on a drone.
- An **attack towards UAV-sensors** seems to be the best choice for hiding from a drone. Combining materials so that they are no longer perceived by the drone is the general approach here. Other ways to attack the sensors include, but are not limited to, setting up camouflage nets, using one-pixel-attacks against automated pattern recognition and blinding cameras with a laser pointer [21, 22].
- An **attack towards the flight-controller** allows for manipulating flight-maneuvers. One way to run such an attack is e.g. to upgrade the firmware of the flight-controller with malicious code over the air. Also, manipulating the upgrade infrastructure or getting access to a shell on drones that use public wifi falls into this category. Another example of an attack towards the flight-controller would be using an EMP generator in order to damage the flight controllers electronics.
- An **attack towards the radio** is the option of choice when gaining full control over a UAV system is the attacker's goal. It has less prerequisites than hacking the flight controller. Taking over the remote control allows for full control over a UAV [23]. Still, if crashing or just stopping a drone fits to the attack goal, jamming the signal, executing a replay attacks or using authentication methods for a denial of service is achievable with pretty

low investment, for example disconnecting the drone from the controllers wifi [24]. Also, drones often depend on another radio signal: GPS. GPS spoofing is especially hard to detect when a drone flies autonomously, whereas a human operator could realize a spoofed GPS signal rather easily.

2.3 Gathering Information

After deciding on a suitable attack vector, the next step in our proposed framework is to gather information about the target system to be able to develop an attack strategy. Usually, this is the part of an attack that requires deep research and investigation and hence takes the most effort. Lateron, Figure 2 shows the steps for information gathering on an example, using the radio attack vector that we will present in section 3:

- First, information that is publicly available is gathered. This information can be gained with the least effort and is often needed in the following steps. Typical information sources for this step contains data sheets, protocol specifications and product specifications that can be extracted from advertisement material and manuals. For the radio attack vector, we would expect to find the used frequencies and hints on how they are used to comply with European and American regulations. It would be expected to find source code if it were an open source project. If enough information has been gathered, the next steps are skipped and the information gathering is complete. This could happen in this step if the drone uses an insecure protocol and an attack on this protocol is already published.
- Next, information that can be retrieved via access to the system is extracted. This is the information that can be gained from inside the system. Configuration options, update options and firmware files can be obtained in this step. For the radio attack vector, if we can find and reverse engineer the firmware, we have all the information we might need. If not, some configuration options might give us hints how the data packets might be structured or on values that will need to be included. This might later on help in doing known-plaintext attacks. If a vulnerability is found in this step, the information gathering process is complete.
- The last step is the extraction of additional information with the help of black box tests: We consider the system as a black box and analyze the signals and data going in and out. Afterwards, we manipulate inputs, investigate outputs, and systemically conclude what the system is doing. For the radio attack vector, this is our last approach, because it is expected to take a lot of time. We are not guaranteed to gather the needed information here either. If the system would use an encryption that follows best practices, the maximum amount of information gathered here might be that the system is not vulnerable to attacks in this way. Thus, we hope to find that a used encryption is weak, applied incorrectly, prone to known-plaintext attacks or has insufficient protection against replay attacks. Through this research based approach, we can gather information even if the other steps did not provide us with sufficient information about the system.

The final goal of the step *Gathering Information* is to identify at least one vulnerability of the targeted system, that can be used to develop an exploit as briefly sketched in section 2.4. As sketched in our framework, if it is not possible to gather sufficient information to craft the targeted attack, then it is necessary to go back to the beginning and investigate again for another attack vector.

2.4 Attack Preparation and Execution

The final step in our proposed framework is to build on the gathered information, i.e. on an identified vulnerability of the target system. First the attack needs to be prepared, e.g. all needed components need to get in place and set up. Finally the very last step remaining is actually exploiting the found vulnerability.

3 Applying our proposed Framework in a Structured Attack to the DJI DT7

In the following paragraphs, we show the application of our proposed framework in an exemplary attack against the DJI DT7. The DT7 is a seven channel remote controller manufactured by DJI. It usually comes with the DR16 receiver, which is capable of up to 16 channels. DJI is the leading drone seller in consumer drones with 70% global market share [25].

3.1 Setting the Attack Goal

First, we define the success criteria for the planned attack. For our showcase we exemplary define as the goals:

- To make a drone operator loose control of his drone while it is already flying.
- No prior hands on of the attacker on the drone before takeoff.
- The drone operator and other people around him should not be able to tell what happened.

These goals were selected as likely goals a real attacker could also choose.

3.2 Selecting an Attack Vector

Next, we discuss selecting the attack vector, following our proposed framework in section 2:

- A physical attack vector does **not** meet our selected attack goal: A physical attack would be too obvious, other people could likely see e.g. a net or another drone that tries to intercept or crash into the target drone.
- An attack towards the UAV-sensors does **not** meet our selected attack goal: Manipulation of the drone's sensors only can be successful when a drone operates fully or partial autonomous. In our use-case the operator is in manual operation mode only, and the attack obsolete.
- An attack towards the flight-controller does **not** meet our selected attack goal: An attack towards the flight-controller requires prior hands on to the drone before takeoff, e.g. through manipulating a firmware update, through USB access to the drone.
- **An attack towards the radio of the UAV does meet our selected attack goal:** A successful attack on the radio communication would look like the operator loses control of the drone or makes mistakes piloting the drone.

3.3 Information Gathering

In this section, we focus on the information gathering process, as presented in Figure 2. The information gathering is the essential sub process of the comprehensive attack framework mentioned in figure 1. We gather the required information about used frequencies and the protocol, which will become the basis for the identification of a vulnerability:

- First, we investigate for public available information as sketched in section 3.3.1.
- Second, we analyze system access as presented in section 3.3.2.
- Finally, we investigate black box testing as outlined in section 3.3.3, where we present the reverse engineering of the communication between microcontroller and radio chip, extract the radio chip frequency information, and investigate data packets.

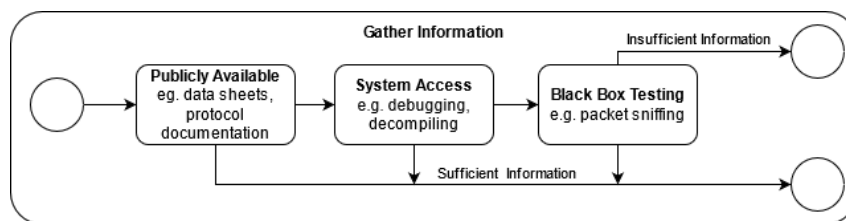


Figure 2: Information gathering process for a radio target.

3.3.1 Information Gathering - Public Available Information

The operating manual from the manufacturer (DJI) gives an overview of the communication system [9]:

- The DT7 system uses Frequency Hopping Spread Spectrum (FHSS) on the 2.4GHz band. By using a broad spectrum of frequencies, this also provides a limited protection against jamming as all used channels would have to be jammed.
- The DJI DR16 Receiver has two antennas and always selects the strongest signal [9].
- The DT7 has two 2-dimensional joysticks that control roll pitch yaw and thrust, two tri-state switches and a wheel that automatically turns back to a standard position. These input devices can be bound to functions by the DJI Assistant Software [18].

A deeper understanding of the system is achieved by opening the enclosure of the RC and analyzing the chips and components mounted on the printed circuit board (PCB), as shown in Figure 3. In a deeper investigation, the scope of functions and the supported protocols of the parts are taken from the corresponding publicly available data sheets:

- **RFMD ML2730DM Transceiver**

The radio chip is placed on the smaller PCB, highlighted in Figure 3 with a red capital B. This card is placed on the underlying MCU board. Both components communicate through a 20 pin connector. The radio chip is generally protected by a metal shield, which has been removed for the purpose of this work. There is a standard antenna port used for the wireless communication located next to the connector used to communicate with the

MCU. The radio chip offers a variety of selectable data rates between 576 kbps and 2.048 Mbps and works in the industrial, scientific and medical (ISM) band between 2.400 GHz and 2.485 GHz [12]. As usual for a transceiver, this chip can be operated as a receiver or a transmitter. A 3-wire control interface is used to control this chip, which is analyzed in more detail in the following.

- **NXP LPC1765FET100 Microcontroller**

The MCU (microcontroller unit) is located on the larger PCB, which is marked in Figure 3 with a blue capital A. This 32-bit Arm Cortex-M3 based MCU offers a high degree of integration through a range of communication interfaces such as general-purpose input/output (GPIO)-Pins, CAN-Bus, universal asynchronous receiver-transmitter (UART), and a few more [11]. In addition to the MCU on this PCB, there are several ports for the peripheral components, e.g., the joysticks and controls as well as the battery.

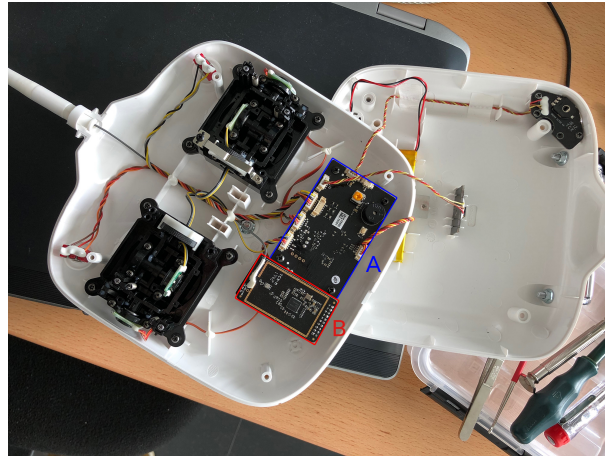


Figure 3: Open remote control with marking of communication components.

In general, the binding process between RC and drone is investigated in this step. For most drones, the binding procedure is only enacted once [9] and it needs to be investigated, whether there could be a weakness in the process that could be used for an exploit. Only if using multiple drones with a single remote control, a rebinding would be necessary.

As in our case the binding process is only enacted once, we skip the investigation as it doesn't fit to our set goal in section 3.1.

3.3.2 Information Gathering - System Access

There are multiple communication interfaces in the UAV that are investigated in this research, as illustrated in Figure 4.

- There is one wireless interface between the radio remote control (which is held by the pilot in the hands), and the radio receiver (located on the UAV).
- All other connections are wired or on the PCB. The LEDs, joysticks, buttons and controls are connected to the MCU via the GPIO pins.

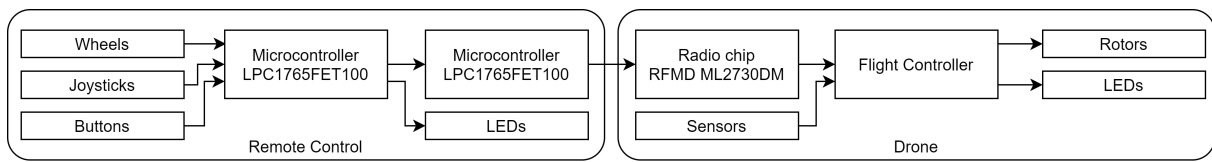


Figure 4: Excerpt of components and communication interfaces in a basic UAV. The left side describes the remote control and the right side the drone.

We also investigated the wired communication interfaces of the microcontroller and the update process of the firmware on the microcontroller. The first investigation yielded the result that interfaces are either closed or are physically inaccessible. The firmware update was encrypted and therefore useless for further analysis by disassembling and decompiling without the corresponding keys. Unfortunately, the keys could not be retrieved from the memory of the microcontroller. While this knowledge does not help to find a new exploit, it is still an important result.

A final investigation delivered results, that can be used in section 3.3.3:

- The investigation of the radio chip showed that no additional functions for encryption or other security features are implemented: the chip just takes the input signal and puts it on the selected frequencies.
- The frequency hopping is controlled by the microcontroller [12].
- Additionally, there was no noise and other communication on the board compared to on the air.

3.3.3 Information Gathering - Black Box Testing

As there was no vulnerability found in the previous sections, we proceed with black box testing, with the goal to get a deeper system insight and finding an exploitable weakness in the system:

- First, we investigate the connection between the radio chip and the microcontroller, in more detail with a reverse engineering approach as sketched in section 3.3.3.1.
- Second, we extract the radio chip frequency information as presented in section 3.3.3.2.
- Finally, we analyze the sent data packets that we could extract between the microcontroller and radio chip as outlined in section 3.3.3.3.

3.3.3.1 Reverse Engineering the Communication of Microcontroller and Radio Chip

Using Reverse Engineering, the following insights of the system was revealed:

- The connection between microcontroller and radio chip utilizes 20 pins, which we analyzed with a multimeter and the datasheet: Table 1 shows, which pin of the PCB-connector is connected to which pin of the radio chip.
- Next to the ground ($VSSD$) and power (VDD), six more pins of the radio chip are used. $XCEN$ and $RXON$ are used to set the operating mode, which can be standby, receive, or transmit. EN , CLK and $DATA$ are used for the serial control bus. Those can be used

to control and monitor the radio chip. The last pin is connected to the chip's DIN pin, which is used as data input for the wireless transmission [12].

- This pinout does not allow to use the chip as a receiver, because the output pins of the radio chip are not connected to any of the 20 pins of the connector plug. The unused pins of the chip are connected to ground.
- Only a stateless unidirectional connection between the remote controller and UAV can be established. Hence there is no backchannel.
- The DR16 selects the strongest signal from its two antennas.
- There are two tasks to accomplish to send data: Configuring the radio chip to the right frequencies and sending the data to the microchip [12].

	1	2	3	4	5	6	7	8	9	10
A				CLK	DATA	EN	RXON	XCEN		
B		DIN					VDD	VSSD	VSSD	VSSD

Table 1: ML2730DM PCB-Connector pinout.

XCEN	RXON	Mode name	Function
0	X	STANDBY	Control interfaces active, all other circuits powered down
1	1	RECEIVE	Receiver time slot
1	0	TRANSMIT	Transmit time slot

Table 2: Table showing the Modes of Operation.

As can be seen in Table 2, for configuration the *XCEN* signal has to be low, while for transmitting and receiving it has to be high. While *XCEN* is high, *RXON* defines the operation mode to either sending or receiving.

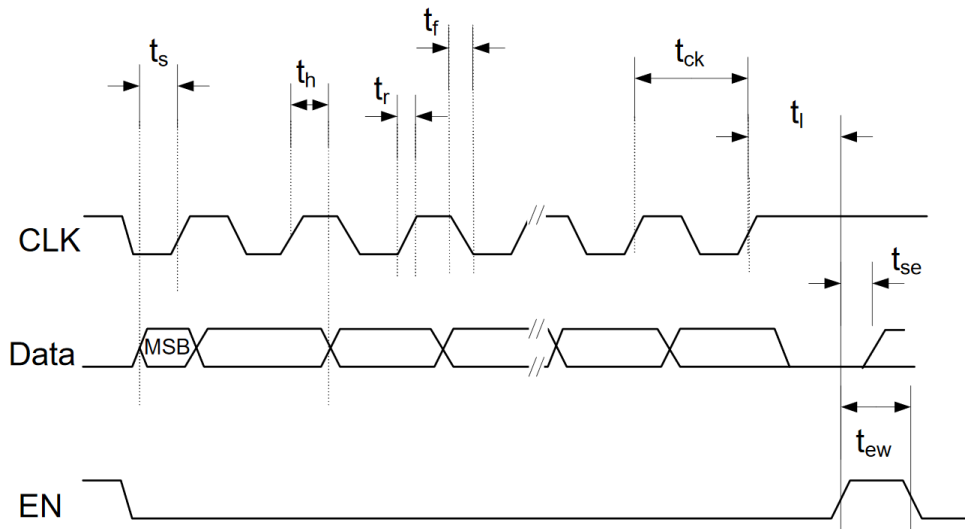


Figure 5: RFMD ML2730 3 wire serial bus timing diagram. Source: RFMD ML2730 datasheet [11].

In figure 5 we see the timing diagram for the 3 wire serial interface of the radio chip. This interface is used to set the configuration registers of the ML2730 radio chip. Beginning with setting EN to low, the data bits are sent, starting with the most significant bit. Every rising edge, the value is read from the data pin. t_{ck} is the clock period with a minimum of 50ns. t_r and t_l describe the clock input rise and fall time with a maximum of 15ns. t_{ew} is the minimum pulse width of EN (enable) with a minimum of 200ns. t_l contains the delay from last clock rising edge to the rise of EN. t_{se} is the enable set up time to ignore next rising clock. t_s and t_h describe the data to clock setup and hold time with a minimum of 15ns, respectively. Setting EN to high after transmitting 24 bit loads the sent values into the appropriate registers.

Bit	23	22	21	20:16	15:0
	0	PLL Frequency Word			
	1	RESET	WE (must be 1 to set write enable)	ADDRESS	CDATA (see table REF)

Table 3: Representation of the Serial Word Format (Frequency or Configuration).

Beginning with the most significant bit, 24 bit words are transferred. As table 3 shows, this can either be a configuration register value or a so called PLL frequency word [12]. This is defined by the first bit. If it is 0 a PLL frequency word. In case it is 1 the 22th bit enables the reset. Bit 21 is the write enable bit (WE). This must be set to 1 to be able to write into the register. Bit 20 to 16 contain the register address and bit 15 to 0 the CDATA.

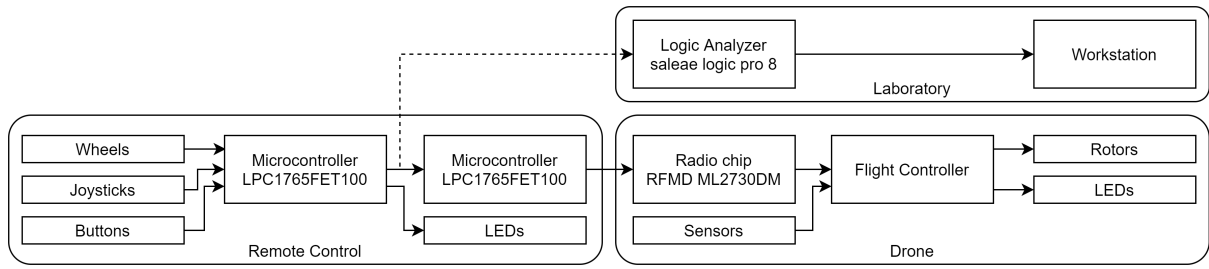


Figure 6: Excerpt of components and communication interfaces in a basic UAS. The left side describes the remote control and the right side the drone. We also added a logic analyzer.

Now that we understand how the packets work, we can add a logic analyzer to the connection between the LPC1765FET100 microcontroller and the ML2730 transceiver. This allows us to intercept the configuration messages as well as the data packets.

3.3.3.2 Extracting the Radio Chip Frequency Information

In this section we investigate the used frequencies, their order and mode of operation. In table 4, we see a complete intercepted configuration of the Registers 0-5 and the PLL frequency word.

Target	Value
Register 0	1110 0000 0000 1010 0111 1110
Register 1	1010 0001 1100 1111 1000 1000
Register 2	1010 0010 0100 0000 1100 0000
Register 3	1010 0011 1000 1000 1000 0100
Register 4	1010 0100 0101 0011 0000 0010
Register 5	1010 0101 1100 0000 0000 1000
PLL Freq. Word	0000 1011 1000 1110 0011 1000

Table 4: ML2730DM register configuration.

Using the data sheet of the radio controller [12], we can calculate the exact values of the used frequencies.

$$f_{ch} = \frac{3}{2} \cdot f_{ref} \left[H + I + \frac{N}{2^{20}} \right] MHz \quad (1)$$

Equation 1 shows the expression for calculating the channel frequency. As register 0 is set to 2, the base frequency is defined as follows:

$$f_{ref} = 12.288MHz \quad (2)$$

The configured rate in register 0 is 2 and the *DIVBASEOFF* in register 3 equals 8. This defines H:

$$H = DIVBASEOFF + 122 = 130 \quad (3)$$

I is defined as the integer part of the PLL frequency word. Bits 3 to 1 represent this value, while the bits 23 to 4 contain the fractional part [12].

Relevant Findings:

- The DT7 uses 36 different frequency words. With this information, we can calculate the used frequencies to start at $2,405.376MHz$, increasing by $2.048MHz$ up to $2,477.056MHz$.
- The ML2730 radio chip is configured every 7 ms.
- Each time, one data packet is transmitted.

3.3.3.3 Data Packets

Finally, we investigate the steps to get access to the data packets and the data packets themselves, in order to detail the vulnerability (and craft an attack on the DT7). With the knowledge of the communication interfaces from previous sections, we capture the transmitted data packets with an oscilloscope and a logic analyzer for further analysis. Here, our goal was to gather information about the protocol in order to be able to understand or identify it.

The following relevant findings could be identified.

- When the registers and the frequency are set up appropriately, the data is sent to the radio chip via the *DIN* Pin. The data is Frequency-Shift Keying (FSK) modulated and sent to the receiver via the antenna.
- With a HackRF One [17] software defined radio, we verify that the data on air is the same as the captured data between the radio chip and the microcontroller. The exact configurations are available on request
- Packets are sent in approximately 1.9ms each and have around 1400 falling and rising edges.
- Packets that follow each other are often identical.
- No counter or timestamp was observed. These might have prevented a replay attack.
- Moving a button or stick on the remote can be attributed to specific places in the transmitted packets.

Additionally, we found more information that we did not need in the next step. Still they are relevant for a security assessment of the DT7 and DR16 system.

- Using two different DT7 controllers with all sticks and buttons in the same (neutral or 0) positions, the packets differed at two locations. One was previously constant, the other seemed to be some kind of checksum or hash of the packet, for which we couldn't reverse engineer the algorithm.
- Also, the unknown checksum algorithm prevents the packets from being tampered with. However, this also guarantees that only valid packets are used. Packets that got disturbed or changed by noise or an attacker get discarded.
- The protocol could be identified as the DJI Enhanced Spread Spectrum Technology (DESST), which is based on the FHSS protocol from the Japanese manufacturer Futaba [14]. Unfortunately, the specifications are not open or available to us.

3.4 Attack Preparation and Execution

In the previous sections, we outlined the results of the data-gathering. We showed that we are able to collect enough data and gain an understanding of the system in order to identify a vulnerability. In this section, we build on this information about the vulnerability and formulate an attack.

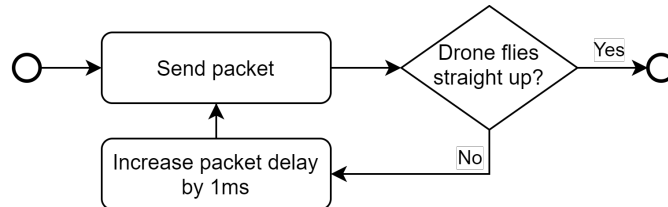


Figure 7: Flow chart showing how to synchronize for a high thrust message.

The attack consists of the following 4 steps:

Gather information about frequencies First, we need to listen to the spectrum and analyze the order of the used frequency channels. This can be done with a software defined radio that supports the relevant frequencies.

Gather packets Next, we need to gather the packets we want to replay. For crashing the drone, we need only one packet. For better effect, we can select one that doesn't have the thrust in a neutral or near neutral position. The more extreme the signal, the better.

Set up hardware Next, we need either a software defined radio or another radio chip and a good and strong antenna. It needs to emit a signal that is stronger than the original signal from the DT7. This way, the DR16 receiver will always select our signal as the strongest one. The stronger the signal, the more range our attack gets. We either use a directed antenna or ignore the government regulations that limit signal strength (most attackers like airport security, police, military or anyone else with ill intents doesn't abide these regulations anyways). Combining both methods works as well.

Synchronizing Now we just need to synchronize our replayed packet with the target's receiver. We need to send in the same slots that the real connection is sending at that time. All slots are filled with the same packet. The procedure for synchronizing a high thrust message is described in Figure 7. For a low thrust, the question would need to be changed to "The drone is falling down?".

The drone's DR16 will then detect a stronger signal on the right frequencies and switches over to that. With the extremely low timings, this will probably have to be done on a realtime capable system.

In our formulated replay-attack, we selected a signal with a non-neutral thrust. As a consequence, the attacked drone will either fly up fast (high thrust) until its battery is empty or it reaches the border of our control range, or it will fall down quickly (low thrust). Hence, we are able to fulfill our defined goals of the attack.

According to our framework, the only thing that is missing is the actual execution of the attack. The proof-of-concept is a task for further research though.

4 Results of Applying our Proposed Framework to the DJI DT7

There was neither enough information available in public to design an attack, nor was it possible to gain access to the system of the microcontroller. However, following our proposed comprehensive attack framework against commercial and private UAV (section 2), we were able to extract and verify the following findings, using reverse engineering techniques during the black box testing to gather the necessary information for a replay attack as described in section 3.3:

- The remote control system uses DJI Enhanced Spread Spectrum Technology (DESST) on the wireless link, a FHSS protocol from the Japanese manufacturer Futaba [14]. The protocol uses frequency hopping on 36 repeating channels. Only the order varies between different remote controllers. Also, DJI Enhanced Spread Spectrum Technology uses a checksum or hash algorithm to verify that packets are valid. This guarantees a very reliable and low latency uplink to the drone. Thus it is very unlikely that this connection is interrupted by accident.
- However, communication from the remote control to the drone is a one way channel link. As the protocol does not implement countermeasures, a replay attack can be conducted with packets sniffed from the air. We suggest using a HackRF One for this.
- Replay packets need to be synchronized to the order and timing of the target DT7 that we want to take the control away.
- The DR16 Receiver always listens to the strongest signal. So if we send the packets with a stronger signal or use a directed antenna to boost our strength in one direction, we can take away control of the drone from the original remote operator.

5 Summary and Outlook

In this section, we present a summary of this work and an outlook to future work.

5.1 Summary

In this paper we described a comprehensive attack framework against commercial and private UAV. Using the framework, we were able to gather enough data about the DT7/DR16 system, so that we could successfully formulate a replay attack. The framework showed to be suitable for this case. It leads through different phases of data gathering. When an approach proves to be yielding insufficient information, it leads to other ways of gathering more information.

5.2 Evaluation & Outlook

Our framework will be tested with other types of attack vectors to evaluate if it needs further adjustments. With this work, we showed a severe vulnerability of the DJI DT7 communication systems. This is only a first step, as the next step would be a full proof-of-concept. The last step would be developing a mitigation of the shown vulnerability.

In order to do this, we suggest the following approach: First, we add a sufficiently large increasing counter to the packet that would pose a simple basic barrier for replay attacks: If the receiver or flight controller would just discard all packets with a lower or equal counter to the last attack, a non-sophisticated replay attack would cease to work.

Enhancing this approach further with encryption would increase the security level: Encrypting the traffic at least partly, and adding a random number to every transmitted packet would make the replay of a packet significantly harder for the attacker.

In this work, we did not fully reverse engineer the protocol, including the checksum algorithm, which is a requirement to fully taking over the drone.

Also, taking over the drone completely could allow the development of a fully autonomous attack toolkit that is able to attack similar drones in real time. Such a framework must know how to detect and breach the drones themselves, the used security measures, especially encryption and simpler countermeasures such as package counters and obfuscation techniques. The development of such a toolkit would require more detailed research in drone communication security, including devices from other manufacturers.

For preventing attacks like these, one might also look into how the discretisation actually works. The remote control has a very limited amount of values it actually uses. For the tri-state switches, only 3 values out of an 8 or 16bit channel are used. This would allow for detecting an attack that does not use the exact same discretisation. For our replay attacks this is irrelevant, as we just reuse the values which were already used by the original remote control.

Still, without implementing autonomous features, a UAV would still be vulnerable to a jamming attack that covers the whole available bandwidth. The default behaviour in such a case does also need further research.

Acknowledgement: This work is funded by a research grant of the German Federal Ministry of Economic Affairs and Energy, as part of the Luftfahrtforschungsprogramm V-3. We also like to thank Prof. Dr. Balzer for his support.

References

- [1] Statista Consumer Market Outlook, "*Drones Report 2020*", Accessed 27. Aug. 2020, <https://www.statista.com/study/78525/consumer-electronics-report-drones/>
- [2] Lykou, G.; Moustakas, D.; Gritzalis, D. "*Defending Airports from UAS: A Survey on Cyber-Attacks and Counter-Drone Sensing Technologies*", *Sensors* 2020, 20, 3537.
- [3] B. Mueller, A. Tsang. "*Gatwick Airport Shut Down by 'Deliberate' Drone Incursions*", *The New York Times* 2018, Accessed 27. Aug. 2020, <https://www.nytimes.com/2018/12/20/world/europe/gatwick-airport-drones.html>
- [4] Simon Calder, "*Drone activity grounds flights at Frankfurt airport, disrupting thousands of travellers*", 2020 *The Independent*, Accessed 27. Aug. 2020, <https://www.independent.co.uk/travel/news-and-advice/drone-frankfurt-airport-grounds-flights-cancelled-diversions-germany-a9369056.html>
- [5] BBC, "*Probe after 'drone made helicopter crash'*", 2018 BBC, Accessed 26. Aug. 2020, www.bbc.com/news/technology-42904204
- [6] Gregory Yee, "*Report: Helicopter crash on Daniel Island may have been caused by drone*", 2018 *The Post and Courier*, Accessed 27. Aug. 2020, <https://www.nytimes.com/2017/10/17/world/canada/canada-drone-plane.html>

- [7] D. Mototolea and C. Stolk, "Software Defined Radio for Analyzing Drone Communication Protocols", 2018 International Conference on Communications (COMM), Bucharest, 2018, pp. 485-490, doi: 10.1109/ICComm.2018.8484821.
- [8] V. Dey, V. Pudi, A. Chattopadhyay and Y. Elovici, "Security Vulnerabilities of Unmanned Aerial Vehicles and Countermeasures: An Experimental Study", 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), Pune, 2018, pp. 398-403, doi: 10.1109/VLSID.2018.97.
- [9] SZ DJI Technology Co. Ltd., "DT7&DR16 RC System.", Accessed 28. Aug. 2020, <https://www.dji.com/dt7-dr16-rc-system>
- [10] International Organization for Standardization: ISO 11898-1, "Road Vehicles - Controller Area Network (CAN) - Part 1: Data Link Layer and Physical Signalling, 2003", <https://www.iso.org/standard/63648.html>
- [11] NXP Semiconductors, "LPC1769/68/67/66/65/64/63" , 2018, Accessed 28. Aug. 2020, https://www.nxp.com/docs/en/data-sheet/LPC1769_68_67_66_65_64_63.pdf
- [12] RF Micro Devices, "ML2730 2.4 GHz Variable Data Rate FSK Transceiver with integrated PA", Accessed 28. Aug. 2020, <https://datasheet.octopart.com/ML2730DM-Micro-Linear-datasheet-8616965.pdf>
- [13] V. Dey, V. Pudi, A. Chattopadhyay and Y. Elovici, "Security Vulnerabilities of Unmanned Aerial Vehicles and Countermeasures: An Experimental Study," 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), Pune, 2018, pp. 398-403, doi: 10.1109/VLSID.2018.97.
- [14] M. Haluza, and J. Čechák, "Analysis and decoding of radio signals for remote control of drones.", 2016 New Trends in Signal Processing (NTSP). IEEE, 2016.
- [15] Hex-Rays SA, "Hex Rays - State-of-the-art binary code analysis", Accessed 28. Aug. 2020, <https://www.hex-rays.com/>
- [16] NSA, "A software reverse engineering (SRE) suite of tools developed by NSA's Research Directorate in support of the Cybersecurity mission", Accessed 28. Aug. 2020 <https://ghidra-sre.org/>
- [17] Great Scott Gadgets, "HackRF - open source hardware for software-defined radio", Accessed 28. Aug. 2020, <https://greatscottgadgets.com/hackrf/>
- [18] SZ DJI Technology Co. Ltd., "DT7 / DR16 Assistant Software", Accessed 31. Aug. 2020, <https://www.dji.com/de/downloads/software/assistant-dt7-dr16-rc-system>
- [19] Peter Govett, "Net gun", Accessed 31. Aug. 2020, <https://patents.google.com/patent/US4912869A/en>
- [20] M. Aagaah, E. Ficanha, N. Mahmoudian, "Drone having drone-catching feature", Accessed 31. Aug. 2020, <https://patents.google.com/patent/US10005556B2/en>
- [21] J. Su, D. Vargas and S. Kouichi, "One pixel attack for fooling deep neural networks", IEEE 2019, doi: 10.1109/TEVC.2019.2890858.

- [22] Jianhao Liu, Chen Yan and Wenyuan Xu, "Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-Driving Vehicles", DEF CON 2016, doi: 10.5446/3625.
- [23] C. Bunse and S. Plotz, "Security Analysis of Drone Communication Protocols", Engineering Secure Software and Systems 2018, doi: 10.1007/978-3-319-94496-8_7.
- [24] S. Raja Gopal et. al., "Deauthentication of IP Drones and Cameras that Operate on 802.11 WiFi Standards Using ESP8266", International Journal of Electronics and Communication Engineering and Technology, 10(2), 2019, SSRN: <https://ssrn.com/abstract=3554188>
- [25] Masha Borak, "World's top drone seller DJI made \$2.7 billion in 2017", Accessed 01 Sep. 2020, <https://technode.com/2018/01/03/worlds-top-drone-seller-dji-made-2-7-billion-2017/>

Leonhard HÖSCH
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
Germany
E-mail: leonhard.hoesch@fhws.de

Max ARNDT
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
Germany
E-mail: max.arndt@student.fhws.de

Lars FICHTEL
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
Germany
E-mail: lars.fichtel@student.fhws.de

Alexander M. FRÜHWALD
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
Germany
E-mail: alexander.fruehwald@student.fhws.de

Vitaliy SCHREIBMANN
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
Germany
E-mail: vitaliy.schreibmann@fhws.de

Helena SCHMIEDL
Friedrich-Alexander-University
Erlangen-Nuremberg
Institute of Psychogerontology
Kobergerstr. 62
90408 Nürnberg
Germany
E-mail: helena.schmiedl@fau.de

Andreas SCHÜTZ
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
Germany
E-mail: andreas.schuetz@fhws.de

Christian BACHMEIR
University of Applied Sciences
Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12
97070 Würzburg
Germany
E-mail: christian.bachmeir@fhws.de

Centralized management web app for real estate developers and HOAs

Felix Husac

Abstract

Homeowners Associations and even big real estate development agencies have always tried to be better at maintaining a close eye over the inner workings of the communities they manage. There may be disgruntled tenants, sloppy, uninterested landlords, or even problems with access to utilities and the condition of the rooms/spaces offered for rent. Most of these problems have a common root: the lack of proper communication. This application allows its users to interact with each other on the platform, making it easy to send messages between users and landlords or administrators, post public announcements, pay the rent or other utilities and even request access to different facilities offered by the association, such as parking places, gym memberships etc. The app has different tiers of membership, that can be used by the landlord to restrict or grant access to different privileges within the app. This is a proof of concept that can be expanded to encompass Smart Home and IoT technologies and even augmented and virtual reality.

1 Introduction

This article responds to the need for a better way to manage medium to large real estate complexes and homeowner associations. The current way of dealing with such tasks is split into multiple apps, that require different accounts, and are bulky and cumbersome for the first-time or inexperienced users. By offering a complete and comprehensive alternative to the old ways of conducting such businesses, this article aims to improve the management side and also the end user experience of the people involved with an HOA or apartment complex. The proposed platform can be improved upon with ease, and can be updated to the latest trends in tech that concern the housing and management markets.

Some real estate managers and management companies sadly still use the “pen and paper” method to manage their balance books, to keep track of their tenants and save important documents. While this doesn’t seem like a big problem for smaller HOA’s, the larger the real estate park, the more documents are needed. This means an increase in volume, in paperwork, in storage size - because paper documents need to be stored securely - and also they require specially trained people to process and file them accordingly. This also means more money spent on salaries, money that eventually comes out of the tenant’s pockets and also affects the profit margins of the property. There is also the downside of more bureaucracy from the end user’s point of view. [1]

Other companies choose to use software solutions to deal with their affairs. Some choose spreadsheets and accounting software for their work, and others use custom made solutions that fit their exact needs. Sadly, none of these solutions provide a very good experience for the tenant. Managers still have to hire secretaries to manually sort through the paperwork, even though this time, it's all digital. No solution currently in use provides the features offered in the proposed application.

There is a gap that can be translated as a business opportunity on the Romanian market for an integrated management system for real estate developers and landlords. For example, a report from 2018, published in “Ziarul Capital”, a Romanian newspaper, states that in Bucharest there are 22.000 HOA's, out of which only 10% are managed by firms and specialised companies, all the rest being left in the care of local administrators and landlords. [2] This figure proves the importance of such a solution like the one proposed in this article. The demand for a greater and greater number of accommodations in larger and developing cities throughout Romania provides the perfect opportunity for such an application to thrive and to be eventually mass-adopted. The fact that most of the landlords also do their own management and accounting for their properties is also a bonus point for the proposed application, because it can help them be more efficient in their work and it provides a reliable and convenient way of dealing with their business.

Another incentive for the development of the application described in this article is the existence of a comprehensive legal framework that governs the activity of HOAs, landlords and tenants. The current legislation provides a guideline for many of the intricate dealings of such a community. [3]

The proposed solution offers a bare-bones platform with enough functionality for a market viable product, while also being capable of adapting and extending to meet the client's requirements. The features currently proposed for the proof-of-concept application are an integrated payment solution, a social platform that allows people to interact directly on the platform and several bulletin boards for important public announcements or advertisements.

The rest of the article is organised as follows. Section 2 contains a description of the business side of the application, a brief overview of the GUI and a short example of use. Section 3 presents the conclusion reached while pursuing this project.

2 Application Design

The application is split into two main workspaces: the “Administrator” panel and the “User” panel. The administrator panel is used by the manager of the HOA or the landlord to supervise and interact with his tenants, to get payment reports and to approve or decline requests made by his tenants for amenities inside the facility. The user dashboard provides a similar functionality, but it is restricted on a tier level access. This means that it can be used as a sanction for late pay and other offences. A user's access to the app or to parts of it can be restricted by downgrading him to a lower privilege level inside the administrator's dashboard.

2.1 Business Logic and Technologies Used

The business logic behind the application is split between the database and its design choices and the PHP implemented functionality.

The application has in mind both the user's side and the admin's side. The administrator can add, remove or edit current users, he can receive invoices and send notices and he can penalize users that are delinquent in payment by lowering their privilege level. He can also chat privately with anyone inside the organization, can post public announcements and can notify people of important upcoming events and maintenance schedules. He also manages the amenities of the facility under his management, offering leases to users.

Users have a limited access to the application, based on the access level set by the administrator individually for any user inside the organization. With the lowest access privilege, the user can only log in to edit his or her credentials, they cannot delete their account and can only visit the “Payments” tab, where they can pay their standing debt and can also download an invoice of financial record for the month in question, in regards to the HOA. With the greatest accessibility level, they have full access to the user side of the application, which means they can also post public messages, they can chat privately with other users and they can send lease requests for the amenities advertised by the landlord.

2.1.1 Database Design and UI Design Choices

The database consists of 8 tables. They are all in relation to one another except the 8th table, that is used as a simple lookup table. They store the data for the public and private message boards, the payment roll, user’s data and login credentials, facility types offered as amenities and also the announcement board.

The database was created with the help of an online visual designer [4]. It is written in MySQL and the server used for implementing it is MariaDB, offered for free from Apache/XAMPP. All the tools and technologies used for the development of this project are a part of the LAMP family.

For connecting the model to the UI, PHP was used for server side processing. No framework was used and the code is written procedurally. At the time of development it was considered the fastest approach, but it can be exchanged for a better, less cumbersome codebase for future maintenance. No JavaScript was used for the user interface, so PHP is also responsible for processing all user generated requests, as well as updating the current page and talking to the database.

Design wise, the approach was to find a balance between complexity and legibility. The app needed to be simple to use, easy to understand by a first time user or an inexperienced, less tech-savvy person. This required larger fonts, contrasting colors on important buttons, an easily accessible navigation bar and a tame background. A pastel color palette was chosen for the visual representation of buttons and backgrounds and a vintage, nostalgic console-type font was chosen for its legibility. This resulted in a pleasant combination of color and information, such that the different interfaces don’t feel crowded but are also well-distinguished from the background and from each other.

2.1.2 Short UI Overview

“My Account” is a common panel both for admin and user. Here, the user data and login credentials can be edited. (Fig. 1)

Pentru a salva modificările, dați click pe 'Salvează'

Nume Utilizator >> Ivan Popescu	Telefon >> xxxxxxxxxx
Adresă >> <input type="text"/>	Email >> popescu_ivan@gmail.com
Parolă >> 1234	Apartament >> Ap. 1
Salvează modificările <input type="button" value="Salvează"/>	

Fig. 1: User’s dashboard

The “Home” panel contains an overview of all the tenants currently living in the complex managed by the admin. The admin can click on the names in the list in the center of the screen, and he will be redirected to a page where he can edit the selected tenant’s credentials and data. He can also add a new tenant, or delete an existing one. The admin also has access to the privilege level settings.

“Add New” is a tab only available for the admin account, and as the name suggests, it is used to add new users.

The “Payment” tab can be used to send new notices to tenants, create PDF invoices and pay via PayPal or create a cash payment request that needs to be approved by the landlord. Also, there are direct links to energy and communications providers that can help tenants to access their own account on those platforms respectively. The Payments tab is common for both admin and user.

The “Private Message” tab is again a common tab both for users and admin. It is used to send and read private messages between any two members of the complex.

The “Notice Board” is used by anyone to share news, advertisements and other information that is considered small advertising. (Fig. 2)

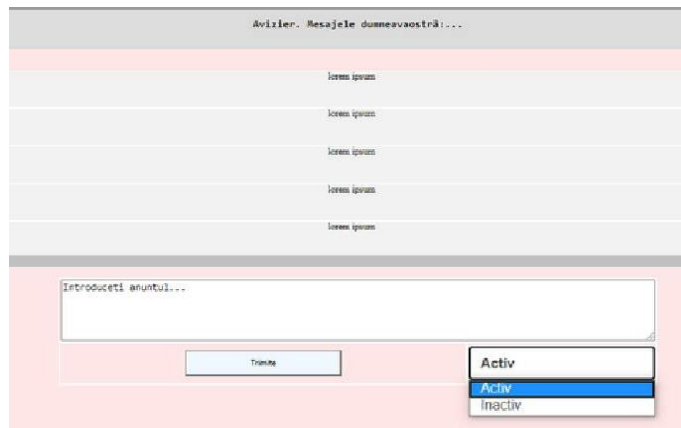


Fig. 2: Public Notice board layout

The “Events” panel is a shared feature, and can be used to schedule and announce important maintenance work, events and other similar occurrences. It is separate from the public message board to underline its importance. Also, the public message board lacks a beginning and ending time-and-date rubric. Each event is also color coded on the basis of its priority, a feature not included in the simple message board. (Fig. 3)

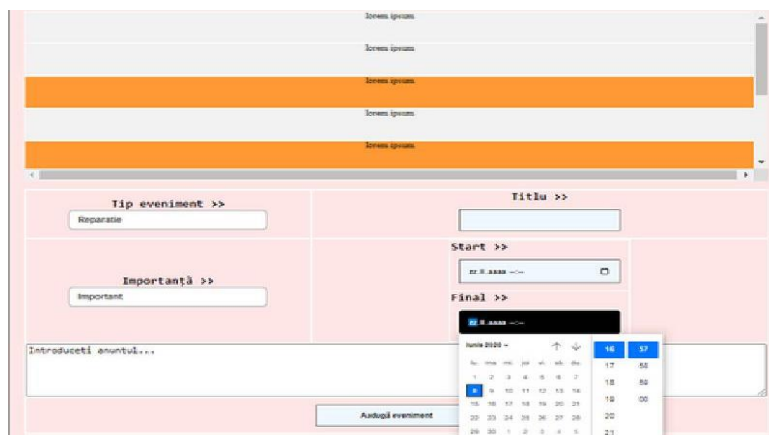


Fig. 3: The Events panel

“Leases” is a panel that has a different layout for admins and users. Admins can accept a lease request, they can delete one, and they can change the price being charged monthly for the lease in question. They can also add or remove new facilities and amenities, such as parking spots, gym memberships and so on for their tenants to choose from and lease. Users can only see the list of currently available amenities and they can issue a lease request. (Fig. 4)

Fig.4: The Leases tab

2.2 Future Developments

The current application is a proof-of-concept that shows what a possible solution to the issues stated above can look like. This means that it can easily be moulded into the desired solution for particular applications and it is flexible enough to be able to support integration with future technologies.

Several directions in which further development can be taken is towards an integration with Smart Home and IoT appliances. From smart light bulbs to smart TV's to heating, everything can be controlled from this app, allowing homeowners to manage their energy consumption and the different appliances in their house. This can also benefit the landlords, because by gathering relevant metrics, they can understand the ways in which their tenants choose to use energy, thus helping to cut costs. [5]

Cleaning and maintenance services, conducted by autonomous cleaning robots, can also be of benefit. They can be useful for cleaning the facilities in which they are deployed after the normal hours of operation, and they can also be used to sanitize and disinfect highly trafficked areas without human interaction. This can prove beneficial if a new epidemic were to arise. [5]

Another area of interest that can be explored in further developing this application can be VR and augmented reality. People can customize and order furniture based on the exact space of their homes, and they can visualize the changes in real time. Another benefit of integrating these

technologies can be in selling, buying or renting a house. The client can preview and take a virtual tour of the property to see if it fits his or her needs. [6]

A very important aspect of owning a property, or lending/renting it is the contract. An important step in creating a complete management solution is to integrate a system to process and store contracts and lease agreements. This is, sadly, dependent on local policies and bureaucracy, so there is currently no easy solution to add this feature.

3 Conclusions

The aim of this article is to propose an easy to use solution that improves the current way of managing HOAs, by offering a comprehensive and flexible platform as a proof-of-concept. A problem of interest for many landlords and managers is to have an easier management solution, to cut unnecessary costs and to keep their tenants content with the services provided. The end goal is to provide a minimum viable product that can help make the business of running a successful HOA easier and more pleasant for both landlords and tenants.

Acknowledgement: This work was supervised by Professor *Mircea Iosif Neamtu*, from “*Lucian Blaga*” University of Sibiu.

References

- [1] *Block Management: The PropTech Proposition*, https://www.bblproperty.co.uk/block-management-proptech-proposition/?fbclid=IwAR3PtVbLZfnqAiJHFnDc9hCUcYQMSAcXQx8NLb_94mP_VjrDRxzWSKXXdbA (retrieved 25.10.2020).
- [2] *EXCLUSIV. Ce se întâmplă în realitate cu banii locatarilor care-și plătesc corect întreținerea. Dezvăluirile unui administrator*, <https://www.capital.ro/exclusiv-cum-dispar-banii-din-conturile-asociațiilor-de-propriet-1.html> (retrieved 25.10.2020)
- [3] *Despre asociatia de proprietari*, <https://fapr.ro/despre-noi/despre-asociatia-de-proprietari/> (25.10.2020)
- [4] *Develop Database Online*, <https://sqldb.com/Home/> (retrieved 25.10.2020)
- [5] *How Technology is Driving the Future of Facilities Management*, <https://softbankrobotics.medium.com/how-technology-is-driving-the-future-of-facilities-management-25908fa7733b> (retrieved 25.10.2020)
- [6] *Lehto launches an ecological block of flats and innovative online shop: for the first time, future residents can design their flats online from interior decor to furniture*, <https://news.cision.com/lehto-group-oyj/r/lehto-launches-an-ecological-block-of-flats-and-innovative-online-shop--for-the-first-time--future-r,c2771630?fbclid=IwAR2QBYmVFkVW-dzJOZ1WUCxG1Pt32NwQx2JvIQtWjyz8xq29VzUZoz75Wx0> (retrieved 25.10.2020)

Felix HUSAC
 “Lucian Blaga” University of Sibiu
 Faculty of Sciences - Informatics
 Dr. Ion Ratiu no 5-7, 550012 Sibiu
 ROMANIA
 E-mail:husacfelix@gmail.com

Automatic recognition of acoustic musical chords

Răzvan-Cosmin Linca

Abstract

The past few years have brought a significant increase in interest for acoustic music, closely related to the parallel evolution of technology globally, with focus on social networks and video streaming. It was found that the evolution determined a distance between performers and the musical sheet, as the process of learning through online tutorials has become much easier. Also, the classical sheet is a musical element with high difficulty, being necessary to know some notions of music theory for a full understanding. Therefore, it proved necessary to use a simple and suggestive notation, specific to acoustic music, namely musical tablature.

Given the existence of these limitations and the desire of guitarists, one solution would be the existence of a platform through which acoustic parts can be transcribed automatically, directly into a tabular representation, using guitar chords.

The first step of the automatic chord recognition system (ACR) is to apply a sound processing method, in order to extract important musical features, by using a suitable representation in the field, namely chromagram. This first step is a vital one in the analysis of a musical sample, as obtaining a correct representation is closely related to the continuous development of the system.

The second part defines the algorithms that underlie the learning processes and differentiate the features of some chords from a musical sample, specifically, machine learning algorithms. The goal is to gradually arrive at a complex and up-to-date machine learning algorithm, able to automatically and independently analyze the audio signal and to classify with high precision each sequence within an acoustic sample.

In order to highlight the good functioning of the ACR, the system will be connected with a mobile application, intended for acoustic music enthusiasts. The application will be able to display, in real time, the results of the automatic recognition of acoustic musical chords, for any desired musical sheet.

1 Introduction

Automatic recognition of musical sheets has been an area actively researched in the field of obtaining musical information for the last 20 years. This category of algorithms is an essential part of many music applications, such as automatic transcription systems for various instruments, learning applications in musical education, or algorithms for recommending music or music genres.

This field follows a precise computational paradigm that involves two distinct steps:

- Step 1: Extract specific descriptors from the audio signal. This stage is called feature extraction. The subsequent extraction and analysis of features are based on algorithms for processing the audio signal. The signal processing domain provides various possible representations of an audio signal that can be used for classification problems, such as recognition of musical acoustic chords.

Let's consider, for example, the use of the Fourier transform in the processing of the sound signal. The process of extracting the characteristics ends with the creation of a suggestive representation for the signal of type *chromagram*. The chromagram provides a sufficiently musically appropriate

description of the audio signal, being used in this case to determine the most likely chord to be played in that interval [1].

- Step 2: Classification of chords based on the representation determined in the first step. A convolutional neural network will be built and optimized, presenting step by step how to achieve and use this machine learning method.

In order to exemplify as clearly and concisely as possible how the algorithm works, a mobile application will be built, which will be able to display, in real time, the results of the automatic recognition of acoustic chords, for any desired musical sheet.

2 Audio processing

2.1 Short-time Fourier transform

The Fourier transform is a type of operation that applies to a complex functions and produces another complex function that contains the same information as the original function, but reorganized according to the frequencies of the components.

It is desired that, using the mathematical principles of the Fourier transform, a representation of the sound wave to be found, on the basis of which useful characteristics can be extracted, applying the Fourier transform or any other derived method (such as the Constant Q transform). The Fourier transform of the function decomposes the signal by frequency and produces a spectrum of it (Figure 1).

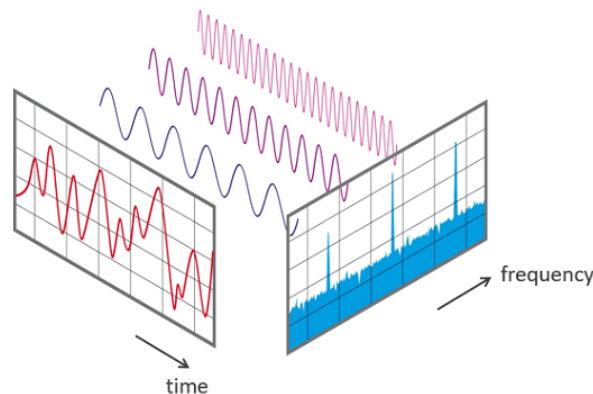


Figure 1: It can be seen in this figure how a time-dependent wave can be mapped into a frequency-dependent representation, by applying the Fourier transform. [16]

The principal property of the transform is given by the reorganization of information by frequencies (temporal, spatial or otherwise), being extremely useful in processing signals of various types, in understanding the properties of a large number of physical systems or in solving equations and systems of equations [3]. Also, the Fourier transform allows the analysis of the signal and the identification of certain desired frequencies, in order to amplify or suppress them, and thus, determining the synthesis of a new signal.

Endowed with these properties, but also with others, Fourier transform is present in many modern technologies and fields (the last field in which it found its applicability being quantum mechanics), because it is more reliable and more robust than other technologies of signal analysis and composition of spectrum.

The main disadvantage is the difficulty and complexity calculations that determine the whole process. In the case of integrating the Fourier transform into a software program, the programmer must translate the process into an efficient algorithm and be aware that some calculations use the processors in an intense way.

2.2 Chromagrama

Chromagram is the representation of a group of features that a program based on one of the audio processing algorithms extracts from an audio signal. The representation is based on the root note profile, known as pitch class profile (PCP), which is a descriptor in the context of a chords recognition system.

The chromagram consists a sequence of characteristic vectors, each having the role of measuring the relative intensity of each note, relative to each frame, in a time interval. This results in a pitch class versus time diagram, i.e. an image of the audio signal, which contains conclusive information that can help determine the most likely chord.

There are different methods in the literature to construct a chromagram. The main steps stated by Bello and Pickens in [1], will be used as a working basis in the processes of this algorithm. Steps for composition of the chromogram are as follows:

- Converting the audio signal into an intermediate representation, called spectrogram, by applying a derived or optimized Fourier transform;
- A filter is applied on the frequencies within the spectrogram, to fit in the range 100Hz - 5000Hz;
- The profile of the root notes is built, based on the estimated frequencies. It is a procedure to determine the pitch class, from the values of the frequencies;
- A normalization of the characteristics is performed, frame by frame, by dividing with the highest value, in order to eliminate the noise, resulting into the image of the chromogram. Figure 2 shows the chromagram obtained by applying the steps on a musical chord.

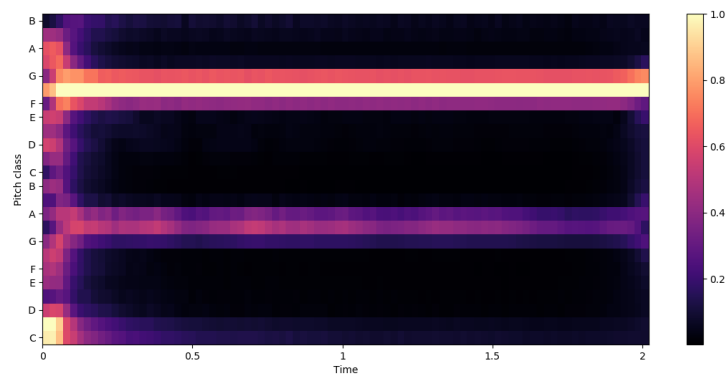


Figure 2: The image obtained by constructing the diagram of chord A, graphical representation using LibROSA and matplotlib.

3 Experimental results

3.1 Description of the proposed solution

A convolutional neural network (CNN) will be used to build the acoustic chord recognition model. CNNs have grown in popularity in recent years, especially in the field of computer vision.

In convolutional neural networks, the constructed or chosen image is used as an input parameter for CNN, and the image is passed through several layers, such as convolutional layers, pooling layers, and activation functions. Convolutional layers are constructed of several filters, which can be interpreted individually, each one learning some superior features of the image.

In the problem of automatic recognition, the detection of chords can be treated similarly to image recognition, because images with chromagram representation can be created. Identifying notes or chords is simpler than classifying images because they do not involve learning certain textures, rotations, or

resizing. However, this problem comes with other challenges. The musical notes in the chromogram are not located in a single region, in the same way that most objects in images are - a note at a certain fundamental frequency will be composed of harmonics at multiples of that frequency.

Despite this challenge, CNN has advantageous properties that can be successfully applied to this issue. Previous experiments have suggested that aggregating information by considering several musical frames from the same sample results in a higher performance prediction. Thus, the convolutions applied to the input data allow the created model to learn valuable polyphonic musical features.

3.2 Dataset

The dataset used to train and evaluate the created convolutionary model consists of the combination of 3 datasets for guitar, available online, on which a series of augmentation algorithms were applied. The dataset is thus composed of:

- 7398 individual chords, grouped in 16 wav audio files, belonging to the Fraunhofer Institute of Musical Semantics, within the Technical University of Ilmenau, Germany. The dataset is called IDMT-SMT-CHORDS;
- 6580 audio recordings, each consisting of a single chord, belonging to the music research laboratory of New York University, USA, called GuitarSet [15];
- 200 audio files for 10 types of chords, being collected by the Motefiore research group of the University of Liège, Belgium.

After applying the augmentation algorithms to the complete dataset, a total of 58.577 wav audio recordings were obtained, grouped into the 24+1 classes (24 chords with known appropriate tags, and one tag for any other chord that it does not fit).

The 24 chords recognized by the system, together with the associated numerical values are the following (musical chord-associated value): A-0, A#-1, A#m-2, Am-3, B-4, Bm-5, C-6, C#-7, C# m-8, Cm-9, D-10, D#-11, D#m-12, Dm-13, E-14, Em-15, F-16, F#-17, F#m-18, Fm-19, G-20, G#-21, G# m-22, Gm-23. For any other chord that does not fit into this sequence, the pair N-24 was created (unknown chord N, with the associated value of 24). The distribution of data by classes can be seen in Figure 3.

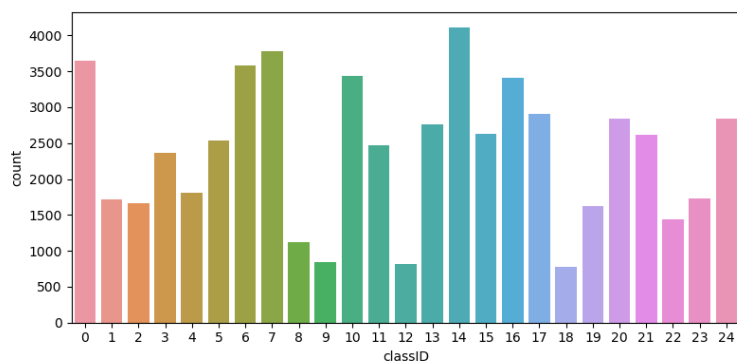


Figure 3: Distribution of data by class, specifying the number of audio files available for each class associated with music chords.

3.3 Extracting musical features

The extraction of musical features is based on one of the sound signal processing algorithms, more precisely, the Constant Q transform.

In order to determine the chromogram using the Constant Q transform method, the processing methods provided by the LibROSA library were used. Thus, the transform method receives as parameters

the wav audio file, a value for hop length equal to 512 and an optional value for the number of distinct semitones at the level of a musical octave, equal to 24.

The accepted time for an audio file (which is the recording of a music chord) is 2 seconds. Thus, knowing the parameters applied to the transformation and the duration allowed for an audio file, the constructed chromagram has a standard size of 24x87, i.e. each of the 24 characteristic vectors has 87 double values in the range [0, 1]. If the record is less than two seconds, values of 0 will be added to the characteristic vectors until the standard size is reached.

3.4 Detection of musical transitions

The automatic musical chord recognition system is built and trained for individual chords. In order to apply the system to complex recordings, which contain a number of different chords (changed frequently), it is necessary to detect exactly when the transition from one chord to another is made (localization). If those points are discovered on the time axis, the recognition algorithm can be applied to all intervals determined by consecutive points.

LibROSA offers a classic detection of musical transitions method, based on the paper of Bock, Krebs and Schedl [2]. The method consists of 3 steps:

- Compose a function that denotes local changes of properties signal, such as energy. It is known as the spectral novelty function.
- The peaks are found in the function of spectral novelty;
- Backtracking is applied from each peak to a previous local minimum to find the segmentation points, so that the transition occurs shortly after the beginning of the segment [13]. Figure 4 shows the peaks of musical events discovered by applying the detection algorithm to an audio recording.

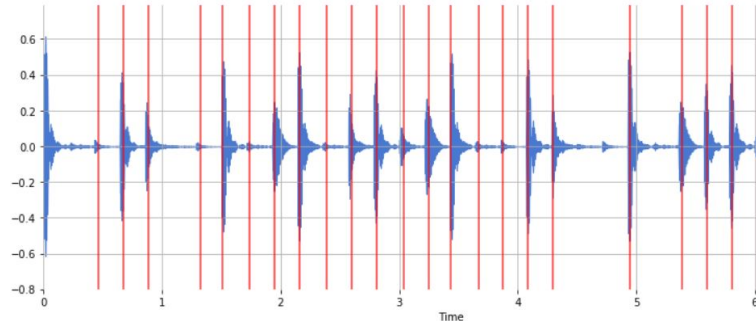


Figure 4: Detected music transitions (represented by red lines) for a recording of about 6 seconds (represented as a blue sound wave) [13].

Since the segments determined by the transitions between musical events discovered, following the use of the algorithm are exact and clearly mark the time intervals with different musical events (musical chords), this algorithm is applied to acoustic musical pieces or recordings, lasting more than two seconds. Thus, the detection of musical transitions is performed as follows:

```
# Consider a complex audio recording
audio_file = "//Let it Be Strum Guitar Cover.wav"
# Load audio file
song, sr = librosa.load(audio_file)
# Compute the frame indices for estimated onsets in a signal
onset_frames = librosa.onset.onset_detect(song, sr=sr,
                                          backtrack=True)
# Convert onsets to units of seconds
onset_times = librosa.frames_to_time(onset_frames)
```

3.5 Creation, training and evaluation of the convolutional neural model

Having the dataset processed after extracting the musical features for each audio file, the architecture of the neural network and the values of specific parameters remain to be established, before the start of the learning process.

The architecture of the constructed neural network is presented in Figure 6, being specified each layer, together with the related parameters or the necessary dimensions.

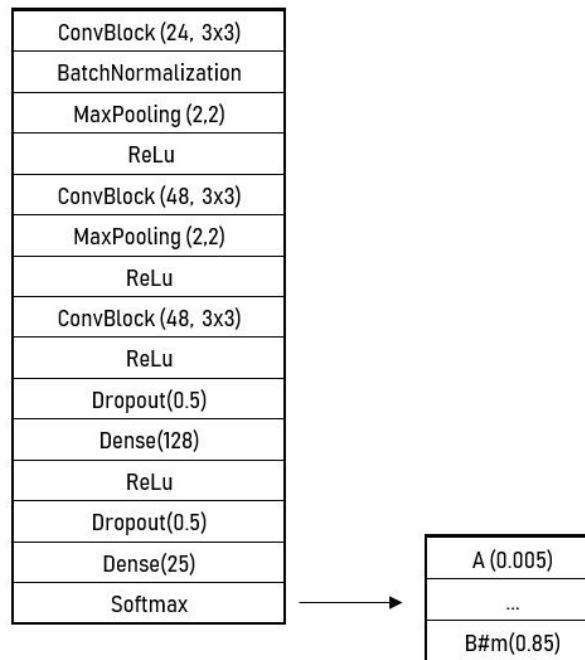


Figure 5: The architecture of the convolutional network built. Number of filters and the size of the convolution core are indicated in parentheses for each ConvBlock. The softmax activation function is used in the last dense layer.

For training and testing, the dataset will be distributed randomly as follows: 80% of the total number of instances will be used for training, while the remaining 20% will be used for testing the convolutional model.

To determine the error or value for loss, a cross-entropy-based strategy was used. Cross-entropy compares the prediction of the model with the value of the corresponding label. Its value decreases as the prediction becomes more accurate. It becomes zero if the prediction is perfect [4]. Thus, cross-entropy is a loss function to create a classification model.

The number of epochs is 25. This value determines how many times the system will repeat the data training process correspondingly, in order to learn from one stage to another, becoming more and more precise, with a smaller and smaller error.

Regarding the optimization algorithm, it was chosen the Adam optimization algorithm, with a batch size of 64. Adam is a method of adapting the learning rate, which means that the learning rate is calculated individually for different parameters [4].

With all these values known, understood and set appropriately for the convolutional neural network, training can begin successfully. The training method used in the project is shown in the following code section.

```

import keras
import logging
import os

def train(self, X_train, Y_train, X_test, Y_test):
    logger.info("Start training cnn model")

    # Compile model
    self.model.compile(
        optimizer="Adam",
        loss="categorical_crossentropy",
        metrics=['accuracy', precision, recall, f1measure])

    # TensorBoard Logging
    log_dir = os.path.join(AUDIO_CONSTANTS.LOGGING_PATH, datetime.datetime.now().
                           strftime("%Y%m%d-%H%M%S"))
    tensorboard_callback = keras.callbacks.TensorBoard(log_dir=log_dir,
                                                       histogram_freq=1)

    logger.info("Tensorboard Logging Started")

    # Start training
    self.__history = self.__model.fit(
        x=X_train,
        y=Y_train,
        epochs=25,
        batch_size=64,
        validation_data=(X_test, Y_test),
        callbacks=[tensorboard_callback,
                  EarlyStopping(monitor='val_loss',
                                verbose=1, patience=2)])
    logger.info("Training completed")

```

3.6 Results

Several calculation metrics will be introduced and used to evaluate the performance of the convolutional model. The functions associated with some calculation metrics are similar to the functions for managing the loss, except that the results of the evaluation metrics are not used to improve the model. The metrics used were accuracy, precision, recall and F score. These metrics were applied at the compilation of the convolutional neural model. The results obtained are presented in Table 1.

Stage	Accuracy	Precision	Recall	F score
Training	97.23%	98.38%	95.85%	97.05%
Testing	89.75%	93.66%	86.67%	89.98%

Table 1: Obtained results after the two stages: training and testing.

Observing and interpreting the results highlights efficiency and the good construction of a convolutional neural network, in the context of automatic recognition of musical chords. All calculation metrics have values over 89%, successfully avoiding overfitting. The high precision in recognizing and differentiating chords is due to a correct convolutionary architecture, with properly placed layers, the network having sufficient depth. The good results are also due to the use of a suitable representation, to successfully learn the most important features at every step.

To observe the evolution of the training and testing processes, and the impact of the whole process on the calculation metrics, two graphs were made to highlight the decrease of the loss and the increase of the accuracy. Thus, Figure 6 shows the values that the loss had, but also the accuracy, during the 25 epochs. A relatively exponential decrease is observed, the values of the loss at training, respectively at testing being close. Regarding the values for accuracy, there is an upward trend, the values for training and testing being close in the last epochs (suggesting a correct and balanced learning).

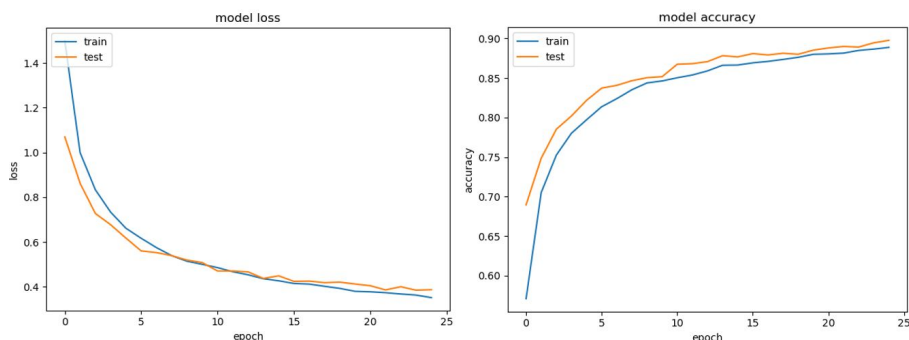


Figure 6: Evolution of values for loss (left) and accuracy (right), when training and testing. The two graphs were made at the end of the learning process, using matplotlib.pyplot.

3.6.1 Cross-validation

The results presented in the previous section are obtained by mixing the dataset, and extracting, randomly, the two groups to perform the training, respectively testing the convolutional neural model. In this way, the validation of the model is a simple one, by using the two disjoint sets.

However, there are methods that can estimate/validate much better the performance of the model, by performing a statistical analysis on it, in relation to the existing dataset. One of the methods is called cross-validation.

Cross-validation is a sampling procedure used to evaluate machine learning models, when the dataset is limited [7]. The procedure has only one essential parameter k , and it refers to the number of groups in which a dataset must be split. As such, the method is known and as k -fold cross-validation.

The procedure is mainly used for machine learning algorithms in order to estimate ability of a model to generalize the learned characteristics on unseen data.

This method was applied to the existing convolutionary model and the existing musical dataset. The value chosen for k was 5, and the results obtained by averaging the k results can be seen in Table 2.

Stage	Accuracy	Precision	Recall	F Score
Training	96.11%	98.03%	93.42%	95.61%
Testing	91.61%	95.43%	88.25%	91.64%

Table 2: Results determined by applying cross-validation.

It is observed that these results are close to those obtained by simple validation. This confirms the initial conclusions regarding the existing model, which is suitable for the problem of automatic recognition of musical chords.

4 Existing musical recognition approaches

Around 2012, at the international machine learning conference in Florida, USA, a paper was presented that would change the direction of addressing the issue of automatic recognition of musical sheets, presenting convolutional neural networks as a solution with high efficiency, by comparison with previous approaches, especially with reference to Markov models.

The paper proposed by Humphrey and Bello, [6], describes two convolutional architectures, presenting layer-by-layer construction and training strategy. The dataset used contains 475 audio recordings, in a total of about 50.000 different agreements.

By training and testing the two convolutional networks, 77.4 % and 76.8 % of test accuracy were obtained, respectively, demonstrating that this approach performs competitively.

Since then, most papers have proposed methods based on convolutional neural networks. The most conclusive paper analyzed, in relation to the proposed solution, is the one presented by Zhou and Lerch, [17], from the center for music technology, within the institute of technology in Georgia, USA. The paper presents the construction of two deep networks with 6 layers, in two different architectures: a classical architecture, with the same number of neurons (1024) in each layer, and a bottleneck architecture. The dataset consists of 317 pieces collected from the discographies of famous bands, each piece being divided into over 1000 frames, for the individual recognition of the chords. The proposed algorithm is able to recognize major and minor chords for each root note, resulting in a dictionary of labels for chords of size 24+1, with 24 chords with known corresponding labels, and a label for any other chord that is not recognized.

The obtained results are presented on both architectures, applying different methods of data pre-processing. The best results were obtained using the spliced filters strategy. The performance evaluation metric is the total duration of the segments with correct prediction. Thus, the results are:

- Common architecture - 98.5 % for training and 87.6 % for testing;
- Bottleneck architecture - 93.6 % for training and 91.9 % for testing.

There is a significant improvement in the results, building more and more efficient convolutional networks, the increase being directly proportional to the increase of the labeled data correctly and in detail.

The evolution of the study in the field continues until around the current period, being presented a very important work about a year ago by Nadar, Abeßer and Grollmisch, [11], at a sound processing conference in Malaga. The paper aims to expand the dictionary of labels from 24 to 84 chords, extending the area of classes of chords that can be recognized. If until now only the major and minor chords of the root notes were recognized, in this paper we want to recognize other classes, among which seven chords or diminished chords. The dataset is complex, being composed both of the classic songs from the discographies of famous bands, and of a special data set for guitar, created by the Institute of Musical Semantics, Fraunhofer, from the Technical University of Ilmenau, Germany.

The convolutional model presented consists of several convolutional type layers, interspersed with max pooling type aggregation layers, the network having a total of 12 layers. The network was built to address two strategies, one that proposes recognition based on the extended dictionary but does not use the entire dataset (S-84), and one that uses the classic dictionary with 24 chords (S-24).

The metric used is f-score. Thus, the best results obtained are:

- S-84 - 76 %, using only the Fraunhofer Institute dataset;
- S-24 - 91 %, using the entire dataset.

5 System application

5.1 Working environment

5.1.1 Back-end

The programming language used to create the back-end project is Python. It is one of the favorite languages among developers, being suitable for many types of applications. In particular, it is considered the best choice for projects involving artificial intelligence.

It is due to the fact that language contains many libraries and external frameworks, which facilitate the coding process and save development time. Machine learning and deep learning are treated very well, through numerous libraries, including NumPy, used for scientific computing, SciPy for advanced computing and scikit-learning for data extraction and data analysis, being among the most popular libraries, working with strong external frameworks, such as TensorFlow, Keras, CNTK or Apache Spark.

5.1.2 Front-end

For the development of the front-end project, the programming language used is Kotlin. It is an open-source programming language which supports both object-oriented and functional programming. Kotlin offers similar syntax and concepts with other programming languages, such as C#, Java or Scala [8]. Kotlin does not aim to be unique, but is inspired by the development of similar languages over the decades, being available in JVM (Kotlin/JVM) variants, JavaScript (Kotlin/JS) and native code (Kotlin/Native).

Some Android APIs, such as Android KTX, are specific to Kotlin, but most are written in Java, and can be called from either Java or Kotlin. Interoperability between Kotlin and Java is essential for language growth. This property allows the programmer to use Java code in a system developed in Kotlin, and vice versa. Kotlin's popularity results in a better development experience on Android, but the development of the Android environment continues with both Kotlin and Java [8].

5.1.3 External packages

The construction of the entire system followed a series of clear steps, which eventually led to the creation of a fully functional application, starting from the functionalities offered to the user through the mobile application, to building a convolutional neural model and putting it into operation. All this would have been much more difficult if the external libraries used had not existed. These include:

- LibROSA: Python library for audio analysis. Provides the necessary packages to build a system for obtaining musical information [9];
- TensorFlow: intended end-to-end open source machine learning platform. TensorFlow is a complex system for managing all aspects of a machine learning algorithm. TensorFlow APIs are arranged hierarchically, with high-level APIs built on low-level APIs [10]. Researchers in the field use low-level APIs to create and explore new learning algorithms;
- Keras: is a high-level, neural network-oriented API, written in Python, being able to run on TensorFlow, CNTK or Theano [10]. It was developed with a focus on performing experiments in a simple and fast way. Keras is recommended for the following:
 - creating simple prototypes, extremely fast;
 - support for convolutional neural networks and recurrent networks, as well as combinations between them;
 - running on the CPU or GPU.
- NumPy: is the main library for scientific calculation in Python. Provides a high-performance multidimensional matrix object and working tools with this type of arrays, including a large collection of mathematical functions [12];
- Pandas: is a Python package that provides fast data structures and flexible, designed to work with structured data (tabular, multidimensional, potentially heterogeneous) as easy and intuitive as possible [14].

5.2 Mobile application architecture

For the construction of the mobile application we chose the design model MVVM (Model-View-ViewModel). It is one of the best design templates that can be used to develop an Android application. It is built from: Model, View and ViewModel [5].

The Model package contains all the classes that model the data processed in the application, APIs but also Repository objects. Thus, the application uses a WebRepository object that implements the functionalities required by the endpoints in WebAPI. It interacts with the package networking, to achieve the desired requests with the help of REST client for Java and Android, namely Retrofit. This package also interacts with phone's local database. The system entities can be saved, at the user's request, in a relational database, through the Room library.

The View package contains the structure and UI aspects with the user can interact. View elements are built through fragments and activities.

The ViewModel package is responsible for translating the data from Model in a format suitable for View. ViewModel and View are associated through the data binding mechanism, so that any changes made at the ViewModel level can be automatically reflected at the View level. Therefore, there is a ViewModel for each activity/fragment existing in the application (apart from MainActivity), through which the visual elements within the fragments and activities are managed.

The dependency relationships between packages are represented in Figure 7.

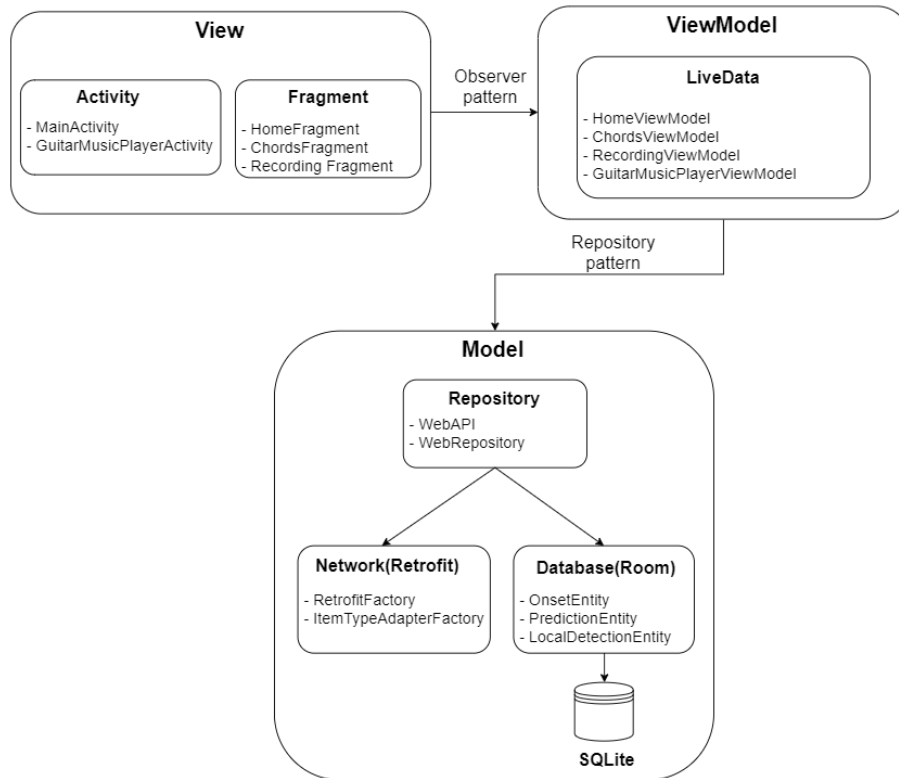


Figure 7: MVVM design model of the application.

5.3 Features

The mobile application is built around the assembly that is dealing with the automatic recognition of acoustic musical chords, with the aim of highlighting the functioning of the system recognition. Also highlighting the system must be in harmony with the wishes of the user so that result in a complex ensemble that meets its needs, primarily by providing accurate results, but also by their interpretation in a simple and suggestive manner, through a graphic representation in the field of music.

The functionalities of the mobile application determine the functional model of the system, each use case being implemented at the level of an activity or a fragment of the application. Thus, the main activity(MainActivity) is the one that opens when starting the application, and includes 3 fragments: HomeFragment, RecordingFragment and ChordsFragment.

The main fragment, which is initially launched, is HomeFragment. This fragment contains a list of music recordings from the mobile application's directory.

To begin the recognition process, it is necessary from the user to press one of the items for two seconds. This gesture will send the chosen audio file to the recognition system, which will perform the necessary operations, and will return the determined predictions in the form of a JSON response. This

gesture also causes the layout to change. Thus, over the created fragment, the activity responsible for the management of the results of the recognition system will be instantiated and displayed. This activity is GuitarMusicPlayerActivity.

The new instantiated activity is responsible for displaying the results obtained in the most suggestive way possible, by managing a MediaPlayer object and by displaying the probabilities associated with each chord recognized by the system.

The summary of the stated functionalities is shown in Figure 8, which presents the two fragments described, HomeFragment and GuitarMusicPlayerActivity.

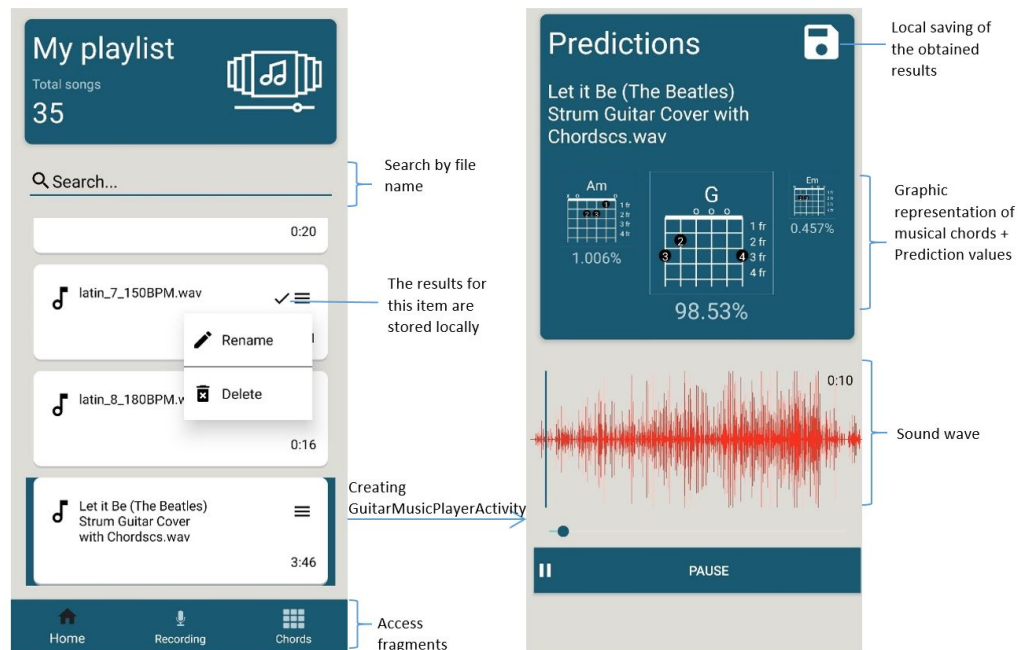


Figure 8: The image on the left shows the elements of the HomeFragment. By holding one of the items, the GuitarMusicPlayerActivity (right image), which is responsible for processing and displaying the results provided by the recognition system, is instantiated and displayed.

These View elements presented are the most important, being closely related to the automatic recognition system of musical chords. Besides them, there also exist two auxiliary fragments, namely RecordingFragment and ChordsFragment.

The recording fragment allows the user to use the phone's microphone to record a music sheet. You can then use the recording to discover the musical chords by sending it to the automatic recognition system.

The last fragment, ChordsFragment, has a didactic role within the application. The fragment contains a list of elements arranged in the form of a grid, each element representing one of the 24 chords recognized by the system. Thus, each element in the list presents the name of the musical chord, together with its graphic representation (drawing), being used the notation by tablature.

Figure 9 illustrates the two fragments described above.

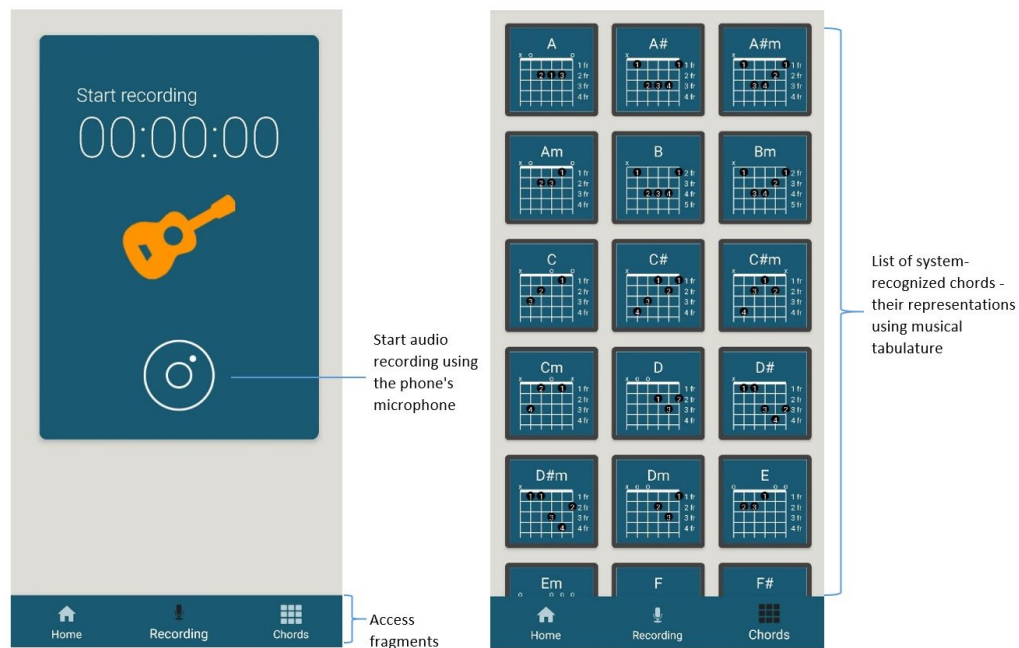


Figure 9: The image on the left is the RecordingFragment, responsible for audio recording, while the image on the right presents the list of musical chords recognized by the system (using tablature).

6 Conclusions and future improvements

This paper presented approaches to the problem of automatic recognition of musical chords. It has been observed that solving the problem depends on going through two main steps. The first step relates to processing methods of sound, and obtaining a meaningful representation. Regarding the signal representation, the chromagram was chosen, presenting the composition technique. The second stage is the introduction of machine learning algorithms, with a focus on deep neural networks. Their role is to create a neural model based on representations of audio recordings, a model that will be used in classifying detected chords.

These two steps were applied to a dataset consisting of wav audio recordings (original and augmented audio files). With the list of chromograms (feature matrices) built, for the next step, a convolutional neural network was built, trained and evaluated. By analyzing the results obtained for the calculation metrics and by analyzing the graphs, it was observed that this model provides excellent results, being a very good choice for this field. The neural model was combined with an algorithm for detecting musical transitions, to make it possible the recognition of chords, on observed musical intervals, thus obtaining complete and correct results for the entire musical content.

Based on the built-in recognition system, a mobile app has been created for users who like acoustic guitar music. The application allows the management of a playlist with acoustic songs, which can be sent, at the user's request, to the recognition system, which will return the predicted results, the application having the role to represent these results, using musical tablature.

The quality of the classification obtained through machine learning techniques is strictly related to the quality of the extracted musical information. For this reason, it is possible to investigate the obtaining and use of a better representation than the chromagram, by analyzing different methods of processing the sound signal. Thus, an improvement of the musical features extracted from the acoustic audio recordings could be obtained.

Another analysis towards system improvement can be made for the detection of musical transitions, which is a separate field from musical chords recognition, with many approaches.

Automatic recognition of musical chords is a theme with multiple approaches, each bringing an element

of novelty to the field. It is clear that the evolution of technology and computing power in terms of machine learning will also be reflected in this field, and in the future there will be more and more accurate recognition models, being able to cover larger areas of contemporary music.

References

- [1] Juan P Bello and Jeremy Pickens. *A robust mid-level representation for harmonic content in music signals*. In processing of ISMIR, 2005, pp. 304–311.
- [2] Sebastian Bock, Florian Krebs and Markus Schedl. “Evaluating the online capabilities of Onset detection methods”. In: *Department of Computational Perception* (2012).
- [3] R. N. Bracewell. *The Fourier Transform and Its Applications (ed. 3rd)*. McGraw-Hill, ISBN-13: 9781124130941, 2000.
- [4] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow*. O’Reilly Media, ISBN-13: 9781492032649, 2019.
- [5] *Guide to app architecture*. URL: <https://developer.android.com/jetpack/docs/guide> (visited on 12/05/2020).
- [6] Eric J. Humphrey and Juan P. Bello. “Rethinking Automatic Chord Recognition with Convolutional Neural Networks”. In: *Music and Audio Research Lab (MARL) New York University* (2012 11th International Conference on Machine Learning and Applications).
- [7] *k-Fold Cross-Validation*. URL: <https://machinelearningmastery.com/k-fold-cross-validation/> (visited on 01/06/2020).
- [8] *Kotlin overview*. URL: <https://developer.android.com/kotlin/overview> (visited on 02/05/2020).
- [9] *LibROSA*. URL: <https://librosa.github.io/librosa/index.html> (visited on 06/04/2020).
- [10] *Machine Learning Crash Course*. URL: <https://developers.google.com/machine-learning/crash-course> (visited on 02/05/2020).
- [11] Christon-Ragavan Nadar, Jakob Abeßer and Sascha Grollmisch. “Towards CNN-based Acoustic Modeling of Seventh Chords for Automatic Chord Recognition”. In: *Sound and Music Computing Conference (SMC), Malaga* (2019).
- [12] *NumPy*. URL: <https://numpy.org/> (visited on 02/05/2020).
- [13] *Onset Detection*. URL: https://musicinformationretrieval.com/onset_detection.html (visited on 28/04/2020).
- [14] *Pandas 1.0.3*. URL: <https://pypi.org/project/pandas/> (visited on 02/05/2020).
- [15] J. Pauwels et al. “Guitarset, A Dataset for Guitar Transcription”. In: *19th International Society for Music Information Retrieval Conference, Paris, France* (2018).
- [16] *Playing with Discrete Fourier Transform Algorithm in JavaScript*. URL: <https://dev.to/trekhleb/playing-with-discrete-fourier-transform-algorithm-in-javascript-53n5> (visited on 28/11/2020).
- [17] Xinquan Zhou and Alexander Lerch. “Chord detection using Deep Learning”. In: *ISMIR 2015, International Conference on Music Information Retrieval* (2015).

Răzvan-Cosmin LINCA
University of Babeş-Bolyai
Faculty of Computer Science
Mihail Kogălniceanu Street 1, Cluj-Napoca
ROMANIA
E-mail: cosmin_linca@outlook.com

Collection of software interfaces

Madalina Marinescu

Abstract

Reverse engineering, which means creating a software architecture starting from existing source code of software components, is **very complex and time-consuming**, requiring **very high initial efforts**. Experienced engineers allocation is needed, based on the software complexity of the components (number of components within the SW, interconnections, workflow, so on) and **UML** has to be learned by the assigned engineer. Collecting the data from an entire project by **analyzing the C code** from all “.c” and “.h” files for **manually completing** the Excel file needed for the usage of EA Import Plugin can be **close to impossible**, at reverse engineering. Our proposed solution collects the data from an **entire project** by analyzing the C code files and writes all the information in an **Excel file**, that will further be used by EA Import Plugin to generate the project’s architecture, resulting in **reduced effort** due to automation phases, no mandatory prior experience in SW Architecture methods needed, and **direct traceability** of SW changes, between SW Architecture modifications and/or SW changes within “.c” and “.h” files.

Keywords: parsing C code, analyzing C code, regular expressions, generation of software architecture, reverse engineering.

1 Introduction

Current software systems, especially those in the automotive industry, are very large in size and highly complex, so it is necessary to structure them by creating architectures [1].

The architecture of a software system involves the definition of architectural elements, called components and the interactions between them, called connectors, along with certain properties of them, to meet the requirements of the system [2]. The architectural diagrams show the input and output relations between the components and the interfaces through which they communicate [3].

Defining a software architecture from existing software components is very complex and time consuming. There is also a need for people specialized in creating architectures. Because of this, the *EA Import Plugin* application was developed within *Drivetrain* department of *Vitesco Technologies*, which automatically generates the architecture of a project starting from the necessary data completed in an Excel file.

Manually completing the Excel file needed to run the architecture generation application requires a lot of effort. Rigorous analysis of C code files from an entire project in order to extract certain information is very difficult and demotivating for the team members assigned for this task.

The *Collection of EA Interfaces* application, presented in this paper, aims to analyze the C code in the “.c” and “.h” files from the directories of a C code project and identify the functions with name, returned type, parameters and their types, description, source component, source

component group, as well as global variables with name, type, description, source component, and source component group. For all global functions and variables, the application identifies the files in which they are used to make connections between components.

Finally, all the extracted information is automatically written to a predefined Excel file, which will then be used by the *EA Import Plugin* application to generate the project architecture.

The import of this information is expected to be done only once at the beginning of a project and should serve as a basis for the final architecture but can also be used for reverse engineering. Reverse engineering [4] is the process of analyzing a system to identify the components that build it, the relationships between them and to create representations of the system in a higher level of abstraction. The objectives of this concept are to analyze and understand the full functionality of a system in order to maintain the system or to develop something new starting from it [4]. It should also be highlighted that not all decisions on architecture are a priority. According to [5] other aspects are considered before the details of the architecture, especially in the case of projects developed by methods such as Agile or spiral.

This paper is divided into 7 chapters, the contents of which are presented below. The first chapter presents the motivation that led to the development of the *Collection of EA Interfaces* application and a general description of its requirements. The second chapter refers to related work.

The third chapter presents the theoretical aspects that formed the basis of application development: the architecture of a project and the UML Enterprise Architect modeling application, object-oriented programming, the Visual Studio programming environment and the C# language, and regular expressions. The fourth chapter provides details about the design of the application, which meets the initial specifications, and the fifth chapter presents the implementation of the proposed solution.

The sixth chapter highlights the results of the test, and the seventh chapter summarizes the ideas presented.

2 Related work

Regarding the comparison with other software products, no similar applications of general utility were found to meet the requirements of the *Drivetrain* department of *Vitesco Technologies Engineering Romania SRL*. It is expected that there may exist such applications developed internally and specific to the needs of other companies.

3 Theoretical aspects

3.1 UML and Enterprise Architect

UML (Unified Modeling Language) is a standardized language for creating models for software products. UML can be used for visualization, specifications, design and documentation of software systems, but also for modeling in business and other non-software systems [6].

The use of UML improves the cooperation between people with technical skills and those with non-technical skills, such as project leaders, analysts, software/hardware architects, designers and developers. This helps to better understand the systems and identify potential risks. Thus, by detecting errors in the analysis and design phase, costs can be reduced in the implementation phase [7].

Enterprise Architect (EA) is an UML-based modeling and design application developed by *Sparx Systems* [7]. The architectures of the projects within the *Drivetrain* department, from *Vitesco Technologies Engineering Romania SRL* are created using this application. The architectural diagram of a software component (Fig. 1) shows in the center the current component,

on the left the components that provide functions and variables as input for it, and on the right the components for which the current component provides functions and variables as output. Such diagrams are automatically generated by the *EA Import Plugin* application, which uses as input the Excel file exported by the *Collection of EA Interfaces* application referred to in this paper.

The blurred areas of Fig. 1 contain confidential information about software components created in *Drivetrain* department of *Vitesco Technologies* and cannot be showed.

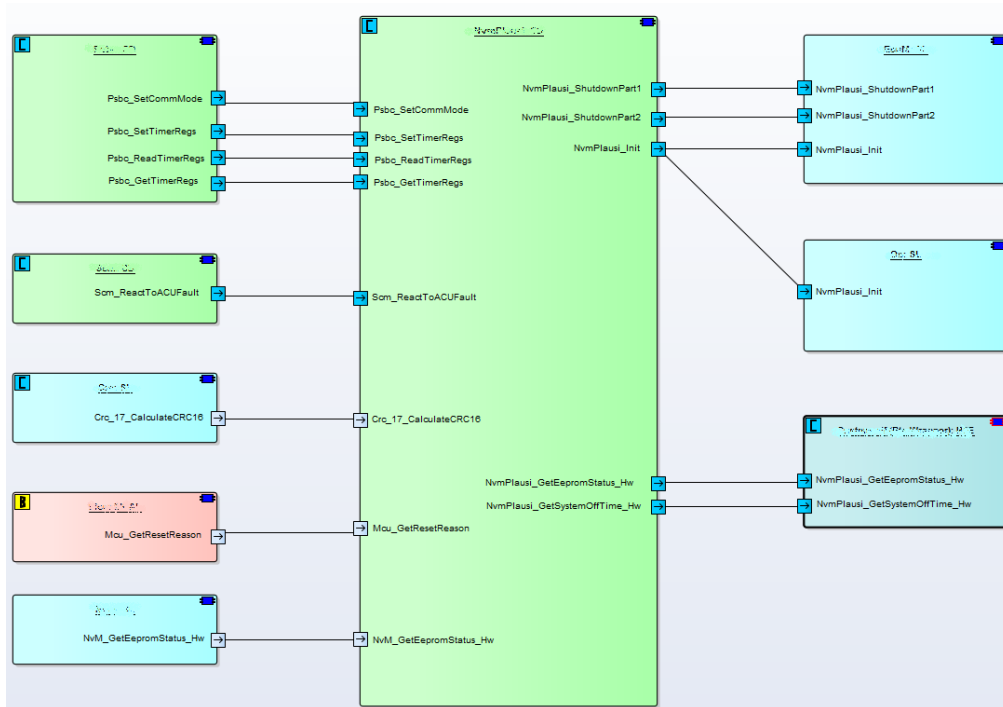


Fig. 1: Architectural diagram of a software component

3.2 Object-oriented programming - the C# language

The object-oriented paradigm emerged due to the fact that the classical paradigm involves techniques based on operations or attributes, but not both. The object-oriented paradigm considers both operations and attributes to be equally important [8].

Using the object-oriented paradigm, in exchange for creating a set of functions that meet certain requirements, one can create software objects that represent the state and behavior of equivalent objects in the real world [9]. For example, for calculating the area and perimeter of a rectangle, we can identify the rectangle object, having as properties the length and width of the sides. These will be used by an area calculation method, respectively by a perimeter calculation method. Thus, the information needed for the calculations is encapsulated in an object.

A class is a template that allows you to define the state and behavior of an object. After creating a class, it can be used to generate objects that represent the features of different objects in the real world [9]. Each object created is an instance of the class. For example, having the rectangle class, you can create several instances of it, but which have different lengths and widths. Variables defined in a class for encapsulating data about each object are called attributes or fields, and actions performed on them to meet certain requirements are called methods [9].

Encapsulation is the technique of wrapping the attributes and methods of an object in a coherent unit that can be used as an indivisible entity. Developers sometimes refer to encapsulation using the term “black box” [10].

The C# programming language was developed as an object and component-oriented language. It is part of the Microsoft Visual Studio package dedicated to developing applications running on computers that are using Microsoft Windows operating system. Unlike other programming languages, C# allows all data to be treated as objects and borrows the principles of object-oriented programming [10]. C# offers the possibility of creating components with properties, methods and events, this aspect making it an ideal language for this century, in which the creation of small reusable components is more important than the creation of huge stand-alone applications.

Some of the advantages of C# [10] language are the following:

- It contains a graphical interface, which makes it similar to the Visual Basic language, but it is considered more concise than Visual Basic;
- It is modeled after the C++ language, but it is easier to learn, the more difficult aspects of C++ being eliminated in C#;
- It is very similar to the Java language, because it was also based on C++, C # being more object-oriented. Unlike Java, all data is treated as objects in C#, providing improved functionality.

3.3 Regular expressions

A regular expression is a string that describes a search pattern [11]. Regular expressions are used in many programming languages, being very useful for finding or replacing certain sequences in large texts, as well as for validating the data entered by the user in a software application. The use of regular expressions reduces software development time, allowing a complex search or verification to be performed in a single line of code [11].

Regular expression engines are software applications that can process these expressions to find a match between the template and the text in which it is searched. There are two such engines: text-oriented (or DFA [12]) and regular expression-oriented (also called NFA - [12]). The latter are more popular because some features can only be implemented through them.

Regular expression-oriented engines always return the leftmost match, even if there is a better match in the continuation of the text. They work like this: they start searching with the first character in the text and try all the permutations of the expression for that character. Only if no match is found, they move on to the next character. If a match is found for a character, it is checked whether the next character in the text corresponds to the next character in the template [12].

The following are the main meta-characters [12] used to construct a regular expression:

- . - The "dot" character means matching any character except the newline character. Included in square brackets represents the character itself.
- () - Round brackets are used to group a series of elements of a template into a single element. Thus, different groups of characters can be searched and then analyzed separately.
- + - The "+" character searches for one or more consecutive occurrences of the item positioned before it.
- ? - The "?" character looks for zero or a single occurrence of the element positioned before it.
- * - The "*" character searches for zero or more occurrences of the element positioned before it.
- [] - Matches with one of the elements inserted in square brackets are searched.
- | - The "or" character chooses the match with one of the elements it separates.
- \s - Search for one of the white characters: tab, line feed, form feed, carriage return, space, next line.
- \S - Searches for any character except white characters.

- `^` - Finds the beginning of a line or string.
- `$` - Finds the end of a line or string.
- `[^...]` - Searches for any character except characters inside square brackets.

4 Development of the application

4.1 The structure of the output excel file

The Excel file resulting from running the application will be used by the *EA Import Plugin* application (an internal tool developed within *Drivetrain* department of *Vitesco Technologies*) to generate the project architecture. All the information that needs to be imported in the software architecture is automatically completed in this file as a result of parsing the C code by the *Collection of EA Interfaces* application. The excel file must contain five types of spreadsheets, each with a specific structure:

- *Component Groups* - spreadsheet in which are presented all the groups of components found in the project to be analyzed;
- *Components* - spreadsheet in which are presented all the components found in the project to be analyzed;
- *Component Interfaces* - spreadsheet showing, for each component, all input and output variables and functions;
- *DV Diagrams* - spreadsheet for which it is not necessary the automatic completion by the application, but only its creation;
- *Data Types* - spreadsheet showing all data types found in the project to be analyzed.

Next, the information to automatically be filled in by the application will be presented for each spreadsheet in the predefined Excel file.

- Completing the *Component Groups* spreadsheet

According to Fig. 2, the *Component Group Name* column must be completed in this sheet with the names of all component groups in the analyzed project. The name of a component group is the name of the directory two levels higher than the “cfg” directory for the current “.c” or “.h” code file.

	Component Group Name	Component Group Owner	Description	ASIL Level	Project (Original)
1	Actuators				
2	App				
3	AppServices				
4	SafetyConcept				
5	Sensors				
6					

Fig. 2: *Component Groups* spreadsheet

- Completing the *Components* spreadsheet

According to Fig. 3, the *Component Name* column must be filled in with the names of all components in the analyzed project and the *Component Group Name* column with the names of the group to which each component belongs. The name of a component is the name of the directory one level higher than the “cfg” directory for the current “.c” or “.h” code file.

1	Component Name	Component Group Name	Component Owner	Description
2	AcBldcM	Actuators		
3	AcBldcM	Actuators		
4	AcSupp	Actuators		
5	AcVlvOnOff	Actuators		
6	Ainfc	App		

Fig. 3: Components spreadsheet

- Completing the *Component Interfaces* spreadsheet

For each function or variable presented, the *Interface* column (Fig. 4) must be completed with the following terms:

1	Status	Interface	Scope	ASIL Level
3		Output		
4		Input		
5		Input		

Fig. 4: Component Interfaces spreadsheet

- *Input* - if the function or variable represents an input for the current component, means that it is defined in another component and it is used in the current component;
- *Output* - if the function or variable is an output for the current component, means that it is defined in the current component and used in other components.

Each function or variable can be an output interface for a component in which it is defined only one time, but it can be an input interface for any component in which it is used.

Next, the *Component Group*, *Component Name*, and *Interface Type* columns are completed for each function or variable, as shown in Fig. 5. The *Component Group* and *Component Name* columns represent the component group and the component in which the current function or variable is used. The interface type is *Variable*. In the case of functions, the Interface Type column will be completed with *Function*.

Fig. 5: Filling in variable information (1)

According to Fig. 6, the following columns to complete are *Source Component Group* and *Source Component Name*, for each function or variable. These represent the component group and the component in which the current function or variable is defined.

In the case of variables, the next column to fill is *Variable Name* (figure 12), where the name of the variable will be written.

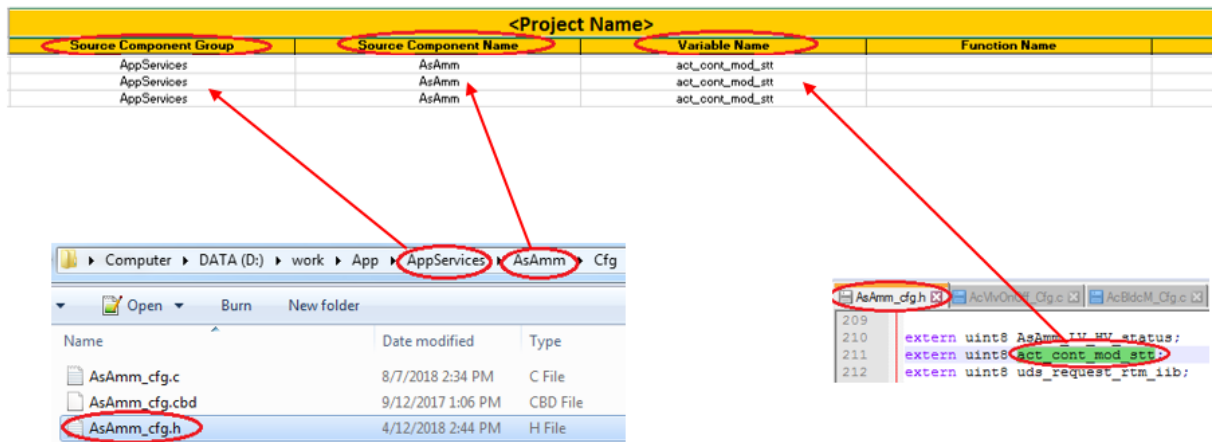


Fig. 6: Filling in variable information (2)

In the case of functions (Fig. 7) the columns to be completed are *Function Name*, *Parameter Name*, *ReturnValue* and *Type*. Three different types of rows must be completed for each function:

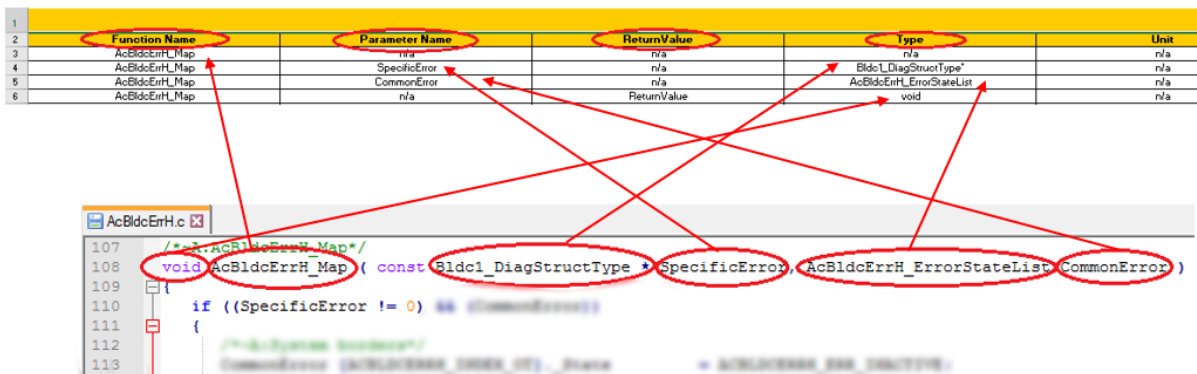


Fig. 7: Filling in function information

- The first type of row, referring to the function name: only the *Function Name* column will be filled in with the name of the current function, for example *AcBldcErrH_Map*, and the other columns will be filled in with n/a;
- The second type of row, referring to the function parameters, if applicable: for each parameter the *Function Name* column will be filled in with the function name, the *Parameter Name* column with the parameter name (for example *SpecificError*) and the *Type* column with the parameter type (for example *Bldc1_DiagStructType**). The other columns will be completed with n/a;
- The third type of row, regarding the value returned by the function: the *Function Name* column will be filled in with the function name, the *ReturnValue* column with the “ReturnValue” string and the *Type* column with the return function type (for example *void* for a function that returns nothing). The other columns will be completed with n/a.

The last column to be completed is *Description* (Fig. 8). For each function or variable, its description must be written, taken from the corresponding comments, if any.

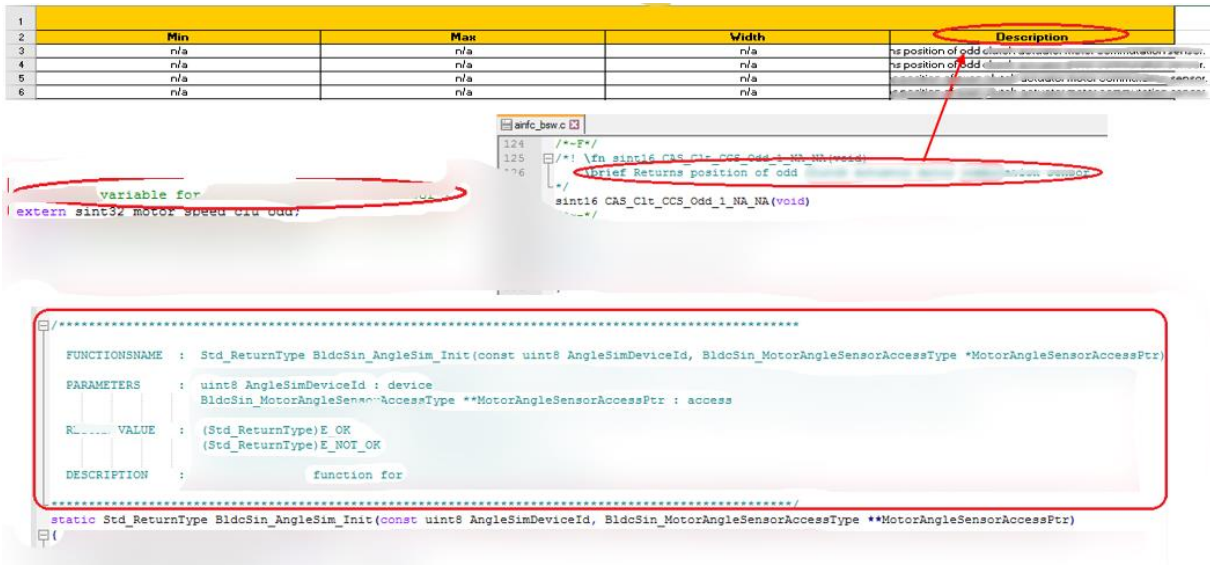


Fig. 8: Complete description column

The *Unit*, *LSB*, *Min*, *Max*, *Width* columns will be completed with n/a.

- Complete the Data Types spreadsheet

According to Fig. 9, the *Data Type* column must be completed in the Data Types sheet with all the data types extracted from the entire project to be analyzed.

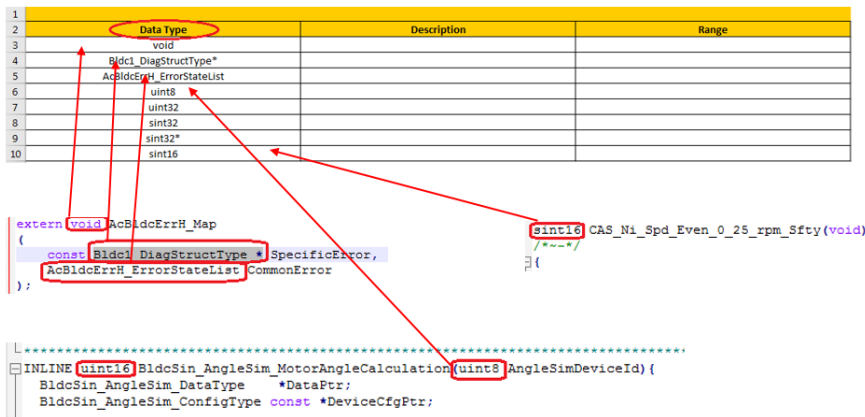


Fig. 9: Data Types sheet

4.2 Description of the application

According to the requirements of the project, the Collection of EA Interfaces application must export an Excel file completed with the information presented in the previous subchapter.

To achieve the specifications, the application must receive as input all the source files of the project for which it is wanted the generation of the architecture, analyze them in order to extract the necessary information about the functions and variables of each component and give as output

the predefined Excel file filled with this information. Fig.10 shows the block diagram of the application.



Fig. 10: Block diagram

The first step in running the application is loading the directory where all the files of the project to be analyzed are located. In the next step, each “.c” and “.h” file in each project subdirectory will be analyzed to extract the necessary information.

The third step is the completion of a *componentPaths.txt* file by the user, with the component name and group name to which the component belongs for each acronym or path written in the file. This is necessary because the names of the components and groups are usually abbreviated, being at the same time names of sub-directors, and in the project architecture it is desired to specify the full names. There are also cases where the “cfg” directory, previously discussed, is missing. Thus, it is not possible to identify the names of the component and the group. For these cases, the path to the parsed file is written in the *componentPaths.txt* file, and the user must complete the names that will eventually appear in the architecture.

In the next step, the user starts the creation of the predefined Excel file. All the information extracted previously will be filled in the Excel file.

In the following figures, the steps of the application are shown through a generic example.

Fig. 11 shows the structure of the project directory to be analyzed: the project consists of 2 component groups, each including 2 components.

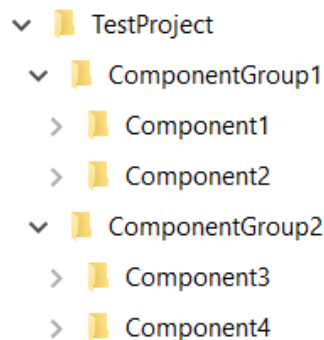


Fig. 11: Project directory

Fig. 12 shows the code from *component1_file.c* from Component1. It contains the function *C1_function_name* that calls *C3_function_name* from Component3.

Fig. 13 shows the *component1_header.h* file from Component1. It has an extern variable defined, *C1_variable_name*, which is used in *C3_function_name* from Component3.

Fig. 14 shows the code from *component3_file.c* from Component3. It contains the function *C3_function_name* which is called in Component1 and which uses *C1_variable_name*.

Fig. 15 shows the *component3_header.h* file from Component3.

All this C code within the presented project is analyzed by the application in order to extract the information of the interfaces from above and to determine the connections between them.

```

component1_file.c component1_header.h component3_file.c component3_header.h
1 #include "component1_header.h"
2 #include "component3_header.h"
3
4 /*
5  Function name: C1_function_name description
6  Parameters: int C1_parameter - parameter that represents "something"
7  Return value: 1 - C3_function_name result is 1
8               0 - C3_function_name result is 1
9 */
10 int C1_function_name (int C1_parameter)
11 {
12     Set of statements - Block of code
13
14     if (C3_function_name(2))
15     {
16         return 1;
17     }
18     else
19     {
20         return 0;
21     }
22 }

```

Fig. 12: C file from Component1

```

component1_file.c component1_header.h component3_file.c component3_header.h
1 extern int C1_function_name (int C1_parameter);
2
3 /* C1_variable_name represents "something" */
4 extern int C1_variable_name;

```

Fig. 13: Header file from Component1

```

component1_file.c component1_header.h component3_file.c component3_header.h
1
2 #include "component3_header.h"
3 #include "component1_header.h"
4
5 /*
6  Function name: C3_function_name description
7  Parameters: C3_parameter - description of C3_parameter
8  Return value: result - integer that is equal to C1_variable_name * C3_parameter
9 */
10
11 int C3_function_name (int C3_parameter)
12 {
13     int result = 0;
14
15     result = C1_variable_name * C3_parameter;
16
17     return result;
18 }

```

Fig. 14: C file from Component3

```

component1_file.c component1_header.h component3_file.c component3_header.h
1
2 extern int C3_function_name (void);

```

Fig. 15: Header file from Component3

The next figures show the generated Excel file, which is automatically completed by the application with the information extracted from the project's code.

Fig. 16 shows the *Components* sheet, which includes all the 4 components from the 2 groups extracted from the code.

Fig. 17 shows the *Component groups* sheet, which includes the 2 groups extracted from the code.

Fig. 18 shows the *Data types* sheet, which includes the data types identified in the project's code.

Fig. 19 shows the *Component Interfaces* sheet, which includes the extracted information about the interfaces, that has been previously explained.

Component Name	Component Group Name	Component Owner	Description
Component1	ComponentGroup1		
Component2	ComponentGroup1		
Component3	ComponentGroup2		
Component4	ComponentGroup2		

Fig. 16: "Components" sheet of Excel file

Component Group Name	Component Group Owner	Description
ComponentGroup1		
ComponentGroup2		

Fig. 17: "Component groups" sheet of Excel file

Data Type	Description	Range
int		N/A

Fig. 18: "Data types" sheet of Excel file

<Project Name>											
Interface	Component Group	Component Name	Interface Type	Source Component Group	Source Component Name	Variable Name	Function Name	Parameter Name	ReturnValue	Type	Description
Output	ComponentGroup2	Component3	Function	ComponentGroup2	Component3		C3_function_name	n/a	n/a	n/a	C3_function_name description
Output	ComponentGroup2	Component3	Function	ComponentGroup2	Component3		C3_function_name	C3_parameter	n/a	int	C3_parameter - description of C3_parameter
Output	ComponentGroup2	Component3	Function	ComponentGroup2	Component3		C3_function_name	n/a	ReturnValue	int	result - Integer that is equal to C1_variable_name * C3_parameter
Input	ComponentGroup1	Component1	Function	ComponentGroup2	Component3		C3_function_name	n/a	n/a	n/a	C3_function_name description
Output	ComponentGroup1	Component1	Variable	ComponentGroup1	Component1	C1_variable_name		n/a	n/a	int	C1_variable_name represents "something"
Input	ComponentGroup2	Component3	Variable	ComponentGroup1	Component1	C1_variable_name		n/a	n/a	int	C1_variable_name represents "something"

Fig. 19: "Component interfaces" sheet of Excel file

pointer (**) must accompany the type, although it often appears separately or concatenated next to the name. For example:

```
STARTUP_INLINE FUNC(sint32 *, STARTUP_CODE) StartUp_ICopyInt32(
    uint32 size,
    sint32 *dst,
    sint32 *src)
```

Many similar challenges arise in extracting the information about parameters as well.

The method *findDescriptions()* uses regular expression templates to identify the function description, the return value description, and the description of each parameter. The search is performed in the comment list previously extracted for each function and for each of its parameters. The templates contain keywords and form groups for extracting useful information.

For example, the regular expression used to identify the function description is:

```
@"(Description|DESCRIPTION|\\par Description|\\brief)((\\.\\s)*?)(Parameters *(in\\)|Parameters *(out\\)|\\par
Parameters *(in\\)|\\par Parameters *(out\\)|In-Parameters|Out-Parameters|Input parameter|Output
parameter|PARAMETERS|Parameters|Arguments|@param|\\param In|\\param Out|\\param In\\Out|Input
Parameters|Input Parameter|Output Parameters|Output Parameter|\\param|NOTE|Note|Service ID| RETURN|
RETURN VALUE|\\return|@return|\\*V)"
```

In the case of parameters, a template is first used to delimit the section of the comment that refers to the parameters. Then another template is applied that looks for the description of each parameter in this section, having as keywords its name and possibly its type.

Extracting descriptions from the comments was one of the biggest challenges in developing this application, imposing the most restrictions. Certain keywords or keyword constructions are needed to delimit the descriptions and not to appear in the actual description. Most of the cases identified so far in the department's projects have been analyzed and treated. All cases for which the application has positive results, as well as those that should be avoided have been presented in the user manual of the application.

As regarding functions calls, various patterns are applied to the lines in the body of each function to determine calls to other functions within them. Several cases are treated with different regular expressions:

- Function calls on a single line of code;
- Function calls extended on multiple lines of code;
- Function calls inside instructions like *while*, *if*, *for*, written only on one line of code or extended on multiple lines.

The *analyzeCFile(string fileName)* method performs the entire analysis of the given file as a parameter. It reads the file line by line and initially look for function prototypes. For this, it is necessary to have at the same time an analysis of the next line to deal with cases where prototypes are extended over several lines of code.

Regular expressions are applied to identify possible prototypes, and if successful, the following lines are concatenated until a possible final prototype is identified. The entire string of concatenated characters is then analyzed by the *findPrototypePattern* method, which provides the final result. For the prototypes found, this method also creates a list with the lines corresponding to the body of each function.

Finally, all the methods presented above are called in the *analyzeCFile* method, after which the information extracted from the current file is added to the global lists of the main class *MainClass*.

The *HeaderProcessing* class parses a “.h” file line by line, in order to extract the defined external variables, as well as the structures and type definitions. Information extracted about

variables is added to the list of objects of type *Variable*, *currentHeaderExternVariables*, those about structures are added to the list of objects of type *Structure*, *currentHeaderStructuresList*, and those about type definitions are added to the list of objects of type *Typedef*, *currentHeaderTypedefs*. The methods of the *HeaderProcessing* class are implemented similarly with the ones of the *FileProcessing* class.

The *ExcelFile* class deals with the creation and completion of the Excel file whose need led to the development of this application. The main properties of the class are the *Excel.Application* object, which creates an instance of Microsoft Excel, the *Excel.Workbook* object, which creates an Excel file, and the *Excel.Worksheet* objects, which creates the sheets required in the predefined Excel file. To access these objects in Visual C#, the reference *Microsoft.Office.Interop.Excel* [13] was used. This allows you to use methods and properties for accessing the elements of an Excel file: creating the file, creating spreadsheets, accessing cells, writing and reading them.

MainClass is the main class of the application, it implements the methods used for the occurrence of various events in the user interface and also implements the methods that find the connections between the analyzed components. For example, for each identified function, *findCalledFunctions()* method goes through the list of the names of the functions called inside it and look for the objects of type *Function* that correspond to these names. Thus, the called functions are obtained with all their specific information. This is necessary to find out the communication between components.

5.2 User interface

The user interface of the implemented application is shown in the following figures. Fig. 20 shows the first step of running the application (choosing the directory of the project which is wanted to be analyzed), the progress of the analysis and the suggestion message to complete *componentPaths.txt* file.

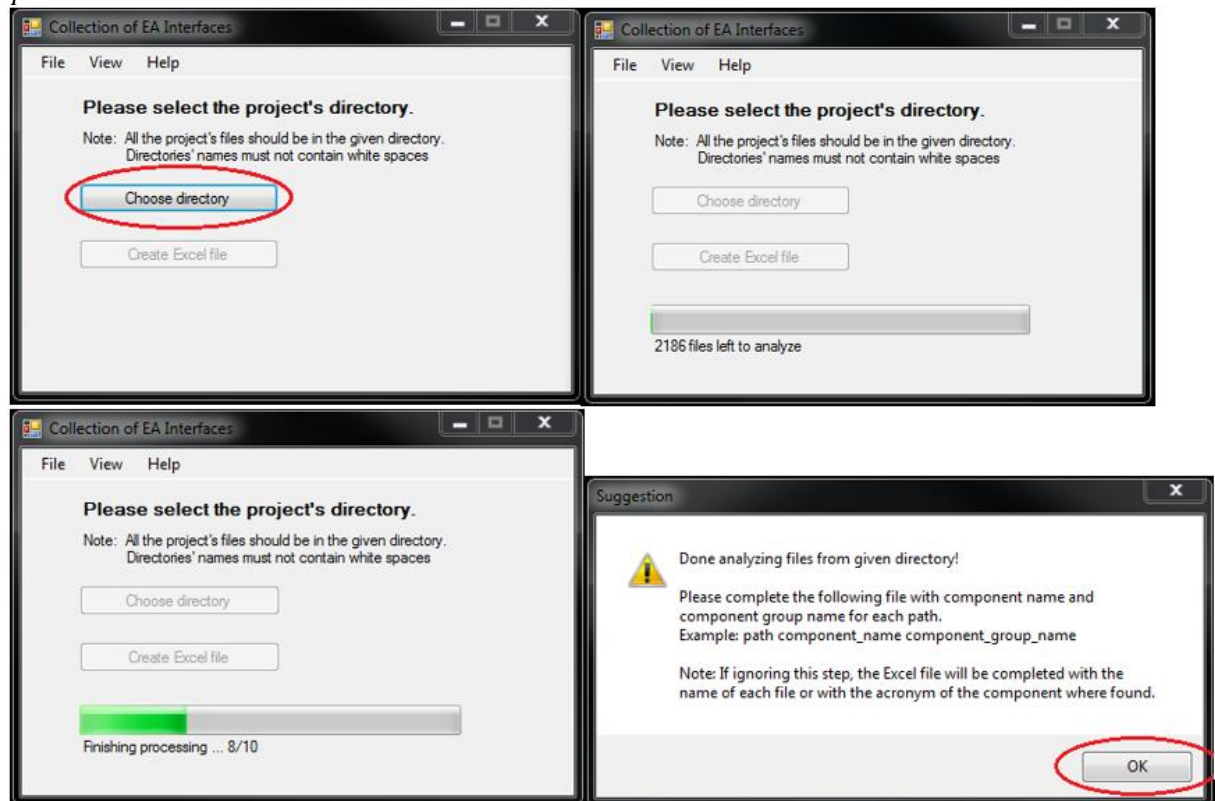


Fig. 20: User interface (1)

Fig. 21 shows the *componentPaths.txt* file, with an example of how it should be completed. For each file path, the user should complete component name and component group name, separated by space, if they want to have the full names in the architecture, not only the abbreviations (e.g. AcBldcErrH).

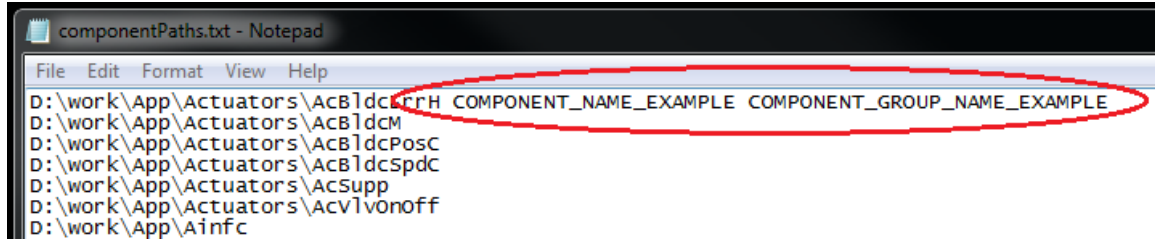


Fig. 21: User interface (2)

Fig. 22 shows the last step of running the application. After completing the *componentPaths.txt* file, the user must click on Create Excel file button. A save file dialog will appear to choose the location where the excel file should be saved, then the confirmation message that the collection of software interfaces has been successfully finished appears.

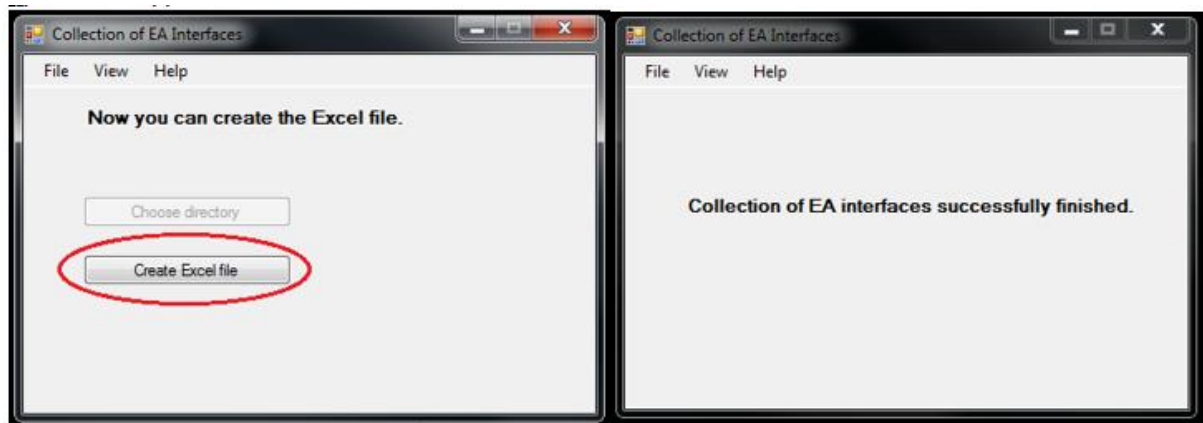


Fig. 22: User interface (3)

6 Results

The *Collection Of EA Interfaces* application was tested on 5 automotive projects, written in C code, within the *Drivetrain* department of *Vitesco Technologies Engineering Romania SRL*. The results summarized in Table 1 were obtained.

Analyzed project	Size	Number of „c” and „h” files	Number of lines written in Excel file	Time spent analyzing files	Time spent creating Excel file	Exceptions
Project 1	Small	457	9.679	1h 21m 0s	0h 0m 32s	0 files
Project 2	Medium	1.027	18.430	6h 0m 54s	0h 0m 34s	1 file

Analyzed project	Size	Number of „.c” and „.h” files	Number of lines written in Excel file	Time spent analyzing files	Time spent creating Excel file	Exceptions
Project 3	Medium	2.189	40.009	8h 20m 40s	0h 0m 41s	0 files
Project 4	Medium	2.190	40.026	8h 26m 31s	0h 0m 33s	0 files
Project 5	Large	6.381	71.088	2d 17h 10m 28s	0h 1m 30s	4 files

Table 1: Testing results

It can be seen that, in the case of project 1, the duration of the analysis is almost 6 times shorter than that of project 2, given that the number of files is only 2 times shorter. Also, in the case of project 3, although an almost double number of files was analyzed compared to project 2, the duration of the analysis was only a quarter longer. This is because the size of a project depends not only on the number of files contained, but also on their size. Comparing the first 4 projects, it can be seen that if the number of analyzed files is doubled, approximately 2 times more lines are obtained in the Excel file. Project 5 is an exception, having about 3 times more files than projects 3 and 4, and the number of lines in the Excel file remaining 2 times higher than in the case of projects 3 and 4. Also the analysis time is unexpectedly high compared to the amount of information extracted. Project 5 has an analysis time 8 times longer than that of projects 3 and 4, although the number of lines written in the Excel file is only almost 2 times greater.

We deduct that a high complexity of the C code can lead to a very long analysis time, even if a relatively small amount of information is obtained compared to other projects. Regarding the duration of the actual writing of the information in the Excel file, this is irrelevant, varying between 30 and 120 seconds.

Regarding the exceptions that occurred during the run, in case of project 2, 1 file was found, and in case of project 5, 4 files were found in which certain exceptions were identified. Making a report, out of the total of 12,244 files analyzed by the application, exceptions were found in 5 of them, i.e. 0.04% chance of encountering unanalyzed cases.

Exceptions do not affect the proper functioning of the application, being treated and reported in the *exceptionsLog.txt* file. The *exceptionsLog.txt* file is saved in the application location for further analysis to identify and treat new cases, which are necessary for the maintenance of the application.

Although it would be almost impossible to manually extract the information needed to complete the predefined Excel file and it would take about 50 times longer than using the *Collection Of EA Interfaces* application, as a further improvement, an attempt is made to identify a way to reduce the running time of the application and to get a 0% chance of encountering a case that has not been analyzed. The intention is to use multiple threads that will run in parallel when analyzing and parsing the C code, so the running time would be significantly reduce.

7 Conclusion

Given the need to develop software architectures for projects within an automotive company, the *EA Import Plugin* application was developed in order to automatically realize these architectures. Due to the phenomenon of code reuse in the case of projects with similar content, it is useful to be able to automatically create the architecture starting from existing software components. Also, in many cases the concept of reverse engineering is used, which involves the realization of the architecture starting from the already implemented system.

In the presented situations, the manual extraction of the information necessary to create the architecture is very difficult to achieve, this being the reason why the *Collection of EA Interfaces application*, that is presented in this article, was developed. The application is able to analyze a large number of C code files of projects and extract the information that defines an architecture, in a much shorter time than by doing it manually. Thus, a predefined Excel file necessary to generate the architecture of a project is created and completed automatically, with a minimum of effort and time from the user.

The development of the *Collection of EA Interfaces* application was a challenge, involving a lot of study in various aspects related to projects in the automotive industry: understanding the structure of a project, how software components communicate, analyzing various implementations of C code, and concepts UML for creating a software architecture. Many problems were encountered in the implementation of the application, especially due to the multitude of cases encountered and the desire to find a generic solution that can cover them all. Multiple regular expressions were created and tested, until discovering of combinations that were able to identify the defining information of a software architecture in most cases of C code implementation.

As improvement, the intention is to find a way to reduce running time and to cover as many cases of parsing the C code as possible.

Acknowledgement: This work benefits from funds given by *Vitesco Technologies Engineering Romania SRL*.

References

- [1] Garlan D, Shaw M. *An Introduction to Software Architecture*. Pittsburgh: Carnegie Mellon University; 1994.
- [2] Perry DE, Wolf AL. *Foundations for the Study of Software Architecture*. Software Engineering Notes. 1992 oct: pp. 40 - 52.
- [3] Kruchten P. *Architectural Blueprints—The “4+1” View*. IEEE Software. 1995 nov.: pp. 42 - 50.
- [4] Chikofsky EJ, Cross JH. *Reverse Engineering and Design Recovery: a Taxonomy*. IEEE Software. 1990 jan.: pp. 13 - 17.
- [5] Bass L, Clements P, Kazman R. *Software Architecture in Practice*. 3rd ed. Pearson Education, Inc.; 2013.
- [6] Booch G, Rumbaugh J, Jacobson I. *Unified Modeling Language User Guide*. 2nd ed. Addison Wesley; 1999.
- [7] Steinpichler D, Kargl H. *Project Management with UML and Enterprise Architect*. Vienna: SparxSystems Software GmbH.
- [8] Schach SR. *Object-Oriented and Classical Software Engineering*. 8th ed. The McGraw-Hill Companies, Inc; 2011.
- [9] Hillar GC. *Learning Object-Oriented Programming*. Birmingham: Packt Publishing; 2015.
- [10] Farrell J. *Microsoft® Visual C#® 2010 An Introduction To Object-Oriented Programming*. 4th ed. Boston: Course Technology, Cengage Learning; 2011.

- [11] Goyvaerts J. *Regular Expressions The Complete Tutorial*. 2007.
- [12] Friedl JEF. *Mastering Regular Expressions*. 3rd ed. Sebastopol: O'Reilly Media, Inc.; 2006.
- [13] Microsoft. *How to: Access Office Interop Objects by Using Visual C# Features (C# Programming Guide)*. [Internet]. [cited 2020 july 22]. Available from: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/interop/how-to-access-office-interop-objects>.

Madalina MARINESCU
Politehnica University Timisoara
Faculty of Automation and Computers
Timisoara
ROMANIA
E-mail: madalinamarinescu96@gmail.com

MARC (Monitored Automated Remote Car)

Christian Melchior

Abstract

As we hear the term robot, we are often thinking of a machine that can work in our service all day long. There are also exceptions, like medical robots which can reach difficult places where no human can. MARC is such a helpful prototype. Imagine that you would lend your car to your child who recently got his driver license, but you don't have the time required to join his every drive. No worries MARC is getting you out of trouble. You would be able to track multiple parameters of your car in real-time telemetry. You could see if he is over revving the engine, over heating the engine, or if he is exceeding the speed limit. At the moment you could also drive MARC remotely, making itself an enjoyable but a future-proof robot.

1 Introduction

This project was developed around the concept which we could remotely be able to know the state of our vehicles. In a modern world, in which we have all the needed information only at a click away, we can't know every moment how our cars are "feeling". In my concept, in the near future, we could know all the parameters in which the car is operating in real-time, with the help of sensors such as thermistors, LIDARs, RPM meters and others, and even be able to drive it remotely, through a reliable data transmission connection, being able to receive the telemetry of the vehicle and sending controls messages back to the car. We all are seeing the technology advancement in similar fields (self-driveable cars, Virtual Reality and road safety).

The existence of other scientific works such as "A REMOTE CONTROLLED CAR USING WIRELESS TECHNOLOGY", "Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino" or "Formula 1 Composites Engineering", helped in creating a clearer picture of the MARC project and developing the final prototype.

2 Description of the project

The project consists of an smart racing car prototype called MARC (Monitored Automated Remote Car) and a CSB (Control Surface Board). Marc is designed to look like a 1980-1990 Formula 1 car. On top of the main chassis I modeled multiple aerodynamical modules which can be changed in case of an accident or if I want to change the performance of the car. On the rear end of the car it can observed

the presence of DRS(Drag reduction system) used in modern racing cars (see Fig. 1). The CSB is designed to look like a master surface control board (see Fig. 2).

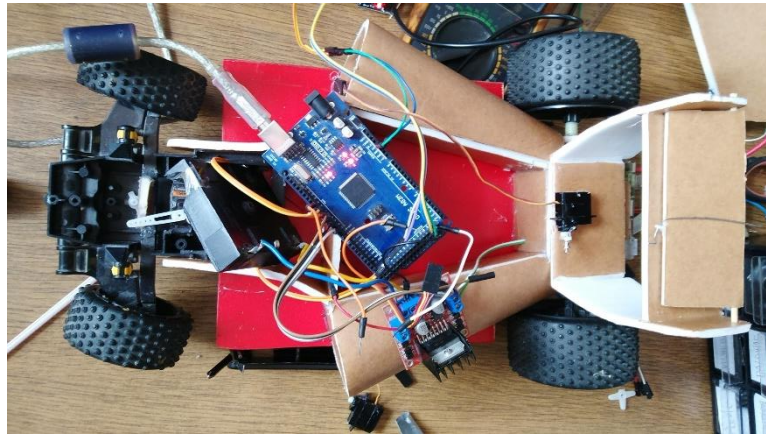


Fig. 1: The main chassis with the aerodynamic elements

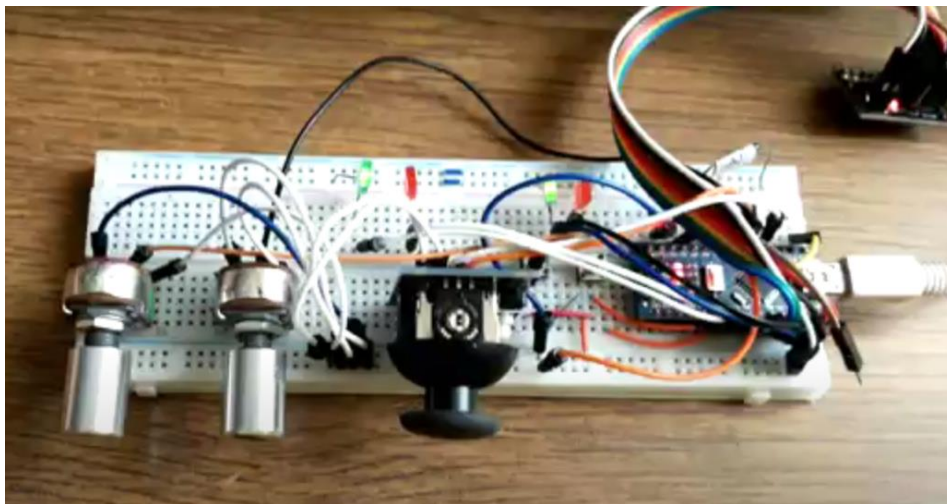


Fig. 2: The CSB

2.1 Hardware

As the hardware side it was used:

- **One Arduino Mega**

Arduino Mega is working as the center of processing and transmitting data to the CSB. This microcontroller has the perfect amount of input and output pins needed.

- **One Arduino Nano**

Arduino Nano is used as the brain of the CSB, transmitting controls to the car, receiving telemetry data back, and transmitting it forward to the PC.

- **One L298N driver**

The L298N driver can control up to two dc motors at a voltage from 5 to 35 V, and at a current up to 3 A. The speed can be controlled using a PWM signal (see Fig. 3), and direction of the motor with a microcontroller, because the driver is based on an H bridge (see Fig. 4). The PWM signal is widely used from electrical engineering to electrical hobbyist, mainly for a more efficient way to control all kind of loads like LED lamps, motors etc. The H bridge is a well-known circuit which change the polarity of the voltage delivered to the load, bringing the capability to change the movement of an motor.

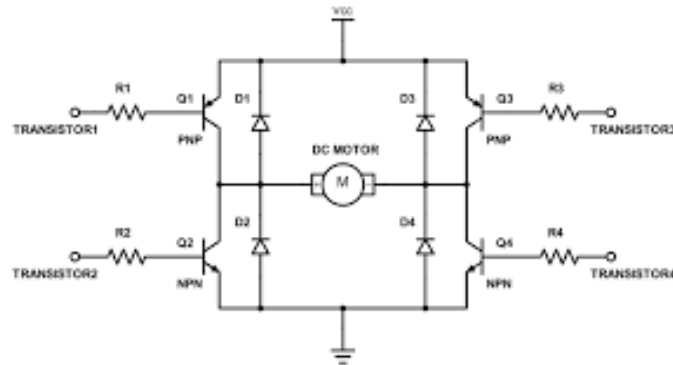


Fig. 3: PWM signal example

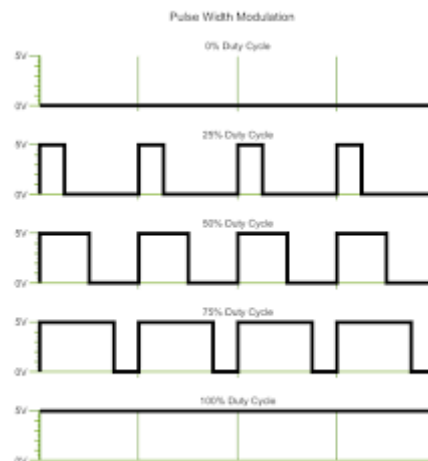


Fig. 4: H bridge schematic

- **Two NRF24L01 modules**

The NRF24L01 is a trans-receiver module, using the 2,4 GHz bandwidth. This module was chosen because the combination of range (up to 1 km in open field) and the amount of data that it could transmit (Max 2 Mbps).

- **Three servo-motors**

I have chosen to use the servo motors because they are easy to be operated (I can tell it a certain angle, and the servomotor will move its crank to the given angle).

- **One IR sensor**

The IR sensor is being used to sense the number of times that the rear wheels rotate, in order to be used as an RPM meter.

- **One NTC thermistor**

The NTC thermistor is used as a thermometer, in order to record the temperature of the motor.

- **Foamboard**

The foamboard is a composite material being used to build RC airplanes because it is very light and in the same time it has a good structural integrity.

- **Four LEDs**

- **Four potentiometers**

- **Three buttons**

- **Two breadboards**

- **One Li-Ion battery pack**

The Li-Ion battery pack contains two cells connected in serial, supplying the very much needed energy to the driver module. The Li-Ion technology was used because of its remarkable propriety to deliver constant power and it can store a large amount of power. Another reason why the Li-Ion technology was chosen is mainly because almost all the modern electric car constructors are using it.

- **One base chassis**

The chassis consists of a rigid platform made of plywood, on which the motor, the Integrated Circuits, the aerodynamical elements and all other components are placed. The chassis is also the heaviest part of the prototype because in order to have a stronger basis, weight compromises had to be made.

- **One motor equipped with a differential gearing**

2.2 Software

2.2.1 Smoothing algorithm

The raw data from the analogue sensors could variate from natural reasons. In order to get accurate data, I was supposed to smoothen the data. In the code I used the basic algorithm that comes with the Arduino Library. The raw data is cautiously stored in an array of five sensor readings. After the array is full of readings, the sum is divided to the amount of the readings, resulting in an average which is the processed data transmitted to the CSB.

```
numReadings = 5;
total = total - readings[arrayIndex];
readings[arrayIndex] = analogRead(recvMessage.NTC);
total = total + readings[readIndex];
arrayIndex = arrayIndex + 1;
if (arrayIndex >= numReadings) {
  arrayIndex = 0;
}
average = total / numReadings;
Serial.println(averageTemperature);
```

```
delay(1);
```

2.2.2 Sensor processing

The prototype is equipped at the moment only with one IR sensor used as an RPM meter and an NTC thermistor, used as a motor thermometer. The CSB is equipped with three buttons used for braking, reverse and neutral, four potentiometers used for speed, direction, brake and DRS sensitivity and 4 LEDs for monitoring the two perimeters (motor temperature and RPM). For the monitoring the temperature I used 2 LEDs 1 green and one red. In the code it can be set up a value as a threshold in order to have a physical user interface. As an example, every motor has a maximum temperature at which it can be operated before it damages itself. In the program, I would add this value and if the motor temperature will rise beyond the threshold, the green led would turn off and the red led will blink intermittently. The monitoring of the RPM is working on the same principal.

2.2.2 User interface

At the moment the prototype can be operated in two ways. The first mode is NT (non telemetry) mode, in which the prototype can be used in flesh bones, being transmitted and received only the important controls of the car such as speed, direction, braking and the rear aerodynamic element to increase the air drag coefficient (DRS- drag reduction system). The second is FT (full telemetry) mode, in which it is not transmitting and receiving only the basic controls, but also RTT (real-time telemetry) such as the temperature of the engine or the RPM of the wheels.

In the FT mode the control surface is transmitting to the pc telemetry data using the next commands, making a user-friendly interface (see Fig. 5).

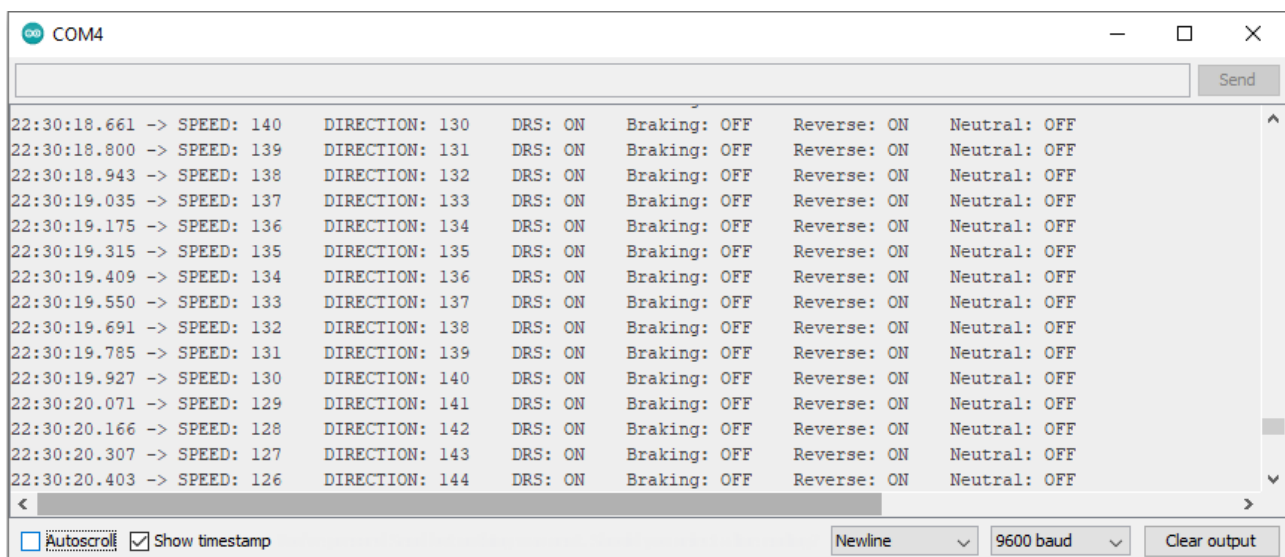


Fig. 5: User friendly graphical interface

3 Result

3.1 Encountered problems

Trough the period of time put in the developing the project I encountered also some problems. Firstly, I had a communication problem, because the both transceiver modules were tuned on the same bandwidth as the Wi-Fi router, so I had to change, in the code, the bandwidth of the modules. Secondly the final code still needs some rewriting, because it still has some annoying glitches.

3.2 Conclusions and further research directions

In such a prototype state, the project proves it concept which it was designed for. But as a prototype, this could be improved in a lot of fields. One of this field could be developing new aerodynamical elements, like a front wing with a variable angle of attack. Another research direction could be increasing the number of sensors and systems, such as a battery management system, energy recovery system or speed limiter.

Acknowledgement: This work was supervised by Professor *Nicolae Steavu*, from “Doamna Stanca” High School, Fagaras.

References

- [1] Self-driving cars - https://en.wikipedia.org/wiki/Self-driving_car 02.09.2020
- [2] 80's-90's Formula 1 car - https://en.wikipedia.org/wiki/McLaren_MP4/4 02.09.2020
- [3] “A REMOTE CONTROLLED CAR USING WIRELESS TECHNOLOGY” scientific paper - <http://e-journal.president.ac.id/presunivojs/index.php/JEEE/article/view/192> 02.09.2020
- [4] “Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino” scientific paper - <https://ieeexplore.ieee.org/abstract/document/8474620> 02.09.2020
- [5] “Formula 1 Composites Engineering” scientific paper - <https://www.sciencedirect.com/science/article/abs/pii/S1350630709001228> 02.09.2020
- [6] Smoothing tutorial - <https://www.arduino.cc/en/Tutorial/Smoothing> 02.09.2020
- [7] PWM signal (see Fig. 3) - <https://www.arduino.cc/en/Tutorial/Foundations/PWM> 02.09.2020
- [8] H bridge (see Fig. 4) - <https://www.build-electronic-circuits.com/h-bridge/> 02.09.2020
- [9] H bridge definition - <https://en.wikipedia.org/wiki/H-bridge> 02.09.2020

Christian MELCHIOR
 “Doamna Stanca” High School, Făgăraș
 Strada Doamna Stanca , No. 14 505200
 ROMANIA
 E-mail: christianmelchior7@gmail.com

Solving critical section problems by using a control thread

Milan Savić

Abstract

This paper shall present the implementation of critical section problem solving algorithms. The critical section is a problem that occurs with concurrent systems. Concurrent systems are collections of processors that communicate by writing in shared space or reading from shared space. This way of programming speeds up the work of certain programs and provides greater utilization of resources. Algorithms that solve this problem allow multiple processes to seamlessly access and modify data in a mutual shared space. So far, several algorithms have been implemented to solve this problem. Initially, these were algorithms that enabled the operation of concurrent systems for two processes. Today, there are algorithms that solve this problem for an arbitrary number of processes. Some of the more well-known algorithms that work for an arbitrary number of processes use the help of the operating system or some additional functionalities that are implemented at the processor level. The algorithms that will be presented in this paper represent a software solution to this problem without the additional help of the operating system or some other functions that are outside the program. The algorithms will be displayed in pseudocode, but they can be implemented in any programming language that supports multithreaded programming. The mutual shared space access simulation will be represented by using a thread on a simple example of the increment of a shared variable in the Java programming language.

1 Introduction

1.1 Concurrent programming

In the beginning of computing development, when computers did not have an operating system, one program was being executed from beginning to end. That program was using all computer's resources. This way of programming was expensive and inefficient. The development of the operating system enabled the execution of several programs at one time, so that each program is executed within the process. Programs were executed independently, and the operating system was allocating resources to them. Today, the concurrent paradigm is the basis of operating systems.

Concurrent systems are a set of processors that communicate by writing in shared space or reading from shared space. These systems enable faster and more efficient execution than single-processor systems.

Unlike sequential programming, where there is only one program execution flow, i.e. one task starts to be executed only if the previous one has come to an end, in concurrent programming there are several program executions flows where several tasks are executed alternately or simultaneously depending on the execution platform and thus progress over time.

Concurrent programming requires support at hardware level, operating system level, programming language level or library level. Depending on the level at which support for concurrent programming is implemented, the performance and portability of the program varies. If it is implemented on hardware level, a more efficient, but less portable code is obtained, while at the library level, the code is less efficient and more portable.

Parallel programming is a type of concurrent programming supported on multiprocessor systems, where multiple tasks can be executed at the same time. Unlike parallel systems, concurrent systems can also be implemented in single processor systems. If the executive platform is not observed, the concurrent system is the same as the parallel one because they have the same syntax and semantics.

The basic problems of concurrent programming relate to the way one access the mutual shared space as well as the communication process.

The goal of concurrent systems is to use more processors in the best possible way. Even on single processor systems, utilization is better.

Today, a large number of client, server and network applications use the principles of concurrent systems.

1.2 Tasks, threads and processes

Some of the basic concepts for a better understanding of concurrent programming are tasks, threads and processes. A task is a unit of a program that can be executed in competition with other units of the same program. Execution of a task within a program can be supported by a control thread or process.

Tasks can be heavyweight tasks and lightweight tasks. Heavyweight tasks have their own address space and can be represented as processes, while lightweight tasks share a common space with other tasks and can be represented as threads. Heavyweight tasks are managed by an operating system that allows them to share CPU time, access files, and address space. Lightweight tasks are part of a difficult task that does not require special computer resources.

In order for a system to work, tasks must communicate with each other or exchange information. Communication can be realized using shared memory or a messaging mechanism. Heavyweight tasks communicate most often using messaging mechanisms.

Lightweight tasks that use shared memory spaces can communicate using shared memory, but the synchronization of writing and reading from shared memory must be taken into account. When communicating using shared memory, there are two ways to synchronize cooperation and contention. Cooperation synchronization is used when one task must complete an operation in order for another task to start, while competitive synchronization is used when two tasks require the same resource that they cannot use at the same time. The problem of task synchronization leads to the problem of the critical section, which is explained in the following text, along with algorithms that solve this problem. Processes that do not assume process speeds are called asynchronous processes. The system must behave the same regardless of the speed of the process. Cooperative processes are processes that share the resources they access. Asynchronous cooperative processes are mainly used in competitive programming.

1.3 Work structure

The second chapter describes the critical section problem. The ways in which it can be realized are presented, as well as the assumptions and features that a software solution to this problem should satisfy.

The third chapter presents the basic algorithms that solve the critical section problem. The first part describes the algorithms that solve this problem for two processes, while the second part describes the algorithms that solve this problem for an arbitrary number of processes. In addition to the description of the algorithms, the basic advantages and disadvantages are also shown.

The fourth chapter presents the implementation of the proposed algorithms. The way these algorithms work is also described. Pseudocodes are presented and a brief description of the operation of these algorithms is given. This chapter also provides an implementation example presented in the Java programming language.

In the last part, a brief overview of the work is performed and possible directions of development of these algorithms are presented.

2 The critical section problem

When two or more processes access and change the shared memory space, it is necessary to synchronize the processes. In case of unsynchronized access, data may be damaged or lost. To avoid this problem a set of commands leading to it should be executed atomically. The part of the code that is executed atomically is called the critical section. The problem of making a critical section atomic is called the mutual exclusion problem. The basic tasks for solving this problem are isolating the critical section and preventing two or more processes from being in the critical section at the same time.

The critical section can be implemented in software, hardware, by using the operating system or by using higher program structures for synchronization (monitors).

The software solution to the critical section problem should satisfy certain assumptions and have certain features.

Assumptions and features of the software solution:

- Mutual exclusion
- Implementation without the help of the operating system
- Processes are asynchronous cooperative processes
- Only processes within the critical section can prevent others from entering the critical section
- The process must not wait indefinitely to enter the critical section
- The process must not remain in the critical section indefinitely
- The algorithm guarantees the possibility of entering a critical section to every process

3 Algorithms for tackling critical section

In the early 1960s, the field of concurrent programming began to develop, and to this day several algorithms have been developed.

The first algorithms were focused on solving this problem for two processes. After several unsuccessful attempts to create an algorithm to solve this problem, in 1964 the Dutch mathematicians Theodorus J. Dekker and Edsger W. Dijkstra came up with the first successful solution to this problem. This algorithm is known as the Dekker's algorithm. This algorithm uses two types of variables to ensure mutual exclusion. One variable represents the process's desire to enter the critical section, while the other variable determines which process is next to enter the critical section. These two variables were known in the strict alternations algorithm, the variable for marking which process is next to enter the critical section and the algorithm without strict alternation, the variable representing whether the process wants to enter the critical section, but were first used in the best way in Dekker's algorithm to make the algorithm work. The basic idea of the algorithm is that the processes initially express a desire to enter the critical section and then in case both processes want to enter the critical section, one of them gives up, i.e. goes into a random delay. This withdrawal is not permanent, but temporary until the second process leaves the critical section.

17 years after Dekker's algorithm, in 1981 Gary L. Peterson introduced his algorithm to solve this problem. Peterson's algorithm represents an improvement on Dekker's algorithm. The

Peterson's algorithm uses two variables in the same way as the Dekker's algorithm. One variable represents the desire of the process to enter the critical section, and the other which process is next to enter the critical section. The idea of Peterson's algorithm is that the process first expresses a desire to enter the critical section, then gives preference to another process. If the process wants to enter the critical section and if it is his turn, he enters. Peterson's algorithm is a more elegant, faster and shorter solution than Dekker's algorithm.

The disadvantage of Dekker's and Peterson's algorithm is that they work for only two processes. Today, there are algorithms that solve this problem for an arbitrary number of processes. Examples of such algorithms are the Test-and-set algorithm and Lamport's bakery algorithm.

The Test-and-set algorithm is a hardware solution to the critical section problem. The idea is that the instructions for testing and setting variables should be atomic, that is that they cannot be interrupted. Test-and-set (TAS) is one indivisible hardware instruction that is implemented at the processor level. There is one variable that the TAS instruction controls and changes and depending on the value of that variable allows the process to enter the critical section. Before entering the critical section, the process is called upon by the TAS instruction and it awaits approval for entry. The disadvantage of this algorithm is that it can only be implemented on systems that have a processor that supports such indivisible instructions. Some modifications of this algorithm are the Fetch and Add algorithm and the Exchange algorithm.

Lamport's bakery algorithm is an improvement on Peterson's algorithm. It is called the bakery algorithm because the basic idea is taken from the way bakery do business. Customers take ordinal numbers to determine the fair order among customers when shopping. The customer with the smallest number had the advantage when shopping. A similar principle is used in Lamport's bakery algorithm. The process that wants to enter the critical section takes its ordinal number. This is achieved by incrementing the largest number chosen so far. As the incremental operation is not atomic, it may occur that two or more processes get the same ordinal number. This problem is solved with the help of the operating system. The operating system assigns a unique process identification number (PID) to each process. If two or more processes have the same ordinal number, then their identification numbers (PIDs) are compared. A process that had a lower ordinal number enters the critical section, and if the processes have the same ordinal number, the process with the lower identification number (PID) enters the critical section.

4 Implementation of proposed algorithm

Algorithms such as Test-and-set and Lamport's bakery are not altogether software solutions to the critical section problem. The Test-and-set algorithm uses the help of a processor to execute a particular instruction atomically, while the Lamport's bakery algorithm uses the help of the operating system. Algorithms that solve this problem for only two processes, such as Dekker's and Peterson's algorithm, can be implemented at the code level without the additional help of other components above the code level (libraries, operating system, hardware components). This is possible because the two processes can control each other. In the case of two processes, each process knows in which state the other process is and in which state that process itself is, and in that way, it is possible to synchronize the process. When there are multiple processes, one process cannot check the states of all other processes in one command, and thus it is impossible to synchronize processes. In order for several processes to be synchronized, the help of a control component (another process, operating system, hardware component) is needed. The role of the control component can be performed by a process that is not in the set of processes that want to enter the critical section. The control process at the request of the processes wishing to enter the critical section determines the order of entry into the critical section. In this way, there can be no conflict between processes.

There are several algorithms that use this critical section approach. The pseudocodes of these solutions will be presented below.

```

flag: array[0..NUM_THREADS] of bool = {false}
turn: integer = {-1}

thread:
flag[i] = true
while(turn != i){/* nothing */}
// critical section
flag[i] = false;
if(executed thread){
    NUM_EXECUTED_THREAD++
}
turn = -1
//end of critical section

control thread:
while(NUM_EXECUTED_THREAD < NUM_THREADS){
    for(i=0; i<NUM_THREADS; i++){
        if(flag[i]){
            turn = i;
            while(turn != -1){/* nothing */}
        }
    }
}

```

The elements of the flag array represent the thread's desire to enter the critical section. Each thread can access only one specific element of the array. The turn variable determines which thread is next to enter the critical section. The thread registers for entry before entering the critical section, then waits its turn. When the thread takes precedence over the control thread then it enters the critical section. When exiting the critical section, the thread shows that it does not want to be in the critical section, checks if it has completed its task to inform the control thread about it, and finally releases the variable turn and gives preference to the control thread. The control thread goes through the flag and checks which thread wants to enter the critical section. When it finds a thread that wants to enter the critical section, it gives it an advantage to enter. The control thread must wait for the thread that is in the critical section to exit it and release the variable turn. The control thread is executed last and must wait for all threads to be executed.

The implementation of this algorithm in the Java programming language is given below. The critical section is represented as an increment variable.

```

class Counter{
    private volatile int value = 0;

    public int getValue() {
        return this.value;
    }

    public void increment(){
        this.value++;
    }
}

class MyThread extends Thread{
    int id;
    Counter counter = null;

    public MyThread(int id, Counter counter) {

```

```

        this.id = id;
        this.counter = counter;
    }

    @Override
    public void run() {
        for(int i = 0; i < Test.NUM_ITERATIONS; i++){
            Test.flag[id] = true;
            while(Test.turn != id);

            counter.increment();

            Test.flag[id] = false;
            if((Test.NUM_ITERATIONS - i) == 1){
                Test.num_executed_thread++;
            }
            Test.turn = -1;
        }
    }
}

public class Test {
    public final static int NUM_THREADS = 5;
    public final static int NUM_ITERATIONS = 100;
    public static volatile int turn = -1;
    public static volatile int num_executed_thread = 0;
    public static volatile boolean[] flag = new boolean[NUM_THREADS];

    public static void main(String[] args) {
        Counter counter = new Counter();
        for (int i = 0; i < NUM_THREADS; i++) {
            flag[i] = false;
            new MyThread(i, counter).start();
        }

        while(num_executed_thread < NUM_THREADS){
            for (int i = 0; i < NUM_THREADS; i++) {
                if(flag[i]){
                    turn = i;
                    while(turn != -1){}
                }
            }
        }

        System.out.println(counter.getValue());
    }
}

```

Solving the critical section problem by using the control thread and by implementing the idea of the Lamport's bakery algorithm is given in the following pseudocode:

```

flag: array[0..NUM_THREADS] of bool = {false}
number: array[0..NUM_THREADS] of integer = {0}

thread:
flag[i] = true
while(flag[i]){/* nothing */}
for(j=0; j<NUM_THREADS; j++){
    while(j!=i && number[j]!=0 && number[j]<number[i]){/* nothing */}
}

//critical section

```

```

number[i] = 0
if(executed thread){
    NUM_EXECUTED_THREAD++
}

control thread:
k = 1
while(NUM_EXECUTED_THREAD < NUM_THREADS){
    for(i=0; i<NUM_THREADS; i++){
        if(flag[i]){
            number[i] = k
            k++
            flag[i] = false
        }
    }
}
}

```

Each thread uses one element from the flag array and the number array. The elements of the flag array represent the thread's desire to enter the critical section. The elements of the array number represent the thread's ordinal number to enter the critical section. The thread applies for entry in the critical section, then waits to get the ordinal number. When it gets an ordinal number, it checks if any of the other threads have a smaller number. If such threads exist, the thread waits its turn, otherwise it enters the critical section. When exiting, it releases its field from the number array. The control thread goes through the flag and checks which thread wants to enter the critical section. When it finds such a thread, it assigns it an ordinal number and informs the thread that it has been assigned a number by setting the flag[i] to false.

5 Conclusion

Concurrent systems are the basis on which today's computers are functioning. A concurrent approach speeds up the operation of many programs. Operating systems use a concurrent approach when allocating resources to processes and thus enable concurrent execution of multiple programs. A large number of client, server and network applications use concurrent programming. The development of this area is of great importance for the improvement of computing. With this approach, computer resources can be used to the maximum.

The paper presents the basic problem of such systems, as well as the ways in which this problem is solved. During the development of this area, several algorithms were developed. Each of them has its advantages and disadvantages. This paper presents new solutions for solving the critical section problem. The advantage of the proposed algorithms is that they are portable and do not depend on the executable platform such as the operating system, processor or some other components.

Solving this problem by using this approach to can lead to finding new algorithms as well as improving existing ones. The paper presents the implementation of the already existing Lamport's bakery algorithm by using this method. Some of the possible improvements are giving more authority to the control thread and thus enabling only it to enter the critical section and perform tasks instead of other threads. In that case, the threads are required to report that they want to use the mutual shared space and then wait for the control thread to execute certain instructions in the critical section instead of them.

Acknowledgement: This work was supervised by Professor *Nebojša Bačanić Džakula*, from *Singidunum University, Belgrade, Serbia*.

References

- [1] B. Goetz, T. Peierls, J. Bloch, J. Bowbeer, D. Holmes and D. Lea, *Java Concurrency in Practice*, Addison-Wesley, 2006.
- [2] D. Lea, *Concurrent Programming in Java*, Addison-Wesley, 2000.
- [3] M. Herlihy and N. Shavit, *The Art of Multiprocessor Programming*, Morgan Kaufmann, 2008.
- [4] G.L. Peterson, *Myths About the Mutual Exclusion Problem*, Information Processing Letters 12(3) 115-116, 1981.
- [5] L. Lamport, *A new solution of Dijkstra's concurrent programming problem*, Communications of the ACM, 17,8 453-455, 1974.
- [6] M.L. Scott, *Programming Language Pragmatics*, Elsevier, 2009.
- [7] Z. Radivojević, I. Ikodinović and Z. Jovanović, *Konkurentno i distribuirano programiranje*, Akademska misao, 2018.
- [8] M. Živković, N. Bačanić Džakula and E. Tuba, *Programski jezici*, Univerzitet Singidunum, 2019.
- [9] M. Raynal, *Concurrent Programming: Algorithms, Principles, and Foundations*, Springer, 2013.
- [10] S. Oaks and H. Wong, *Java Threads: Understanding and Mastering Concurrent Programming*, O'Reilly, 2004.

Milan SAVIĆ
Singidunum University
Faculty of Technical Sciences
32 Danijelova St., Belgrade
SERBIA
E-mail: milan.savic.16@singimail.rs

Software application for extracting information from Romanian identity cards

*****Constantin-Marius Stanciu

Abstract

The aim of this paper is to propose a method based on Machine Learning technics for detecting and extracting informations from the Romanian identity cards, such as: last name, first name, personal identification number and picture portrait. The final shape of the informaitons are saved in the text format (last name, first name and personal idetification number) and image format (picture portrait). The outcome of the recognition is over 98%. Unlike other systems that allow to scan the identity card through a scanner device, this system have capacity to detect and extract the information from a distance of up to one meter using a webcam integration of 1 MegaPixel. In the present contex given by the pandemy with Corona virus, this system plays an important role in keeping the physical distance while assuring the protection of personal data used often in companys.

1 Introduction

Nowadays the easier access to information and in the same time the need of analysis of the huge available quantity of data have dramatically increased. Software systems based on Machine Learning and Data Mining techniques exploit the information and correlations hidden inside the data. The accuracy and computational cost of these techniques are the most important features aimed to be optimized. In this paper I managed to highlight needs, which in the past were not so important, such as social distance and protection of personal data. The system I managed to develop is able to extract information from the Romanian identity card (ID card) from distance. This distance is debatable, because it can be interpreted depending on the performance of the web cam that we use. The extracted information is then saved in a file or in a database. All this procedure can be done in a second or maybe less depending on the specifications of the computer.

The proposed software system is based on supervised Machine Learning (ML) techniques [12] which learn, based on a set of training examples, to recognize the information from the Romanian identity cards (images). We built our own training data set by collecting, processing and labeling photos with Romanian identity cards. This was one of the most consuming task from our work. We used images publically available in the Internet together with images of identity cards from known people for whom we requested consent for the use of this information.

The implementation of the proposed software are realized mostly in Python [13], but also JavaScript [8] was used to display the data extracted through a table in a web page.

The rest of the article is organized as follows.

2 Methodology

The results reported in the last literature regarding image recognition emphasize the high quality results obtained using artificial neural networks (ANN) [11]. More precisely, recurrent convolutionary neural networks (R-CNN) brought a new level of image recognition. Neural networks aim to solve problems in a similar way to the human brain, using many layers of artificial neurons having weighted connections between them. The weights are adjusted within the learning process, using different algorithms. One of the first and most common is the backpropagation algorithm [2, 11]. The results of ANN can be improved using many layers of neurons. Very good results in image recognition were achieved using Deep Learning (DL). Deep Learning algorithms [5] use multiple layers of neurons to progressively extract features. The term "Deep Learning" firstly referred to the number of layers between the input layer and the output layer [9].

To achieve our purpose, i.e. recognition of information from images containing Romanian identity cards, I used different technologies, the most important being TensorFlow and the Faster R-CNN algorithm [1, 14]. TensorFlow is an open-source software library used to solve ML problems. It is very flexible and offers a wide variety of libraries and resources allowing a easy development and implementation of ML applications [1, 16].

For object detection I used the architecture of the Faster R-CNN algorithm [18] well known for slow detection, but with higher accuracy. The architecture of Faster R-CNN is complex. We start with an image from which we want to obtain:

- a list of bounding box;
- labels assigned to each bounding box;
- probability for each label of the bounding box.

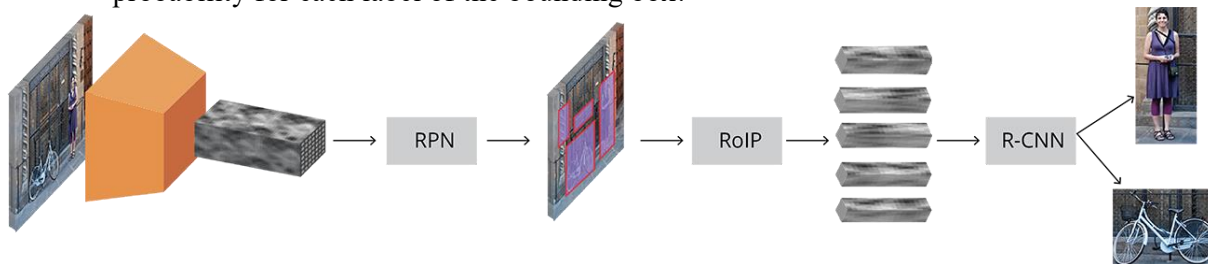


Fig. 1: The complete architecture of Faster R-CNN [18]

In our case the input is represented by an image of an identity card. Using Faster R-CNN we obtain the following bounding boxes: CNP (the unique identifier of a person), Last name, First Name, Photo.

We use Tesseract to recognize the text from the boxes (images) separated using Faster R-CNN. Tesseract is a free software, which deals with optical character recognition for different operating systems, being considered one of the most accurate engines of optical character recognition (OCR - Optical Character Recognition) [9].

3 Application architecture

The architecture of an application mainly depends of its goals.

Our system should be capable to complete the following tasks:

- Taking as input an image; it can be in jpg / png format, video or webcam;
- Detecting from the identity card the classes (labels): Picture (portrait), CNP (personal identification number), last name and first name. Once identified, each class has to be framed in a box with a distinct color;

- Providing the class name and the accuracy of the class recognition (in percentages);
- Cutting the bounding boxes from the initial image;
- Saving the box with label Picture, in image format with the .jpg extension;
- Extracting the text from the bounding boxes for the CNP (personal identification number), last name and first name.

Therefore firstly we need an API for object detection.

The source of the code written in Python [13] containing the object detection API can be found publicly on GitHub [15]. The models for detecting already pre-trained objects can be used if we are interested in categories from the available data sets, being already trained. They are also useful for initializing models when we want to drive new datasets.

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet50_lowproposals_coco	64		Boxes
rfcn_resnet101_coco	92	30	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes
faster_rcnn_resnet101_lowproposals_coco	82		Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	620	37	Boxes
faster_rcnn_inception_resnet_v2_atrous_lowproposals_coco	241		Boxes
faster_rcnn_nas	1833	43	Boxes
faster_rcnn_nas_lowproposals_coco	540		Boxes
mask_rcnn_inception_resnet_v2_atrous_coco	771	36	Masks
mask_rcnn_inception_v2_coco	79	25	Masks
mask_rcnn_resnet101_atrous_coco	470	33	Masks
mask_rcnn_resnet50_atrous_coco	343	29	Masks

Fig. 2: The COCO (Common Object in Context) list of the training models [3]

For building the system in order to detect the information from the identity card, I chose from the table the model **faster_rcnn_inception_v2_coco** (classifier pre-trained with specific neural network architecture), based on the Faster R-CNN algorithm [14], these being characterized by a slow detection, but with much better accuracy.

The application is divided into two parts:

1. Detecting of personal data and extracting them from the identity card. The main working files are located in:

idExtractor

detectLabelsFromImage.py

detectLabelsFromWebcam.py

2. Displaying the information that has been extracted from the identity card in a representative format as possible. The file for displaying the information being located in:
idExtractor
listaInregistrari
index.html

Currently, the application is designed to receive as input data the identity card in two ways:

- in jpg / png image format from the *detectLabelsFromImage.py* file;
- by connecting to the webcam or a video camera, from the *detectLabelsFromWebcam.py* file. In this case the detection will take place in real time, and then pressing the "s" key (from save), this will extract the personal data when we consider that we have positioned as correct as possible the identity card in front of the webcam.

Once the input data has been received from the webcam or in image format, all detected labels: Picture, CNP (personal identification number), First and Last Name, including the whole picture with all detected labels, will be extracted and saved as follows:

- the picture (the portrait from the identity card) and the whole picture with all the detected labels, are saved in the *extracted_images* folder;
- from the CNP (personal identification number), First and Last Name labels, the text will be extracted from the bounding boxes using the Tesseract OCR library, and saved in JSON format (JavaScript Object Notation) in the *data.json* file.

After this process, all the information will be displayed in a web page, opening the *index.html* file from the *listaInregistrari* folder, being represented as a table.

4 Model building and testing

4.1 Creating the training set

4.1.1 Collecting the training ID images

We collected 290 pictures with ID cards in .jpg format, from 9 known people. For each person we collected about 15-35 photos. We respected the European Union Regulation on the Protection of Personal Data (GDPR), and I obtained everyone's consent for including their photos in the training set.

In order to create a representative training set for picture identification in the initial image with the ID card we included images with different backgrounds and different lighting conditions. I chose the resolution of all images to be 1024 x 1024 pixels. The study undertaken by us with different resolutions, revealed that this resolution can include enough information so that the proposed information can be successfully extracted.

Once we pass the configuration stage of the object detector from TensorFlow, we will have images that we can use to train a new personal data detection classifier.



Fig. 3: Photos resized to 1024x1024 pixels

4.1.2 Labeling images

Supervised learning requires a huge amount of manually labeled data. The model compares the data already labels with the recently received data to find errors, and then the model is modified accordingly. This method is used later in model formation. Labeling objects in the image means giving the specific coordinates by drawing a bounding box around the objects that will be predicted in the future. During this time-consuming labeling process, a mistake made in labeling can also lead to wrong predictions. Depending on the project, the number of labels may vary. Some projects require a single label to represent the content of an image. In the current project four different labels are labeled in a single image: Picture (portrait in the identity card), CNP (personal identification number), First Name and Last Name. Basically, the more objects in different categories to be labeled in an image, the higher the working time and the risk of making mistakes.

Labeling is done with the help of various software being available on online labeling platforms or simply a program is installed on the operating system. The labels are of several types, the best known being: 2d, 3d, polygons, lines, etc.

For this project, I will use the 2d labeling type, this is distinguished by a four-sided delimitation box that can have the shape of a square or rectangle. The software I will use for image labeling is Labellmg [10], written in the Python, available for free on Github [15].

For each image that has been labeled, a file with the .xml extension will be created with the Pascal VOC format, that contains the image information (label coordinates, label name, etc.).

4.1.3 Generation of training data

After labeling all the images, each image will correspond to a .xml file with the coordinates of the bounding boxes generated by LabelImg. Next we need these coordinates to generate TensorFlow records also known as TFRecord. To read the data efficiently, it can be useful to have the data serialized and stored in a set of 100-200 MB files each, which can be read linearly. TFRecord is a simple format for storing a sequence of binary records [17]. TensorFlow uses than TFRecord as input data to drive the object detector.

4.2 Network training

The last step we will have to take before we start training the network, is choosing the configuration of the Faster R-CNN model. The configuration is made from a file named "faster_rcnn_inception_v2_coco.config" (the contents of the archive downloaded earlier from the list of pre-trained models Fig.2). In this file we have to make some configurations to define which model and which parameters we will use for training.

The model was trained around 21.5 h. For each training epoch the loss is reported. The accuracy decreases with the number of epochs (or with the training time) After around 140,000 epochss were made the loss fell below the threshold of 0.03.

After successfully prepared the model, all we have to do is to generate an inference graph. This is a file that can be used in applications that want to run our model. It can now be run and implemented in almost any application. This is a very good thing because I used my personal laptop for training, requiring different configurations.

Now the trained model can be run on almost any system at least with a working environment installed from Python and TensoFlow library, without the need for all related and system-dependent configurations.

4.3 Model testing

For testing the model, we need identity cards, that have never been "seen" by our model. As I mentioned in Section 2, the system accept two ways for data input: in image or video format. We will also test our model using these two way of input the the data (ID card).

4.3.1 Model testing with data images of jpg/png type as input

A larger Google search found some publicly accessible identity cards. I collected five subjects, that I will use to check the generalization capabilities of our proposed system.

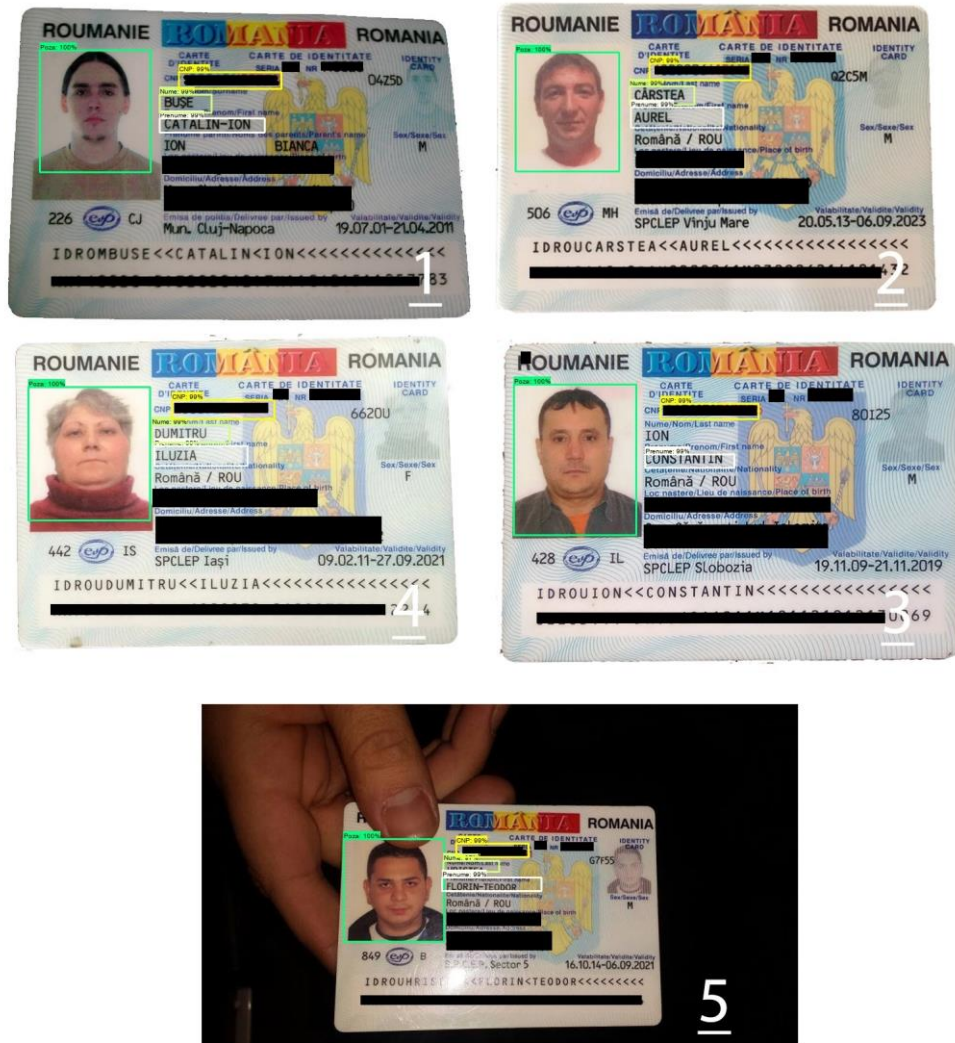


Fig. 5: Labeling of personal details

The detection was successful in almost all cases, except for the subject number 3, where the "name" label could not be detected. For the other labels the detection accuracy was over 98%.

4.1.2 Model testing with data from a webcam as input

For testing our model using as input data obtained from a webcam I used the *detectLabelsFromWebcam.py* option. In this file you can set the address (path). The image source can come from a video recording or in real time from a webcam. For the following example I used a phone from 2015, through which I installed an IP Webcam application available for free in Google Play [7], so I could transmit the image in real time, giving for detection an identity card, from a person who was agreed with the use of his ID card for this test.



Fig. 6: Detection of an identity document never seen before, using the mini-studio

This identity card was "seen" for the first time in the mini-studio. As you can see, the detection was over 99% accurate.

5 Conclusions

We proposed a system capable to identify the features from a Romanian ID card. Currently the detection system is not 100% accurate for all labels because the training set was relatively small. The images have been resized to a resolution of 1024 x 1024 pixels. From what I noticed in most of the cases I tested, the detection of the label "Picture" (portrait picture) from the ID card had the highest rate of accuracy, because this label takes the largest space in an ID photo. Therefore more information is included in this label thus leading to a fairly high accuracy.

For the other labels, "CNP" (personal identification number), "Last Name" and "First Name" the accuracy was not high enough because these labels do not contain enough information.

As future directions of development of the system, I want to build a larger and more diverse training set. I will consider different backgrounds, lighting modes and even black and white images from prints, in order to cover as many scenarios as possible.

I am convinced that this system for detecting and extracting personal data could be very useful in various institutions, especially in the current pandemic situation. The main advantage of presented system is that it can transfer personal data in electronic format, quickly, easily, without transcription errors and with a considerable social distance. The distance can be increased if we use a better video camera to extract personal data from the identity card.

The system could be used in the voting process, automating data extraction from an ID card fast and easy while keeping the physical distance.

Acknowledgement: I thank Prof. univ. dr. Dana SIMIAN for the support provided and useful discussion in the development of this work and the opportunity to study this field of Machine Learning.

References

- [1] Bharath Ramsundar, Reza Bosagh Zadeh, *TensorFlow for Deep Learning*, O'Reilly Media, Inc, USA, 2018.
- [2] Charu Aggarwal, *Neural Networks and Deep Learning: A Textbook*, Springer, 1st ed. 2018.

- [3] COCO (Common Object in Context) list of the training models, https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md, (accessed date: 15.09.2020).
- [4] Li Deng, Dong Yu, *Deep Learning: Methods and Applications*, Now Publishers, 2014.
- [5] Ahamed Hafiz, Alam Ishraq, *Deep Neural Network based Image Recognition*, Scholars' Press, International Book Market Service Ltd, 2019.
- [6] Hosting service for software, <https://github.com>, (accessed date: 15.09.2020).
- [7] IP Webcam, <https://play.google.com/store/apps/details?id=com.pas.webcam>, (accessed date: 15.09.2020).
- [8] JavaScript, <https://en.wikipedia.org/wiki/JavaScript>, (accessed date: 15.09.2020).
- [9] Kay Anthony, *Tesseract: an Open-Source Optical Character Recognition Engine*, Linux Journal, 2007.
- [10] Labeling images with LabelImg, <https://github.com/tzutalin/labelImg>, (accessed date: 15.09.2020).
- [11] Laurene Fausett, *Fundamentals of neural networks*, Prentice-Hall, 1994.
- [12] Tom Mitchell, *Machine Learning*, McGraw-Hill Science/Engineering/Math, 1997.
- [13] Python (programming language), [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)), (accessed date: 15.09.2020)
- [14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.
- [15] Source code of the models from TensorFlow for object detection, <https://github.com/tensorflow/models>, (accessed date: 15.09.2020).
- [16] TensorFlow, <https://en.wikipedia.org/wiki/TensorFlow>, (accessed date: 15.09.2020).
- [17] TFRecord, https://www.tensorflow.org/tutorials/load_data/tfrecord, (accessed date: 15.09.2020).
- [18] The architecture Faster R-CNN, <https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/>, (accessed date: 15.09.2020).

Constantin Marius STANCIU
"Lucian Blaga" University of Sibiu
Faculty of Science
Sibiu
ROMANIA
E-mail: mmscm22@gmail.com

Access control system based on QR Codes

Sebastian Stoica

Abstract

Access control systems are security systems that allow a company to control the access of people in different buildings, offices. It is a fundamental concept to increase security and minimize risks for businesses or various organizations. In many of the company's headquarters, employees use cards to gain access inside the building or to exit the building. In large businesses, access control software is often interfaced with a turnstile system to make unauthorized access impossible. For office spaces, access may be restricted by an access door locking system. These cards can be replaced through an application, which creates a unique QR code for each employee. In the absence of a hardware infrastructure, in this paper we will address the software component of the access control system, based on QR codes, as a cheaper alternative to the card-based solution. Each employee will scan their own QR code, generated through an application installed on the mobile phone, at certain control points installed at the entrance or exit of the company headquarters. This eliminates the cost of physical cards, and scanning the QR code can be done with a simple webcam.

1 Introduction

Preventing the access of unauthorized persons in a company headquarters and also monitoring the working hours of employees are some basic aspects, extremely important in a company. The methods by which these things are done evolve with technology, improving from year to year, starting from locks, simple keys, clocks for measuring time, used in the past, to the newest and most sophisticated electronic systems of today. Granting access can be done by scanning a card, via Bluetooth smartphones and more recently by biometric authentication (fingerprint, facial recognition). Implementing a time tracking system within a company can bring many benefits such as increased productivity, increased accountability and increased transparency. At the same time, a timekeeping system can bring many disadvantages such as low employee morale, stress and high costs for implementing such a system. People who take their time into account are more productive and organized, follow their work in more detail and adjust their productivity barometer. A time tracking system also means greater responsibility at work from the point of view of each employee. Each employee must plan his working hours in the best possible way to complete on time and as efficiently as possible the tasks he received in a project. Effective employee time tracking gives managers a clearer understanding of the work process for each employee. This helps them to plan projects and reserve the resources needed to solve the tasks in a much more efficient way. The purpose of access systems is to minimize the risk of unauthorized access inside a company. Some of the advantages that come with implementing an advanced access control system in a company are:

- remote control;
- avoiding the problems created by losing or stealing the keys and
- day or time restrictions that may be imposed by the employer.

Authentication, through several factors, is often an important part of implementing a multi-level security system. Thus, access / security control systems identify a person, check if that person is who they claim to be, authorizing the level of access and the set of actions associated with the username. The importance of the existence of an access control system within a company is major, in order to increase the degree of security and secure the safety of employees, but also to be able to protect the data and assets that the company owns, in a much more efficient way. Also, the data collected by the access control system can be used successfully in the implementation of a time tracking system. The objective of this paper is to develop an access control system, implemented as a Web application, using Oracle APEX, for granting the employees access to the company's headquarters, based on a QR code. Along with the access control module itself, the application also includes a reporting module used to perform the time tracking and allows the analysis of data by graphical visualization of them. With the help of the Android Studio IDE, the application has also been adapted for mobile phones, which have Android as their operating system. The rest of this paper is organized as follows. Section 2 aims to present the necessary concepts to develop our approach, related to the QR code generation and QR code recognizing, respectively. In Section 3 we present the Oracle APEX, the low-code framework used to implement the Web component of our application. Section 4 contains technical details about the implementation of the Access control and Timekeeping system. Conclusions and further directions of study are formulated in Section 5.

2 Using QR codes for access control

The access control system must ensure the logging of the entrances and exits in / from the monitored premises for the authorized persons, but at the same time it must forbid the passage of the checkpoint of the unauthorized persons. For this purpose, for use in real cases, the application will be interfaced with a turnstile system.

An ordinary user, after logging into the application (Web or Android version) has the possibility to display the QR code based on which the authentication at the control point is performed.

In administrator mode, the application has multiple functionalities:

- Allows the management of departments and staff;
- Managing a daily calendar to specify days off;
- Management of a monthly calendar for calculating the number of days off / working days;
- Daily / monthly interactive reports for calculating the number of hours worked;
- Interactive report for displaying all input and output events;
- Daily / monthly charts to display hours worked / overtime / missing with multiple filtering options for detailed analysis;
- Reports in PDF format for daily / monthly timekeeping at company or department level;
- Administration interface.

2.1 Generating the QR Codes

To each person within the organization is assigned a unique, 6-digit identification number (not a restriction imposed for technical reasons, but a convention within the application). To generate the corresponding QR code, we used a PL / SQL package integrated in an APEX plugin [1]. The solution is based on two components: a PL / SQL package (ZT_QR) with which we can quickly and efficiently generate QR codes directly from the Oracle database and an APEX plugin that simplifies the use of codes generated in Web pages. The QR code can be displayed on the Web

page in two ways: as an HTML table or as a BMP image. In the application we chose to display the QR code as an HTML table and used the highest level for error correction.

The Capability of QR codes to correct errors (restore rate for encrypted data)	
Level L	≅ 7%
Level M	≅ 15%
Level Q	≅ 25%
Level H	≅ 30%

Table 1. Error correction levels for QR codes

QR codes have the ability to restore encoding data if those codes are dirty or damaged. Four levels of correction are available, shown in Table 1. A higher level of correction involves increasing the size of the QR code. Levels Q and H are indicated in industrial environments, and level L is used in clean environments in the context of encoding a large volume of data [2]. Given that it is expected to use the client application on the smartphone and the amount of data stored is small, we considered that using the H level for error correction is the best option.

2.2 Recognition of the QR Codes

The application integrates the jsQR JavaScript library which is available under an Apache license [3]. The library implements a function that extracts the QR code from an image and contains over 10,000 lines of source code. The analyzed image can be extracted from a video stream. This functionality is implemented in the application, the JavaScript code being briefly presented below:

```
var video = document.createElement("video");
var canvasElement = document.getElementById("canvas");
var canvas = canvasElement.getContext("2d");
var outputData = document.getElementById("outputData");

// video stream with the Webcam
navigator.mediaDevices.getUserMedia({ video: { facingMode: "environment" } }).then(function(stream) {
    video.srcObject = stream;
    video.setAttribute("playsinline", true);
    video.play();
    // Decoding the QR code from the frame
    requestAnimationFrame(tick);
});

function tick() {
    if (video.readyState === video.HAVE_ENOUGH_DATA) {
        ...
        canvasElement.height = video.videoHeight;
        canvasElement.width = video.videoWidth;
        // getting the video frame
        canvas.drawImage(video, 0, 0, canvasElement.width, canvasElement.height);
        // the image that will be transmitted to the jsQR library
        var imageData = canvas.getImageData(0, 0, canvasElement.width, canvasElement.height);
        // getting the QR Code from the image
        var code = jsQR(imageData.data, imageData.width, imageData.height, {
            inversionAttempts: "dontInvert",
        });
        ...
        // displaying the QR Code on the Web Page.
        outputData.innerText = code.data;
        ...
    }
    requestAnimationFrame(tick);
}
```

The control of the QR scanner is performed within the application by 3 events, provided by the APEX plugin [4]:

Event	Action
scannerPlay	Starts QR code recognition
scannerPause	Places the scanner in stand-by mode
resetValue	Resets the scanner to allow the recognition of the following QR code

Table 2. Events for controlling the QR code scanner

3 Oracle APEX

To implement the application we opted for Oracle APEX, a low-code framework for developing Web applications in a data-centric architecture. Although APEX is practically integrated into the Oracle database, due to the openness offered by the REST Web services that it can provide or call, the platform is not limited to the Oracle data source. Low-code platforms are designed to accelerate software delivery by quickly building applications for common business use scenarios.

There are currently several low-code platforms, including:

- Zoho Creator;
- Appian;
- PowerApps (Microsoft);
- Mendix;
- Google App Maker.

According to an analysis by Forrester Research [5] in 2020, low-code platforms will bring revenues of over \$15 billion (licensing costs). A Gartner report (an American IT consulting and research company) predicts that by 2024, low-code platforms will be responsible for more than 65% of software development activities [6]. We chose Oracle APEX for development due to our previous experience with Oracle databases, but also due to the following arguments:

1. It is the only free platform with unlimited number of users in Oracle XE.
2. It offers the possibility to install on-premise applications (on your own server) and in the cloud. For example, PowerApps and Google App Maker only have cloud versions (and are not free).
3. Provides great flexibility, from no-code use to advanced Web application development [7]:
 - **No code**: Can be used to build forms, repots, graphs, with predefined components and access tables / views in the database;
 - **Low code**: A developer with experience in PL / SQL can add business logic and additional functionality without having advanced knowledge of Web technologies;
 - **Full code**: It involves the complex use of the platform, in the back-end (SQL, PL/SQL) and front-end (JavaScript, CSS, HTML, Oracle JET, SOAP/ REST Web services).

Oracle APEX is also an extensible platform, through the ability to integrate and develop plugins within the framework.

In the case of the access control application, we used the platform in full code manner, the application covering the back-end spectrum, front-end and plugin integration.

An application developed using Oracle APEX is structured in three levels: customer level, business level, data level. The client level is materialized in the application interface, which runs in a Web browser, the business level is the engine / logic of the APEX application and the data level is

represented by the database itself and REST API interfaces for accessing different systems or data locations.

The application is accessed by the user through a Web browser. From the Web browser, HTTP (Hypertext Transfer Protocol) requests are sent, which are directed to ORDS (Oracle Rest Data Services). ORDS runs on a Web server such as Tomcat. Requests to the APEX engine are sent to PL / SQL packets, which make up this engine. The APEX engine runs in an Oracle database, which can be a licensed version (Standard, Enterprise) or the free version of Oracle Database Express Edition (XE).

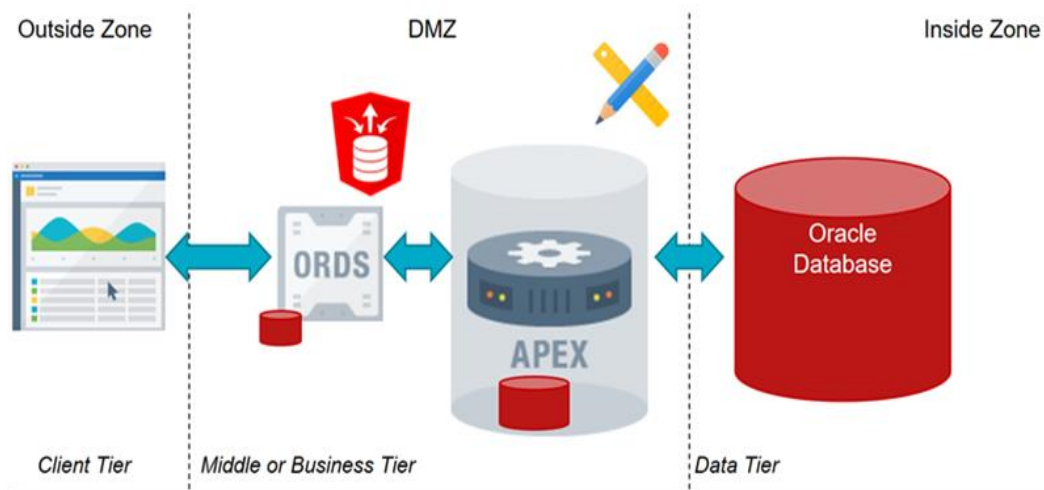


Figure 1. Multi-level architecture of Oracle APEX application [8]

Oracle APEX offers many features and various tools such as: *SQL Workshop*, *Application Builder*, *Team Development*, *App Galery*. *SQL Workshop* comes with an extremely powerful and very useful set of tools. The tools are: *SQL Commands* and *Object Browser*, for directly manipulating the database, *SQL Scripts* for script management, *Utilities* for data import / export, *RESTful Services* for providing REST services through ORDS (Oracle Rest Data Services) for accessing objects from the database. *Application Builder* allows the developer to create a custom application, very quickly, after completing a few simple steps. *App Galery* allows the user to download and install many pre-existing applications in Workspace, which he can run or modify and use as needed. *Team Development* is a tool used by the development team to track new tasks or known issues.

Oracle APEX allows the use of jQuery in Web pages as JavaScript code executed in the browser. jQuery selectors are also used in defining dynamic APEX actions. Oracle APEX has its own JavaScript API [9] that uses jQuery intensively.

The book [10] presents advanced techniques for using jQuery in APEX applications. Oracle APEX uses AJAX to implement *Ajax Callback* processes. These processes contain PL / SQL code and are invoked from the browser via AJAX calls.

4 Access control and time tracking system

4.1 The access control module

In normal working mode, after logging into the application, to the user is presented on the main page the option to generate the QR code in order to present it at the control point.

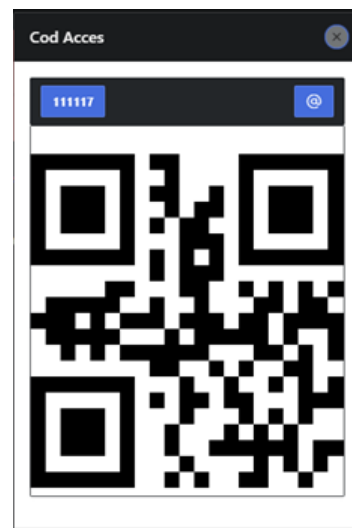
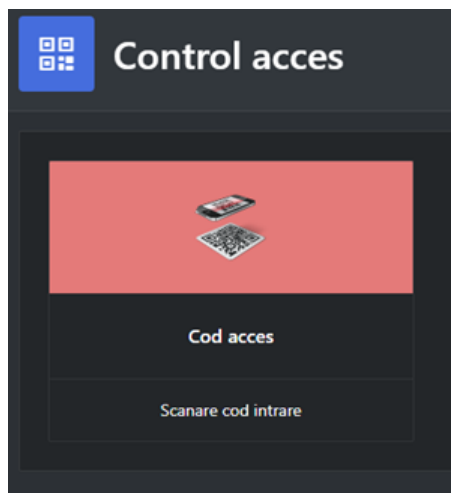


Figure 2. The option to generate the QR code Figure 3. The generated QR code for the access control

When the QR code is presented to the webcam at the control point the application recognizes the QR code:

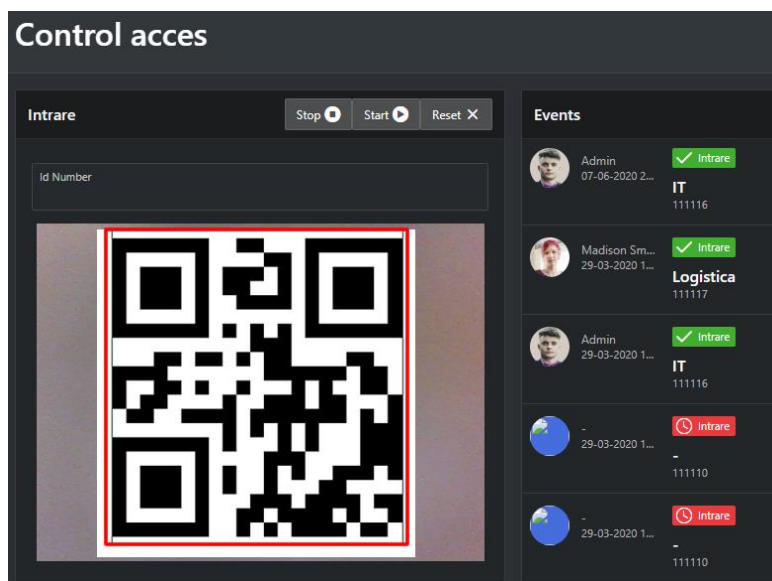


Figure 4. Scanning the QR code at the checkpoint

The information representing the user's ID is decoded and a window is displayed for 4 seconds confirming the user's identification and granting access (Figure 5).

The self-closing function is implemented in JavaScript, by simulating a click on a hidden button:

```
setTimeout(function(){
  document.getElementById('cancel').click();
}, 4000);
```

If the QR code is not recognized (the user is not registered in the database), a window with the message "No access!" is displayed (Figure 6).

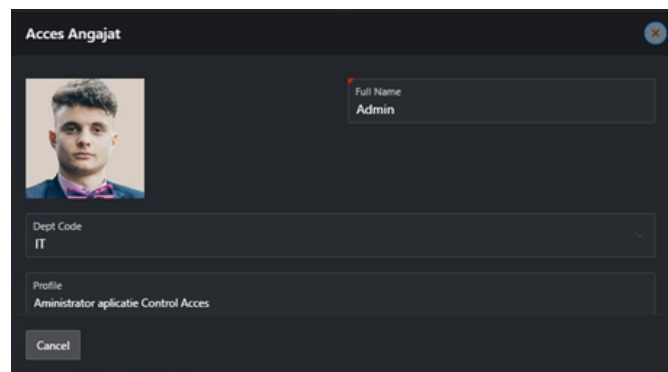


Figure 5. Recognizing the QR code at the checkpoint and granting access

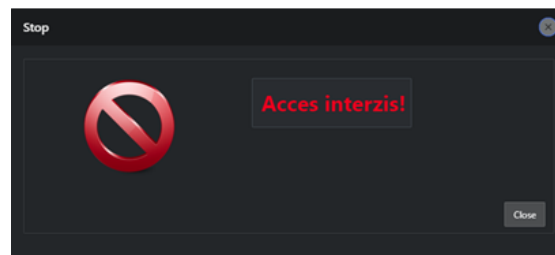


Figure 6. Denying the access to an unregistered QR code

In the administration mode, there is the possibility to set the two modes of the QR scanner: input and output, respectively. This can determine the time spent by the company's staff within the institution and can prepare various reports for timekeeping and other analysis. Selecting the Input or Output option on the Home page will set the hidden field P6_IN_OUT on page 6 (Access Control) to the value Input or Output, respectively.

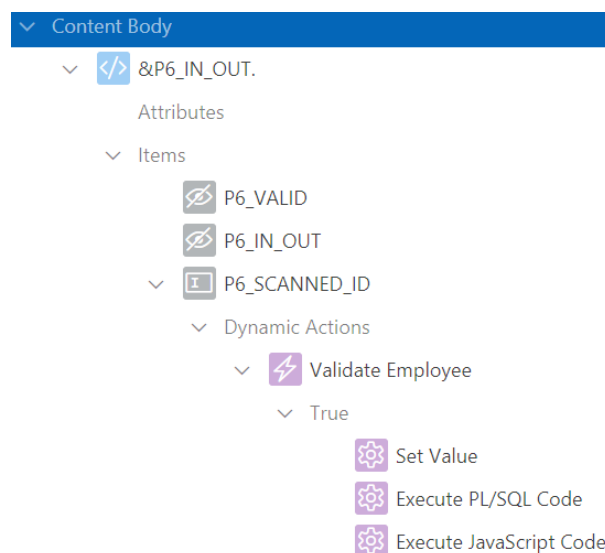


Figure 7. The *Control Access* page in design mode

The decoded value of the QR code is stored in the P6_SCANNED_ID page field. For this field there are set 3 dynamic actions that are executed successively when changing the value of the respective

field. The first dynamic action is implemented in PL / SQL code and has the role of determining if the scanned QR code is the code of a registered user:

```

declare
  l_result varchar2(2);

BEGIN
  select case when nvl(username,'-') = '-' then 'NO' else 'YES' end into l_result from employees
  where ID_NUMBER = :P6_SCANNED_ID;
  return l_result;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    return 'NO';
END;
```

The value returned by the dynamic action is stored in the P6_VALID page field. The second dynamic action is also implemented in PL / SQL code and has the role of journaling the attempt to pass through the control point:

```

declare l_in_out varchar2(1);
  l_valid varchar2(1);
BEGIN
  APEX_UTIL.SET_SESSION_STATE('P9_ID_NUMBER',:P6_SCANNED_ID);
  if :P6_IN_OUT = 'Input' then
    l_in_out := 'I';
  else
    l_in_out := 'O';
  end if;

  if :P6_VALID = 'YES' then
    l_valid := 'Y';
  else
    l_valid := 'N';
  end if;

  if not(trim(:P6_SCANNED_ID) is null) then
    insert into EVENTS
      (ID_NUMBER, DATE_TIME, IN_OUT, VALID)
    values
      (:P6_SCANNED_ID, sysdate, l_in_out, l_valid);
  end if;
END;
```

The following information will be stored in the database:

- Access mode (I = Input or O = Output);
- Valid user (Y) or unrecognized QR code (N);
- Decoded QR code value (P6_SCANNED_ID).

The third dynamic action is to confirm the granting of access and the prohibition of access. The implementation is performed in JavaScript and depending on the validation of the QR code, one of the two hidden buttons "go" and "stop" is activated, which displays the windows for granting access (Figure 5) and for banning access (Figure 6), respectively.

```

var valid = $v('P6_VALID');
var id_scanned = $v('P6_SCANNED_ID');

apex.region('events').refresh();

if (valid == 'YES') {
  $("#go")[0].onclick();
}
```

```

else
  if (id_scanned.length > 0) {
    $("#stop")[0].onclick();
  }

```

4.2 Operational, interactive reports

The display of events (input / output) is done through a *Classic Report* for which a *Timeline* template has been set (Figure 8).

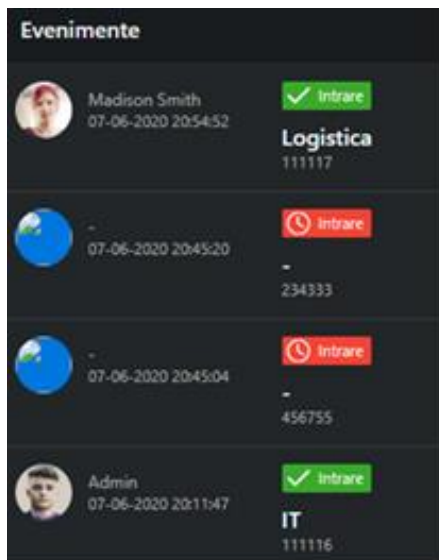


Figure 8. *Timeline* report

```

select e.id event_id
, t.id
, null event_modifiers
, null event_attributes
, null user_color
, dbms_lob.getlength('PHOTO_BLOB') user_avatar
, t.full_name user_name
, to_char(e.date_time, 'DD-MM-YYYY HH24:MI:SS') event_date
, (case e.valid
  when 'N' then 'is-removed'
  when 'Y' then 'is-new'
end) event_status
, case when e.in_out = 'I' then 'Input' else 'Output' end event_type
, (case e.valid
  when 'N' then 'fa fa-clock-o'
  when 'Y' then 'fa fa-check'
end) event_icon
, d.dept_name event_title
, e.id_number event_desc
, null event_link
from events e left join employees t on t.id_number = e.id_number
left join departments d on t.dept_code = d.dept_code
where e.IN_OUT = case when :P6_IN_OUT = 'Input' then 'I' else 'O'
end
order by e.date_time desc

```

Figure 9. Data source of the logging report

Three operative reports were implemented to display the entry / exit events and to determine the time spent by employees at the company's headquarters, daily and monthly, respectively. The reports are of the *Interactive Report* type (allow customization by the user at runtime, saving the settings made). In Oracle APEX, interactive reports have various configuration options that can be set when the report is displayed:

- displaying and hiding the columns;
- filters that can be applied at row / column level;
- sorting records;
- generating new columns (*Compute* option);
- data aggregation and grouping;
- highlighting the records according to a specific criteria (*Highlight*);
- creating an ad-hoc chart / pivot, etc.

Figure 10 shows the daily report of hours worked, in which 3 interactive configuration options were set:

1. Filter by Department column;
2. Grouping by Department column;
3. Highlight for insufficient hours / day (Hours < 8).

Report Zilnic Raport Lunar Intrari - Iesiri

From 01-06-2020 18:48:00 To 05-06-2020 18:48:00 Report Zilnic

Department = 'Logistica'

Department

Ore insuficiente

Department : Logistica

Ziua ↑=	Angajat	Ore	Minute
02.06.2020	Harold Youngblood	8	9
02.06.2020	Madison Smith	8	27
03.06.2020	Harold Youngblood	8	18
03.06.2020	Madison Smith	8	0
04.06.2020	Madison Smith	7	37
04.06.2020	Harold Youngblood	8	17
05.06.2020	Madison Smith	7	52
05.06.2020	Harold Youngblood	7	25

Figure 10. Interactive report – hours worked (daily)

4.3 Graphs for data analysis

The component *Faceted Search* allows easy data filtering (contains check box fields, list of values, etc.). In APEX, filtering is applied by default to a *Classic Report* component. To use the filtering mechanism to display an interactive graph that displays data filtered through the *Faceted Search* component, we will use the API function:

```
FUNCTION OPEN_QUERY_CONTEXT (
  p_page_id      IN NUMBER,
  p_region_id    IN NUMBER)
  return apex_exec.t_context;
```

With this function we can access the filtered data, normally presented by the *Classic Report* component. For this purpose we have implemented the functions:

```
create or replace function get_faceted_search_data_days(
```

```
  p_page_id      in number,
  p_region_static_id in varchar2 )
  return t_worked_hours_day_table pipelined
```

```
create or replace function get_faceted_search_data_months(
```

```
  p_page_id      in number,
  p_region_static_id in varchar2 )
  return t_worked_hours_month_table pipelined
```

to build a daily and monthly schedule. The *p_page_id* parameter is the page number where the *Faceted Search* component resides, and *p_region_static_id* is the static ID of the *Classic Report* component.

Figure 11 shows the daily schedule of hours worked at the department level.

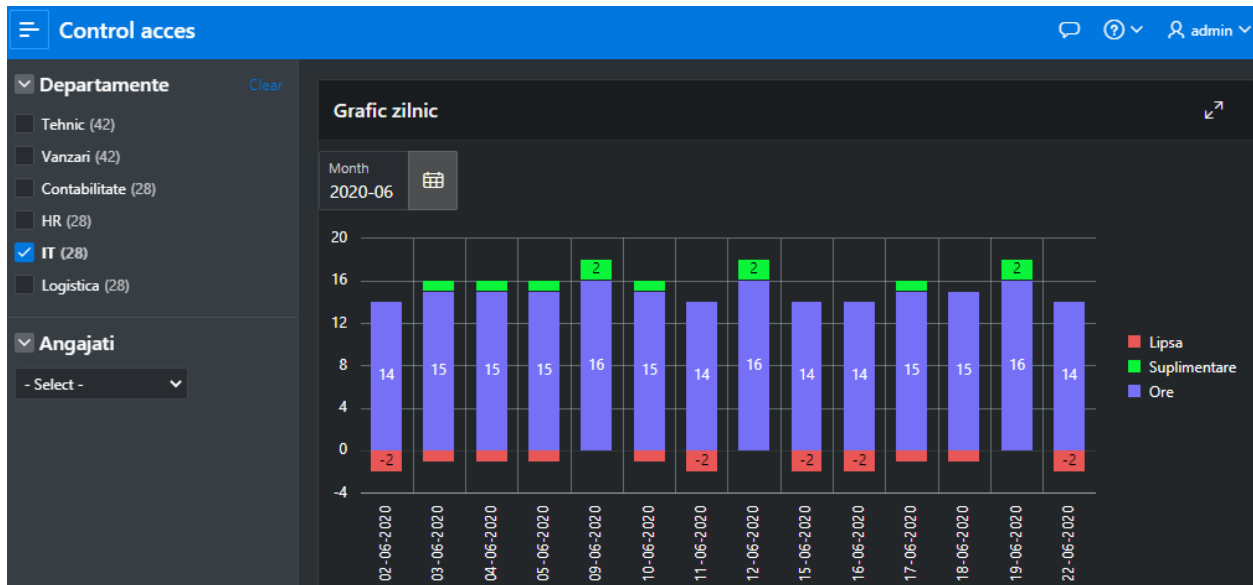


Figure 11. Graph of hours worked daily – filter by department

Figure 12 shows the daily graph in which the filtering was performed at employee level:

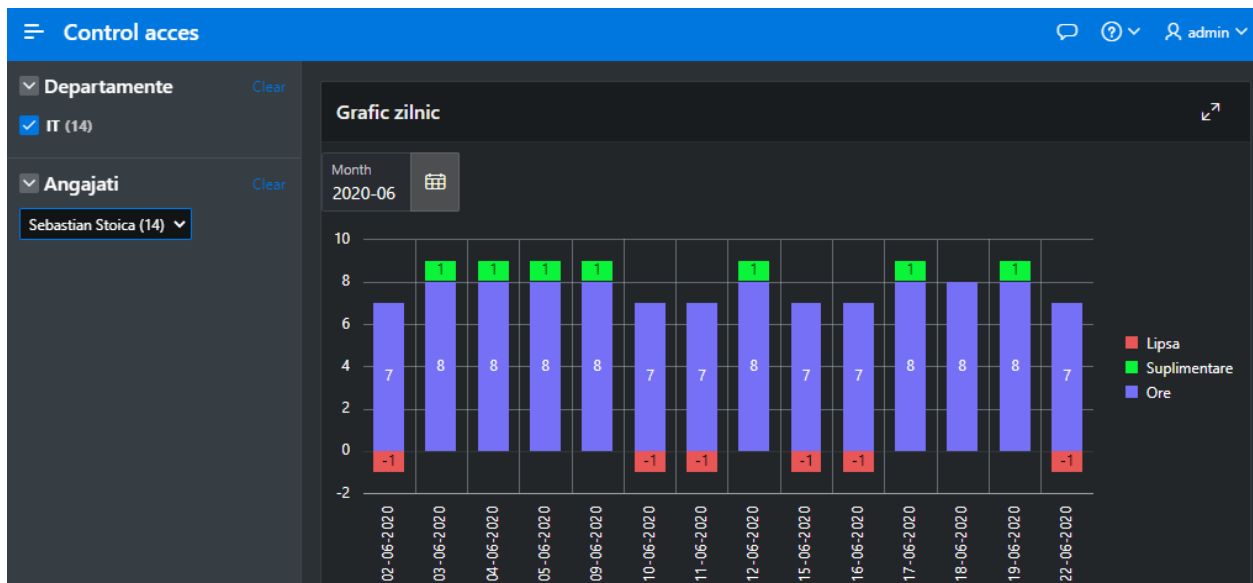


Figure 12. Graph of hours worked daily – filter per employee

5 Conclusions

Through this application, a new approach to access control systems is proposed, a modern, practical and at the same time much more efficient approach, which keeps up with the pace at which technology is evolving. Nowadays, the idea of transferring physical cards to the mobile phone, using a Wallet application, is increasingly supported. The same can be done through this type of access control system, which eliminates the cards that employees have to wear to access the office, with an application that generates a unique QR code for each employee. The access control application was developed using Oracle APEX, one of the most popular and powerful low-code platforms in Web

application development. Oracle APEX is in full development and is a low-code platform that will continue to evolve radically in the years to come. This low-code platform offers a set of standardized but configurable, flexible and extensible tools, to the developers eager to create modern applications, powerful and as efficient as applications made using traditional programming methods, all in a much shorter time. The application can be accessed by the HR department, employees or managers, from a Web browser, using a desktop, or from the phone in the format of an Android application, developed in Android Studio. The application also contains a time tracking system, which records the entrances and exits of employees and provides a set of information related to the hours worked daily and monthly by each employee. As a future direction of development, the application can be expanded by interfacing with a turnstile system. The access control system will also provide the facility to set detailed authorization levels (by departments and / or individuals) at the access points.

References

- [1] Oracle PL/SQL Package and APEX plugin for QR Code Generation
<https://github.com/zorantica/plsql-qr-code>
- [2] QR Code Correction Feature
https://www.qrcode.com/en/about/error_correction.html
- [3] JavaScript QR code reading library
<https://github.com/cozmo/jsQR>
- [4] APEX QR Code Scanner
<https://github.com/RonnyWeiss/APEX-QR-Code-Scanner>
- [5] The Best Low-Code Development Platforms
<https://www.pcmag.com/picks/the-best-low-code-development-platforms>
- [6] Gartner Report: Low-Code Development Technologies Evaluation Guide
<https://www.mendix.com/resources/low-code-development-technologies-evaluation-guide/>
- [7] Holger Mueller, Oracle APEX Brings No Code and Low Code to the Oracle Database Ecosystem, 2018,
<https://www.oracle.com/assets/apex-oracle-db-ecosystem-4491616.pdf>
- [8] Lucas Jellema, Oracle APEX: the low-code and low-cost application middle tier, 2018
<https://technology.amis.nl/2018/11/01/oracle-apex-the-low-code-and-low-cost-application-middle-tier/>
- [9] Oracle Application Express JavaScript API Reference
<https://docs.oracle.com/en/database/oracle/application-express/19.2/aexjs/index.html>
- [10] Scott Wesley, *Pro jQuery in Oracle Application Express*, Apress, 2015, ISBN-13: 978-1-4842-0961-5

Sebastian STOICA
Lucian Blaga University of Sibiu
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str.
ROMANIA
E-mail: sebastian.stoica@ulbsibiu.ro

Analysis of the Operating Costs of a Decentralized App in the Ethereum Blockchain

André Stollberger, Tobias Fertig, Andreas E. Schütz, Karsten Huffstadt, Nicholas H. Müller

Abstract

The open source blockchain Ethereum is most popular public blockchains. Unlike Bitcoin, Ethereum supports the development and execution of decentralized applications (DApps) as well as smart contracts. However, there is a lack of information about the operating costs of such a DApp on a public blockchain like Ethereum. In case of Ethereum, the costs of a DApp is based on the amount of transactions and their so-called gas price. The gas price depends strongly on the calculation effort of the respective transaction. In order to get an overview of these costs, we analyzed 350 DApps which are deployed on the Ethereum blockchain. Moreover, we classified those DApps in different categories, such as games or finance. We found out that the average costs in some of the categories differ widely from the others. Furthermore, we propose different approaches how to obtain the required data from the Ethereum blockchain and we discuss the advantages and disadvantages of every approach.

Keywords: Blockchain, Ethereum, Smart Contracts, Decentralized Applications, DApps, Cost Analysis, Ethereum Analytics, Gas Price

1 Introduction

The public blockchain solution Ethereum is well-suited for the development and deployment of DApps. Tang, Huimin, Yong Shi, and Peiwu Dong showed in their paper [16] that Ethereum is among the top three public blockchains. Ethereum supports the storage of source-code in the blockchain and comes with an own virtual machine. This enables the development and execution of decentralized applications (DApps). The following analysis will refer to DApps of the Ethereum blockchain.

Unlike private blockchains, in a public blockchain you do not have a direct provider who sets the prices for storage space and computing capacity. Much more the costs depend on the individual transactions and the current exchange rate of the respective cryptocurrency. Within our research, we have the goal to create transparency by analyzing 350 different DApps that will be executed on the Ethereum Blockchain until the 04/27/2020 and determining their respective costs dependent on the category of the software.

In this paper we will first explain the basics of the Ethereum Blockchain, which are needed to understand the later steps. Then we will discuss where the data for the analysis comes from. Afterwards we will evaluate the collected data and discuss the following research questions: What are the costs to operate a DApp on the Ethereum blockchain? Are the costs of a

DApp/Transaction dependent on the category? Finally, the summary concludes the paper with an outlook on future research topics that have emerged in the course of this study.

2 Related Work

Massimo Bartoletti and Livio Pompianu show in their elaboration [2] how widespread the use of smart contracts in different categories on the Ethereum and Bitcoin blockchain is. In addition, they present different design patterns for smart contracts on the Ethereum blockchain and provide an analysis of how they are used in those categories.

Using a transaction graph, Wren Chan and Aspen Olmsted described in their paper [3] different problems in anonymizing the users of the blockchain. Sergei Tikhomirov et al. also discovered security problems with the use of smart contracts in their work [18]. Sara Rouhani and Ralph Deters investigated the execution speed of transactions on the Ethereum blockchain [13].

However, there is a lack of information about the actual costs of a DApp on the Ethereum blockchain. Therefore, we conducted an analysis based on the transactions of several DApps. Moreover, we performed an analysis for each of the different categories of DApps.

3 Ethereum Basics

In order to carry out an analysis of the costs of a DApp on the Ethereum Blockchain, we had to look at the structure and function of the Ethereum Blockchain. Gavin Wood described exactly this structure and setup in his paper [19]. The current state of the blockchain results from the state of all accounts that are registered in the blockchain. There are two different types of these accounts: The first are the externally owned accounts, which can be controlled by one person. The second are the so-called contract accounts. These are controlled by program code that is executed in the blockchain. A DApp can consist of one or more such contract accounts.

These accounts can communicate with each other using transactions. For example, one account can transfer a certain amount of Ether to another account. Each transaction executed on the Ethereum Blockchain costs a certain fee. This fee is based on the amount of work that the transaction generates on the blockchain and is called Gas. A single transaction does not include any data field to determine how much gas this transaction has consumed. This is due to the fact that it is only possible to determine how much gas was actually consumed after the transaction has been executed on the blockchain. After the execution a so-called transaction receipt is created. Here you can see how expensive this transaction was and if the transaction was completed successfully. Even if a transaction fails, it still costs the user a certain amount of gas. However, this amount is usually not as high as it would have been if the transaction had been completed successfully. Nevertheless, if a developer accidentally uses the assert mechanism the gas costs will exceed the costs of a successful transaction.

Gas - the transaction fee - is calculated in Wei. Wei is the smallest in the Ethereum blockchain and can be imagined like the cent of the dollar. 100 cents add up to a dollar, while 1,000,000,000,000,000 Wei equal 1 Ether. For every 1000 steps there is a separate term. Table 3 represents these terms.

The issuing contract and the receiving contract are referenced in a transaction with their respective addresses. In this way, the costs incurred by a transaction can be assigned directly to a particular contract. To create a contract on the blockchain, an initial creation transaction

Unit	Amount in Wei
Wei	1
Kwei / Femtotether	1.000
Mwei / Picoether	1.000.000
Gwei / Nanoether / Nano	1.000.000.000
Szabo / Microether / Micro	1.000.000.000.000
Finney / Milliether / Milli	1.000.000.000.000.000
Ether	1.000.000.000.000.000.000

Table 1: Overview of the individual units of ether

is executed. In this transaction, the system checks which address can be assigned to the new contract. This transaction also costs a certain fee on the blockchain. However, since this address is not yet known before the execution of the transaction (whereas it is possible to determine the address manually beforehand), it can only be provided in the transaction receipt.

To check the transactions for validity, they are verified by each participant in the blockchain. If a certain number of participants declare the transaction valid, it is accepted and executed by the blockchain. Each participant then inserts the transaction into its copy of the blockchain. To become a participant in the blockchain, you must first download it. This approach makes it clear that the Ethereum blockchain data management is decentralised [9].

4 Crawling Data from the Ethereum Blockchain

To determine the costs of a contract, as described in Section 3, all transactions addressing that contract that are on the blockchain and their respective transaction receipts have to be considered. There are several ways to access these transactions, which are described in more detail in this section.

4.1 Etherscan.io

Etherscan [6] describes itself as a block explorer-, search-, API- and analysis platform for Ethereum. There it is possible to view all blocks, transactions, transaction receipts, and contracts. It would be possible to develop a web crawler, which automatically crawls all required transactions and transaction receipts for the analysis. Since all transactions in the Ethereum Blockchain must be searched for the analysis, more than 716,362,148 [8] pages on Etherscan must be parsed. Assuming an average time of 10ms to parse a page, it would take 83 days to parse all transactions. Due to the high frequency query of the Etherscan website the IP address of the crawler would probably be blocked due to a Denial of Service [12] Suspicion.

Moreover, Etherscan offers an API to query the various data of the blockchain. However, it is only possible to query the 500,000 most recent transactions. Since all transactions of the blockchain are required to determine the costs of a DApp. This approach is not suitable for our purpose.

4.2 JSON RPC

To retrieve data directly from a node of the blockchain, it is possible to use JSON RPC [1] to send individual requests to that node. The node then returns the corresponding result from the

blockchain. There are different requests for each element in the blockchain. Be it a transaction, block, or transaction receipt.

The blockchain data is freely available on the Web and can be downloaded free of charge. There are different modes [14] for the download:

- Light Node: Saves only the header chain. The complete content of each block is missing and is only downloaded on request from another Archive Node. Therefore, this mode is not suitable.
- Full Node: Stores the complete blockchain data on hard disk and can supply the network with any data on request. All states can be derived from a complete node.
- Archive Node: Stores everything kept in the full node. Also builds an archive of historical states. This mode contains irrelevant data for our purpose.

Currently the size of a full node is around 360 GB of data [7]. With an average download speed of 12 MB/s this download would take about 21 days. Alternatively, there are external providers, such as Infura [5], which grant access to a node of the blockchain. But there is a fee for each query sent to the Infura node. The total number of requests for all transactions and transaction receipts is 1,432,724,296. For the largest subscription of 5,000,000 requests per day you pay monthly 1000\$. This means that 10,000\$ will be charged for 287 days to process all requests.

4.3 Google BigQuery

BigQuery [17] is a fully-managed, serverless data warehouse that enables scalable, cost-effective and fast analysis over petabytes of data. BigQuery is a serverless Software as a Service (SaaS) that supports querying using SQL. Moreover, creating, editing and managing your own databases is possible. Besides the self-created databases there are also a number of public databases. These can be viewed and queried by every user of BigQuery. Within the Web application it is possible to write SQL queries, execute them and view the results. In addition, there are various APIs which can access these databases via various programming languages, such as C#, Go, Java, Node.js, PHP, Python, Ruby. The pricing strategy of Google BigQuery behaves differently than that of Infura. In addition to the costs for the used storage space of the own databases, a fee of 5\$ for each processed TB is charged. The first processed TB per month is free.

Among the public databases a full copy of the Ethereum Blockchain [11] is available. The blockchain data is updated daily. Moreover, the blockchain data is splitted into different SQL tables. There are tables for blocks, transactions and contracts. All transactions are stored with the corresponding transaction receipts. The transactions and transaction receipts together occupy approximately 307.6 GB of disk space. With a suitable SQL query, the amount of data processed can be reduced to approx. 77.1 GB. The time needed to process this query is about 4 seconds.

5 Gather Information about DApps

A DApp can have more than one contract. Therefore, a reliable source from which this information can be obtained is required. State of the DApps [15] is a not-for-profit curated directory of DApps, which runs on various several blockchains. State of the DApps was initially created to

categorize and showcase developed projects built on the Ethereum Blockchain, but more recently added support for EOS, Steem, Hive and many more blockchains.

Every DApp project can register itself on State of the DApps and deposit information about their respective DApp there. Besides the name and a description of the DApp, there is information like the blockchain on which the DApp is running, the status of the DApp, the contract addresses and a category of the DApp. With the contract addresses it is possible to assign the costs that can be assigned to each contract directly to the corresponding DApp. The costs of a single DApp are not really meaningful. Therefore, it is useful to group them by their category. The existing categories are:

- **Development** DApps where users can create and edit various things like contracts. Examples: Band Protocol, Unstoppable Domains, etc.
- **Energy** Here are all DApps listed, which deal with the management of electricity in any kind. Examples: SunContract, Charg (CHG), etc.
- **Exchange** All DApps that deal with token trading are listed here. Examples: KyberNetwork, Uniswap, etc.
- **Finance** Be it trading or observations of financial markets, then these DApps are included here. Examples: MakerDAO, Aave Protocol, etc.
- **Gambling** DApps that have to do with any kind of gambling fall into this category. Examples: FunFair, PoolTogether, etc.
- **Games** The most popular type of DApps are probably games. Therefore they have their own category. Examples: Axie Infinity, My Crypto Heroes, etc.
- **Governance** DApps which take care of collaboration and management of teams belong to this category. Examples: Aragon, Colony, etc.
- **Health** Everything to do with health, whether it is the storage of health data or fitness programs can be found here. Examples: BEAT, etc.
- **High risk** Shares or crypto currencies, here you will find DApps where investments in these currencies can be made. Examples: Forsage, HEX, etc.
- **Identity** The Management of your own identity on the Internet, can be done by the DApps in this category. Examples: SelfKey, Dock, etc.
- **Insurance** Insurance or juridical matters, like flight delays, all these DApps are at home in this category. Examples: Nexus Mutual, Etherisc Flight Delay Insurance, etc.
- **Marketplace** The DApps in this category are dedicated to the sale of real estate and other items. Examples: OpenSea, Origin Protocol, etc.
- **Media** Music, videos, advertising or other media can be distributed via DApps from this category. Examples: Livepeer, AdEx, etc.
- **Property** The secure storage and encryption of personal data and files, is the task of the DApps in this category. Examples: Decentraland, Ethereum Name Service, etc.

- **Security** Data or transactions that need to be managed in a secure manner are made through these DApps. Examples: Chainlink, V-ID, etc.
- **Social** DApps like Facebook or Instagram, but with decentralized data management, can be found here. Examples: 2key, Foresting, etc.
- **Storage** Cloud storage and data management, that's what DApps in this category are responsible for. Examples: Storj, X Cloud, etc.
- **Wallet** Crypto currencies and other digital trading objects can be managed via these DApps. Examples: Status, Basic Attention Token, etc.

State of the DApps provides a ranking of all DApps of a blockchain. These rankings can be sorted with different filters. The DApps can be sorted by the number of active daily users, by the ratio of transaction volume to DApp contracts or by developer activity. To find a large number of active DApps in daily use, the filter sorted by user activity is best.

In order to effectively collect the required information on the respective DApps, it is possible to parse them via a web crawler. The first 350 DApps are searched within the ranking and then the name, the category and all contract addresses on the Mainnet blockchain are parsed via the corresponding DApp page. There are also contracts on other networks than Mainnet. Ropsten, Kovan and Rinkeby are test blockchain networks. However, Mainnet is the currently active live network of the Ethereum blockchain. Ropsten is a Proof of Work (PoW) [10] Blockchain where Kovan and Rinkeby are Proof of Authority (PoA) [4] blockchains and are deployed due to attacks on the Ropsten blockchain. Because Ropsten, Kovan and Rinkeby are only test networks, those contracts are not needed.

6 Results

As described in Section 5 the name, category and contract address of the first 350 DApps will be crawled from the ranking with the most active users. The reason for only crawling the first 350 DApps is that the other DApps did not have any user activity at the time of our crawling. Therefore, we could not analyze their costs. Afterwards, all transactions of the Ethereum Blockchain are searched using Google Big Query and their costs are assigned to the respective contract addresses.

6.1 Problems

While parsing the DApp data, it is noticeable that some DApps use the same contract as another DApp. This can have several reasons:

- The two DApps are from the same developer team and can therefore communicate with each other.
- Two DApps have two contracts in common, with one contract being used for communication in each direction.
- In one of the DApps you trade or gamble with different currencies. To access a wallet directly, the DApp uses the same contract as the Wallet DApp for communication.
- The functionality of two DApps are handled by the same contract.

The information that is available for a transaction and the respective contract is not sufficient to assign it to the right DApp if the contract is used by more than one DApp. Therefore, it must be decided individually for each contract how to allocate the costs.

- If the DApps are in the same categories, the costs can simply be assigned to one of the two DApps, since the evaluation is performed on the individual categories at the end.
- If both DApps use the same contract but are not in the same category, the costs must be statistically distributed based on the size of the respective DApps.

Another problem in matching the data, is that State of the DApps stores contract addresses case sensitive and Google Big Query stores these addresses in lower case only. With hash values, upper and lower case should not really matter. This was the case at the beginning of the Ethereum blockchain. To ensure a certain security when entering a contract address, the EIP-55 [1] encryption was introduced in the addresses of the Ethereum blockchain. EIP-55 is a small error detection for the addresses. This is used to prevent, that a user has mistyped a contract address and may have sent ether to the wrong address. Once the ether has been sent, it cannot be retrieved. Since this functionality was not available at the beginning of the Ethereum blockchain, the checksum check must not necessarily be implemented by every DApp.

Therefore, when checking whether a transaction belongs to one of the contract addresses, the case sensitivity must be ignored. In this way, each transaction can be assigned to the respective contract.

6.2 Evaluation

As mentioned at the beginning of this section, the first 350 DApps with the most user activity are part of the evaluation. Of the 350 DApps, 9 DApps are not included, either because they have not yet made a transaction on the blockchain or because they have shared all contract with another DApp in the same category. Figure 1 shows the distribution of DApps across the categories. For the category Insurance no DApp could be found, since all of them are in a development status and therefore have not yet executed any transactions on the blockchain.

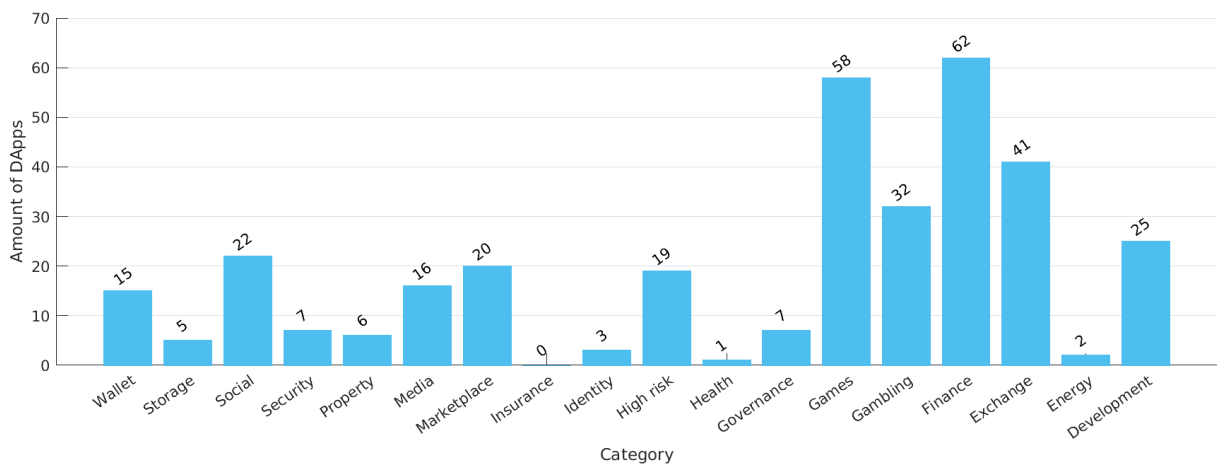


Figure 1: Amount of DApps per Category

All DApps together made 71,907,411 transactions. Figure 2 shows the distribution of transactions across the various categories. Since there are only a few DApps in the Health category,

this category offers the fewest transactions with only 2780 transactions. Due to the high-speed transactions of individual tokens, a large number of transactions are generated in the Exchange category. With 29,012,133 transactions, this category clearly stands out from the other categories. All transactions have a total value of 944893,754991901 ether. The total costs per category are shown in Figure 3. 4 can be used to determine the average cost of a transaction in each category.

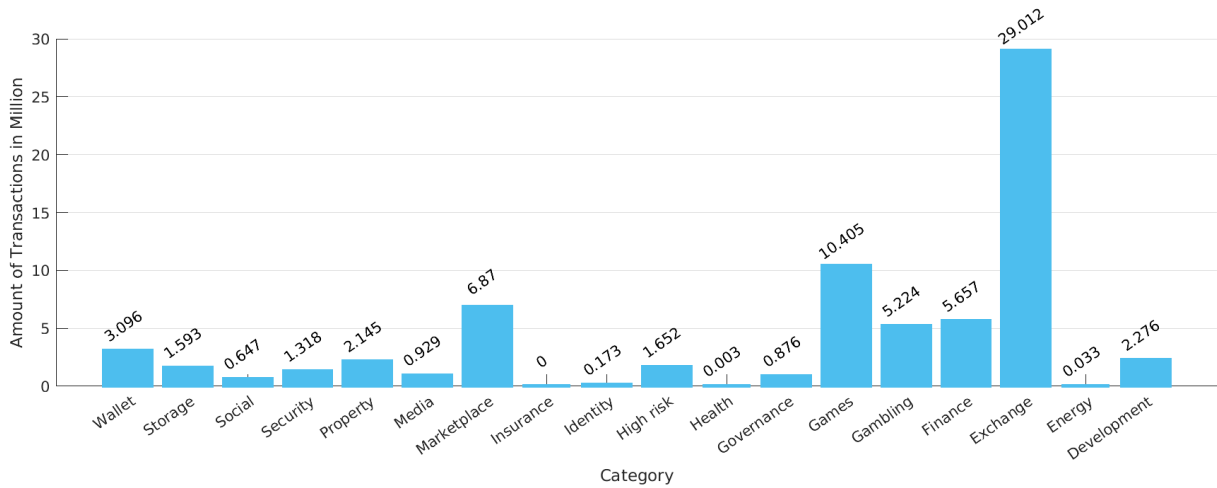


Figure 2: Amount of Transactions per Category in Million Rounded to Three Decimal Digits

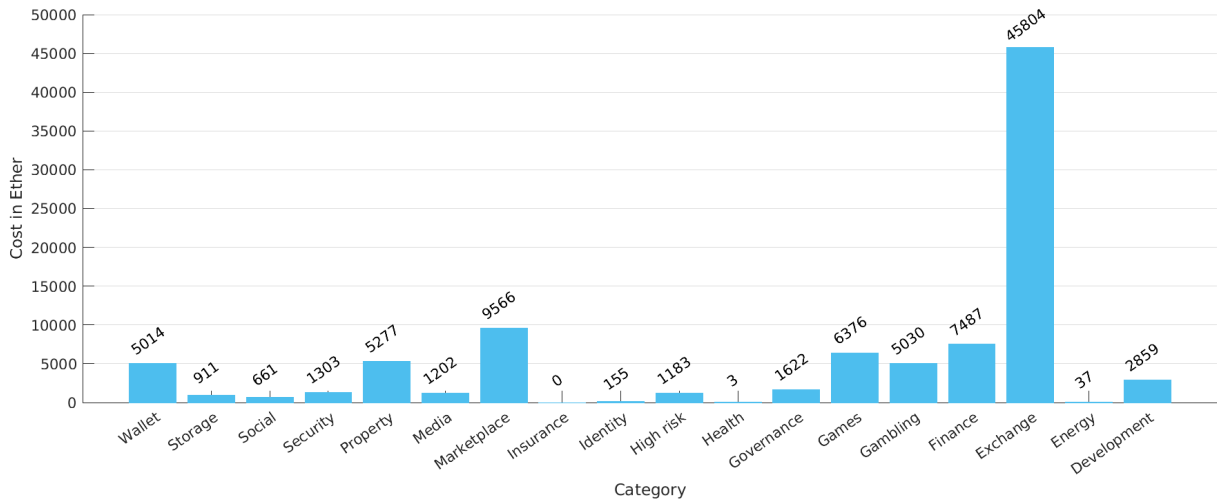


Figure 3: Total Costs by Category rounded to total Ether

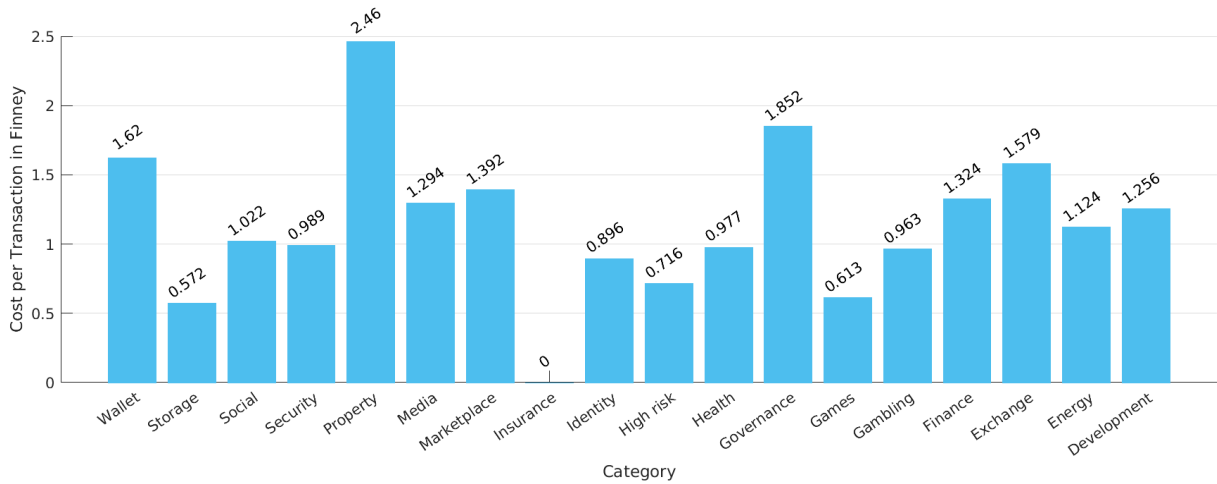


Figure 4: Average Costs per Transaction in Finney Rounded to Three Decimal Digits

7 Discussion

In this section, we will discuss the findings of the previous section. Figure 1 shows that most of the DApps on the Ethereum Blockchain are in the categories finance and games. Ethereum is the central platform for Decentralized Finance (DeFi) applications, which are classical finance products based on smart contracts. The integration of the ERC721-standard added assets on the ethereum blockchain, which enable the development of several games, like trading card games. Also the categories Exchange, for the trading of different tokens, and Gambling, for applications like lotteries, are very popular with developers. Taking a look at Figure 2 shows that the more DApps are in a category, the more transactions are executed. This is no surprise and simply shows, that the interaction in popular categories is higher, than in less popular categories. The comparison of Figure 2 and Figure 3 shows that the more transactions are executed in a category, the higher are the summed up costs of all the DApps in the category. With the huge amount of transactions in the category Exchange, it is no surprise that this category is also leading in the total costs. Even if the number of transactions plays a huge role for the summed up cost per category, we've found out that the average costs of a transaction are higher in some of the categories (cf. Figure 4). This could be due the different tasks that arise in the individual categories. In the category Property, for example, more processing power might be required due to the high cryptographic effort during encryption.

This data provides us with valuable information for answering the questions asked at the beginning. We cannot give exact information about the costs for running a DApp, but we gained some insights towards this goal. The costs seem to be dependant of the number of transactions made and the Use Case of the DApp. To get an approximate cost forecast for the DApp, one can multiply the number of transactions in a given time period by the estimated costs per transaction for that category from Figure 4. This also answers our second question: The costs seem to be dependant on the category of the DApp. However, currently only assumptions can be made about why these differences between the categories occur.

8 Conclusion and Future Work

In this paper we analyzed the operating costs of a DApp in the Ethereum Blockchain. We discussed that the general costs are dependant of the number of interactions with the DApp and the category of the DApp. With the calculated average costs of a category it is possible to calculate an approximate estimate for the cost of the app. Because of the dependency of costs on interaction, future work should take into account the differences of internal costs and external costs. This will allow operators to calculate what costs they have to cover themselves and what costs are delegated to the users. This could also help to find reasons why the differences of average costs in the categories occur. Since computational effort leads to higher gas consumption, the costs are increased. Whether the categorization of an app indicates an increased computing effort has to be investigated in future work. Since the price of one Ether can vary greatly, it might make sense to calculate it in dollars in future work. For this, the dollar exchange rate from the beginning of the Ethereum blockchain in 2015 until today is needed. In addition, each transaction must be stored with the time of execution and the used Gas.

In addition to the knowledge about costs, our work also provides other researchers with a possible approach to analyze costs of DApps. To analyze the information on the blockchain, Google Big Query is the best choice time and money wise, if no access to a full node is possible. If one does not trust the data of Google Big Query regarding completeness and correctness, there is no way around to rely on a full node copy. State of the DApps offers analysts easy access to data from various DApps already published on the Ethereum Blockchain. These can be filtered according to various criteria. The addresses of these DApps can be used for queries in Google Big Query. However, the completeness and correctness of the information provided by State of the DApps is also not guaranteed. The data originates from the respective DApp development team and does not necessarily have to be complete. But as a source for contract addresses of the respective DApp or their more detailed information, this website is well suited.

References

- [1] Andreas M. Antonopoulos and Gavin Wood. *Mastering Ethereum: Building Smart Contracts and Dapps*. O’reilly Media, 2018.
- [2] Massimo Bartoletti and Livio Pompianu. “An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns”. In: *Financial Cryptography and Data Security*. Ed. by Michael Brenner et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 494–509. ISBN: 978-3-319-70278-0. DOI: 10.1007/978-3-319-70278-0_31.
- [3] Wren Chan and Aspen Olmsted. “Ethereum Transaction Graph Analysis”. In: *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST). 2017, pp. 498–500. DOI: 10.23919/ICITST.2017.8356459.
- [4] Stefano De Angelis et al. “PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain”. In: *Italian Conference on Cyber Security* (2018).
- [5] *Ethereum API — IPFS API Gateway — ETH Nodes as a Service — Infura*. 2020. URL: <https://infura.io/> (visited on 06/10/2020).
- [6] etherscan.io. *Ethereum (ETH) Blockchain Explorer*. 2020. URL: <http://etherscan.io/> (visited on 06/10/2020).

- [7] etherscan.io. *Ethereum Full Node Sync (Default) Chart* — Etherscan. 2020. URL: <http://etherscan.io/chartsync/chaindefault> (visited on 06/10/2020).
- [8] etherscan.io. *Ethereum Transactions Information* — Etherscan. 2020. URL: <http://etherscan.io/txs> (visited on 06/10/2020).
- [9] Adem Efe Gencer et al. “Decentralization in Bitcoin and Ethereum Networks”. In: *International Conference on Financial Cryptography and Data Security*. Springer, 2018, pp. 439–457.
- [10] Arthur Gervais et al. “On the Security and Performance of Proof of Work Blockchains”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 3–16.
- [11] Google Inc. *Ethereum in BigQuery: A Public Dataset for Smart Contract Analytics*. 2020. URL: <https://cloud.google.com/blog/products/data-analytics/ethereum-bigquery-public-dataset-smart-contract-analytics/> (visited on 06/10/2020).
- [12] Felix Lau et al. “Distributed Denial of Service Attacks”. In: *Smc 2000 Conference Proceedings. 2000 Ieee International Conference on Systems, Man and Cybernetics. 'cybernetics Evolving to Systems, Humans, Organizations, and Their Complex Interactions' (Cat. No. 0. Vol. 3. IEEE, 2000, pp. 2275–2280*.
- [13] Sara Rouhani and Ralph Deters. “Performance Analysis of Ethereum Transactions in Private Blockchain”. In: *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2017, pp. 70–74.
- [14] *Running an Ethereum Node - EthHub*. 2020. URL: <https://docs.ethhub.io/using-ethereum/running-an-ethereum-node/> (visited on 06/10/2020).
- [15] *State of the DApps — A List of 3,118 Blockchain Apps for Ethereum, Hive, EOS, and More*. 2020. URL: <https://www.stateofthedapps.com/> (visited on 06/10/2020).
- [16] Huimin Tang, Yong Shi, and Peiwu Dong. “Public Blockchain Evaluation Using Entropy and TOPSIS”. In: *Expert Systems with Applications* 117 (2019), pp. 204–210.
- [17] Jordan Tigani and Siddartha Naidu. *Google BigQuery Analytics*. John Wiley & Sons, 2014.
- [18] Sergei Tikhomirov et al. “Smartcheck: Static Analysis of Ethereum Smart Contracts”. In: *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*. 2018, pp. 9–16.
- [19] Gavin Wood. “Ethereum: A Secure Decentralised Generalised Transaction Ledger”. In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

André STOLLBERGER
University of Applied Sci-
ences Würzburg-Schweinfurt
Faculty of Computer Sci-
ence and Business Informa-
tion Systems
Sanderheinrichsleitenweg 20,
97074 Würzburg
GERMANY
E-mail: *andre.stollberger*
@student.fhws.de

Tobias FERTIG
University of Applied Sci-
ences Würzburg-Schweinfurt
Faculty of Computer Sci-
ence and Business Informa-
tion Systems
Sanderheinrichsleitenweg 20,
97074 Würzburg
GERMANY
E-mail: *tobias.fertig@fhws.de*

Andreas E. SCHÜTZ
University of Applied Sci-
ences Würzburg-Schweinfurt
Faculty of Computer Sci-
ence and Business Informa-
tion Systems
Sanderheinrichsleitenweg 20,
97074 Würzburg
GERMANY
E-mail: *andreas.schuetz*
@fhws.de

Karsten HUFFSTADT
University of Applied Sci-
ences Würzburg-Schweinfurt
Faculty of Computer Sci-
ence and Business Informa-
tion Systems
Sanderheinrichsleitenweg 20,
97074 Würzburg
GERMANY
E-mail:
karsten.huffstadt@fhws.de

Nicholas H. MÜLLER
University of Applied Sci-
ences Würzburg-Schweinfurt
Faculty of Computer Sci-
ence and Business Informa-
tion Systems
Sanderheinrichsleitenweg 20,
97074 Würzburg
GERMANY
E-mail:
nicholas.mueller@fhws.de

IoT moisture monitoring system for improving indoor plants' growing conditions and optimizing maintenance routines

Minodora Suilea, Teodora Popa, Iuliana Buruiana

Abstract

Plants play a major role in human well-being. Recent studies suggest that human health is greatly impacted by the presence of plants and green spaces. Plant production and maintenance can be improved by adjusting the growing conditions in an effective way and by optimizing the resource consumption. As water supplies represent a rising concern in the current age, the paper addresses sustainability by providing a starting point of an IoT solution with low-cost sensors and an Arduino microcontroller. Soil moisture level is being collected into time series data to be further analyzed and interpreted. Relations to room temperature and humidity are discussed in order to support the improvement of plant growing conditions and maintenance routine.

1 Introduction

Plants have become very important resources for human well-being. Due to the constantly increasing population, the need for urbanization has turned green spaces and forests into buildings and cities. The modern lifestyle results in a considerable amount of time spent indoors [13] and industrialization is contributing to low air quality [1]. Besides health concerns, productivity is greatly impacted by the indoor air quality also [12]. Recent studies suggest that plants and green spaces improve air quality [9] and reduce mortality [10]. The increasing food demand, the need for urban green spaces and sustainable improvement of indoor air quality with the use of plants [5], turns the focus on the very limited water supplies and the need for sustainable solutions regarding the way we manage them [16]. The quantity of water contained by the soil has a major impact on plant development, as water stress is a major concern when it comes to growing indoor and outdoor plants [8]. Temperature, light and air humidity are the most important factors that influence water consumption. Bright light and high temperature increase water consumption, as well as low air humidity [7]. Optimal growing conditions influence the rate and efficiency of volatile organic compounds removal [9]. Monitoring irrigation represents the first step towards sustainability, healthier plants and more efficient maintenance routines. As technology advances, more complex, sustainable solutions for plant maintenance are becoming increasingly available to the general population. Since most of the existing similar monitoring systems are suitable for heavy duty industrial use, domestic users are lacking the opportunity to benefit from the latest advances in technology. The future goal of this project is to develop an easy to use, low-cost sustainable IoT solution for domestic plant growers, that will help the users acquire healthier plants and more efficient maintenance routines.

The paper is organized in four main sections. The first section addresses the challenges we are facing as a society, our motivation as a team and the insight provided by the already existing research on the topic. In the second section, a low-cost and sustainable solution is presented in order to improve

the rate and efficiency of volatile organic compounds removal and the plants' maintenance routines. The experimental materials and methods are presented in the third section, along with the obtained results. The last section provides our conclusions and the discussion regarding the results.

1.1 Related works

The already existing research concerning agriculture and plant production have proposed many suitable solutions to agricultural sustainability and productivity. Tackling the climate change, resource depletion and urbanization, Beddows and Mallon brought significant knowledge regarding power consumption efficiency, durability and compact packaging in their research paper [11], that focused on an Arduino-based long-term monitoring system suitable for harsh environments.

Abba et al. [14] developed an IoT-based irrigation monitoring and control system, in order to facilitate the automatic supply of adequate water for domestic crops from a reservoir. The authors of the paper provided insight regarding the typical challenges faced in the development process.

Addressing the most important factors that influence water consumption in greenhouse crops, other relevant authors [15] observed that in order to automate the watering schedule, additional factors such as soil type, ventilation and plant species must be taken into consideration.

The systems that were mentioned are mostly designed for greenhouses and controlled environments in plant production. When it comes to systems that are meant to be used in personal homes, the most important aspects are simplicity and accessibility. The proposed system focuses on these aspects, facilitating a fast and easy set up at a low price.

2 System Design

The most important factors influencing photosynthesis and plant growth should be monitored in order to support the decision-making process for a more suited environment for plants. Adjusting the watering schedule according to the constantly changing growing conditions can improve the water consumption and prevent water stress by over or under watering. A sustainable solution to plant maintenance routines ensures smart water management and promotes healthy growth. Optimal growing conditions increase the rate of photosynthesis, maximizing the efficiency of volatile organic compounds removal [9].

For the indoor growing conditions of plants to be monitored effectively we propose a starting point of an IoT solution with low-cost sensors and an Arduino microcontroller. For the collected data to be relevant, these parameters should be monitored on long term. In order to be able to collect and access remotely large volumes of data, an open source database server is proposed. The soil moisture sensor placed in the pot and the temperature and air humidity sensor that should be placed right next to the pot collect the most important parameters of the microclimate in which the plant is growing.

The proposed system built with readily available components, aims to collect and store data automatically on a given interval. When needed, data can be collected manually. Important considerations such as size, mobility, and ease of use were taken into account when designing the system. Another important aspect is the beginning of an IoT approach and the future migration from a physical server to its Cloud-hosted replacement.

2.1 Hardware

An Arduino Nano microcontroller was used to collect the data from the sensors and send it by serial communication to a computer. Power is supplied through the same USB cable used for data transmission. The Nano microcontroller offers exactly the same features and performance as the standard UNO board, but has a very small size in comparison.

In order to read the plant's soil humidity, an FC-28 soil moisture sensor was used. This sensor provides soil moisture reading capabilities and ease of use at a low cost. The probes of the sensor are connected to an intermediate board that converts the voltage into a value between 0 and 1023. It is connected to the Arduino board using one wire for data and two wires for power supply.

A fully calibrated SHT21 sensor is used to measure air temperature and relative humidity. The values are communicated to the Arduino board through the i2c serial protocol. The sensor uses two wires for communication and two for power supply. The SHT21 sensor is a very precise air temperature and relative humidity sensor that is very affordable and reliable. With a precision of $\pm 0.3^{\circ}\text{C}$ and $\pm 2\% \text{RH}$, a resolution of $0.04\% \text{RH}$ and 0.01°C , the SHT21 sensor's response time is less than 30 seconds. Considering the current study's particularities, the sensor provides air temperature and relative humidity detection with an adequate level of precision. Although other sensors offer better specifications, the SHT21 has proved to be the best option from a cost/precision standpoint.

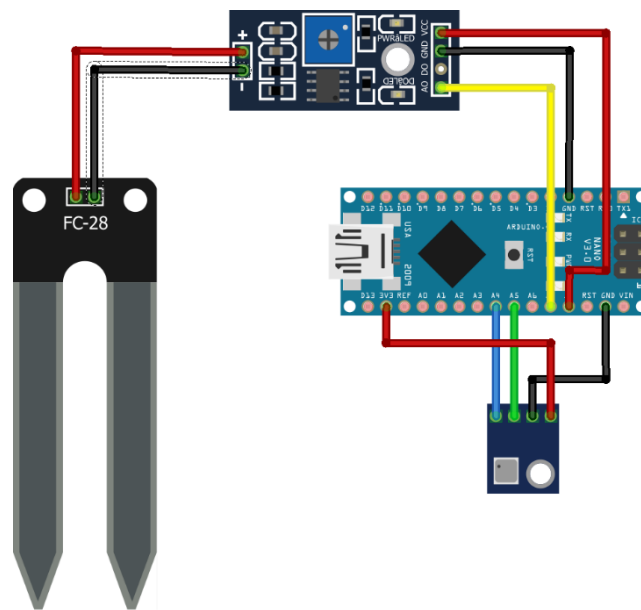


Fig. 1: Simplified schematic of the monitoring hardware

2.2 Software

The Arduino board was programmed using Arduino IDE, which provides compilation, upload of the code to the board and easy access to software libraries for a variety of sensors.

The desktop application was developed using the java programming language and Eclipse IDE. The data collected by the application was stored in a MySQL database.

The application consists of 3 main modules. The ArduinoConnection module provides functions that read the data emitted by the Arduino board, for each monitored parameter there is an implemented function.

The DBConnection module provides functions that retrieve and send data to the database.

The main module contains the logic that manages the interaction of the user with the graphical interface.

The GUI allows the user to collect data automatically or manually, and to easily manage the information stored in the database. Over a relatively short period of time, the trends can be easily observed either in the application or by exporting the data into tabular format from the database.

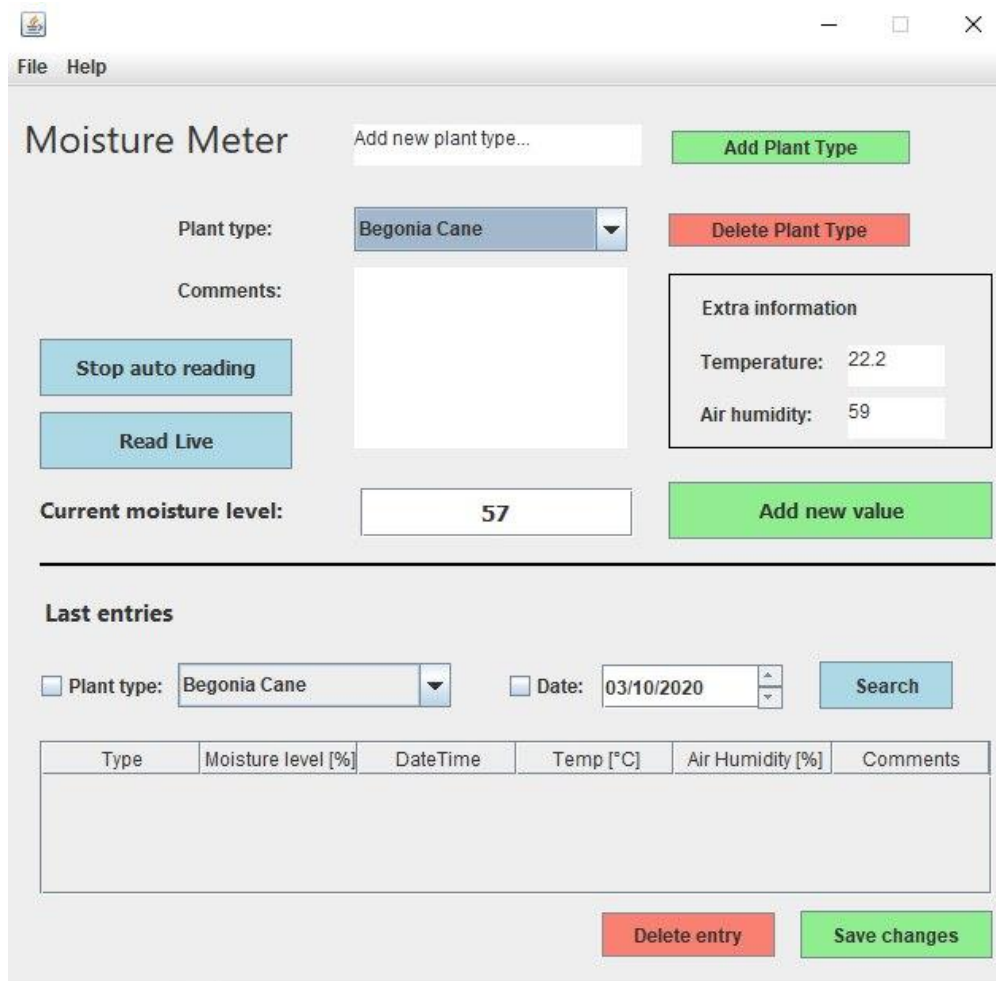


Fig. 2: Control Panel GUI

3 Experimental materials and methods

The current study was performed on two different species of indoor growing plants: we applied our system on *Spathiphyllum Wallisii* and *Ficus Pumila*. After that, represented the data visually and interpreted the results of this experiment. The plants used for this study were in good condition, well-established in their pots. *Spathiphyllum Wallisii* was placed in a north-west facing room, close to the window, in a 12 cm pot, as well as *Ficus Pumila* that was placed on the windowsill in a 9 cm pot. The study was conducted in early autumn, on rainy and sunny weather.

In order to collect the time series data, each soil moisture sensor was placed in a pot, and the SHT21, GY-21 sensor was placed right next to the pot, in order to collect as precise data as possible. In the first part of the experiment, data samples were collected automatically each hour, from *Spathiphyllum Wallisii* for five days in a row. In the second part of the experiment, data samples were collected from *Ficus Pumila*, each minute for three consecutive days.

Spathiphyllum and *Ficus* are tropical plants that have been proved to be very efficient air-purifying indoor plants, according to a NASA experiment [2] and other more recent studies [4]. Due to these species' popularity, they are readily available in most flower shops at an affordable price. Both of these plants are easy to grow indoors and need a constantly moist soil, so the watering schedule has to be precise enough in order to avoid over or under watering. Due to the fact that

Spathiphyllum is a resilient plant that can recover from under watering [3], it has higher chances of surviving a defective maintenance routine. Ficus Pumila is a more vulnerable plant that does not tolerate dry soil. In this case, monitoring the moisture level is crucial. Because the soil needs to be moist at all times, overwatering can easily occur, in which case the leaves will fall off [6]. Given the fact that over and under watering are the most common reasons for short-living indoor plants, this study can offer insight over any other plants that fall in these categories.

3.1 Results

Two datasets were collected using the implemented system: one for Spathiphyllum and one for Ficus. These datasets consist of values for variables that represent the moisture level of the soil, the temperature in the room and air humidity in every moment of data collection. The quality of the data obtained proved to be of a reasonable quality, considering the affordability of the hardware components. The collected datasets were exported into tabular format from the database.

plant_type	moisture_level	temperature	air_humidity	date_time
Spathiphyllum Wallisii	45	19.4	62	30-09-20 19:41
Spathiphyllum Wallisii	45	19.4	63	30-09-20 21:01
Spathiphyllum Wallisii	43	19.1	63	30-09-20 22:01
Spathiphyllum Wallisii	43	19	63	30-09-20 23:01
Spathiphyllum Wallisii	42	18.9	63	01-10-20 0:01
Spathiphyllum Wallisii	42	18.9	63	01-10-20 1:01
Spathiphyllum Wallisii	41	18.9	63	01-10-20 2:01
Spathiphyllum Wallisii	41	18.9	63	01-10-20 3:01
Spathiphyllum Wallisii	40	18.8	63	01-10-20 4:01
Spathiphyllum Wallisii	40	18.7	63	01-10-20 5:01
Spathiphyllum Wallisii	40	18.7	63	01-10-20 6:01
Spathiphyllum Wallisii	40	18.6	63	01-10-20 7:01
Spathiphyllum Wallisii	39	18.7	63	01-10-20 8:01
Spathiphyllum Wallisii	39	18.8	62	01-10-20 9:01

Table 1- Sample of the collected dataset

Table 1 shows a sample of the collected dataset, containing the three parameters monitored, the name of the plant and the timestamp. Then, based on the collected data, graphs were generated in order to represent and highlight the evolution of the three parameters in time and a possible connection between them. The evolutions of moisture level, temperature and air humidity are represented in Figure 3 (for Spathiphyllum Wallisii) and in Figure 4 (for Ficus Pumila). The x – axis shows the timestamps, the y – axis represents the three parameters as percents.

The moment of watering Spathiphyllum is September 30th 2020 at 01:41 a.m. In this moment, the soil moisture has a value of 45%. On September 30th at 10:01 p.m., the level of moisture in the soil starts decreasing from a value higher than 40% to a value lower than 20%. So, after about 20 hours from the timestamp of watering Spathiphyllum, the level of moisture starts decreasing. The level of moisture in the moment of watering Spathiphyllum is higher than in the levels of moisture on September 29th, at 10:41 a.m. (about 15 hours before), when it has values between 35% and 41%. Air humidity starts increasing in the moment the moment of watering the plant, on September 30th at 01:41 a.m, reaching values of 63%. Air humidity starts decreasing on September 30th at 08:41 a.m. from higher than 60% to lower than 60%. The value of air humidity is always increasing and decreasing between 44% and 64%. The temperature in the room has values between 18.6% and 29% and during the five days.

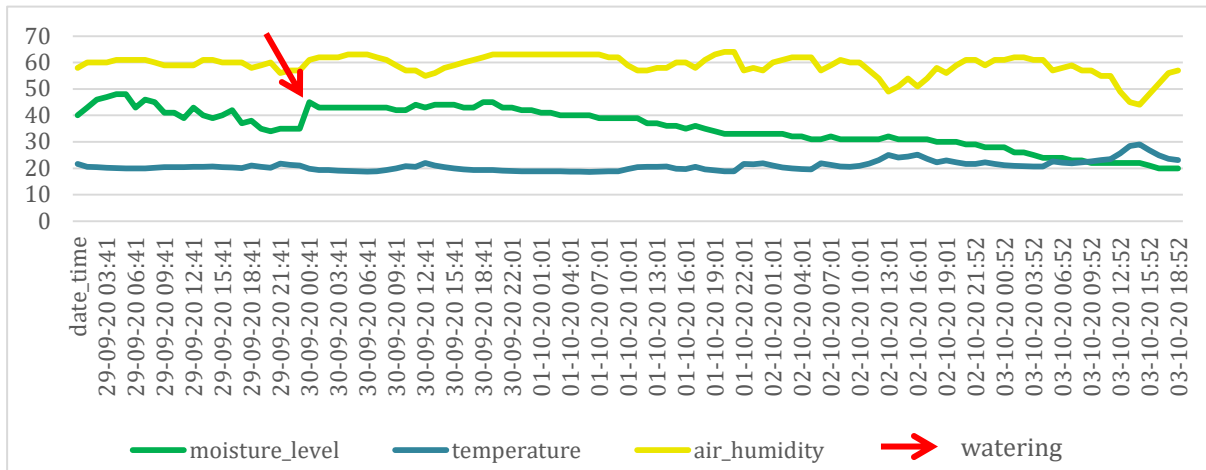


Fig. 3: The evolution of moisture level, temperature and air humidity for *Spathiphyllum Wallisii*

The moment of watering *Ficus* is October 3rd 2020 at 08:19 p.m. In this moment, the soil moisture has a value of 64%. On October 3rd at 09:04 p.m., the level of moisture in the soil starts decreasing from a value of about 60% to a value lower than 60%. So, after about 23 hours from the timestamp of watering *Ficus*, the level of moisture starts decreasing. Air humidity has values between 59% and 60% between October 3rd 2020 at 08:19 p.m. and October 3rd at 09:04 p.m. The value of air humidity is always increasing and decreasing between 45% and 63%. The temperature in the room has values between 21.6% and 28.8% and during the three days.

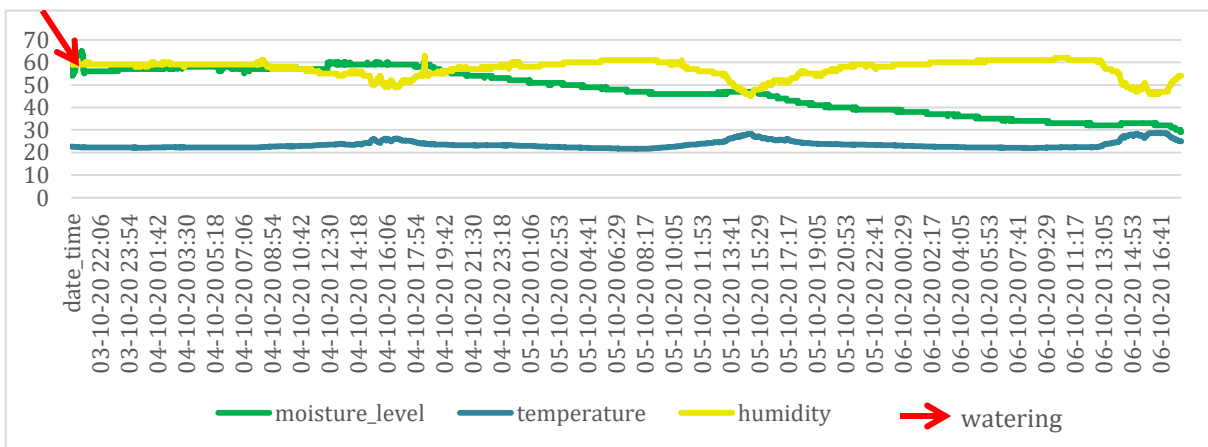


Fig. 4: The evolution of moisture level, temperature and air humidity for *Ficus Pumila*

4 Conclusions

Monitoring the important parameters that influence plant growth and development brings valuable insight for growing strong, long-lasting plants. Optimizing the maintenance routine also plays an important role in the plant growing activity. With the provided user-friendly moisture monitoring system, domestic plant growers are able to benefit from sustainable solutions that industrial plant growers are using successfully ever since the technological advancements allowed.

Compared to the other existing works, the proposed system focuses on readily available low-cost components and an easy to use software solution. The implemented system proved to be reliable and successful in monitoring and storing all the data collected from the sensors.

The time series data obtained can be further analyzed in order to obtain meaningful statistics regarding the rate of water consumption in houseplants. A future goal of our system is prediction of important events in the life of a plant, based on data analysis techniques rather than empirical approaches. A mathematical model can help explain the data in order to facilitate prediction and thus improve the maintenance routine.

The current design can be further improved by replacing the Arduino nano board with a wireless connection board. In order to increase mobility, the wired power supply can be replaced by a battery. Light intensity should also be added to the list of monitored parameters, as it influences the rate of photosynthesis.

4.1 Discussion

As this study is still in an early stage, part of an ongoing development project we are researching, the current representation does not act as conclusive evidence over the matter. For more reliable associations to be established, the data collection process should extend to a minimum period of six months, up to one year. Our long-term goal is to be able to use data analysis techniques and machine learning in order to gain a better understanding of the biological phenomena. Such data will provide support in the decision-making process regarding the plants' maintenance routines.

Acknowledgement: The results presented in this paper are part of an ongoing study that constitutes the foundation of several larger projects of the authors and was supervised by Ralf Fabian, from Lucian Blaga University of Sibiu.

References

- [1] Arnold D. Bergstra, Bert Brunekreef and Alex Burdorf. *The effect of industry-related air pollution on lung function and respiratory symptoms in school children*. Environmental Health, vol. 17, 30, 2018.
- [2] B. C. Wolverton, Willard L. Douglas and Keith Bounds. *A Study of Interior Landscape Plants for Indoor Air Pollution Abatement*. National Aeronautics and Space Administration, 1989
- [3] Bodie V. Pennisi. *Growing Indoor Plants with Success*. University of Georgia Cooperative Extension, 2020
- [4] C. Gubb, T. Blanusa, A. Griffiths and C. Pfrang. *Can houseplants improve indoor air quality by removing CO₂ and increasing relative humidity?*. Air Qual Atmos Health, vol. 11, 2018.
- [5] Federico Brillì, Silvano Fares, Andrea Ghirardo, Pieter deVisser, Vicent Calatayud, Amalia Muñoz, Isabella Annesi-Maesano, Federico Sebastiani, Alessandro Alivernini, Vincenzo Varriale, and Flavio Menghini. *Plants for Sustainable Improvement of Indoor Air Quality*. Trends in Plant Science, vol. 23, 2018.
- [6] Fran Bailey, Zia Allaway and Christopher Young. *Royal Horticultural Society Practical House Plant Book*. Dorling Kindersley Limited, 2018
- [7] Geraldo Chavarria, Henrique Pessoa dos Santos. *Plant Water Relations: Absorption, Transport and Control Mechanisms*. IntechOpen, 2012.
- [8] Kelley Drechslera, Isaya Kisekkaa, Shrinivasa Upadhyayaa. *A comprehensive stress indicator for evaluating plant water status in almond trees*. Agricultural Water Management, vol. 216, 2019.
- [9] Kwang Jin Kim, Md. Khalekuzzaman, Jung Nam Suh, Hyeon Ju Kim, Charlotte Shagol, Ho-Hyun Kim and Hyung Joo Kim. *Phytoremediation of volatile organic compounds by indoor plants: a review*. Horticulture, Environment, and Biotechnology, vol. 59, 2018.
- [10] Mireia Gascon, Margarita Triguero-Mas, David Martínez, Payam Dadvand, David Rojas-Rueda, Antoni Plasència, Mark J. Nieuwenhuijsen. *Residential green spaces and mortality: A systematic review*. Environment International, vol. 86, 2016.

- [11] Patricia A. Beddows and Edward K. Mallon. *Cave Pearl Data Logger: A Flexible Arduino-Based Logging Platform for Long-Term Monitoring in Harsh Environments*. Sensors, vol 18, 2018.
- [12] Pawel Wargocki. *Productivity and Health Effects of High Indoor Air Quality*. Encyclopedia of Environmental Health, 2011.
- [13] Sabine Brasche, Wolfgang Bischof. *Daily time spent indoors in German homes – Baseline data for the assessment of indoor exposure of German occupants*. Int. J. Hyg. Environ.-Health vol. 208, 2005.
- [14] Sani Abba, Jonah Wadumi Namkusong, Jeong-A Lee, and Maria Liz Crespo. *Design and Performance Evaluation of a Low-Cost Autonomous Sensor Interface for a Smart IoT-Based Irrigation Monitoring and Control System*. Sensors, vol 19, 2019.
- [15] Scott Henderson, David Gholami and Youbin Zheng. *Soil Moisture Sensor-based Systems are Suitable for Monitoring and Controlling Irrigation of Greenhouse Crops*. HortScience, vol. 53, 2018.
- [16] The United Nations World Water Development Report 2015. *Water for a sustainable world*. UNESCO, Paris, 2015.

Minodora SUILEA
Lucian Blaga University of Sibiu
Department of Mathematics and
Informatics
Informatics
Doctor Ion Ratiu, 5-7, Sibiu
ROMANIA
E-mail: minodora.suilea@ulbsibiu.ro

Teodora POPA
Lucian Blaga University of Sibiu
Department of Mathematics and
Informatics
Informatics
Doctor Ion Ratiu, 5-7, Sibiu
ROMANIA
E-mail: teodora1.popa@ulbsibiu.ro

Iuliana BURUIANA
Lucian Blaga University of Sibiu
Department of Mathematics and
Informatics
Informatics
Doctor Ion Ratiu, 5-7, Sibiu
ROMANIA
E-mail: iuliana.buruiana@ulbsibiu.ro

Stellar Pointer

Eduard-Traian Stefanescu

Abstract

This paper intends to present a robotic arm designed to point celestial bodies in both observable and unobservable fields using celestial coordinates. Since this robotic arm is created for educational purposes, this paper aims to present its components and the related research so that it will be easier by teachers to use it and teach their students where the celestial bodies are positioned. The position of celestial bodies is based on the celestial coordinates reference system and on the geocentric elliptical coordinates provided by the Institute of Celestial Mechanics and Ephemeris Computation through a Web Application Program Interface, that accepts certain parameters so that the returned data to be valid and relative to the observer. This robotic arm system due to the used materials and the open-source codebase is reducing the cost of buying a complete stellar pointer, but it requires the knowledge of setting it up to be fully-functional and to provide accurate parameters, such as the latitude, longitude and altitude of the observer for the Web Application so that to point in the right direction. The paper will also illustrate the robot components and scalability characteristics.

1 Introduction

The human nature curiosity all the time wanted and want to discover the mysterious universe. As in the seventeenth century Johannes Kepler, Galileo Galilei and Isaac Newton developed a modern understanding of physics that emphasized that the Earth is not stationary and moves around the Sun so that every object or body in the universe is governed by the same physics that govern Earth.

In the same century, the first telescopic observations of the Solar System were conducted by telescope, when Galileo Galilei discovered the physical details about the individual bodies of the Solar System, that the Earth's Moon was cratered, that the sun was marked with sunspots and the Jupiter had four satellites in orbit around it.

After discovering the celestial bodies of the Solar System and their moons using telescopes from Earth, scientists started the spacecraft observations also known as the Space Age explorations. They performed robotic spacecraft missions to explore the characteristics of the planets in depth by being closer to them so that until now all planets of the Solar System are visited from varying degrees by spacecraft launched from Earth. The spacecraft consists into various telescopes that gravitates Earth or other planets, satellites that are passing nearby planets as was Luna 1 that successfully past the Moon in 1959, car-sized rover designed to explore the Gale crater on Mars [1] and even crewed exploration who stepped on the Earth's Moon on July 21, 1969, during the Apollo 11 mission [2].

As a result of the curiosity into exploring and finding more objects of the universe, pupils of all ages are becoming more interested into discovering where planets or other celestial bodies like

dwarf planets or comets are positioned on the sky. So that people start buying a telescope to observe these celestial bodies, but there are hard to find just by looking up at the sky or by searching them randomly.

Given the importance of this subject, this paper is based into creating a robotic arm structure that uses already defined mathematical formulas and astrophysics rules that are easy to set up to be used by people of all ages, but especially for educational purposes to find planets of our Solar System without difficulty. For the fact, the universe is an infinite grace it must be easily observed by anyone knowing very few astrophysics concepts.

Currently, there are two types of telescopes, the optical telescope that collects and focus light to create a magnified image for direct view or to make a photograph [3] and the robotic telescope that is initiated by a human to make observations but after that, it doesn't need intervention to follow comet orbits or supernova light curves [4]. These two types of telescopes are built in many forms, from the astronomical telescopes that have a substantial dimension to the smaller ones that are used in schools or for personal purpose. Few of the telescopes nowadays have integrated robotic telescopes that points the telescope to the requested celestial body.

Although this robotic arm purposes a different paradigm in this area, by being scalable and easy to use. Scalable by the fact that in the current state can be used by all sorts of users but investing in the built parts it can be used by astrophysicists in their research studies. Firstly, its easiness of using it is that it only needs to be positioned the North and plugged into an outlet, and secondly by using a responsive application with modern design components.

The paper begins to present a short overview of similar studies in the Related work section. After that, the reference coordinates system of the angular and celestial coordinates will be described in The coordinates in a reference system section. In the fourth section, after describing what is the coordinate reference system and which type of coordinates will be used, in section Locating celestial bodies using the IMCCE API, a description will be given about how these coordinates will be requested from an API. And in the last section Integrating Arduino and embedded components, the previously mentioned sections are combined so that the celestial coordinates will be translated into numerical ones for the robotic arm to point the requested celestial body through the User Interface.

2 Related work

The Stellar Pointer paper was treated differently in the “A new star tracker concept for satellite attitude determination based on a multi-purpose panoramic camera” paper which introduced an advanced algorithm developed for attitude determination of a space platform. The algorithm takes advantage of photographs taken from a panoramic multi-purpose camera fitted with a hyper-hemispheric lens and used as a star tracker. The designed sensor in this paper is an original creation since state-of-the-art star trackers correctly visualize as many stars as possible within a narrow to medium size field-of-view, whereas an incredibly significant portion of the celestial sphere is viewed by the considered sensor, but its detection capacities are constrained by the optical device features. The proposed solution in this paper incorporates algorithmic principles, such as template matching and point cloud registration, inherited from the fields of computer vision and robotic science [5].

Also the “Optimal estimation for the satellite attitude using star tracker measurements” paper researched an optimum estimate scheme which uses the gyro readings and star tracker measurements of a widely used satellite attitude measuring unit to assess the satellite attitude. The method is primarily based on the exponential Fourier densities that are trained to have the desired

closure property and the conditional probability density, which is an exponential Fourier density, is recursively defined by updating a finite and fixed set of parameters. Besides the estimation for the satellite attitude, the paper also presents a suitable solution for systems that have high randomness of poor observability levels and systems for which precision is of utmost importance [6].

Further in the “Azimuth method for ship position in celestial navigation” suggests an innovative solution that uses the azimuth of the celestial body as the celestial bodies tend to determine the broad circle equation corresponding to the observable body to the location of the ship. Furthermore, as with the previous two related work, this approach does not include the horizon and sextant instruments. The main value that separates this strategy from prior ones is its ability to assess the location of the ship during the night when the horizon is invisible. By examining the relationship between the ship location and the great-circle azimuth of the observable body, the vector calculus is applied to find the mathematical equation for the ship location. And based on the ship location equation system that is generalized into a regular system in which the great-circle azimuth and the coordinates of the observed body are the input for the suggested mathematical system [7].

3 The coordinates in a reference system

To locate a geographical point on a three-dimensional surface as the Earth, a two-dimension surface is needed to represent the coordinate reference system which defines a specific map projection, as well as transformations between different spatial reference systems [8]. So this reference system is also called the spatial reference system because is represented as a set of coordinates (x, y, z) or more often used for practical application as latitude, longitude and altitude [9].

So that the map projections are used to portray the surface of the earth, or a portion of the earth, on a flat piece of paper or computer screen. In a simplified form, map projections try to transform the earth from its spherical shape (3D) to a planar shape (2D) [10].

3.1 Angular coordinates

The earth has an irregular spheroidal shape. The natural coordinate reference system for geographic data, as was mentioned before is longitude and latitude. This is an angular system. The latitude of a point is the angle between the equatorial plane and the line that passes through a point and the centre of the Earth. The longitude is the angle from a reference meridian (lines of constant longitude) to a meridian that passes through the point.

These angles cannot be measured. But these can be estimated. To do so, a model of the shape of the earth is needed. Such a model is called a datum. The simplest datums are a spheroid (a sphere that is flattened at the poles and bulges at the equator). More complex datums allow for more variation in the earth’s shape. The most used datum is called WGS84 (World Geodesic System 1984). This is remarkably like NAD83 (The North American Datum of 1983). Other, local datums exist to record locations more precisely for a single country or region.

The order of the coordinates is significant, and they are sometimes identified by their position in an ordered tuple and sometimes by a letter, as in "the x-coordinate". The coordinates are taken to be real numbers in elementary mathematics but may be complex numbers or elements of a more abstract system. The use of a coordinate system allows problems in geometry to be translated into problems about numbers and vice versa [11].

So, the simple way to record a location is a coordinate pair in degrees and a reference datum. In Fig. 1, ϕ represents the latitude and λ represents the longitude [12].

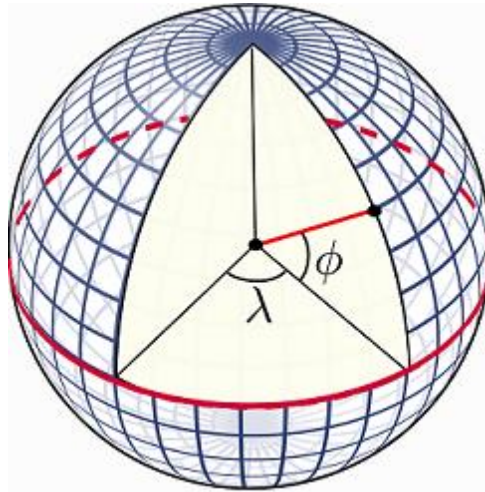


Fig. 1: The latitude and longitude represented on a spheroidal shape

3.2 Celestial coordinate system

Definition 1 *Celestial coordinates are a reference system used to define the positions of objects on the celestial sphere.* There are two main coordinate systems in use.

The first one is the equatorial coordinate system is by far the most common for astronomical observations and is an extension of the latitude and longitude coordinate system used on Earth. By defining a right ascension, a declination and an epoch, every astronomical object is identified with a unique position on the sky.

And the second one is the horizontal coordinate system uses the observer's horizon as a reference and measures an object's altitude (height above the horizon) and azimuth (angular distance from due north measured eastwards). Unlike the equatorial coordinate system, the position of each object depends on the location of the observer and the time of the observation. This coordinate system [13].

There is a significant difference between the equatorial and geographic coordinate systems: the geographic system is fixed to the Earth; it rotates as the Earth does. The Equatorial system is fixed to the stars, so it appears to rotate across the sky with the stars, but of course, it is the Earth rotating under the fixed sky.

The latitudinal angle of the Equatorial system is called Declination and measures the angle of an object above or below the Celestial Equator. The longitudinal angle is called the Right Ascension and measures the angle of an object East of the Vernal Equinox. Unlike longitude, Right Ascension is usually measured in hours instead of degrees, because the apparent rotation of the Equatorial coordinate system is closely related to Sidereal Time and Hour Angle. Since a full rotation of the sky takes 24 hours to complete, there are $(360 \text{ degrees} / 24 \text{ hours}) = 15^\circ$ in one Hour of Right Ascension. In the following paper, this formula will be also used to convert the horizontal coordinates to the ecliptic coordinates.

The equatorial coordinates for deep-sky objects and stars do not vary appreciably over short durations of time since they are not affected by the diurnal motion (the daily apparent rotation of the sky around the earth. However, note that this takes 1 sidereal day, as against 1 solar day).

They are suitable coordinates for making catalogues of stars and deep-sky objects. However, some effects cause the Right and Declination of objects to vary over time, namely Precession and nutation, and proper motion, the latter being even less important. Equatorial coordinates are thus generally specified with an appropriate epoch, to account for precession. The Popular epoch used in this paper was J2000.0 or the Julian Year 2000 [14].

3.2.1 Horizontal coordinate system

The horizontal coordinate system, also known as the topocentric coordinate system, is a celestial coordinate system that uses the observer's local horizon as the fundamental plane. Coordinates of an object in the sky are expressed in terms of altitude (or elevation) angle and azimuth. This coordinate system is dependent on the observer's latitude and longitude.

This celestial coordinate system divides the sky into two hemispheres: the upper hemisphere, where objects above the horizon are visible, and the lower hemisphere, where objects below the horizon cannot be seen since the Earth obstructs views of them. The great circle separating the hemispheres is called the celestial horizon, which is defined as the great circle on the celestial sphere whose plane is normal to the local gravity vector. In practice, the horizon can be defined as the plane tangent to a still liquid surface, such as a pool of mercury. The pole of the upper hemisphere is called the zenith. The pole of the lower hemisphere is called the nadir.

Using the observer's local horizon as a reference plane, the position of an object on the celestial sphere at a particular time is given by [15] the altitude and the azimuth.

As mentioned early, the altitude, sometimes referred to as elevation, is the angle between the object and the observer's local horizon. For visible objects, it is an angle between 0° and 90° . Alternatively, zenith distance may be used instead of altitude. The zenith distance is the complement of altitude so that the sum of the altitude and the zenith distance is 90° .

Azimuth is the angle of the object around the horizon, usually measured from true north and increasing eastward. The previously two mentioned coordinates can be seen in Fig. 2.

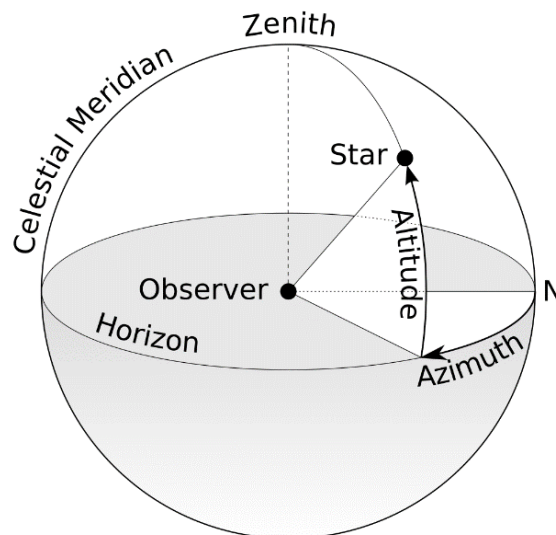


Fig. 2: The altitude and azimuth exemplified on a celestial sphere

Considering the previously mentioned about the altitude and azimuth, a northern observer can determine whether altitude is increasing or decreasing by instead considering the azimuth of the

celestial object. If the azimuth is between 0° and 180° then the object is rising and if the azimuth is between 180° and 360° the object is setting [16].

3.2.2 Equatorial coordinate system

The Equatorial Coordinate System is the favoured way astronomers use to monitor the positions of objects in the sky. Space experts imagine that the Earth is encircled by a huge circle called the celestial sphere. And the Earth's equator and the plane of the Earth's orbit are extended onto this circle.

When the plane of the earth's orbit is projected onto the imaginary celestial sphere, it is called the ecliptic. The rotation axis of the earth forms an angle of 23.5° with the earth's orbital plane, so the celestial equator and the ecliptic also form an angle of 23.5° , as shown in Fig 3 [17].

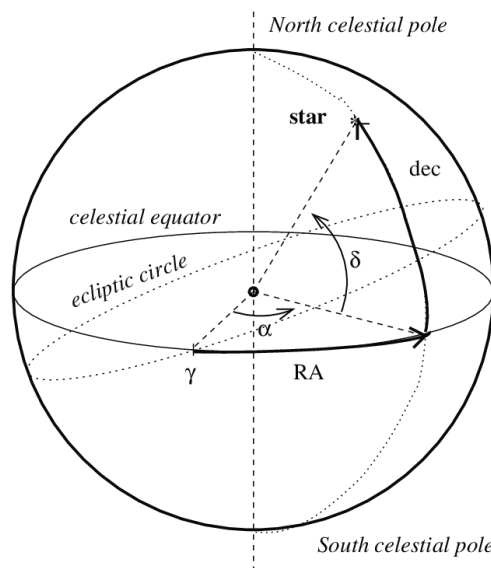


Fig. 3: The equatorial coordinate system

3.2.2.1 What is declination?

The declination symbol δ (abbreviated as DEC) measures the angular distance of an object perpendicular to the celestial equator, which is north, south, and negative. For example, the magnetic declination of the north celestial pole is $+90^\circ$. The origin of declination is the celestial equator, which is the projection of the earth's equator onto the celestial sphere. Declination is like the latitude of the earth.

3.2.2.2 What is right declination?

The right ascension symbol α , (abbreviated as RA) measures the angular distance of an object eastward alongside the celestial equator from the vernal equinox to the hour circle passing via the object. The vernal equinox factor is one of the two where the ecliptic intersects the celestial equator. Analogous to terrestrial longitude, the right ascension is usually measured in sidereal hours, mins and seconds rather than degrees, a result of the technique of measuring the right ascension by way of timing the passage of items across the meridian as the Earth rotates. There are $360^\circ/24\text{h} = 15^\circ$ in one hour of right ascension, and 24h of right ascension round the complete

celestial equator. When used together, right ascension and declination are usually abbreviated RA/Dec [18].

4 Locating celestial bodies using the IMCCE API

This paper uses the IMCCE API to locate celestial bodies, in this way the calculation formulas are used indirectly obtaining the data from this API. The other way was to use those calculation formulas directly but was in the scope of this paper because as was previously mentioned, this paper wants to offer a simple way of locating celestial bodies.

By keeping with this goal, the Institut de mécanique céleste et de calcul des éphémérides (IMCCE) is the scientific research institute of the Paris Observatory and the research unit of CNRS which will provide the needed data to locate a celestial body. Also, the IMCCE is accountable to create annual reports, calendars, tables, and diverse documents, for the Bureau des longitudes which has the mission to distribute those the French public. The scientific work carried out by IMCCE mainly covers the dynamics and planetary studies of the sun, extrasolar bodies, and the earth's environment. The core business of IMCCE is mathematics, celestial mechanics, and astronomical observation of planets, exoplanets, natural satellites, asteroids, comets, and meteoroids.

Additionally, the goal of the Solar System Portal is to apply its knowledge and expertise to planetary system dynamics and physics through databases, ephemeris calculation services, simulation tools and numerical calculations for use by the astronomy community and the public. The computing service is fully in line with the interoperability concept of the Virtual Observatory (VO).

4.1 The positional Ephemerides API

The positional ephemerides, also known as the ephemcc method is suitable for this paper to calculate the ephemeris of solar system objects. This method offers a tremendous number of parameters to obtain the needed data.

By default, the planetary ephemeris is calculated using IMCCE's INPOP 4-D planetary theory. The ephemeris of natural satellites is calculated based on various special planets. The ephemeris of asteroids and comets are calculated by numerical integration of the disturbance problem of n objects and the dynamics of asteroids come from the ASTORB database of the Lowell Observatory or the MPCORB database of the Minor Planet Center. The dynamics of comets are taken from the COMETPRO database of IMCCE.

Another reason that this paper is using this API is that it provides valid and accurate data so that the dynamics of asteroids and comets are updated weekly. Once a new solution is publicly provided, the dynamics of the planet and its natural satellites will be updated [19].

4.1.1 Usage of the Ephemerides API

To the use, this API certain parameters are required to be sent to get the needed data. The API as will be explained in the next subsection returns more data, but only the Declination and Right Ascension will be used to point the celestial body with the robotic arm. This API can be used in many ways, such as using the query form available on the Solar System portal of IMCCE, by implementing an own Miriade Web service ephemcc method in own software or by a non-

interactive file transfer program such as `wget` or `curl` to call HTTP requests on the command line interface.

But for this paper, an ASP.NET Core application was created to request data from the IMCCE API and send them to the robot. Later in this paper, the entire communication system will be explained. For ASP.NET, the Web API was a simple way to implement RESTful Web services using the .NET framework. RESTful web services are those services that use HTTP as the basic method of communication and by ASP.NET Web API this is defined as a framework that enables the development of HTTP services to be extended to client entities such as browsers, as will be explained in the next subsections [20].

To start using this API, the base URL is needed, in this case, the URL will be <https://ssp.imcce.fr/webservices/miriade/api/ephemcc.php>, and will be succeeded by the following parameters.

The first parameters refer to the name or the designation of the searched object and it consists of two parts, the first one is the prefix and can have one of the following values `a` to point an asteroid, `c` to point a comet, `s` to point a natural satellite, but in this paper, only the `p` value was used to point a planet and the `dp` value to point the dwarf planet, Pluto.

The restrictions are related to the names of asteroids and comets must be the official names adopted by International Astronomical Union or IAU. And for the natural satellites, only the ones for which an ephemeris is available are recognized. And the second part of the first parameter describes the name of the official number or name, or temporary name from the Solar System Open Database Network or abbreviated as `Sso`.

The second is the epoch, that it is used to get the coordinates for a certain date and time, but for this paper, the current time as Julian date will always be used. This parameter also accepts the date to be sent as a textual English date in accordance with the GNU syntax of dates or as an ISO 8601 date. The main reason of which the Julian date was used is based on the restriction of this parameter which relates to the seconds that must be an integer number in the ISO format so for a time resolution better than a second the Julian period is recommended to be used.

The third is represented by the location of the observer, by using latitude, longitude, and altitude, because the current location will be used. Otherwise, the IAU code of the observatory can be used. The restrictions apply to the latitude and longitude that must be expressed in decimal degrees, and altitude must be expressed in meters above mean sea level. Longitude is negative for the west. So that the sign `+` for longitude and latitude can be omitted.

The fourth refers to the type of coordinates, as was mentioned in the second section, the spherical coordinates are needed to find the location of a celestial body using a stellar pointer, a telescope or just be simple looking at the sky. So, the parameter value, in this case, will be `1` which refers to the spherical coordinates.

The fifth is the type of ephemeris representing the relativity of the coordinates. For this, the parameter value will be `2` representing that these coordinates will be relative to the epoch date. And the last parameter refers to the type in which the data will be returned, and here the type is JSON because it is easier to parse and some libraries easily deserialize such a type into an object.

4.1.2 The API response structure

Since the mime type of the response was set to JSON, which refers to the last parameter, the ephemerides response is structured into multiple values, in the following manner. The `sso` value provides information about the requested celestial body, it is followed by `coosys` that provides information about the celestial reference frame, then by the `ephemeris` value specifying the

information about the ephemeris computation, afterwards by the `data` containing an array with the ephemerides of the body and the last value is `unit` which designates the quantity units.

For this paper, only the `ra` and `dec` values from the `data` parent value are needed to indicate the position of a celestial object. These are provided in hours, minutes, and seconds and in degrees, hours, and minutes, respectively but afterwards this will be converted into degree values [21] [22].

4.2 Program presentation

The main role of this application is to facilitate user interactions with the robotic arm. By using the Angular framework and a responsive design, a web application was created, displaying the planets of our Solar System, the Earth's Moon, and the dwarf planet Pluto as can be seen in Fig 4. To create the responsive design, Angular Material components were used, such as card containers, buttons and their guidelines for alignments and colours. This application runs on the local network and sends data to an API created in .NET Core, which is also on the local network [23].



Fig. 4: The web application running on the local network

By pressing any button seen on the Fig 4., it will send a request to the API, then the API will request the celestial body position from the IMCCE API, afterwards, the data was received, the local API will convert the data and will send the coordinates to the robot, that in turns will send the received values to the stepper motor and servo motors as is described in Fig 5.

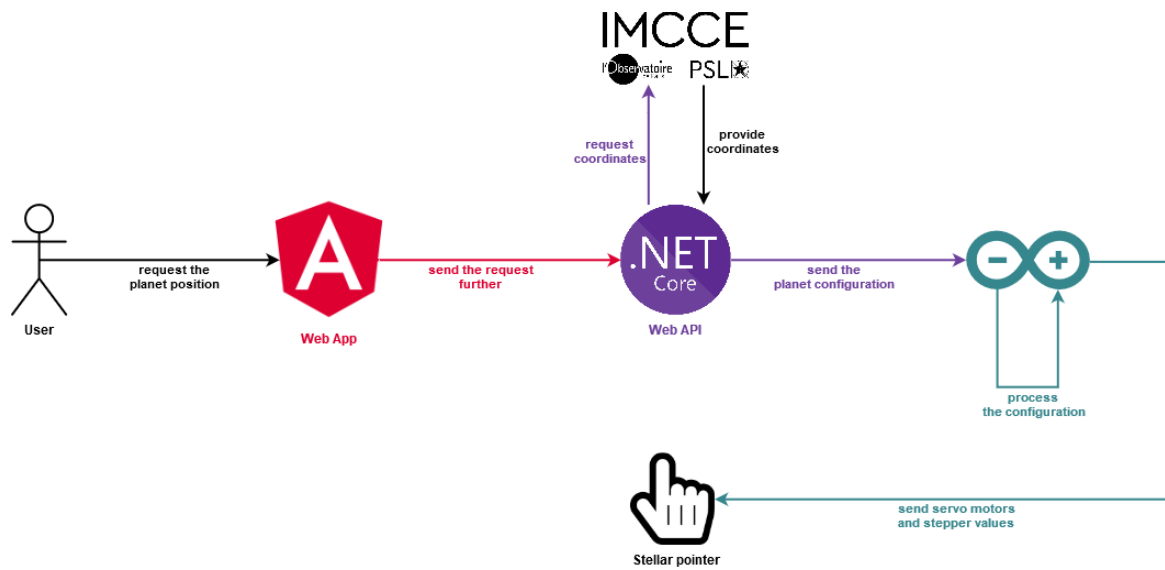


Fig. 5: System communication diagram

4.3 Converting the Right Ascension and Declination into degrees

Both the longitude and the Right Ascension begin at the Greenwich Meridian, which simplifies the conversion from one coordinate system to another. The meridian is an imaginary line along which the coordinates have a constant value and extend from north to south. The Right Ascension meridian falls on the celestial body, and the longitude meridian falls on the earth. The Right Ascension is measured to the east, in hours, minutes and seconds, and the value range is 0 to 24 hours. Longitude extends east and west, measured in degrees. Greenwich's zero value is -180 degrees for the west and +180 degrees for east [24].

To convert the right ascension into decimal the following formula was used:

$$\alpha = (RAh * 15 + RAm/4 + RAs/240) \times \pi/180 \quad (1)$$

where α represents the Right Ascension, degrees expressed in a decimal form, so it is called the azimuth value. The RAh , RAm , RAs are the Right Ascension values expressed in hours, minutes and seconds and is multiplied by $\pi/180$ because the returned values must indicate if the value is towards east or west. RAh is multiplied by 15 because it represents the conversion factor that comes from the rotation speed of the Earth which is equivalent to the duration of the Earth's rotation in one hour, stated in *degrees/hours*. The same rule applies also to RAm and RAs that is divided by 4 and 240, so that it represents the Earth's rotation in *degrees/minutes* ($60/15 = 4$) and *degrees/seconds* ($3600/15 = 240$) [25].

Because the Declination value is expressed in hours, minutes and seconds, the conversion formula does not require any extra conversion as the Right Ascension one required [26].

$$\delta = (DEd \pm DEM/60 \pm DES/3600) \times \pi/180 \quad (2)$$

this time δ represents the Declination degrees expressed as the Right Ascension value in a decimal form and is called the altitude value. DEd , DEM and DES represent the value Declination value expressed in hours, minutes, and seconds. And is multiplied by $\pi/180$ because the value should specify if the celestial body is towards north or south.

5 Integrating Arduino and embedded components

Arduino is a ready-to-use structure that adopts a complete package, including 5V regulator, burner, oscillator, microcontroller, serial communication interface, LED, and connector. The developer or engineer only needs to think about the connections for programming or any other interface and after that the Arduino board it will be plugged it into the USB port. This revolutionary board changes the world with just a few sentences of coding [27].

But for educational purposes, it can come as a whole created by someone else, which is the case for this robotic arm, or it can be developed or improved by someone else. If young students that are not passionate about technology are considering, most of them will not be interested in this category. Most children are usually interested in model making and can like it among their peers. By imagining electronic scorers or interactive house models to make a board game-gender and self-image safety projects can be completed through Arduino, as is happening for the stellar pointer.

Because of these target audiences, using this Stellar Pointer is easy and efficient-you plug and play board, but at the same time powerful features can be introduced, such as the students can be encouraged to add a new feature to the robot, such as supporting new planets or tracking satellites.

Another targeted audience is the hobbyist astrophysicists and physicists that want to use and develop a robot for their home project that involves hardware control and reaction to sensors. This is what the Stellar Pointer aims to do. But also, the astrophysicists and physicists that see this robot as a prototype and may use them for their research or even to scale it so that they may use other materials to build it, keeping the same concept.

5.1 The advantage of embedded components

Besides the Arduino component, the robotic arm consists of three servo motors, one stepper motor and one IR sensor. As was described in the second section there are two coordinates needed to point the celestial body. To do that, the three servo motors are used to translate the altitude value from 0° to 90° and from 0° to -90° . The servomotors were was in this scope because it offers a broad range of purposes and because their micro-stepping capabilities that make it possible for most modern drive electronics to step or raise a stepper motor to a large number of steps per revolution or higher resolution.

The stepper motor was used to translate the azimuth values from 0° to 360° . The reasoning behind using the stepper motor for this translation was that it has become easier and easier to use due to open source libraries. Furthermore, the growing "maker revolution" has made them more popular at the same time and reduced their expense. Stepper motors do not need tuning to maximize their efficiency, unlike servo motors. And using the stepper motor, scaling and motion commands are usually easy to perform.

Another possible solution was to use another servo motor instead of the stepper motor to translate the azimuth values. But after researching this option, it came to be not a feasible one because servo motors are available as traditional rotary motors, as well as linear servo motors, much like the stepper motors. The several advantages of the stepper motors lead to an increasing number of instrument makers preferring servo motors to improve the efficiency of their instruments, such as higher speed, greater precision, and more torque.

The IR sensor was used to keep track of the servo motor movements and to reset the robotic arm to the initial location to point another celestial body. So that the bottom of the robot has a small gap and where that small gap is detected by the IR sensor by reading a greater value than

that indicates that the stepper is the initial position such that it can point another celestial body [28].

6 The purpose of 3D printing

Now a day's 3D printing is a popular technology for prototyping and manufacturing. The main point of 3D printing is the process of making physical 3D objects from digital computer-aided design files. The printer adds successive layers of material together until the final object is created. Since this robotic arm wants to be an effective teaching tool, the role of 3D printing in the learning process has been given high attention. 3D printers are visual things; therefore, it is very important to realize its potential when wanting to build scalable structures.

3D printing has the following the advantage of rapid prototyping that can shorten the product development cycle, reduce waste, and shorten the time to market. The robot parts can be printed on-demand almost anywhere, and the need to transport products is greatly reduced. Additively if a part will break or if the teacher or student wants to change its colour, they can do that very easily just by printing a new one with another colour.

To scale a part of the Stellar Point up, there will be a less polygonal model there will be a model that is half the size of the detailed limit, but that means that the model may not still look good when will be enlarged. And to overcome this problem is to divide each polygon into smaller parts.

Instead, when scaling a part of the robot down will cause another problem. Similarly, wall thickness restrictions also play a role. If the thickness of a certain part of the model is, for example, 1mm, and the entire model is reduced by 4 times, the part is now 0.25mm thick, which is too thin to be printed. Again, here, the model should be redesigned and be to account different thickness limitations.

Scaling can be done quickly; the effects of polygons and prices should be considered as the main deciding factor. For miniatures, details are critical, so although the entire object does have to be greater than the minimum wall thickness and the section size should be kept below the "detail limit". Similarly, when zooming in on the model, scaling up will exaggerate the surface of the 3D model [29].

Overall, 3D printing is a very interesting technology applied to scientific visualization but since 3D printed scientific models are so easily scalable and there are not expensive, the robot's parts have a great potential into this 3D printing area helping altogether teachers and students.

7 Conclusions

In conclusion, the universe is a fascinating area to research, even by look without any instruments at the sky, some planets and their moons can be seen. By using a telescope more details can be seen such as twinkling stars, particular aspects of our Solar System planets and even planets that are beyond.

The Stellar Pointer is focused on building a robotic arm framework that uses already established mathematical formulas and rules of astrophysics that are easy to create in order to be used without difficulty by people of all ages, but in particular for educational purposes, to locate planets in our Solar System. Our Solar System is the next stage in the human evolution, so before moving to another planet, these can be seen from our Earth by using a telescope and the Stellar Pointer to locate their positions.

Another advantage of the Stellar Pointer technologies is the desire of reducing the cost of a built-in satellite or star tracker. For this, the teacher and students can benefit from because of the low-cost components and its ease of connecting all the parts without a vast experience in this field. For scientist or astrophysicist, its scalability property can be a real advantage. And even for people or students that use a telescope and wants to see where a planet is, this Stellar Pointer represents a pragmatic approach.

Acknowledgement: This work was supervised by Ralf Fabian from the Department of Mathematics and Informatics.

References

- [1] *Curiosity (rover)* Wikipedia [Online]. Available: [https://en.wikipedia.org/wiki/Curiosity_\(rover\)](https://en.wikipedia.org/wiki/Curiosity_(rover)). [Accessed 11 10 2020].
- [2] *Discovery and exploration of the Solar System* [Online]. Available: https://en.wikipedia.org/wiki/Discovery_and_exploration_of_the_Solar_System. [Accessed 11 10 2020].
- [3] *Optical telescope* Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Optical_telescope. [Accessed 11 10 2020].
- [4] *Robotic telescope* Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Robotic_telescope. [Accessed 11 10 2020].
- [5] Opromolla, Roberto, Giancarmine Fasano, Giancarlo Rufino, Michele Grassi, Claudio Pernechele, and Cesare Dionisio. *A new star tracker concept for satellite attitude determination based on a multi-purpose panoramic camera*. Acta Astronautica 140 (2017): 166-175.
- [6] Lo, JT-H. *Optimal estimation for the satellite attitude using star tracker measurements*. Automatica 22, no. 4 (1986): 477-482.
- [7] Nguyen, Van-Suong, Nam-Kyun Im, and Quang-Dan Dao. *Azimuth method for ship position in celestial navigation*. International journal of e-Navigation and Maritime Economy 7 (2017): 55-62.
- [8] *Spatial reference system* [Online]. Available: https://en.wikipedia.org/wiki/Spatial_reference_system. [Accessed 11 10 2020].
- [9] *Coordinate reference systems* [Online]. Available: <https://inspire.ec.europa.eu/theme/rs>. [Accessed 11 10 2020].
- [10] 8. *Coordinate Reference Systems* [Online]. Available: https://docs.qgis.org/3.10/en/docs/gentle_gis_introduction/coordinate_reference_systems.html. [Accessed 12 10 2020].
- [11] *Coordinate system* Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Coordinate_system. [Accessed 12 10 2020].
- [12] *Coordinate Reference Systems* [Online]. Available: <https://rsatial.org/raster/spatial/6-crs.html>. [Accessed 12 10 2020].
- [13] *Celestial Coordinates* [Online]. Available: <https://astronomy.swin.edu.au/cosmos/C/Celestial+Coordinates>. [Accessed 12 10 2020].
- [14] J. Harris, *Celestial Coordinate Systems* [Online]. Available: <https://docs.kde.org/trunk5/en/extragear-edu/kstars/ai-skycoords.html>. [Accessed 12 10 2020].
- [15] *Horizontal Coordinate System* [Online]. Available: <https://astronomy.swin.edu.au/cosmos/H/Horizontal+Coordinate+System>. [Accessed 12 10 2020].
- [16] *Horizontal coordinate system* [Online]. Available: https://en.wikipedia.org/wiki/Horizontal_coordinate_system. [Accessed 12 10 2020].
- [17] *Cosmic Coordinates* [Online]. Available: <https://lco.global/spacebook/sky/equatorial-coordinate-system/>. [Accessed 17 10 2020].

- [18] *Equatorial coordinate system* [Online]. Available: https://en.wikipedia.org/wiki/Equatorial_coordinate_system. [Accessed 17 10 2020].
- [19] *Presentation* [Online]. Available: <https://ssp.imcce.fr/webservices/>. [Accessed 17 10 2020].
- [20] *Few Good Reasons for Why you Need Web APIs for ASP.NET Application Development* [Online]. Available: <https://www.brainvire.com/few-good-reasons-for-why-you-need-web-apis-for-asp-net-application-development/>. [Accessed 17 10 2020].
- [21] *How to do? Miriade .ephemcc Positional ephemerides* [Online]. Available: <https://ssp.imcce.fr/webservices/miriade/howto/ephemcc/>. [Accessed 18 10 2020].
- [22] *Miriade ephemcc Positional ephemerides* [Online]. Available: <https://ssp.imcce.fr/webservices/miriade/api/ephemcc/>. [Accessed 18 10 2020].
- [23] Eduard-Traian Stefanescu. [Online]. Available: <https://github.com/StefanescuEduard/StellarPointer>. [Accessed 24 10 2020].
- [24] S. Leathem, *How to Calculate Longitude from Right Ascension* [Online]. Available: <https://sciencing.com/calculate-longitude-right-ascension-6742230.html>. [Accessed 18 10 2020].
- [25] *Lecture 1: Introduction to Astronomy 250* [Online]. Available: http://ircamera.as.arizona.edu/astr_250/Lectures/LECTURE_01.HTM. [Accessed 18 10 2020].
- [26] *How can I convert Right Ascension and declination to distances?* [Online]. Available: <https://physics.stackexchange.com/questions/224950/how-can-i-convert-right-ascension-and-declination-to-distances>. [Accessed 18 10 2020].
- [27] I. Sarwar, *Advantages and Disadvantages of Using Arduino* [Online]. Available: <https://engineerexperiences.com/advantages-and-disadvantages.html>. [Accessed 18 10 2020].
- [28] *Advantages of Servo Motor and Direct Drive Technology* [Online]. Available: <https://www.techbriefs.com/component/content/article/tb/supplements/mcat/features/articles/25416>. [Accessed 21 10 2020].
- [29] *Design tips for shrinking and enlarging models to scale* [Online]. Available: <https://support.shapeways.com/hc/en-us/articles/360023732334-Design-tips-for-shrinking-and-enlarging-models-to-scale>. [Accessed 20 10 2020].

Eduard-Traian STEFANESCU
University Lucian Blaga of Sibiu
Faculty of Sciences
Ion-Ratiu 5-7, Sibiu 550012
ROMANIA
E-mail: eduard.traian.stefanescu@gmail.com

Using Machine Learning in retail industry. A case study on Lidl fresh food market

Römer Walter

Abstract

The aim of this paper is to present a solution for predicting the orders for fresh goods especially in fresh market supply chain, in our case in Lidl market. The proposed solution is based on Machine Learning (ML) techniques, and uses the Prophet Software system produced by the Facebook Data Science team. Using the prediction capabilities of this algorithm with real historical data from LIDL, I established a starting point for building an Artificial Intelligence (AI) based architecture for stock predictions within the company. The Prophet algorithm gives a robust prediction, has a high tolerance for data noises and can be easy customized. From a technical point of view, the proposed solution uses Python with the inclusion of the Pandas class to be able to interact with the data given in a csv. format.

1 Introduction

Machine Learning techniques in the retail industry, especially in the field of fresh produce is more relevant than ever, while with the help of special designed algorithms retailers achieve to obtain better prediction for supply chains and within that they manage to be more efficient productive and earn more, consolidating the position on market.

Today, Machine Learning works all around us, when we interact with banks, buy online or use social media, machine learning algorithms come into play to create an efficient, smooth and safe experience [1].

Machine learning and the technologies that support it are developing rapidly, although we are only at the beginning of using their functionalities.

For decades, traditional data analytics have worked perfectly for the retail industry. However, AI and ML have introduced a new level of data processing which leads to a new perspective on business in general and a paradigm shift in the field of retail market.

Between 2013 and 2018, the AI and ML used by Amazon recorded gains of 1.8 billion dollars, so Amazon launched a new challenge on the market, thus forcing all economic agents in the retail market to use these techniques to be able to remain efficient in the market [2].

Thus, it can be stated that the implementation of smart systems by retailers becomes more than an option representing a matter of adaptation or loss of market share and thus bankruptcy.

In the retail industry, food stores in particular are constantly facing the challenge of maintaining a balance between purchases and sales.

Any imbalance in this delicate balance sheet brings financial losses, affects the performance of internal processes and incapacitates the retailer to sell to the maximum potential that the profile market offers at that time. Errors in this segment generate losses of 1.1 trillion euros per year [3].

Nowadays, the offline market must be able to compete with the challenges imposed by the online market and retailers to meet the customers who are increasingly accustomed to the multitude of benefits that the online domain offers, becoming dependent on fast and adapted services their needs are quickly reoriented in the traditional offline market where these possibilities do not exist.

The supply chain is an area that can be greatly transformed by implementing the solutions offered by AI, as there will always be a need for faster product delivery and better inventory control. Artificial Intelligence can provide a clear view of how a supply chain works and can quickly detect its inefficiencies by creating multiple ways to improve them by providing better predictions.

The problem becomes much more complex when a retailer has several stores that are placed in completely different areas in terms of purchasing options and consumption levels as well as their number of customers or revenue. A closer analysis increases the complexity as the marketing of some products is perceived differently depending on the area and population of a region and the fact that in the food industry there is a considerable difference in consumption depending on the season, so the consumption of some foods is influenced by many factors [4].

AI and ML techniques applied in the supply chain can be used to reshape and calculate the demand for certain products taking into account several variables such as: sales history, geographical area of the store, weather context in a certain range, trends from a certain period, the profile of the population in the region, the incomes, the preoccupations in the respective area and many others.

The desideratum of the applicability of ML techniques in the supply process in the retail industry is represented by the desire of all retailers to obtain a better ability to order products without involving human interaction and to have the guarantee that their volume is optimal, avoiding financial losses.

ML algorithms will analyze the patterns of requests from historical data and will provide actionable information that will lead to a better understanding of customers, their interests and needs [5].

Using ML approaches it will be possible to automatically transmit inventory needs and predict stock incidents. Consequently the supply chain will be faster, more efficient and less expensive.

Certainly in the market, who will have access to superior stock prediction systems will be one step ahead of the competition and will be able to ensure the needs of customers even before they have them.

The convergence between AI and the Internet of Things (IoT), will facilitate the next step forward in areas such as trade and production, and the beginning of so called Industry 4.0 revolution. The goal in this area of innovation is to provide better, smarter and faster automated services, based on a more accurate understanding of certain specific environments.

The e-commerce is the area where companies are created on digital foundations. In this ecosystem, notions such as artificial intelligence, data, personalized, proximity and unified trade are the main competitive strengths. But gradually, all areas of retail will be affected. Proof of this is the significant increase in smart store concepts, without cashier or human operator, or Amazon's attempt to go offline, through the Amazon Go concept, where customers can enter by scanning a QR code and

get what they want, leaving the store without being verified, because algorithms that use artificial intelligence monitor their activity [6].

At the same time, reducing food losses globally can ensure better predictability for producers and a real and fair level of price for food, less food thrown away means ultimately a higher performance of the whole chain leading to a standard of living. Better and stability by relaxing economies and economic policies by standardizing production capacities and bringing performance to producers who will not have to produce goods that will never be consumed.

In the last decade, most retailers in Romania have become interested in the solutions available in the field of Artificial Intelligence, but there is still a low degree of their use in business processes. The reluctance comes from the lack of availability on the internal software market of customizable and reliable products and the high costs that these technologies entail for the retailer related to the need to recruit qualified staff in a cutting-edge field. The convenience and considerable effort involved in changing internal processes and workflows are also reasons that make it difficult to engage these smart techniques in retailers IT systems.

Most current IT systems offer a standard structure of the main module and the possibility of adding other modules necessary for various operations: fixed assets management, working with the cash register, the module for financial analysis, etc. Even if most companies producing applications in the field of AI and ML promise the compatibility of the software offered with any type of internal work process, this is usually not entirely true.

Despite these limitations, offline retailers, especially those in the food industry that sell perishable products, are becoming increasingly interested in this vast field that offers multiple possibilities and are aware that ignoring them will inevitably lead to decline as the inability to adapt to everyday realities market means bankruptcy.

Therefore, the evaluation of smart IT solutions capable of adding value in a business needs to be adapted and harmonized with internal work processes and at the same time internal work processes need to adapt to the new capabilities offered by smart systems.

The global retail market is about to undergo a major change and the mechanisms involving Artificial Intelligence are there to transform stores of all kinds.

A desideratum of all retailers is to offer customers what they want even before they realize that they need it. In an increasingly connected world, the relationship between customers and stores will be made more and more through artificial intelligence.

As consumers increasingly look for fast, high-quality services, it is certain that retailers need to be one step ahead of customer needs, and this means digital transformation backed by artificial intelligence.

Retailers need to adapt their applications to the technologies already used by their customers (mobile, portable devices) to help them when they are in the store, while also finding a way to log in easily from the moment they enter in the environment of that retailer. Thus, artificial intelligence transposes the personalized experience of online shopping with physical retail, offline, instantly finding product recommendations and rewarding customer loyalty. Artificial intelligence supports store employees by giving them all the information they need for each customer. The result is a broader perspective on buyer behavior for retailers and a seamless customer experience, regardless of the retail channel in question.

In this article I propose a solution based on Machine Learning for stock predictions within the LIDL company. The solution is built using the Prophet Software system and real historical data from

LIDL. The Prophet algorithm gives a robust prediction, has a high tolerance for data noises and can be easily customized. The next of the article is organized as followed. Section 2 describes the proposed solution. The results are presented in Section 3. Conclusions and future directions of study are highlighted in Section 4.

2 The proposed solution

In the chosen solution I analyzed the time series with Prophet algorithm and made predictions for our supply chain. The main advantage of this algorithm is that it is very flexible when it comes to the data that is fed to the algorithm. The Prophet algorithm is an additive model, which means that it detects the following trend and seasonality from the data first, then combine them together to get the forecasted values. Therefore it is a perfect choice for our predictions since we want to get the overall trend, the daily, weekly and yearly seasonality and also the holidays effect in fresh food market.

With this assets of the Prophet is possible to gain quite a lot of useful insights from the model of our predictions. The Prophet algorithm is used for forecasting data from time series and is based on an additive model in which non-linear trends match different situations that are influenced by uneven demand. It works very well with data that have a daily periodicity of at least one year, being at the same time robust in case of lack of data in a certain interval, at changes in trend and values well above the average of a period. Prophet is an open source software and is provided by the Facebook Data Science team being developed by Facebook for predictions related to Facebook's internal applications.

Prophet provides accurate predictions for a wide range of products based on sales history and various attributes such as: price, promotions and stocks along with external factors such as days off, weather forecast, local holidays, etc.

From the Prophet's strengths that led to his qualification as the best in our case study we list the following:

- Accurate and fast: It is commonly used by Facebook in several applications to obtain reliable predictions and over time has proven to be the best tool for this task within Facebook.
- Automatic: we can obtain reasonable predictions using a heterogeneous data set without the need for intervention on the standardization of source data.
- Customizable predictions: It offers a wide range of customizations for various predictions, and supports parameters from various fields to improve predictions.
- Scalable: The Prophet procedure, is implemented in Python or R and can use multiple languages to obtain predictions [7].

In our study we have had access to a data set that represents the sales of two stores over a period of 365 days. With the help of the Prophet algorithm we made sales predictions for the next 365 days from the date of the last registration.

We performed a validation of our model for several goods and the predictions are very close to real demand for this goods by comparing with real sales. The results are presented in Section 3.

3 Results

First, we trained the algorithm in console on the available training data. The trained model was used to make predictions on a desired product. The training data were historical data with the sales of this product on the past 365 days. We predicted the sales for the next 365 days.

The results are saved as a .csv file. The algorithm provides the date, the trend and the rounded prediction. The trend column indicates the arithmetic average of sales until the date for which the prediction is made, and the rounded prediction column contains the prediction made by the Prophet and then rounded for an easier interpretation.

The accuracy on the first 20 training data set is presented in Figure 2. Figure 1 contains a snapshots from console with the prediction commands.

The prediction is made by using the method *predict*.

The instruction *forecast = forecast[['ds', 'trend', 'yhat']]* select the 3 columns we are interested in: date, product and predicted quantity.

In order to simplify the prediction result for an easy reading by the end user of the prediction, we will make a rounding to the whole obtained result: *forecast['approximation roundness'] = round(forecast['approximation roundness'])*. The last step in the prediction is to export it to a csv file that can be used in any other computer system or can be an input for another application: *forecast[['date', 'trend', 'aproximare rotunjita']].to_csv(product + '.csv')*.

```

39 forecast = m.predict(future)
40
41 print(last_data)
42
43 forecast=forecast[forecast['ds'] > last_data]
44 forecast=forecast.reset_index()
45
46 forecast=forecast[['ds', 'trend', 'yhat']]
47 forecast.columns=['date', 'trend', 'aproximare rotunjita']
48
49 forecast['aproximare rotunjita']=round(forecast['aproximare rotunjita'])
50
51 forecast[['date', 'trend', 'aproximare rotunjita']].to_csv(product + '.csv')

```

Figure 1: Snapshots from console with the main prediction commands

	Date	Trend	Rounded Prediction	Real Sale	Accuracy %
0	05.12.2019	45,5074263	42	39	92,8%
1	06.12.2019	45,58323591	45	48	93,7%
2	07.12.2019	45,65904552	47	40	85,1%
3	08.12.2019	45,73485512	43	47	91,4%
4	09.12.2019	45,81066473	47	33	70,2%
5	10.12.2019	45,88647434	46	50	92%
6	11.12.2019	45,96228395	50	38	76%
7	12.12.2019	46,03809355	43	50	86%
8	13.12.2019	46,11390316	45	42	93,3%
9	14.12.2019	46,18971277	48	55	87,2%
10	15.12.2019	46,26552238	43	38	88,3%
11	16.12.2019	46,34133198	48	52	92,3%
12	17.12.2019	46,41714159	46	40	86,9%
13	18.12.2019	46,4929512	51	57	89,4%
14	19.12.2019	46,56876081	43	28	65,1%
15	20.12.2019	46,64457041	46	50	92%
16	21.12.2019	46,72038002	48	50	96%
17	22.12.2019	46,79618963	44	42	95,4%
18	23.12.2019	46,87199924	48	48	100%
19	24.12.2019	46,94780884	47	39	82,9%

Figure 2: The resulted csv file with predictions

4 Conclusions and future directions of study

As the accuracy of the predictions made with AI and ML techniques are by far better than older methods, the use of these tools becomes a real necessity in the retail market for trade outside the online environment for any profit-oriented retailer. The use of Artificial Intelligence and Machine Learning technology can be applied not only in the case of stock predictions but also in the case of prices or in other areas that do not necessarily require predictions.

By using methods similar to our case study it will be possible to obtain many advantages, some of which can be mentioned: reduced response time for obtaining valuable information to inventory management in logistics centers, reduced workload, reduced costs related to staff in charge of inventory management, reduction of costs with stock handling (transport costs calibrated to the need for consumption, efficient use of storage space, etc.).

The implementation of this type of techniques for predictions including the one proposed by me in this paper, can be done taking into account the opinions, indications and suggestions of all those involved in the process even from end customers. Only through an effort combined with the desire to create a powerful tool for data management and prediction can better results be achieved.

On the one hand, retailers can be helped by AI to turn the enormous amount of data they have available into clues to guide them in terms of consumer desires, and on the other hand, to automate

some repetitive tasks, leaving store employees the opportunity to do work with a greater impact on customer satisfaction.

Also noteworthy are the offers that the field of Artificial Intelligence and Machine Learning offers in areas such as marketing, studying customer behavior, consumption trends and augmented reality.

A strong temptation is represented by the outsourcing of these tools to companies that offer customized solutions and are certainly viable solutions for many retailers in the market and is a step forward in the technological development of the company and internal work systems but comes with a loss of data control and the inherent problems brought by the complexity of customizing in an external environment solutions that correspond to internal systems.

The intelligent solutions developed within the IT departments of retailers should meet the needs of the company as they can be quickly calibrated to the profile of each field of consulting software solutions.

Thus, with the internally developed solutions, the data are better protected, the possibility of data leaks in this field representing a risk for the company as currently the information is the most valuable that an actor can have in the global market.

An advantage in addition to the traditional outsourcing of AI services that more and more retailers turn to is the possibility of current and fast knowledge of the situation of the company's processes and the possibility of rapid recalibration of deficiencies.

The trends of the last years are manifested on the line of purchasing these systems and software architectures that have integrated modules specific to each field but the problems of these external solutions are related to the criteria underlying the choice of these products. It is usually found that without specialist advice, retailers fail by purchasing software solutions that are inappropriate for the profile, internal processes and requirements of future customers, and these systems will be abandoned in time in favor of the old ones for reasons related primarily of operability and / or poor results.

Despite these shortcomings, the future of Artificial Intelligence solutions is promising, in the coming years most retailers will use a smart IT solution for some of their internal processes.

The topic of this paper remains an open point of discussion as the fields of Artificial Intelligence and Machine learning are vast and there are still possible extensions of this project in several directions.

Acknowledgement: I thank Prof. Dr. Dana SIMIAN for the useful discussions and suggestions provided in the development of this work.

References

- [1] Artificial Intelligence in Retail Market by Type (Online, Offline), Technology (Machine Learning and Deep Learning, NLP), Solution, Service (Professional, Managed), Deployment Mode (Cloud, On-Premises), Application, Region - Global Forecast to 2022, <https://www.marketsandmarkets.com/Market-Reports/artificial-intelligence-ai-retail-market-36255973.html>
- [2] Artificial Intelligence for Retail in 2020: 12 Real-World Use Cases, Roman Chuprina, 20.12.2019, <https://spd.group/artificial-intelligence/ai-for-retail/>
- [3] Artificial Intelligence for Retail in 2020: 12 Real-World Use Cases, Roman Chuprina, 20.12.2019, <https://spd.group/artificial-intelligence/ai-for-retail/>

- [4] Artificial Intelligence in Retail Market - Growth, Trends and Forecast (2020 - 2025) Mordor Intelligence, <https://www.mordorintelligence.com/industry-reports/artificial-intelligence-in-retail-market>
- [5] How AI can save the retail industry, Macy Bayern, 13.09.2019 <https://www.techrepublic.com/article/how-ai-can-save-the-retail-industry/>
- [6] AI Retail Playbook, Transformation Strategies for Intelligent Retail, Microsoft with PSFK, Piers Fawkes, Scott Lachut, Conner Dial, <https://info.microsoft.com/rs/157-GQE-382/images/Final%20AI%20Retail%20Playbook.pdf>
- [7] Forecasting at scale, Prophet, Facebook Data Science Team, <https://facebook.github.io/prophet/>

Römer WALTER
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Dr. Ion Ratiu no 5-7, 550012 Sibiu
ROMANIA
E-mail: waltercezar.romer@ulbsibiu.ro

LIST OF AUTHORS

- Max ARNDT** University of Applied Sciences Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12, 97070 Würzburg
GERMANY
E-mail: max.arndt@student.fhws.de
- Christian BACHMEIR** University of Applied Sciences Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12, 97070 Würzburg
GERMANY
E-mail: christian.bachmeir@fhws.de
- Eugen BECKER** University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: eugen.becker@student.fhws.de
- Roland BOLBOACĂ** "George Emil Palade" University of Medicine, Pharmacy,
Sciences and Technology of Târgu Mureş.
Faculty of Engineering and Information Technology
Str N. Iorga no. 1, Târgu Mureş
ROMANIA
E-mail: roland.bolboaca@umfst.ro
- Iuliana-Maria BURUIANĂ** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: iuliana.buruiana@hotmail.com
- Stel CARAMIHAI** Ovidius University of Constanta
Faculty of Mathematics and Computer Science
124, Mamaia Blvd. Constanta
ROMANIA
E-mail: caramihaistela@gmail.com
- Arina Ioana CAZACU** Babes-Bolyai University
Computer Science
Mihail Kogălniceanu 1, Cluj-Napoca 400000
ROMANIA
E-mail: arina.cazacu@gmail.com
- Paul-Robert CEOLCA** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: paul.ceolca@ulbsibiu.ro

- Alexandru-Mihail CRĂCIUN** Babes-Bolyai University
Faculty of Mathematics and Computer Science
Mihail Kogalniceanu nr.1, 400084, Cluj-Napoca,
ROMANIA
E-mail: alex.mihail.craciun@gmail.com
- Valentin-Gabriel CRĂCIUN** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: gabriel.craciun@ulbsibiu.ro
- Ligia – Izabela CRĂCIUNESCU** Politehnica University Timișoara
Faculty of Automation and Computers
Timișoara
ROMANIA
E-mail: craciunescu.ligia@yahoo.com
- Alexandru DANCAU** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: alexandru.dancau@ulbsibiu.ro
- Andreea DOGARU** Transilvania University of Brasov
Faculty of Mathematics and Computer
Science
Iuliu Maniu 50, 500091 Brasov
ROMANIA
E-mail: andreea.dogaru.d@gmail.com
- Tobias FERTIG** University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97070 Würzburg
GERMANY
E-mail: tobias.fertig@fhws.de
- Lars FICHTEL** University of Applied Sciences Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12, 97070 Würzburg
GERMANY
E-mail: lars.fichtel@fhws.de
- Răzvan Gheorghe FILEA** Samuel von Brukenthal National College
Sibiu, ROMANIA
E-mail: razvan.filea@gmail.com
- Andreas FILINGER** University of Applied Sciences Landshut
Faculty of Computer Science
Am Lurzenhof 1, 84036 Landshut
GERMANY
E-mail: andreas.flinger@gmail.com

- Alexander M. FRÜHWALD** University of Applied Sciences Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12, 97070 Würzburg
GERMANY
E-mail: alexander.fruehwald@fhws.de
- Matei-Florin GRAURĂ** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: grauramatei@gmail.com
- Bogdan-George GROS** Aurel Vlaicu University of Arad
Str. Elena Drăgoi, nr. 2,
310330, Arad
ROMANIA
E-mail: gros.bogdan@yahoo.com
- Simon HAAS** University of Applied Sciences Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12, 97070 Würzburg
GERMANY
E-mail: simon.haas@student.fhws.de
- Janik HEMRICH** University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: janik.hemrich@student.fhws.de
- Leonhard HÖSCH** University of Applied Sciences Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12, 97070 Würzburg
GERMANY
E-mail: leonhard.hoesch@fhws.de
- Karsten HUFFSTADT** University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97070 Würzburg
GERMANY
E-mail: karsten.huffstadt@fhws.de
- Felix HUSAC** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: husacfelix@gmail.com
- Marina – Larisa INDRECAN** Ovidius University of Constanta
Faculty of Mathematics and Computer Science
124, Mamaia Blvd. Constanta
ROMANIA
E-mail: maryna_larysa@yahoo.com

- Steffen KASTNER** University of Applied Sciences Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12, 97070 Würzburg
GERMANY
E-mail: steffen.kastner@student.fhws.de
- Stella KONIECZEK** University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: stella.konieczek@student.fhws.de
- Răzvan-Cosmin LINCA** University of Babes,-Bolyai
Faculty of Computer Science
Mihail Kogălniceanu Street 1, Cluj-Napoca
ROMANIA
E-mail: cosmin_linca@outlook.com
- Mădălina MARINESCU** Politehnica University Timisoara
Faculty of Automation and Computers
Timisoara
ROMANIA
E-mail: madalinamarinescu96@gmail.com
- Christian MELCHIOR** “Doamna Stanca” High School, Făgăraş
Strada Doamna Stanca , No. 14 505200
ROMANIA
E-mail: christianmelchior7@gmail.com
- Nicholas H. MÜLLER** University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97070 Würzburg
GERMANY
E-mail: nicholas.mueller@fhws.de
- Teodora POPA** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: teodora1.popa@ulbsibiu.ro
- Milan SAVIĆ** Singidunum University
Faculty of Technical Sciences
32 Danijelova St., Belgrade
SERBIA
E-mail: milan.savic.16@singimail.rs
- Helena SCHMIEDL** Friedrich-Alexander-University
Erlangen-Nuremberg
Institute of Psychogerontology
Kobergerstr. 62, 90408 Nürnberg
GERMANY
E-mail: helena.schmiedl@fau.de

- Anna-Maria SCHMITT** University of Applied Sciences Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12, 97070 Würzburg
GERMANY
E-mail: anna-maria.schmitt.1@student.fhws.de
- Vitaliy SCHREIBMANN** University of Applied Sciences Würzburg-Schweinfurt
Institut Digital Engineering
Münzstraße 12, 97070 Würzburg
GERMANY
E-mail: vitaliy.schreibmann@fhws.de
- Andreas SCHÜTZ** University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97070 Würzburg
GERMANY
E-mail: andreas.schuetz@fhws.de
- Justin SEEGETS** University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: justin.seegets@student.fhws.de
- Constantin Marius STANCIU** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: mmscm22@gmail.com
- Sebastian STOICA** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: sebastian.stoica@ulbsibiu.ro
- André STOLLBERGER** University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97070 Würzburg
GERMANY
E-mail: andre.stollberger@student.fhws.de
- Minodora SUILEA** Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: minodora.suilea@ulbsibiu.ro

Eduard-Traian ȘTEFĂNESCU

Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: eduard.traian.stefanescu@gmail.com

Daniel WAGNER

University of Applied Sciences Würzburg-Schweinfurt
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: daniel.wagner.2@student.fhws.de

Römer WALTER

Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: waltercezar.romer@ulbsibiu.ro

SPONSORS (in alphabetical order)



AUSY Technologies Romania



Asociația BIT



CodexWorks technologies



Fundația Academia Ardeleană



Global Solutions for Development



IQuest



Keep Calling



NTT Data



PAN FOOD



Omeron Technologies, Romania



ProIT



ROPARDO



Top Tech



VISMA