

Proceedings in
Informatics and Information Technologies

Student Research Conference 2021

Ján Lang and Rastislav Bencel (Eds.)

Keynote by Peter Dolog

Ján Lang and Rastislav Bencel
(Eds.)

IIT.SRC 2021: Student Research Conference

17th Student Research Conference
in Informatics and Information Technologies
Bratislava, April 22, 2021
Proceedings



SLOVAK UNIVERSITY OF
TECHNOLOGY IN BRATISLAVA
FACULTY OF INFORMATICS
AND INFORMATION TECHNOLOGIES

Proceedings in
Informatics and Information Technologies

IIT.SRC 2021
Student Research Conference

Editors
Ján Lang and Rastislav Bencel
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2
842 16 Bratislava, Slovakia

© 2021 Contributions are printed as delivered by authors without substantial modifications.

For internal use only.

Visit IIT.SRC on the Web: <https://www.fiit.stuba.sk/iit-src2021>

Executive Editors: Rastislav Bencel

Copy Editor: Ján Lang
FIIT STU

FIIT STU
Ilkovičova 2, Bratislava, Slovakia
Published in Slovak University of Technology in Bratislava in SPEKTRUM STU Publishing
2021
ISBN 978-80-227-5161-2

Preface

The criteria of universities quality appraisal are primarily focused on their research activities. One of the crucial indicators is the research results publication. So, it is obvious a university activity setting – not except our faculty – the Faculty of Informatics and Information Technologies of the Slovak University of Technology in Bratislava – oriented to the closest possible research and education connection. There is a natural effort of researchers to name research problems, analyse existing solutions and approaches in close confrontation with the design and implementation of their own solution. State of the art in a domain, as well as their own approach in solving problems naturally together with evaluation, are usually part of the final work of students at all three levels of university study. Joint research activities of students and their teachers, researchers are also documented by other publications. Partial results are often published at thematically related conferences in the form of research articles. The research orientation degree of a student is directly proportional to the university study degree. The highest rate is expected for students at the doctoral level of study, which is primarily focused on research in the field. The lowest but not negligible is also clear at the bachelor study level of university study. Bachelor students also use research techniques although their final theses are more of an implementation nature. The documented research output, exposed in the community of stakeholders, presupposes factual criticism in the form of a review. Assessment of papers is a natural process of selecting the relevance of the issue and the solution offered at conferences as well. The IIT.SRC Student Scientific Conference is an opportunity for students to experience the feeling of attending the conference, to experience the flavour of an international scientific conference and to receive feedback on published research results. The conference is an opportunity for these comments to lead to a real improvement in the scientific and research nature of the article. That is why it was considered particularly important to prepare at least two or more reviews for each article. Both internal researchers and external researchers were involved in the assessment, and finally, the students themselves, who had at least one such experience of publishing at another conference.

A total of 67 papers were submitted to the conference. At least two reviews per paper were applied. There were 39 program committee members and 20 additional reviewers. Finally, 66 papers were accepted and so the acceptance rate is 98.51%. The conference was attended by 112 authors - students at all three levels of study. Exactly there were 7 doctoral students, 82 master students, 21 bachelor students, and two high school students. Given the number of articles published in this collection, the experienced feeling and feeling in the context of the provided feedback has shifted the quality of articles so that they are published on other forums. Selected young researchers were supported by special travel grants that cover partially their travel expenses to conferences. The best paper awards were also presented. The conference was opened by Professor Markus Rupp from Vienna University of Technology, Austria, and his keynote entitled: Wireless Communications of the Future.

Bratislava, April 2021

Ján Lang and Rastislav Bencel

Conference Organisation

	The 17th Student Research Conference in Informatics and Information Technologies (IIT.SRC), held on April 22, 2021 in Bratislava, was organised by the Slovak University of technology (and, in particular, its Faculty of Informatics and Information Technologies) in Bratislava. All from FIIT STU in Bratislava, Slovakia.
--	--

Steering Committee

Valentino Vranić (Steering Committee Chair)

Katarína Jelemenská

Ivan Kotuliak

Lukáš Šoltés

Program Committee

Ján Lang (Program Chair)

Rastislav Bencel (Program Vice-Chair)

Vanda Benešová

Peter Kapec

Fedor Lehocki

Martin Bobák

Lukáš Kohútka

Dominik Macko

Pavel Čičák

Kristián Košťál

Martin Madaras

Ján Genčí

Ivan Kotuliak

Giang Nguyen

Pavol Helebrandt

Michal Kováč

Marta Prnova

František Horvát

Tomáš Kováčik

Martin Šeleng

Xiaolu Hou

Monika Kováčová

Lukáš Šoltés

Ján Hudec

Tibor Krajčovič

Peter Trúchly

Ladislav Hudec

Štefan Krištofik

Jozef Tvarožek

Vladimír Janiš

Eduard Kuric

Valentino Vranić

Katarína Jelemenská

Ján Laštinec

Jan Zelenka

Local Organizing Committee

Zuzana Marušincová (Local Chair)

Zuzana Horniaková

Ľubica Palatinusová

Roman Stovíček

Katarína Mršková

Branislav Steinmüller

Additional Reviewers

Jaroslav Abaffy

Ivan Kapustík

Igor Stupavský

Peter Bakonyi

Matej Kompánek

Waheedullah Sulaiman

Pavle Dakić

Miroslav Laco

Khail

Tomáš Frťala

Viktor Lančarič

Alexander Valach

Stefan Grivalský

Peter Lehoczky

Michal Valiček

Patrik Honíšek

Lukas Mastil'ak

Juraj Vincúr

Lukáš Hudec

Juraj Petřík

Shakirullah Waseeb

Supporting industry partners



Table of Contents

Keynote:

Understanding and Predicting User Behavior by Data Analysis and Machine Learning

Peter Dolog

Object recognition using the convolutional neural networks

Lukáš Haninčík 1

Olive oil classification

Simona Klučková, Maroš Kollár, Jakub Kučečka, Zuzana Popovcová, Mária Rajníková, Alena Valová 10

Overview of Some Windows Server Security Features in Accordance to ISO/IEC 27002 – System Operation Management

Martin Pavelka 17

Analysis of market data using technical indicators and decision trees

Robert Gers 26

Music analysis using deep learning

Matej Halinkovič 31

Data Collection System

Vladimír Bachan, Maroš Čerget', Veronika Čipelová, Ludovít Popelka, Robert Yamkovyi, Tomáš Zátka 43

VR Lab

Patrik Tománek, Lubomír Kurčák, Viktor Beňo, Erik Paľa 51

Web-based teamwork support for the SystemVerilog language

Matúš Pilňan 58

Web system intrusion detection based on computational intelligence methods

Jakub Hubert 70

Controlled generation of realistic human faces

Róbert Belanec 78

Scheduling Tool for Deterministic Communication in Distributed Real-Time Systems

Veronika Búcsiová 87

Visualization and Manipulation of a Volumetric Representation for the Needs of 3D Scan Processing

Štefan Schindler 100

Lines detection via Hough Transform

Kamil-Martin Somrak 115

Tool for testing web application security

Patrik Bobocký 122

Understanding and Predicting User Behavior by Data Analysis and Machine Learning

Peter Dolog

Aalborg University, Denmark
dolog@cs.aau.dk

Abstract. Almost any human activity has been affected by digitization. We interact with each other by means of e-mail or social networks, we seek and publish information online, we share our opinions online, we shop online, and we pay by credit cards or by mobile payment solutions, and so on. Digitization simplified our lives and made parts of our life more effective. By such interactions online, we are also leaving our trails behind. These trails in the logs and databases are signals of our preferences, interests, needs, and not least partial signals of our behavior. Current digitized world with traces of our activities online provide rich data and gives us possibilities and opportunities to help people at different levels of activities. Computational user modeling gives us possibilities to understand the user behavior and preferences from online electronic data. Personalization models and algorithms help users in their tasks and predict their behavior based on learned user model. In this talk, I will touch upon some approaches we have developed. First, I will start with recent work on context aware recommendation where we developed multi-view latent factor models based on interaction data with a recommender system. I will present our work on joint collective matrix factorization and collective neural embedding as computational/machine learning models for predicting user preference in user context. I will further touch upon impact of ratings on side information on recommendation system algorithms. I will also present the work where we studied impact of user activities in personalized word clouds as means for navigation in social media. I will also try to sketch some interesting directions for future work.

Object recognition using the convolutional neural networks

Bc. Lukáš Haninčík*^[0000–0002–1589–2780]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ľkovičová 2, 842 16 Bratislava, Slovakia
lukas.hanincik@gmail.com

Abstract. We introduce modern approach of the evaluation of images similarity based on the object captured on these image. In this task we don't work with entire image but we use image patches capturing local features. Our goal is reliable evaluation of similarity of image patches. For this purpose we utilize novel approaches based on machine learning, in particular a deep siamese convolutional neural networks and residual networks.

Keywords: Image patches · Local features · Feature descriptors · Siamese networks · Convolutional networks · Deep learning · HPatches

1 Introduction

Theme of this work belongs to the scientific discipline called computer vision. Objective of the computer vision is finding of approaches of high level understanding of digital image by computer. Researches in computer vision are utilizable in different problems of real world, e.g. cyber-security (face recognition), bio-medicine (searching for anomalies), transportation (traffic sign recognition), e-commerce (searching by images), etc.

Goal of this work is an evaluation of similarity of two or more images. We decided to solve this problem so that we do not work with an image as a whole but we use just image patches (see Fig. 1 a). Each image patch captures a single specific characteristic feature (local feature) of the entire scene. These features are also called key points. Correct encoding of detected key points leads to responsible evaluation of the similarity of key points and later to evaluation of similarity of the images where the given features belong to. Encoded key point presents a vector of real numbers, called a descriptor. Lot of well known frameworks rely on the approach consisting of *1. Key point detector*, *1. Key point orientation estimator* and *1. Key point descriptor*. In this work, we will especially focus on the key point description.

* Master study programme in field: Intelligent Software Systems

Supervisor: Ing. Marek Jakab

Institute of Informatics, Information Systems and Software Engineering

Faculty of Informatics and Information Technologies STU in Bratislava

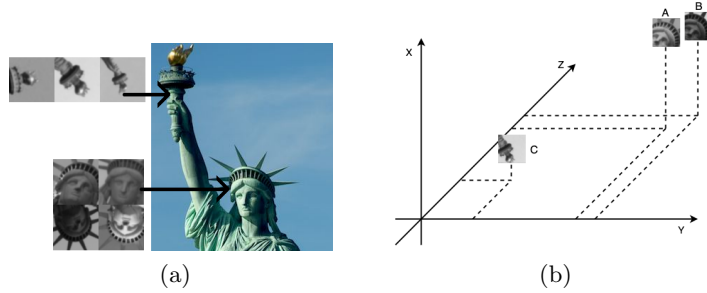


Fig. 1: (a) Visualization of the key points captured on the image patches. Visualized image patches originate to Brown dataset [3]. (b) Suitable placement of descriptors in vector space. Positives A and B lay pretty close in vector space meanwhile the negative C lays far away of them.

2 Key points

As we mentioned in the introduction, a key point refers to a low-level feature of the entire image. Our hypothesis is: if we extract all key points from the image and encode them correctly (based on the properties mentioned in Sec. 3), then we can compare them towards the great database of encoded key points extracted from the large amount of images. In addition, if we can responsibly compare key points of the input image towards the key points from different images, than we can evaluate the similarity of the input image towards the images from database. This approach is more robust and invariant to affinity distortion.

During the key point detection and description we must take a care about the issues of the image. *L. Chen et. al* [5] divided image issues to two categories, photometric transformations and geometric transformations. Since all captured images are different, we must resolve both kinds of deformation before the final evaluation, otherwise we cannot responsibly evaluate the similarity of key points.

3 Descriptors

Image description refers to the transformation of image patch to the vector of real numbers with fixed size, also known as descriptor. Realization of the image description depends on the designed approach. We know two kinds of approaches, classic ones and modern ones. Both kinds are more detailed in Sec. 4 and Sec. 5. In both cases, the descriptors must satisfy a set of requirements. Descriptors must be invariant to all photometric and geometric deformation and sufficiently descriptive with respect to the length of descriptor. Too small descriptors are not sufficiently descriptive but they are computationally effective. Too large descriptors are sufficiently descriptive but they are computationally ineffective. Descriptors of the key point of the same local feature (also called as positives) must lay as close as possible in vector space (see Fig. 1 b). Descriptors of the

key point of different local features (also called as negatives) must lay as far as possible in vector space (see Fig. 1 b).

We can divide descriptors to two categories based on the values of those descriptors: numerical descriptors and binary descriptors. Similarity of numerical descriptors can be evaluated by Euclidian distance or Mahalanobis distance meanwhile the binary descriptors can be easily compared by XOR operation.

4 Classic approaches

Classical approaches are mostly based on very complex deterministic mathematical models. Well known approaches like SIFT [9] or SURF [2] are composed of several modules like: **Scale space estimator** - generates input image on different scales (scale space), **Key point detector** - detects key points from generated scale space and select just the most relevant ones, **Orientation estimator** - estimates an orientation of each key point and rotates the key point, **Key points descriptor** - encodes rotated key points to vector.

Both mentioned approaches SIFT and SURF produce integer descriptors. Well known binary descriptors are BRIEF [4] or ORB [11].

5 Modern approaches

Latest researches in artificial intelligence, especially in machine learning based on deep neural networks, showed the convolutional or siamese convolutional neural networks are suitable instruments for the computation of feature vectors (descriptors) from the input image without the respect of the image's origin. Therefore, modern approaches utilize deep convolutional neural networks for different tasks of computer vision. As shown in multiple sources [12][10][7], modern approaches, based on machine learning, can highly overcome the performance of classic ones.

LIFT [12] presents an analogy of SIFT or SURF approach, but it is composed just of the machine learning components. They introduced a pipeline composed of key points detector, key points cropper, orientation estimator, rotator and descriptor. Entire concept is encapsulated to one huge four channel siamese network and it can be trained as a whole by single forward and backward propagation. End-to-end differentiability is secured by Softargmax functions. Source [6] used two channel siamese network with the Contrastive loss function for the face verification task. Great results achieved HardNet [10] that utilized a Triplet loss function with the online triplets selection before the loss computation. HardNet became a state-of-the-art, in particular because it proves high performance on different tasks of computer vision (in contrast with other works) validated by HPatches benchmark framework [1].

6 HPatches benchmark

HPatches benchmark [1] is a benchmarking framework developed for the image descriptors evaluation. It validates descriptors on three independent tasks - Image matching, patches retrieval and patches verification. In addition, HPatches contains its own dataset composed of image patches and precondition for the descriptor validation is a computation of descriptors from the HPatches dataset. HPatches dataset aggregates different geometrical and photometrical deformations of the image patches with an application of different amount of noise. Results are calculated in Mean Average Precision metrics, since this metrics is not sensitive to highly unbalanced observations.

In this work, we used HPatches benchmark as the only decisive approach to the evaluation of our descriptors because it covers the precision with respect to the most image issues mentioned in Sec. 2. In addition, achieved results can be directly compared with results of another competitors within the HPatches benchmark.

7 Design and implementation

As we mentioned in introduction, the goal of this work is computing the feature vectors (descriptors) from the input image patches, with respect to requirements written down in Sec. 3. As researches showed, utilization of deep machine learning can bring better results and is better scalable then classic ones. Therefore we decided to use a siamese convolutional neural network, that showed as a good choice by lot of sources [10][6][12][7].

While the designing of the network architecture we have been mostly inspired by HardNet [10]. Architecture of the network is better visualized on the Fig. 2. We trained model on Brown dataset composed of three collections of image patches: Statue of Liberty, Notre Dame, Yosemite. Image patches of Statue of Liberty collection is show on Fig. 1 a). In first experiments we trained our model on Yosemite collection but later we found a way, to merge two dataset (including the computation of common mean and standard deviation for patches standardization). Then we trained on Yosemite (with 633587 samples) and Notre Dame (with 468159 samples) collections together. Each collection contains 64×64 grayscale image patches that have been resized to 32×32 using the interpolation. During the batch generating, following augmentation operations are applied with 50% probability: vertical flip, diagonal flip, random rotation (90° , 180° , 270°). This simple augmentation allowed us to extend the training set and force the network to better generalization.

We experimented with loss functions. At first we tried a Contrastive loss function because it turned out to be good choice in case of high amount of observation classes but just a few samples per each class [6]. As the Contrastive loss function expects pair of descriptors and label (whether they are positives or negatives), our siamese neural network was composed of two channels (one for each member of the pair). Then we experimented with a naive implementation

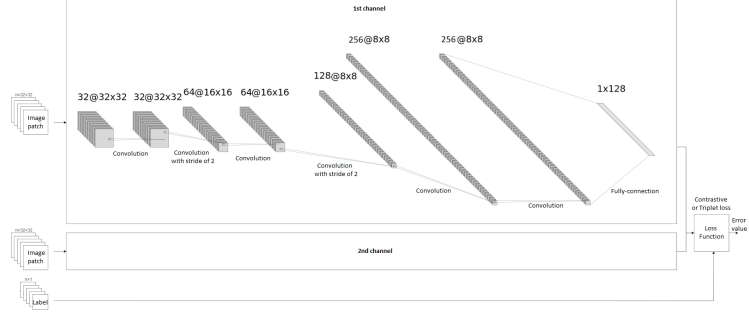


Fig. 2: Visualization of the architecture of the convolutional neural network used in experiments of this paper.

of Triplet loss function. For Triplet loss function we added another channel to our siamese neural network (3 overall) - 1st channel for anchor, 2nd channel for positive of the anchor and 3rd channel for negative of the anchor. In last experiment we enriched the Triplet loss function by the online hard mining. This approach allowed us to use only single channel. After each forward pass, the hardest positives and negatives have been chosen from the batch. Triplet loss is calculated just from the chosen (hardest) triplets. Hardness of descriptors is calculated by Euclidean distance and for each anchor is selected the closest negative and the farthest positive.

On the input of the network comes batches of 32x32 patches in grayscale. Each patch is normalized first, then go through the network and the output of the network is normalized by L2 normalization. Final output of the model is a decimal vector with size 128. We applied dropout at the penultimate convolutional layer and involved L2 regularization of the weights. Also we exponentially decayed the learning rate during the training. Batch size, learning rate and weights optimization method have been changed with respect to experiments and used loss functions.

As our network is composed of 8 layers, we noticed an occurrence of vanishing gradient during the training process. Therefore we decided to integrate a residual shortcut connection as recommended in He's research [8]. It allowed us to attach another convolutional layers with greater number of filters that increased network's potential to learn new features. We utilized a model similar to ResNet-50 [8] where we removed several convolutional and identity blocks (see Fig. 3). Removing of redundant blocks led to better generalization and better performance on HPatches validation. We kept input normalization layer and L2 normalization output layer. Kernels of all layers are initialized by Xavier uniform initializer. ResNet has been trained exclusively using the triplet loss with online hard mining that proved to be the more suitable for our task and dataset than Contrastive loss (see Sec. 8).

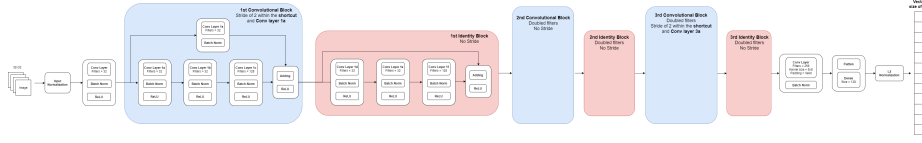


Fig. 3: Visualization of the architecture of the Residual neural network (ResNet) used in experiments of this paper.

8 Experiments

As we already introduced, we performed four the most significant experiments. Regardless the experiments oriented to the hyperparameter tuning, we experimented with loss functions and network architecture. Within the individual experiments, the network architecture have been slightly adjusted in compare to the reference architecture visualized on Fig. 2. In all experiments we tried to reach at least the same performance as SIFT.

8.1 Contrastive loss function

In this Contrastive loss experiment we involved the same architecture as described on Fig. 2. After the hyperparameter tuning we consider following as the best setup: learning rate - 0.1, batch size - 2048, optimizer - Momentum with beta equal to 0.9.

Results of the experiment are visualized on Fig. 4 (dotted lines). As we can see, the Contrastive loss achieved the worst performance in all tasks of HPatches benchmark. On the other side, its growth have been the least significant. We assume, more training iterations wouldn't increase the performance anyway, but network architecture adjustment is needed.

8.2 Naive triplet loss function

Since the naive triplet loss works with triplets of the descriptors - anchor, positive and negative, therefore we added another channel to the architecture described on Fig. 2. We chose following configuration of hyperparameters: batch size - 512, learning rate - 0.01, optimizer - adam.

Results of the experiment are shown on Fig. 4 (dashed lines). As we can see, naive implementation of triplet lost achieved an order of magnitude better performance than Contrastive loss in all metrics. In addition, it overcomes the triplet loss with hard mining in few metrics. As all metrics are still growing, we assume, more iterations of training can bring even better performance.

8.3 Triplet loss function with online hard mining

The best performance was achieved by the training using triplet loss with online hard mining. In this experiment we used just one channel of the siamese network

and we selected the hardest triplets online from the batch of output descriptors. It better prevents the overfitting and forces the network to better generalization. In this experiment we used following configuration: batch size - 64, learning rate - 0.001, optimizer - adam.

Results of the experiment are shown on Fig. 4 (dot-dashed line). In this experiment we trained the network on 1 mil. of iterations but we visualized only 500k. However, triplet loss with hard mining significantly overcame the naive triplet loss in two tasks of HPatches benchmark. We have great expectations from the triplet loss with hard mining and therefore we decided to proceed with the improvement of this approach. We assume, this approach can achieve comparable performance as competitive researches and highly overcame the classical approaches.

8.4 Residual network with triplet loss function with online hard mining

As in last experiment (described in Sec. 8.3) we found out the Triplet loss with online hard mining has great potential but we identified two bottlenecks. First one is small training set and second one is limitation of the network. As the current network hasn't enough capacity, we replaced it for ResNet model (see Fig. 3). Bottleneck of small training set has been resolved by merging two collections of Brown dataset [3] (Yosemite and Notre Dame), standardized with common mean and standard deviation. ResNet is trained with following hyperparameter setup: batch size - 128, learning rate - 0.01, optimizer - adam.

Results of the experiment are shown on Fig. 4 (solid line). We can see that ResNet significantly overcome general convolutional network from the experiment 8.3 in all tasks, especially in task of image matching, where we achieved double score. This approach not only overcomes our recent experiments, but also it overcome SIFT in several metrics (see Tab. 5). Another experiments, where we tried to manipulate with the depth and width of the ResNet, were unsuccessful.

9 Conclusion

In this paper we presented modern approach of the evaluation of the similarity of images. We experimented with the data preprocessing, used architecture of the neural network but also with different configuration of loss function. In this paper we introduced only those experiments that achieved the best performance on the HPatches benchmark. As the worst setup we consider two-channel siamese network trained with Contrastive loss function. Significant improvement came with involving of Triplet loss function. We experimented with simple, naive implementation but also we utilized online hard mining. As the Triplet loss function with online hard mining achieved the best results, we utilized it for the training of 21-layered Residual network that proved its dominance in this task and achieved the best results in this paper.

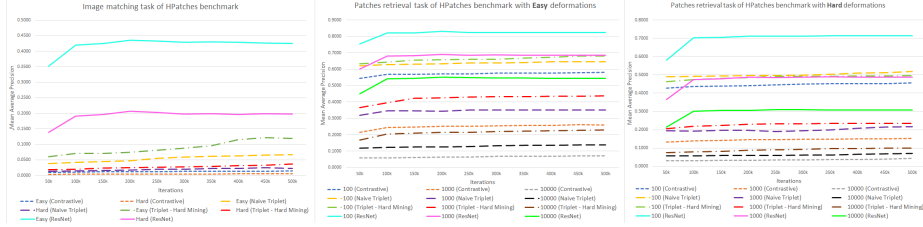


Fig. 4: Growth of HPatches metrics with respect to the training iterations. From the visualization we can see the best performance achieved ResNet with triplet loss with hard mining training in all cases (tasks). The worst performance achieved a training with Contrastive loss. Very slight growth is caused by limited chart scale. Easy and Hard denotes to the degree of deformation of the input patches. 100, 1000 and 10000 denotes to the noise level applied to the input patches.

	Image matching			Patches retrieval					
	Easy	Hard	Mean	Easy			Hard		
Contrastive loss	0.0142	0.0052	0.0073	0.5824	0.2611	0.0713	0.4551	0.1532	0.0429
Naive Triplet loss	0.0674	0.0256	0.0261	0.6471	0.3514	0.1392	0.5188	0.2171	0.0697
Triplet loss w/ hard mining	0.1211	0.0361	0.0517	0.6817	0.4371	0.2291	0.4951	0.2349	0.1002
ResNet	0.4356	0.2065	0.2438	0.8312	0.6913	0.5510	0.7136	0.4886	0.3097
SIFT	0.4532	0.1930	0.2442	0.8332	0.6979	0.5768	0.6238	0.4229	0.2926
HardNet	0.6814	0.5016	0.5007	0.9213	0.8477	0.7720	0.8785	0.7581	0.6360

Fig. 5: Comparison of achieved results for different setup of our network with competitive approaches SIFT [9] and HardNet [10].

Unfortunately, we were not capable to overcome current state-of-the-art like HardNet by Mishchuk [10] or LIFT [12] by Yi, but at least we overcome classical approaches like SIFT by Lowe [9] in tasks with hard deformation of input images (see Tab. 5). Better performance can be achieved by more training iterations, more detail hyperparameter tuning but mainly more thorough tuning of the network’s depth and width. Insufficient performance for image matching task of HPatches benchmark can be resolved by involvement of spatial transformer at the input of the network.

References

1. BALNTAS, V., LENC, K., VEDALDI, A., AND MIKOLAJCZYK, K. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors, 2017.
2. BAYA, H., ESSA, A., TUYTELAARSB, T., AND VAN GOOLA, L. Speeded-up robust features (surf). *Computer vision and image understanding* 110, 3 (2008), 346–359.
3. BROWN, M., AND LOWE, D. G. Automatic panoramic image stitching using invariant features. *International journal of computer vision* 74, 1 (2007), 59–73.
4. CALONDER, M., LEPETIT, V., STRECHA, C., AND FUA, P. Brief: Binary robust independent elementary features. In *European conference on computer vision* (2010), Springer, pp. 778–792.
5. CHEN, L., ROTTENSTEINER, F., AND HEIPKE, C. Invariant descriptor learning using a siamese convolutional neural network. In *XXIII ISPRS Congress, Commission III 3 (2016)*, Nr. 3 (2016), vol. 3, Göttingen: Copernicus GmbH, pp. 11–18.
6. CHOPRA, S., HADSELL, R., AND LECUN, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)* (2005), vol. 1, IEEE, pp. 539–546.
7. DETONE, D., MALISIEWICZ, T., AND RABINOVICH, A. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2018), pp. 224–236.
8. HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.
9. LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.
10. MISHCHUK, A., MISHKIN, D., RADENOVIC, F., AND MATAS, J. Working hard to know your neighbor’s margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems* (2017), pp. 4826–4837.
11. RUBLEE, E., RABAUD, V., KONOLIGE, K., AND BRADSKI, G. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision* (2011), Ieee, pp. 2564–2571.
12. YI, K. M., TRULLS, E., LEPETIT, V., AND FUA, P. Lift: Learned invariant feature transform. In *European Conference on Computer Vision* (2016), Springer, pp. 467–483.

Olive oil classification

Simona KLUČKOVÁ, Maroš KOLLÁR, Jakub KUČEČKA, Zuzana
POPOVCOVÁ, Mária RAJNÍKOVÁ, Alena VALOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovicova 2, 842 16 Bratislava, Slovakia
mikasa.fiit@gmail.com

1 Introduction

Olive oil is a fundamental component of the Mediterranean diet because of its nutritional values. It is also rich in antioxidants, and therefore consumption of olive oil can help prevent cellular damage caused by free radicals. However, not all olive oils are equally beneficial for the human body because they do not contain the same amount of antioxidants and nutrients [1].

The quality of olive oil depends on the technological processes of its production and natural factors, including the type of soil and its composition or climate.

According to the regulations from European Union standards, there are three categories of olive oil, "extra virgin" (EVOO), "virgin" (VOO), and "lampante" (LOO) [2]. The quality of olive oils is evaluated by a panel test, which assesses taste, appearance, and aroma. In the panel test, the experts assign a score to the oils. Subsequently, the statistical analysis is applied to the score, which the experts gave to the sample. Finally, they classify olive oil into one of three classes. It is a demanding, expensive, and time-consuming task, so there is a need to automate this process.

We propose the system for automatic spectrum recognition, which is using the methods of machine learning. In comparison to the other approaches, the application can also visualize the olive oil samples.

2 Related works

The authors of *Quality control of olive oils using machine learning and electronic nose* [3] dealt with the recognition of olive oil authenticity. They used standard classification methods for identifying the olive oil and falsification, for instance, sunflower oil. The accuracy of the proposed method was in the range of 56-70%, and the best results were achieved by classifier Naive Bayes.

In the *Laser-based classification of olive oils assisted by machine learning* [4], the

* Master study programme in field: Intelligent Software Systems

Supervisor: Ing. Marta Šoltésová Prnová, PhD., Mgr. Martin Sabo, PhD., Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

authors obtained the data with Laser-Induced Breakdown Spectroscopy (LIBS). They chose to combine LIBS and machine learning algorithms, specifically SVM, LDA, and random forest, to classify olive oil and its origin. According to them, it was used for the first time, and they achieved competitive results. The results of the used classification methods were in the range of 90 and 99.2%.

3 Our approach

The main goal of our project is to create a system to analyze the quality of olive oils. To achieve that goal, we propose the architecture of the application named ASR for automatic spectrum recognition, which can process data from AIMS (Advanced Ion Mobility Spectrometer). The AIMS allows us to analyze not only gaseous substances but also liquid and solid substances. The application is developed in cooperation with research company MaSaTech, s.r.o¹, which provides the data from the spectrometer. The proposed application has three main functions, i.e., data processing, classification, and data visualization. The system architecture is shown in Figure 1.

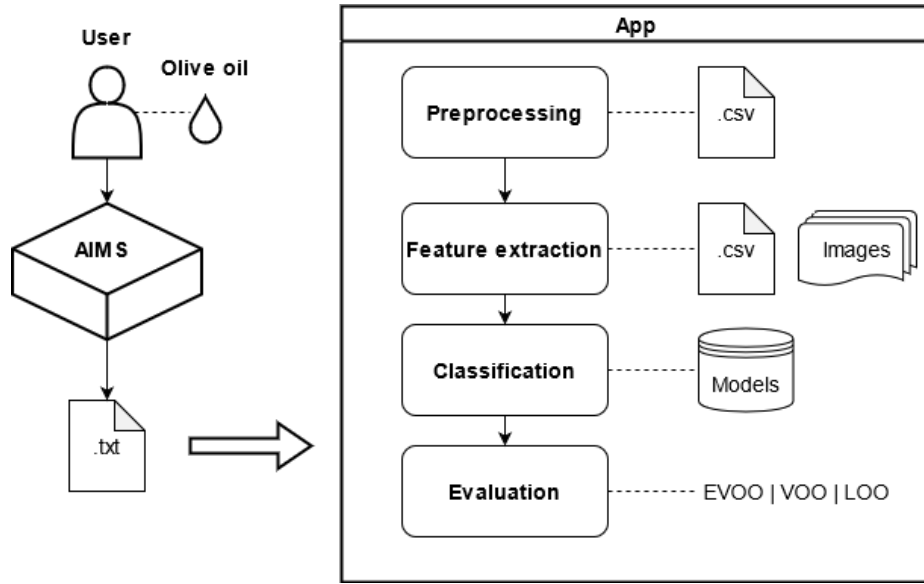


Fig. 1. System architecture.

The trigger in the application is loading text data with measured intensities by a user. Main features of our application includes:

¹ <https://www.masatech.eu/>

- load text data generated by a spectrometer,
- process data,
- show images from data,
- evaluate the class of the sample with one or multiple classification models,
- train new models,
- save images and classification report.

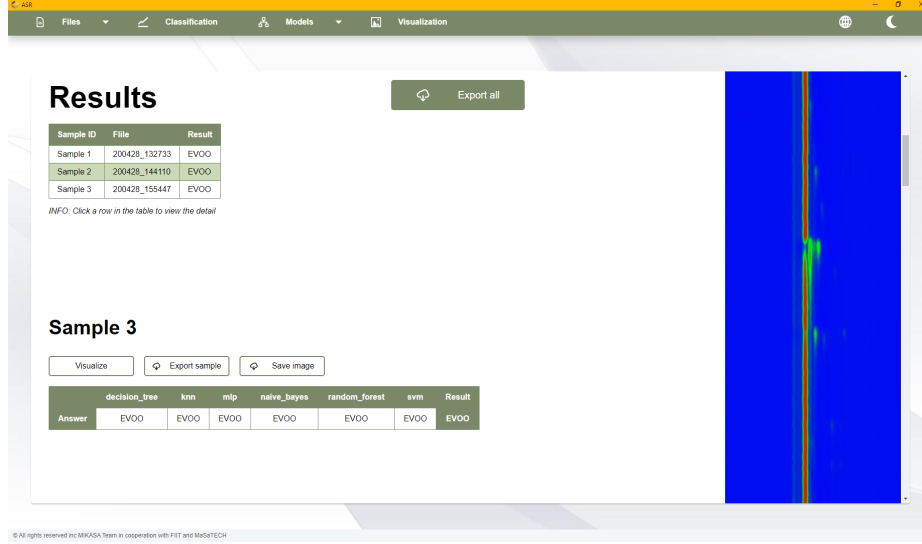


Fig. 2. Application screen preview - evaluation view.

3.1 Data

Data used in our application are from chemical analysis of olive oils performed by an ion mobility spectrometer. Spectrometer records drift field intensity, pressure, temperature, drift tube length, analysis time, the time when the intensity was measured, and intensities values.

The data were classified in two panel tests, which have an accuracy of classification around 80%. Eventually, only data that had the same predicted class in both panel tests were selected. Therefore, the data has an accuracy of 96%, which we computed using Formula 1.

$$data_accuracy = 1 - (panel1_error * panel2_error) \quad (1)$$

The panel_error was computed as a complement of panel accuracy (Formula 2).

$$panel_error = 1 - panel_accuracy \quad (2)$$

The original dataset consists of 400 samples, which belong to three olive oil grades in a ratio of 2:1:1 (EVOO: VOO: LOO).

3.2 Extracting the features

To achieve more accurate results, we decided to train our models on computed features from input data. We calculate some relevant information from the original data:

- the average for each oil,
- the average value for each column in a given oil sample,
- the maximum for each column in a given oil sample,
- the minimum for each column in a given oil sample,
- the median for each column in a given oil sample,
- the standard deviation for each column in a given oil sample,
- delta for each column in a given oil sample (maximum-minimum),
- the sum of the values for each column in a given oil sample.

In addition to these standard features, we came up with an idea to extract differential maps and point features. The differential maps represent the difference between the oil sample and created a template for every specific oil grade. Template for each oil is generated from average in values enabling calculation of difference from all types. Point feature is computed as the mean and max value from significant areas, which we find during the analysis of differential maps.

We transform original data into images, which we use in an experiment with a convolutional neural network. Furthermore, the images can be displayed by users in the application for better visualization of data.

The feature selection is applied to created features because of the amount of new data. Thus, the proposed method is trained only with 500 calculated features, which are significant for the classification of an oil category. The extraction algorithm evaluates the accuracy of random forest on the selected features. Subsequently, the data on which the model has achieved the best success are selected. A smaller subset of features speeds up the calculating process while preserving the accuracy of classifiers.

3.3 Classification

In the classification phase, we decided to use these algorithms:

- k-nearest neighbors (KNN),
- support vector machine (SVM),
- decision tree (DT),
- random forest (RF),
- naive bayes (NB),
- multilayer perceptron (MLP),
- VGG-16.

The classifiers were chosen based on similar works, where they achieved excellent results in the field. The best hyperparameters for each classifier are selected by the grid search algorithm. Our application provides one trained model for each classifier and also option for training new models. To combine predictions

of multiple classifiers, we selected category with highest score. This score is calculated as sum of prediction weight for each model, represented in Formula 3.

$$weight = \sum_m^{models} (accuracy[type(m)] - 0.5)^4 \quad (3)$$

We evaluated accuracy for all six classifier types, getting values in the range 0.6-0.9. To calculate weight, we select accuracy depending on classifier type, subtract 0.5 for values closer to zero. This value was powered by 4 to make differences more significant.

4 Results

We evaluate every model individually on test data. Training and test datasets are acquired by splitting original data in a ratio of 4:1 for every class to provide even distribution. We choose two approaches for the classification of olive oils. In the first one, we predict one of three categories of olive oils. From all classification models, the random forest achieved the best results with an accuracy of 80.95%. We obtain these results on data with 96% accuracy, as we mentioned in Section 3.1. In comparison to the panel tests with an accuracy of 80%, we achieved better results. This approach has one significant problem, specifically unbalanced classes. As we mentioned earlier, the original data contains three different olive oil categories. The dataset includes twice more extra virgin oil samples than virgin oil or lampante oil samples.

Therefore, we came up with the idea of firstly predicting if the olive oil belongs to the extra virgin olive oil, the highest quality grade. In case the sample is not extra virgin oil, it is classified into the remaining two classes. We get an evenly distributed dataset by combining virgin oil and lampante oil samples. Using this method, we achieved better results, which are in Table 1. The random forest achieved the best results with an accuracy of 89.29%. For classification between VOO and LOO, we decided to use the random forest because it gained the best results. The trained random forest achieved 82.14% accuracy in predicting virgin oil and lampante oil.

Table 1. The accuracy of classifiers predicting oil category for EVOO and combined classes of VOO and LOO. The second row includes results of accuracy after firstly predicting EVOO, and subsequently predicting VOO and LOO with random forest classifier.

	Decision tree	k-NN	MLP	Naive Bayes	Random forest	SVM	Total
EVOO vs combined	73.81%	70.24%	66.67%	70.24%	89.29%	84.52%	89.29%
EVOO vs VOO vs LOO	67.86%	65.48%	60.71%	66.67%	84.52%	77.38%	82.14%

5 Discussion

We proposed a solution of classification by using traditional classifiers and neural networks. The classification models were trained on selected features from original data. We chose features based on the accuracy of the random forest because this algorithm achieves the best results from the beginning. Simultaneously, it can provide information about data that was used for the training process.

The results indicate that the random forest classifier is most successful in categorizing data. On the other hand, MLP and VGG-16 did not obtain competitive results compared to standard classifiers. The lower success rate of MLP may be due to the small number of samples. The multilayer perceptron achieves better accuracy on numerous data because it can better generalize in the learning process. The accuracy of VGG-16 could be affected by the spatial reduction, which could lead to information loss. The original image size (approximately 10000x448px) had to be reduced due to hardware limitations.

The biggest problem we had to deal with is that we don't have 100% correct data labels. Therefore, our classifiers can predict the correct oil grade, but the sample was labeled erroneously with a panel test, which can affect the resulting accuracy. In our application, we noticed cases when all classifiers predicted the same class with high confidence, but the panel test classified different oil type. We believe the classifiers could find some pattern in samples, which could not be discovered by panel test.

The prediction if the olive oil belongs to the extra virgin olive oil class improves the accuracy of the proposed models. The fact that we gain two evenly distributed groups, which facilitates finding significant differences between the olive oil classes. The correct prediction of extra virgin olive oil is also crucial for business purposes. This class has the highest quality because it contains a lot of antioxidants and nutritions.

6 Conclusion

Nowadays, the panel tests for olive oil classification are expensive and time-consuming because the oil is assessed by 10-20 panelists. In contrary to the panel test, the proposed application provides a more accurate, faster, and cheaper way to predict the olive oil grade. Our solution includes not only the precise classification method but also a user-friendly and easy-to-use application. The application allows users to visualize the oil samples as images for a better understanding of measured data. The users also can train new models.

In future work, we want to devote improving the accuracy of proposed models. We believe that there are some possibilities to give other classification methods a try. The convolutional neural network could be used not only for classification but also for feature extraction, which can improve the accuracy of this approach. Simultaneously, there is an opportunity for scaling up an application for use in other areas of spectrum recognition, such as wine classification.

References

1. Gaforio JJ, Visioli F, Alarcón-de-la-Lastra C, et al. Virgin Olive Oil and Health: Summary of the III International Conference on Virgin Olive Oil and Health Consensus Report, JAEN (Spain) 2018. *Nutrients*. 2019;11(9):2039. Published 2019 Sep 1. doi:10.3390/nu11092039
2. Beatriz Quintanilla-Casas, Julen Bustamante, Francesc Guardiola, Diego LuísGarcía-González, Sara Barbieri, Alessandra Bendini, Tullia Gallina Toschi,Stefania Vichi, and Alba Tres. Virgin olive oil volatile fingerprint and chemo-metrics: Towards an instrumental screening tool to grade the sensory quality.*LWT*, 121:108936, 2020
3. Emre Ordukaya and Bekir Karlik. Quality control of olive oils using machine learning and electronic nose.*Journal of Food Quality*, 2017:1–7, 10 2017.
4. Odhisea Gazeli, Elli Bellou, Dimitrios Stefas, and Stelios Couris. Laser-based classification of olive oils assisted by machine learning.*Food Chemistry*,302:125329, 20220.

Overview of Some Windows Server Security Features in Accordance to ISO/IEC 27002 – System Operation Management

Martin PAVELKA^{*}, MTA[†]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava
pavelkax@gmail.com

Abstract. This paper focuses on several techniques and tools contained in Windows Server 2019 that might be used to implement organization's compliance with ISO/IEC 27002 standard. There are many tools and sources available that assist companies to fulfill security requirements, but there is a gap in the information sources about implementing ISO/IEC 27002 standard in recent versions of Windows Server 2019 and Windows 10 20H2 (or newer). Family of Windows operating systems contains a lot of built-in or easily addable tools which might be used to address counter measured areas required in ISO/IEC27002 standard. Lack of implementing best practices in system operation management of Windows and Active Directory might result in complications in performed audit procedure when trying to achieve ISO/IEC27002 certification. The main goal of this paper is to summarize the state of the art and introduce our developed tool for automated securing Microsoft Windows/Server environment.

Keywords: System Operation Management, Active Directory, Windows Server, Domain Controller, GPO, Integrity, Availability, Information Security, Compliance, Countermeasure

Introduction

In today's world, many companies use several techniques to eliminate risks involved in operating systems technology. Security managers and local IT departments might have developed system operation management procedures that cover the most vulnerable procedures and parts of their IT infrastructure. Trying to meet security and legal requirements, companies are giving a huge effort to implement an Information Security Management System and to gain some kind of certification. Recent Decree of the Office of the Deputy Prime Minister for Investment and Informatization No. 179 from

^{*} Bachelor study programme in field: Informatics – Information Security.
Supervisor: Mgr. Peter Lehoczky, Institute of Computer Engineering and Applied Informatics,
Faculty of Informatics and Information Technologies STU in Bratislava

[†] MTA – Microsoft Technology Associate, certification

June 2020 [5] sets out the obligations for public administration institutions on ways to secure assets and prevent security threats. Not only this decree but also other legal regulations are based on setting international standards from the international standards of the ISO 27000 series. Our thesis deals with the analysis of the ISO/IEC 27001 [1] and 27002 [2] standards with regard to its implementation in the enterprise environment using Windows Active Directory technology and other. We will focus on the selection of some provisions of the ISO27001 standard and its subsequent implementation in the enterprise environment using available tools integrated in the Windows operating system or easily installable components.

1 Identification of ISO/IEC27001 controls in the context of the Windows operating system

The first activity involved in building an information security management system is usually a proper risk analysis and threat identification [3]. The purpose of this step is to provide a consistent risk management framework in which the risks will be identified, considered and reviewed. Each risk should be classified not only by severity and probability. A description of the impact that the risk event will have on the organization should be included.

We have studied ISO/IEC27001 with Annex A and with accordance to minimum viable scope[‡] [4] selected the following controls and risks that might be addressed in our Windows/Server/Active Directory countermeasures. Risk contained in the last column is subjectively described and will vary depending on company size, branch of business, type of employees (programmers, accountants), etc. Therefore, a numerical assessment of the probability of risks is irrelevant in this case.

Table 1. ISO/IEC27001 controls for Windows Environment

Annex A reference	Control Title	Risk
A8.1.1	Inventory of assets	Lack of knowledge about location of assets; Losing CIA [§] attributes of assets.
A8.1.3	Acceptable use of assets	Users unaware of this policy cannot be punished in disciplinary procedure.
A9.1.1	Access control policy	User might gain access to assets or information without legitimate reason

[‡] Minimum viable scope is the minimum number of documents required by the certification body when trying to gain ISO/IEC27001 certification.

[§] CIA stands for confidentiality, integrity and availability as the basic principles of information security

A9.1.2	Access to networks and network services	Users might gain access to network/services that they have not been explicitly granted.
A9.2.1	User registration and de-registration	Users might use their accounts without legitimate reason, i.e. during maternity leave
A9.2.4	Management of secret authentication information of users	Administrator or other user might have knowledge of universal, initial password for newly created user – masquerade attack might take place.
A9.2.5	Review of user access rights	User accounts might resist a long time after the person left the company or their role has changed respectfully.
A9.2.6	Removal or adjustment of access rights	User might be granted more permission than needed to perform their role.
A9.4.2	Secure log-on procedures	Improper handling user credential during unsafe log-on procedure might result into breach of credentials or identification of user.
A9.4.3	Password management systems	Using shared account might result into loss of accountability – difficulties when performing security evidence incident collection
A9.4.4	Use of privileged utility programs	Users might change default, standardized operating system environment and expose devices/systems/information to public network or expose overall risk
A12.1.1	Documented operating procedures	Users, especially IT management might use wrong procedure when dealing with some situation/problem
A12.4.1	Event logging	Lack of awareness about system malfunction – errors or security incidents – unauthorized access.
A12.4.3	Administrator and operator log	Administrators might be able to destroy evidence needed when performing security incident inspection
A15.1.1	Information security policy for supplier relationships	External parties might be granted greater access rights than needed to perform contracted operations

2 Risk treatment

We have used best practices described in ISO/IEC27002 to mitigate identified risks. Our goal is to provide the kind of countermeasures that will not require commercial, proprietary solutions but will rely on integrated parts of Windows operating system or freely available Microsoft tools. Described countermeasures are valid for recent versions of Windows 10 20H2 and Windows Server 2019. The following table contains the main tools, that we will use to establish adequate countermeasures and controls according to Table 1.

Table 2. Tools for implementing countermeasures in Microsoft Windows Environment

Tool	Description - usage
Active Directory Users and Computers	User Accounts Management. User accounts creation, deletions, user groups and organizing into hierarchy (organizational units). Disabling user accounts.
Local Group Policy	Defining settings for standalone workstation. Complex tool for customizing policies, options of integrated parts of Windows Operating system. Applies on the local computer only, no synchronization with central server or other tools.
Group Policy Management	Controlling standardized environment in organization's domain. More complex settings than Local Group Policy, options for automatic software installations, defining policies based on users and computers in organizational units, using hierarchy of inheritance.
Folder Properties View – ACL	Maintaining proper access rights for information sources, folders and files. Addressing several layers of access to exact users, user groups for many kinds of operations (standard operations such as read/write, extended operations such as metadata writing)
Sysinternals Autologon	Utility for automated logon. Secure way to enable automatic user account logon on computer startup. Might be used in special occasions – for example kiosk in warehouse.

The following sub-chapters provide some advices that should be considered when trying to implement referenced Annex A controls in place, mentioned in table 1. Described suggestions are based on best practices, generally described in ISO/IEC27002.

2.1 Inventory of assets

Active Directory Users and Computers tool might be used to create a hierarchy of organizational units based on geographical locations of devices or user accounts. This hierarchy might be extended into device categories (computers, servers, virtual servers, groups, etc.). Special attention should be paid to maintaining a list of disposed or inactive devices (for example old desktop computer) and user accounts – these must be disabled and password for them must be changed after user account is no more needed. Printers might be centrally installed on dedicated server – print server, from which printers are deployed to end user's computers.

2.2 Acceptable use of assets

Group Policy setting for displaying warning message at user log-on might be used to inform users about organization's information security policies and regulations that they should be aware of

2.3 Access control policy

This overall policy should be based on principle “what is not allowed is denied”. Employees should have dedicated accounts according to their working roles. This is especially important for people with cumulative functions (for example HR and payroll). There should be formal procedure for IT department for handling access requests when managing user accounts. IT should use proper classification of user account and its membership in Active Directory organizational unit and security groups.

2.4 Access to networks and network services

Organization's services, for example shared folders must be accessible via proper network only. It is unacceptable to have file server exposed to guest wireless network. There should be a described way explaining how to work with these services and working instructions for common situations – for example accessing shared folder from remote network, VPN, etc. Windows Firewall policies might be used for controlling network traffic combined with hardware-based firewalls, VLANs and network segmentation.

2.5 User registration and de-registration

Each user should use their own user account(s) with unique password, that only they have knowledge of. IT department must take measures when blocking, deleting and maintaining user accounts, especially if the employee is in the job termination process. Active Directory user *Disabled* attribute might be used, combined with proper classification of user account into other domain organizational unit for disabled accounts. Password of such account might be changed to a different one, because in case of accidental re-enabling account, the employee will not be able to log in.

2.6 Management of secret authentication information of users

Along with operational procedure for keeping user authentication information, especially password confidential, it is needed to avoid using old, weak password hashing techniques. Attribute *Store password using reversible encryption* should not be used in user account properties, combined with proper Group Policy setting regarding password hashing. Older forms of hashing are reversibly engineered and an adversary is able to gain the plain-text form of user account passwords.

Control for changing new employee's password should be used when creating new user account. Attribute *User must change password at next logon* should be used.

2.7 Review of user access rights

NTFS (file system) permission mapper utilities (third party executables or scripts) might be used to generate report of folder permissions on file servers.

PowerShell scripts and snippets might be used to grab list of users inactive in the last 30 days.

Formal process and guidelines for IT department should be created and used. Described tasks might be fully-automated, so that administrators might get monthly email report about described artifacts.

2.8 Removal or adjustment of user access rights

Countermeasures from 2.7 and 2.5 apply.

2.9 Secure log-on procedure

According to workstation and information classification (for example "confidential workstation" for processing personal data – working contracts) there should be procedure created for secure log-on. Standard method is using user account name and password, more secure methods such as USB-token and smart-card might be used, if the classification of information is high. In Group Policy, there should be CTRL+ALT+Del policy set as mandatory for all user accounts. If there is a need for automated, unattended login (such as kiosk – attendance system), utility Sysinternals Autologon (free available) might be used. This utility is able to bypass CTRL+ALT+Del sequence and perform automatic logon. Password is saved on local computer, encrypted properly.

For login to information system, single sign on should be implemented using Active Directory user accounts and permission. LDAP protocol might be used.

2.10 Password management system

Regular users should be able to change their password according to password complexity rules. Password expirations should be carefully considered, because more

often the password must be changed, the higher probability of writing the password down is. Each user must use their own account and password when applicable. Special IT accounts, such as disaster domain admin account password should be generated in two parts, by two people and these parts should be stored in a written form, securely saved in organization's safe.

Group Policy for password complexity should be used, combining with user security training to avoid weak password. MS Windows does not provide proactive password checking by default. If needed, external third-party solutions might be used.

2.11 Use of privileged utility programs

Group Policy must be set for the *User Account Control* functionality as enabled. Users should work using standard user account, without admin privilege. If the kind of user's task should need the administrator's rights, there should be a special, dedicated local user account to switch and perform the necessary task there.

By default, utilities that change operating system's behavior are not available to regular users with standard permissions. Active Directory sets new user account as regular *User*.

2.12 Event logging

Windows Event Log saves all events locally, meaning in case of computer hard drive failure or malicious activities, needed event log will not be available. This issue needs to be addressed by using some kind of external logging functionality or using script for automated Event Log backup to external location. Group Policy might be used for enlarging size of Event Log and for forwarding logs to external server. This also applies for Administrator and Operator Logs that must be saved to different server by default, as administrators are able to delete the whole local Event Log.

3 Semi-automated approach when addressing risk mitigation

Security Administrators in enterprise environment are usually using integrated parts of Operating System to reach the established goal of secure Active Directory Environment. More experienced Administrators can use some scripts developed in PowerShell or Batch to simplify their work and reduce time-spent during performing routine tasks.

There are some commercial products or product suites available focusing on several ISO/IEC27002 controls, for example ADAudit [8] or Qualys [9] tool for Event logging and endpoint device controlling. These tools might be considered as ready-made solutions covering some of the identified controls/areas. From our experience, we can confirm that they are used primarily in corporate environments and in larger domains built on Active Directory technology. Their use in smaller domains is also questionable given the cost of these programs.

Our bachelor thesis, “A comprehensive tool for automated security of the operating system in a corporate environment”, describes in more detail some of the examined countermeasures and provides their automated or semi-automated implementation in the form of group policies in the Windows Server operating system.

Topics covered in our thesis contain:

- Secure methods of user login, login policy and password protection,
- System services, their role in operating system, management of system services accounts in corporate environment,
- Scheduled tasks as integrated part of operating system
- Group policies options for overall security countermeasures

These topics were selected upon our research. We have studied several materials and guidelines regarding Windows Server, Active Directory and Windows 10 security.

Some of the studied materials contain best-practices and detailed description for implementation of countermeasures. Especially in *Hardening Microsoft Windows 10 version 1909 Workstations* guide [6] from Australia Cyber Security Centre, many advanced features and tweaks are provided. These are more complex and in-depth advices than general advices in ISO/IEC 27001 Annex A or ISO/IEC 27002 provisions. Publication *Active Directory and Related Aspects of Security* from Saudi Arabia [7] generally describes Active Directory environment and common vulnerabilities and security incident scenarios.

We have examined in more detail the protection of login passwords, integrated operating system services, scheduled tasks and overall security of a domain based on Microsoft Active directory technology. At the end of our research, we found that the described materials [6] [7] deeply discuss the implementation of countermeasures, which means that they can be used as specific instructions in business practice, administrators can follow exactly them. For comparison, ISO standards represent general procedures and conceptually approach for several areas of information security but do not provide specific instructions for solving security issues in operating systems and therefore it is necessary for security administrators to study enough other information sources that will provide a detailed procedure on how to implement countermeasures in specific cases of enterprise environment.

Part of our bachelor thesis is implementation of tool in a PowerShell environment that allows enterprise security administrators to automatically apply measures to reduce risk and increase overall enterprise Active Directory Domain security. Not only ISO/IEC 27002 guidelines but also studied materials [6] [7] provide deeper overview of security countermeasures in Microsoft Windows environment. In our tool, we have covered login password countermeasures, scheduled tasks and configuring system services. Many of these require in-depth knowledge of operating system configuration, enterprise environment, applications used, and therefore we cannot provide universal guidelines for setting enterprise Windows environment in accordance with ISO27001 standards.

Creation of ready-made solution is not the goal of our bachelor thesis nor of this paper. This paper and concept of our solution of “A comprehensive tool for automated security of the operating system in a corporate environment” might be a good start-

ing point for deeper analysis and development of such an in-depth tool covering most of the Windows/Server environment in accordance to ISO/IEC 27001 and 27002 controls.

Conclusion

We have analyzed ISO/IEC 27001 Annex A regulations regarding system operations management. Based on best practices from ISO/IEC 27002 we have created a list of controls, which are addressable and relevant for Windows operating system environment and provided countermeasures based on default Windows integrated or easily installable tools. We have developed our own tool, which might partially help system security managers to deploy some policies and countermeasures in an easier way. This tool covers only some of the ISO/IEC controls and does not go in depth defense practices. We have developed an idea how to start thinking about implementation of ISO/IEC 27001 in enterprise Windows environment. A deeper, detailed guide or product might be created based on our initial research. Enterprise administrators might use our conclusions as a first-step for building secured, ISO/IEC 27001 compliant Windows environment.

References

1. ISO/IEC 27001:2013, Information technology — Security Techniques — Information Security Management System - Requirements
2. ISO/IEC 27002:2013, Information technology — Security Techniques — Code of practice for information security controls
3. Hudec, L.: Manažment informačnej bezpečnosti. In: Vzdelávanie v oblasti informačnej bezpečnosti – prednášky, <https://www.csirt.gov.sk/bezpecnostna-studovna/mfsr-vzdelavanie-89f.html> .(2014)
4. Kosutic, D.: List of mandatory documents required by ISO 27001 (2013 revision). In: Advisera, <https://advisera.com/27001academy/knowledgebase/list-of-mandatory-documents-required-by-iso-27001-2013-revision/>
5. VYHLÁŠKA Úradu podpredsedu vlády Slovenskej republiky pre investície a informatizáciu z. 22. júna 2020, ktorou sa ustanovuje spôsob kategorizácie a obsah bezpečnostných opatrení informačných technológií verejnej správy, <https://www.slovlex.sk/pravne-predpisy/SK/ZZ/2020/179/20200630> (2021)
6. A. Binduf: Active Directory and Related Aspects of Security, 2018 In: 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, 2018, pp. 4474-4479, doi: 10.1109/NCG.2018.8593188.
7. Hardening Microsoft Windows 10 version 1909 Workstations. In Cy-ber.gov.au, <https://www.cyber.gov.au/acsc/view-all-content/publications/hardening-microsoft-windows-10-version-1909-workstations>
8. ADAudit – Real-Time Auditing for Active Directory. In ManageEngine.com, <https://www.manageengine.com/products/active-directory-audit/>
9. Qualys Cloud Agent for Windows. In Qualys.com, <https://www.qualys.com/docs/qualys-cloud-agent-windows-install-guide.pdf>

Analysis of market data using technical indicators and decision trees

Róbert Gers*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xgers@stuba.sk

Abstract. The subject of this work is to analyze the possibilities of automating financial market decision making, compare them to manual analysis methods and introduce basic concepts used to analyze financial data, including technical indicators, oscillators and classification and regression trees. This work focuses specifically on cryptocurrency markets, but the proposed concepts should, to a certain degree, apply to other financial markets. The analysis of market data is traditionally a manual process, using a combination of methods including manual data analysis and fundamental analysis (financial reports, mentions in news sources, etc.). However, given the abundance of freely available, reliable, real-time data from cryptocurrency markets, more methodical, data-driven approaches may be viable.

Keywords: Data Analysis, Financial Data, Cryptocurrency, CART.

1 Introduction

The analysis of financial markets is traditionally a manual process, where analysts utilize a combination of technical and fundamental analysis methods. Fundamental analysis uses external sources of information, like public news sources, financial reports, or public attitude, for instance. With the emergence of computerized trading platforms, however, technical analysis, which uses past and current market data as its main point of reference, has become equally prominent[1][2]. Technical analysis provides an opportunity to explore the possibility of automating trading behaviour since the data it requires is abundant, very precise, reliable, and freely available[3].

* Bachelor study programme in field: Informatics

Supervisor: Vladimír Kunštár, Institute of Computer Engineering and Applied Informatics,
Faculty of Informatics and Information Technologies STU in Bratislava

2 Related work

In the field of financial market prediction, technical analysis has been the focus of a lot of work [1][2][4], mostly focusing on manual technical analysis methods of the stock market.

When it comes to computerized technical analysis, various approaches have been introduced, for instance, focusing on pixel-based visual analysis [5], or portfolio visual analysis [6]. There is also some work studying possible optimization of technical indicators [7] or studying the actors of a stock market using a network approach [8].

In a similar approach to the one described in this article, L. Khaidem, S. Saha, and S. R. Dey [9] have used an ensemble learning method – random forest, which essentially combines multiple decision trees in a single model – to form relatively accurate predictions on the stock market for several stocks.

3 Proposed solution

Financial market data history can be represented as an array of objects. The continuous time dimension is therefore divided into discrete intervals, where each object in the array contains the data describing a single interval. It contains 5 attributes:

- Open – price at the start of the interval
- High – maximum price during the interval
- Low – minimum price during the interval
- Close – price at the end of the interval
- Volume – total volume of assets transferred during the interval

These objects are often referred to as candles because they are the direct representation of the candlestick chart, which is often used in financial market visualization charts. This data model, despite being very precise and exhaustive in the way it describes past market events, is not optimal for a machine learning model. A time interval has no value in predicting the future market changes by itself but instead needs to be evaluated relative to its predecessors. This aspect is solved by a concept in technical analysis – technical indicators and oscillators.

3.1 Technical indicators and oscillators

A technical indicator is essentially a mathematical function, which transforms an array of candles into an array of numbers, where each element represents the candle at its point in time, with regards to several of its predecessors. A common example of a technical indicator is a moving average, used widely even outside the field of financial analysis. The moving average calculates an average of closing prices with a set number of its predecessors (14 previous candles for instance), therefore providing us with an interpretation of the candle's value relative to other candles in its proximity, which is much more relevant for a machine learning model than just an isolated candle.

An oscillator is also a mathematical function, but in contrast to an indicator that preserves the scale of the data source, oscillator's results are on a scale of its own, usually oscillating around 0, or between the values of 0 and 100. A commonly used oscillator is, for instance, the stochastic oscillator. It compares the current closing price with a number of past closing and opening prices (usually 14). Its results are in a range between 0 and 100. The stochastic oscillator[9] is calculated by the following formula:

$$\frac{close - Ln}{Hn - Ln} \times 100 \quad (1)$$

Where *close* is the closing price of the current candle, *Ln* is the lowest low over the past *n* intervals and *Hn* is the highest high over the past *n* intervals.

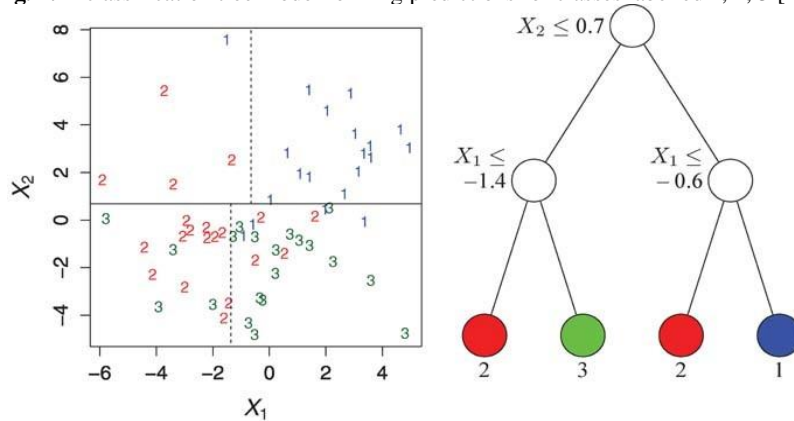
3.2 Decision making

Now that we have established a way to efficiently reduce a discrete time-based series into simple numerical values, let us talk about how to interpret them to motivate trading decisions.

Traditional technical analysis is a very non-deterministic process, where each analyst employs their own strategy and revises it over time, and for specific markets and assets. This is directly caused by the nature of the field [10], where no general ultimate strategy can exist because once a critical mass of trades is traded using the same strategy, or even similar strategies using similar methods, it becomes increasingly less profitable.

A machine learning model using decision trees is well suited for this task because it is designed to be used specifically in problems where a solution cannot be reasonably found using traditional methods but instead focuses on finding a solution with as little error as possible. Classification and regression trees (CART) are machine learning methods for constructing prediction models from data. The models are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition [11].

Fig. 1. A classification tree model forming predictions for classes labelled 1, 2, 3 [11]



Finally, for any machine learning model to be able to formulate effective predictions, an accurate and well-formed dataset is vitally important. We have already explored how we convert time into a discrete set of time intervals, and how to reduce market data into numerical values taking into account values in its time proximity. The only thing that is left, is to mark these data points, in order to establish whether they lead to a reasonably distant future price change. This can be achieved by calculating the difference between the current price and a weighted moving average [12], similar to the indicator described in chapter 3.1, but taking into account values from the future relative to the current time interval. The weighted moving average also differs from a regular moving average by assigning a weight to each time interval, allowing us to give more importance to values in close proximity.

4 Conclusion

Trying to computerize the identification of trends of cryptocurrency markets in an attempt to predict future price movements is a difficult task given the non-deterministic and volatile nature of the market. Traditional algorithms usually fall short of providing a reliable solution for following market trends and analysts are left to devise their own strategies based on, in part, some of the technical analysis methods described in this article. However, given the amount of freely available data accurately representing all past market events we have access to, it is worth pursuing the creation of a machine learning model suited for this task.

This article described the structure of market data, why it is not directly suitable to be used in a machine learning model and explored the use of technical indicators and oscillators to transform this data into a more meaningful format. It also went through a method of labelling source data with the following price changes and a method for classifying this data – classification and regression trees.

In future work, it is advised to assess different indicators and oscillators based on their effectivity, collect a reasonably large dataset and train a model in order to compare its predicting performance with actual market price changes.

References

- [1] S. J. Brown, W. N. Goetzmann, and A. Kumar, “The Dow Theory: William Peter Hamilton’s Track Record Reconsidered,” *J. Finance*, vol. 53, no. 4, pp. 1311–1333, 1998, doi: 10.1111/0022-1082.00054.
- [2] C. Dow, *Dow Theory Unplugged: Charles Dow’s Original Editorials & Their Relevance Today*. Traders Press, 2009.
- [3] “Coinbase Pro | API Reference.” <https://docs.pro.coinbase.com/#introduction> (accessed Dec. 29, 2020).
- [4] R. J. Bauer and J. R. Dahlquist, *Technical Markets Indicators: Analysis & Performance*, vol. 64. John Wiley & Sons, 1998.
- [5] H. Ziegler, T. Nietzsche, and D. A. Keim, “Visual Analytics on the Financial Market: Pixel-based Analysis and Comparison of Long-Term Investments,” in *2008*

- 12th International Conference Information Visualisation*, Jul. 2008, pp. 287–295, doi: 10.1109/IV.2008.80.
- [6] X. Yue *et al.*, “sPortfolio: Stratified Visual Analysis of Stock Portfolios,” *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 601–610, Jan. 2020, doi: 10.1109/TVCG.2019.2934660.
 - [7] K. Bakanov, I. Spence, and H. Vandierendonck, “Stream-Based Representation and Incremental optimization of Technical Market Indicators,” in *2019 International Conference on High Performance Computing Simulation (HPCS)*, Jul. 2019, pp. 833–841, doi: 10.1109/HPCS48598.2019.9188212.
 - [8] S. George and M. Changat, “Network approach for stock market data mining and portfolio analysis,” in *2017 International Conference on Networks Advances in Computational Technologies (NetACT)*, Jul. 2017, pp. 251–256, doi: 10.1109/NETACT.2017.8076775.
 - [9] L. Khaidem, S. Saha, and S. R. Dey, “Predicting the direction of stock market prices using random forest,” *arXiv Prepr. arXiv1605.00003*, 2016.
 - [10] J. Bacidore, K. Berkow, B. Polidore, and N. Saraiya, “Cluster Analysis for Evaluating Trading Strategies,” *J. Trading*, vol. 7, no. 3, pp. 6–11, 2012.
 - [11] W.-Y. Loh, “Classification and regression trees,” *Wiley Interdiscip. Rev. data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 14–23, 2011.
 - [12] J. S. Hunter, “The Exponentially Weighted Moving Average,” *J. Qual. Technol.*, vol. 18, no. 4, pp. 203–210, 1986, doi: 10.1080/00224065.1986.11979014.

Music analysis using deep learning

Matej Halinkovič^{*}[0000–0002–8002–9887]

Faculty of Informatics and Information Technologies STU in Bratislava, Ilkovičova 2
842 16 Bratislava 4
`xhalinkovicm@stuba.sk`

Abstract. This paper presents a method of applying deep neural networks in the field of music analysis, specifically genre recognition. Our proposed model utilizes a convolutional recurrent neural network (CRNN) architecture to classify audio clips presented in the form of a mel spectrogram. The CRNN uses a convolutional neural network for general feature extraction and a gated recurrent unit to supplement temporal information for the extracted features. This CRNN model was tested in a serial and parallel configuration on the complex Free Music Archive dataset where it showed promising results, and on a simpler custom dataset to explore the upper boundaries of the model’s capabilities. Overall, we discovered that the proposed CRNN model is capable of achieving favourable results on multiple datasets and that it is able to be retrained to adjust to new data quickly, due to its relatively low hardware demands.

Keywords: Deep learning · Music information retrieval · Convolutional neural networks · Recurrent neural networks · Artificial intelligence.

1 Introduction

In recent years, deep learning has shown to be immensely useful in a wide area of applications. Its approaches are not unified and there exists a variety of models, each with its own benefits [8]. Deep learning has earned its popularity by being robust, universal in its learning approach, capable of generalization, and highly scalable [1].

In 2016, Google DeepMind unveiled a project called WaveNet, a deep neural network for generating raw audio waveforms [10]. It demonstrated state-of-the-art performance when applied to text-to-speech, with human listeners rating it significantly more natural sounding compared to its predecessors. It was also applied to the problem of music modeling, yielding highly realistic musical fragments, thus demonstrating the capabilities of deep learning in the field of acoustic signals.

Extracting useful information from audio signals poses a significant challenge that has only been tackled successfully in recent years. We can see this advancement in the rapid propagation of speech recognition systems [12].

^{*} Bachelor study programme in field: Informatics Supervisor: Assoc. Professor Vanda Benešová, Institute of Computer Engineering and Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

While music analysis bears many similarities to speech recognition, it also presents new constraints [12]. Music typically consists of numerous sound sources that form more complex dependencies and span a wider range of frequencies compared to speech.

Many music analysis models rely on using existing image recognition architectures with modifications to accommodate audio representations, without accounting for the properties of music. In our work, we present a model that utilizes a convolutional recurrent neural network to capture the dependencies found in music and characterize audio clips without needing additional metadata.

2 Related work

While Music Information Retrieval, and specifically Genre Classification, is a widely studied area, there is no single best approach we can rely on. A challenge [4] organised in 2018 by the CrowdAI organisation provides a good overview of many different approaches to this problems. The contestants were tasked with creating a method of recognizing genres from the data provided by the medium version of the Free Music Archive dataset [2]. Attempts were made to repurpose successful image recognition models [3] into music analysis models. While these methods can be competitive in terms of accuracy (a modified Xception model achieved an F1-Score of 0.60), they tend to be outperformed by other approaches if sufficient computational power isn't available [4].

A study [9] points to the fact that ensemble learning methods using numerical feature extractors can significantly outperform the traditional Deep Neural Network approach. With only a lower-end GPU available (GTX 1060, 6GB VRAM), the XGboost classifier managed to achieve a significantly lower loss than a CNN model capable of training on this card, resulting in an F1-Score of 0.60

The internal structure of music requires an architecture that specializes in retrieving information that is specific to music. We can see this example in a model [7] that uses additional metadata about artists in addition to clips of audio to learn these so-called 'Artist Group Factors' (AGF). This method allows for the usage of a model, that can train a simple network for each AGF in parallel and then combine their outputs for a final genre prediction. This model was also the most successful model entered into the competition, achieving an average F1-Score of 0.63.

3 Theoretical background

The utilization of deep learning methods has proven to be incredibly successful in the areas of image processing, speech recognition, art, medical imaging, cybersecurity, and many more [1]. The field of music analysis (Music Information Retrieval) is primarily inspired by the accomplishments of Convolutional Neural Networks (CNNs) in computer vision.

3.1 Convolutional Neural Networks

CNNs replace the standard matrix multiplication seen in classic neural networks with convolutions, reducing the number of weights required, and reducing the overall complexity of the network. In addition to convolutions, max-pooling layers are used to summarize the values of several nearby outputs and replace them with the maximum output in a rectangular neighbourhood with a goal to reduce the dimensions of data representation [1,8].

The output of these summarizations is passed through activation functions (e.g. Sigmoid, SoftMax, rectified linear, ...) to produce feature maps, which can later be converted to classifications. These feature maps can be defined as:

$$\mathbf{x}_j^l = \mathbf{f}(\sum_{i \in M_j} \mathbf{x}_i^{l-1} * \mathbf{k}_{ij}^l + \mathbf{b}_j^l) \quad (1)$$

Where x_j^l is the output of the current layer, x_j^{l-1} is the output of the previous layer, k_{ij}^l is the kernel for the current layer, b_j^l are the biases for the current layer, and M_j represents a selection of input maps [1].

3.2 Gated Recurrent Units

Unlike in image processing, the features extracted by CNNs may not be sufficient for a model to successfully classify music. Music relies heavily on temporal dependencies that can be captured by utilising the fact that recurrent neural networks allow the operation over a sequence of vectors over time [1]. Adding a recurrent component to a conventional CNN model should allow it to better analyze the relationships between the extracted features. The output of such a model should therefore better reflect the nuanced differences between genres.

In our model we implement the recurrent component through a Gated Recurrent Unit (GRU). Compared to a traditional recurrent neural network, it does not suffer from long-term memory loss (information getting lost over time) and it has a lower computational cost compared to a long short-term memory (LSTM) recurrent model [1].

The GRU retains information through its hidden state. The hidden state carries over information from the previous iteration, and this information is altered on each layer by new input. This process is controlled by a reset gate and an update gate.

Mathematically, we can define this process as [1]:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z * [\mathbf{h}_{t-1}, \mathbf{x}_t]) \quad (2)$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r * [\mathbf{h}_{t-1}, \mathbf{x}_t]) \quad (3)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W} * [\mathbf{r}_t * \mathbf{h}_{t-1}, \mathbf{x}_t]) \quad (4)$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{z}_t) * \mathbf{h}_{t-1} + \mathbf{z}_t * \tilde{\mathbf{h}}_t \quad (5)$$

Where z_t represents the update gate, r_t represents the reset gate, W is a weight matrix, h_t is a hidden state vector, x_t is the input vector, and σ is the sigmoid function.

3.3 Signal Processing

The success of models in deep learning relies heavily on the data we provide for it. Sound is a continuous signal that needs to be transformed into a sequence of discrete values using sampling [15]. The higher the rate of sampling, the more precise data we get, but a high sampling rate also increases the computational expense required [15].

However, the sampling rate we choose must adhere to the Nyquist theorem - all the frequency components of the signal we are sampling must be at most half the sampling frequency [13].

Mathematically, we can define it as [13]:

$$f_s > 2 * f_m \quad (6)$$

Where f_s is the sampling frequency and f_m is the maximum frequency in the signal.

Any frequency components above this threshold could be interpreted as negative frequencies and introduce error [13]. Such misrepresentation of high frequencies as lower frequencies is referred to as aliasing.

Fourier transform We can decompose our signal into its constituent frequencies by using the Fourier transform. When analyzing music this can be extremely useful. For example, we can use it to decompose a chord into individual frequencies of notes that constitute it.

The discrete Fourier transform is well suited for computational needs. It allows discrete sampling of a signal at constant intervals. Applying the discrete Fourier transform allows us to use the fast Fourier transform algorithm to obtain results with $O(N \log N)$ complexity [13]. The discrete Fourier transform is defined by [13]:

$$\mathbf{F}(\mathbf{n}) = \frac{1}{N} \sum_{\mathbf{k}=0}^{N-1} \mathbf{f}(\mathbf{k}) e^{-\frac{2\pi i \mathbf{k} \mathbf{n}}{N}} \quad (7)$$

Where N represents the number of frequency components, k represents the frequency values and $i \equiv \sqrt{-1}$.

Spectrograms Spectrograms let us represent audio data in greater detail. They allow us to show time, frequency, and amplitude in one graph [11]. Spectrograms are also useful tools for detecting issues with our audio samples.

They are created by splitting an audio signal into smaller frames. This way, each frame can represent a set amount of time. We can apply the fast Fourier transform to each frame to extract the frequencies contained in the given frame. It is considered good practice to overlap frames to prevent the loss of some frequencies [11].

If we want to classify audio data based on the way it sounds to humans for example, into music genres, it is beneficial to represent it in the Mel Scale. The Mel Scale bases the distance between tones on the way human hearing works. Distances between frequencies depend on how far apart they would sound to humans, not on the numerical difference between them [17].

This conversion between a linear Hz scale and a ‘human’ scale is easily implementable using a nonlinear transformation. Mathematically, it can be defined as [17]:

$$\text{Mel} = 2595 * \log_{10}(1 + \frac{\text{freq}}{700}) \quad (8)$$

Using this transformed scale to interpret audio the way a human would could significantly increase the effectiveness of our model. The performance of mel spectrograms in the area of music information retrieval seems to be superior to other methods of representing audio, outperforming traditional spectrograms and models based on processing raw audio [5,16].

4 Our solution

In our approach, we focused on creating a convolutional recurrent neural network (CRNN) model that attempts to predict genres based on audio data transformed into mel spectrograms. To preserve as much information as possible, the data was stored and used in models as unaltered tensors.

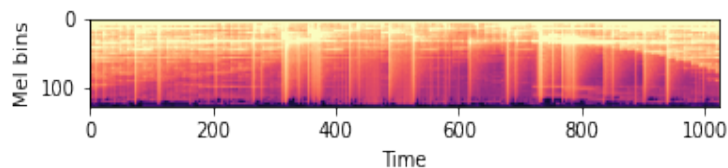


Fig.1: An example of a mel spectrogram fed to the network. Parameters: `n_fft=2048`, `hop_length=1024`, `n_mels=128`, `win_length=n_fft`, `win_function=torch.hann_window`

We explored several different implementations of a GRU paired with convolutional layers to increase the reliability of our results.

The best results were achieved by appending a GRU to a 4 layer convolutional model to create a serial Convolutional Recurrent Neural Network (CRNN).

The output of the GRU was then passed through a fully connected layer and a SoftMax function to create a final genre prediction Fig. 2.

Only slightly worse results were achieved by a parallel CRNN implementation. This parallel network consists of two separate models - a CNN and a GRU, each of which received the same spectrogram as input. The output tensors of both of these models were then concatenated, passed through fully connected layers and a SoftMax function to create a final genre prediction Fig. 3.

The number of channels in the convolutional layers of our models increases in the powers of 2, which is inspired by the VGG-16 architecture [14]. Both models include a layer that does not increase the number of channels and has a kernel size of 1 at the end of the CNN architecture. The inclusion of this layer lead to more accurate predictions in our experiments.

Both models feature a pooling layer just before the GRU that serves as encoding of sorts. This approach allows us to reduce the size of the GRU and it reduces the overfitting of our models.

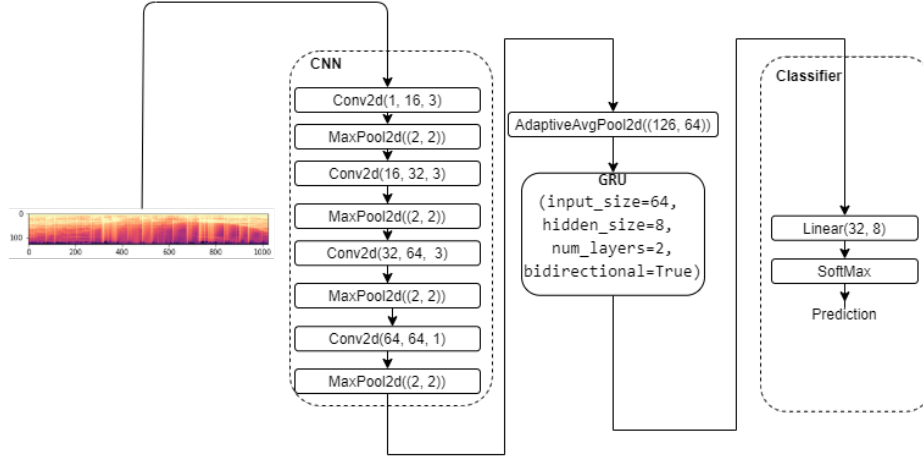


Fig. 2: Serial CRNN model

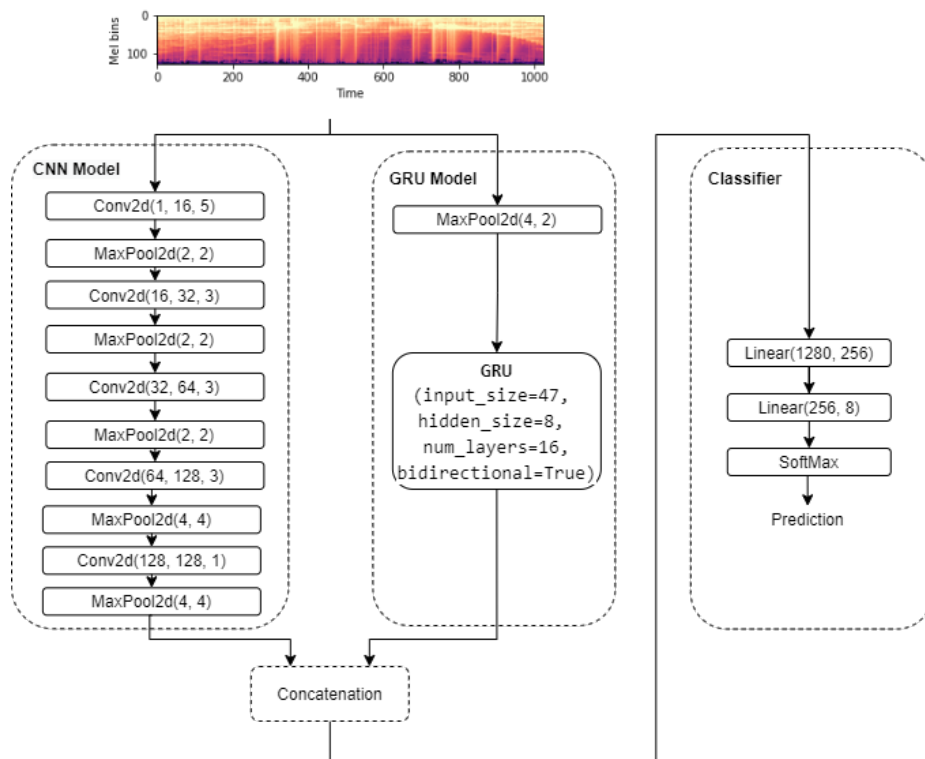


Fig. 3: Parallel CRNN model

4.1 Training

Our approach is implemented using the PyTorch framework for deep learning and our experiments run on hardware provided by the platform Google Colab. PyTorch is also used to create spectrograms from audio data Fig. 1.

The spectrograms provided for our models were created from the Free Music Archive dataset [2]. It sources its data from the Free Music Archive platform, where independent artists upload their creations under a Creative Commons license. The authors of this dataset used the API to create a compilation of metadata combined with .mp3 audio files to create a complex dataset with tools, creating an efficient platform that can be used in all kinds of music analysis applications.

Our model utilizes the 'top-genre' labeling of clips and the audioclips themselves. Due to our limited access to hardware, we decided to use the 'small' version of this dataset, which contains 8000 30 second clips split evenly between 8 genres. The .mp3 files are transformed into mel spectrograms Fig. 1 and saved for further use as a train/validation/test split with a ratio of 80/10/10 %.

Additionally, we created a custom, simpler dataset that consists of 600 full songs split evenly between 6 genres. The songs were sourced from Top 100 lists of their respective genres and clipped to 30 seconds when creating spectrograms to better match the data from the FMA dataset and make comparisons more meaningful.

This dataset is included to provide a more controlled exploration of the capabilities of our models. Since the FMA dataset is annotated by individual artists and contains hard to define genres such as 'International' and 'Experimental' it is likely that it contains a degree of labeling noise [2].

4.2 Results

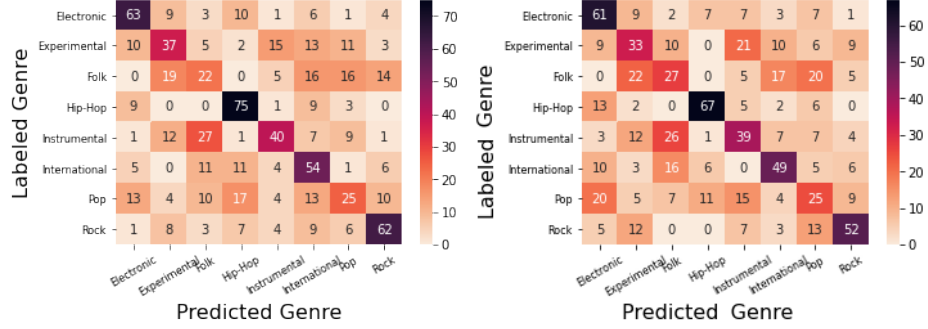
In addition to overall accuracy of the model, we decided to include the metrics *Precision*, *Recall*, and *F1-score* to evaluate our models on a per-genre basis. These metrics were calculated as follows [6]:

$$\text{Precision} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}} \quad (9)$$

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (10)$$

$$\text{F1 - score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

Both of our models achieved a similar performance on the FMA dataset, with the serial version gaining a slight edge with an accuracy of **49.22%** compared to the parallel model with an accuracy of **45.96%**.



(a) Serial CRNN model confusion matrix (b) Parallel CRNN model confusion matrix

Fig. 4: Confusion matrices on the FMA dataset

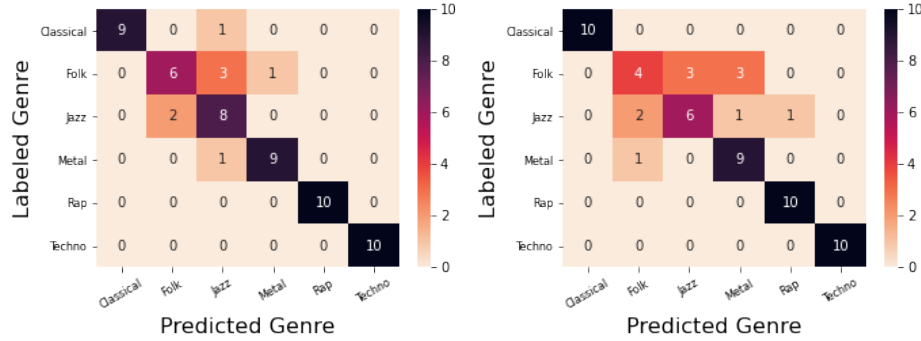
Table 1: Serial CRNN evaluation metrics

Metric Genre	Recall	Precision	F1-Score
Electronic	0.649485	0.617647	0.633166
Experimental	0.385417	0.415730	0.400000
Folk	0.239130	0.271605	0.254335
Hip-Hop	0.773196	0.609756	0.681818
Instrumental	0.408163	0.540541	0.465116
International	0.586957	0.425197	0.493151
Pop	0.260417	0.347222	0.297619
Rock	0.620000	0.620000	0.620000

Table 2: Parallel CRNN evaluation metrics

Metric Genre	Recall	Precision	F1-Score
Electronic	0.628866	0.504132	0.559633
Experimental	0.336735	0.336735	0.336735
Folk	0.281250	0.306818	0.293478
Hip-Hop	0.705263	0.728261	0.716578
Instrumental	0.393939	0.393939	0.393939
International	0.515789	0.515789	0.515789
Pop	0.302083	0.322222	0.311828
Rock	0.260417	0.280899	0.270270

We also decided to run both of our models on the custom dataset to see if they are capable of performing more accurate predictions on simpler data. On this dataset, the serial model reached an accuracy of **86%**. The parallel model performed slightly worse once again, with an overall accuracy of **81%**.



(a) Serial CRNN model confusion matrix (b) Parallel CRNN model confusion matrix

Fig. 5: Confusion matrices on the custom dataset

5 Conclusions

Both of our models came close to 50% accuracy on the FMA dataset. While this accuracy is not groundbreaking, it is important to note that the FMA dataset has some drawbacks. The genres are set by the musicians themselves when they upload their music to the website. The website is curated, but it does not guarantee that all clips are labeled correctly.

There is also the question of genres themselves. Genres do not have specific, clearly defined borders. Especially when we consider the included genres *Instrumental* and *International*. During testing, we listened to clips that our model classified incorrectly and often failed to classify the genre ourselves.

When presented with a simpler dataset that consisted of genres that we judged to be sufficiently distinct, the model achieved a significantly higher accuracy (86%). This seems to indicate that our model is capable of highly accurate predictions if provided with data of adequate clarity.

An additional benefit of our model could be seen in the fact that it is able to fully train in about 5 hours using a relatively inexpensive Tesla K80 GPU. This means that it can be adapted to new data fairly quickly.

It appears that we would need to rely on a supporting model that utilizes additional metadata (such as artist information) to further increase the accuracy of our model on more complex data.

References

1. Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Hasan, M., Van Essen, B.C., Awwal, A.A.S., Asari, V.K.: A state-of-the-art survey on deep learning theory and architectures. *Electronics* **8**(3), 292 (3 2019). <https://doi.org/10.3390/electronics8030292>, <http://dx.doi.org/10.3390/electronics8030292>
2. Benzi, K., Defferrard, M., Vandergheynst, P., Bresson, X.: FMA: A dataset for music analysis. *CoRR* **abs/1612.01840** (2016), <http://arxiv.org/abs/1612.01840>
3. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1800–1807 (2017). <https://doi.org/10.1109/CVPR.2017.195>
4. Defferrard, M., Mohanty, S.P., Carroll, S.F., Salathé, M.: Learning to recognize musical genre from audio: Challenge overview. In: Companion Proceedings of the The Web Conference 2018. p. 1921–1922. WWW ’18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). <https://doi.org/10.1145/3184558.3192310>, <https://doi.org/10.1145/3184558.3192310>
5. Dieleman, S., Schrauwen, B.: End-to-end learning for music audio. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6964–6968 (2014). <https://doi.org/10.1109/ICASSP.2014.6854950>
6. Handelman, G.S., Kok, H.K., Chandra, R.V., Razavi, A.H., Huang, S., Brooks, M., Lee, M.J., Asadi, H.: Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods. *American Journal of Roentgenology* **212**(1), 38–43 (2019). <https://doi.org/10.2214/ajr.18.20224>
7. Kim, J., Won, M., Serra, X., Liem, C.C.S.: Transfer learning of artist group factors to musical genre classification. In: Companion Proceedings of the The Web Conference 2018. p. 1929–1934. WWW ’18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). <https://doi.org/10.1145/3184558.3191823>, <https://doi.org/10.1145/3184558.3191823>
8. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. *Neurocomputing* **234**, 11–26 (2017). <https://doi.org/10.1016/j.neucom.2016.12.038>
9. Murauer, B., Specht, G.: Detecting music genre using extreme gradient boosting. In: Companion Proceedings of the The Web Conference 2018. p. 1923–1927. WWW ’18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). <https://doi.org/10.1145/3184558.3191822>, <https://doi.org/10.1145/3184558.3191822>
10. Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016)
11. Prasomphan, S.: Detecting human emotion via speech recognition by using speech spectrogram (2015). <https://doi.org/10.1109/dsaa.2015.7344793>
12. Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S., Sainath, T.N.: Deep learning for audio signal processing. *CoRR* **abs/1905.00078** (2019), <http://arxiv.org/abs/1905.00078>
13. Randall, R.B.: Frequency analysis. *Bruel et Kjaer* (1987)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference

- on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), <http://arxiv.org/abs/1409.1556>
15. Udo, Z.: Digital Audio Signal Processing 2e. John Wiley & Sons Incorporated (2008)
 16. Ullrich, K., Schlüter, J., Grill, T.: Boundary detection in music structure analysis using convolutional neural networks. In: ISMIR. pp. 417–422 (2014)
 17. Zabidi, A., Mansor, W., Khuan, L.Y., Sahak, R., Rahman, F.: Mel-frequency cepstrum coefficient analysis of infant cry with hypothyroidism. 2009 5th International Colloquium on Signal Processing & Its Applications pp. 204–208 (2009)

Remote Secure Data Collection In Intelligent Households

Vladimír BACHAN, Maroš ČERGETĚ, Veronika ČIPELOVÁ, Ľudovít POPELKA,
Robert YAMKOVYI, Tomáš ZÁTKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
veronika.cipelova@gmail.com

Abstract. In this paper, a solution is proposed for a problem that started to be the most relevant during the world pandemic. While it is safer when people do not get visits from waterwork or power plant employees, household consumption of heat, water, electricity still needs to be tracked and properly billed. This idea opens a new realm of possibilities and even more problems that need to be addressed. Whereas it starts from the fact that there needs to be a way by which those analog measurements are being tracked, the true innovation comes from the fact that units that are responsible for this cannot rely on regular power outlets, yet they need to make use of the surrounding environment and harvest its energy from it. It is important to also denote the added value of a lesser ecological footprint and increased comfort for all parties - starting from a developer who can keep track of his projects to measured subjects who can view summarized information about their consumption. The whole solution needs to be secure, not letting end customers tamper with it, which will result in information about the particular person, being sent to the responsible person.

Keywords: Data Collection, Utility Measurement, Z-wave, Rest API, MySQL, HTTPS, Security.

1 Introduction

In recent years, many people have turned their focus to environmental issues like wasting water and using sustainable energy sources. Nowadays with the busy digital era, people value their time and prefer to look information up digitally, which is the reason why there is a need for a simple effective solution for tracking certain utilities like used hot water or heating. An energy autonomous system for collecting data from utility meters along with comprehensible graphical visualization of the collected data is proposed. The system is primarily meant for developers and aims to save their time

* Master's degree study programme in field: Information systems
Supervisor: Vladimír Kunštár, Institute of Informatics, Institute of Computer Engineering and Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

by improving the utility measuring logistics. Using the system will let the user check the data remotely from the comfort of his home or any place he prefers, as long as he has access to a device that can connect to the internet, without the need to physically check the utility meters. One of the advantages of the system is that the user does not have to remember and keep a mental note of the utility-related information or write it down using analog tools since he can find this information easily by using the system. A good example is keeping a record of the used hot water over a course of time which is more tedious when done by old methods like writing down in a notebook compared to the digital means.

2 Related work

There are many research papers written about this topic, each of them using a different approach when it comes to communication technologies to suit the specific needs of a project – ranging from Zigbee [1], WiFi [2] to Sigfox [3]. Sigfox is a long-range low-power wide-area network (LPWAN) with small message bandwidth. In our work, we decided to use Z-Wave as opposed to the other mentioned technologies.

Zigbee allows for encryption and creating a full-mesh topology which means that a neighboring node can be used for transmitting the message to the master unit by hops between individual nodes. Similar to BLE and WiFi networks, Zigbee is a protocol that uses a 2.4GHz bandwidth. Even though it provides a better collision solution, there is still a negative impact and the maximum range in buildings is 10-20 meters.

Using WiFi as means of communication for collecting data from households requires a router and an internet connection and it is not guaranteed every person will be able to meet these requirements. Other concerns are the stability of the internet connection, internet outages caused by the internet providers, crowding the network, etc., since WiFi devices most commonly operate on a bandwidth of 2.4GHz. WiFi devices mostly have even larger energy consumption than ZigBee modules.

Z-Wave is a low-power, low bandwidth protocol [4] that operates on a different radio frequency than both Zigbee and WiFi, which is 800-900MHz, so it will not interfere with the WiFi signal in the households. It has good interoperability between Z-Wave members and requires a separate hub just like Zigbee. Additionally, using Z-Wave provides good signal penetrations into buildings. Finding and selecting the right Z-Wave modem allowed us to meet the energy usage requirements of our project. The costs of Z-Wave devices are higher, but Z-Wave is more favorable because of its properties. Summarized, we decided to use Z-Wave instead of the other mentioned technologies because of a large combination of factors that were taken into account, which are its ability to create an encrypted full-mesh topology, its broad coverage in buildings, and the fact that a loss of signal quality due to excessive interference caused by other services is not a problem because it uses a different frequency.

There is a lot more research done about data collection systems for utilities in single households as opposed to solutions meant for multiple households, developers, and large companies. Hardware and software system that assists to control the household energy consumption and reduce the electricity bill is considered as a home ener-

gy management system. [5] The key features of such systems are real-time energy consumption data collection, visualization of energy consumption, web access or Android or Apple OS applications to monitor consumption and control, i.e. activation/deactivation, of home appliances in a single household. Real-time energy consumption data collection is also a key feature of our work with the difference of sending the collected data periodically instead of in real-time. The reason for this is lowering energy usage of our system since one of the requirements is energy autonomy.

A paper written by Jongwoon Hwang et al. [1] is a good example of implemented visualization of energy consumption, available either from personal computers or mobile devices while using Zigbee to transfer the collected data.

The work from 2020 [2] describes the smart water consumption measurement system, which is responsible for monitoring water consumption in households and visualization of the collected data. Opposed to previously mentioned works, this system also provides an anti-tampering security mechanism. In contrast with our system, this solution does not provide any statistics and is not meant for multiple customers or developers and uses Raspberry Pi with a WiFi connection, while we are designing and making a functional prototype of our device to achieve better control over the hardware part of our system.

3 Topology

Our data collection system consists of three main parts, which ensure onsite 24/7 measurement availability with autonomous measurement data storage and transportation to the host application.

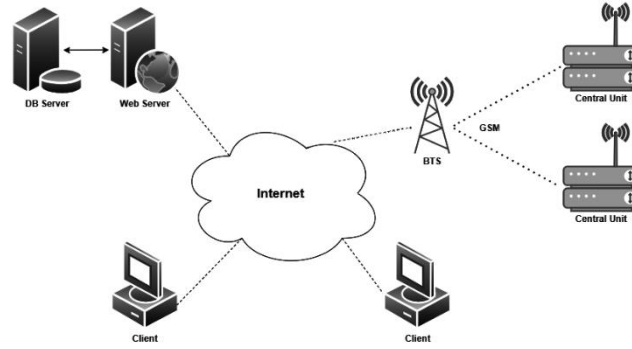


Figure 1. External topology

There are two types of embedded solutions installed at the customer's side. The first one, called the central unit (CEU), is the main unit at the customer's side. The second one, called the controller unit (CU), is responsible for collecting the data from utility meters and temporarily storing them for further synchronization with CEU. Embedded CUs are also equipped with a self-charging unit consisting of Peltier's plates. This may provide enough electricity for charging the internal lithium battery pack and extend the durability.

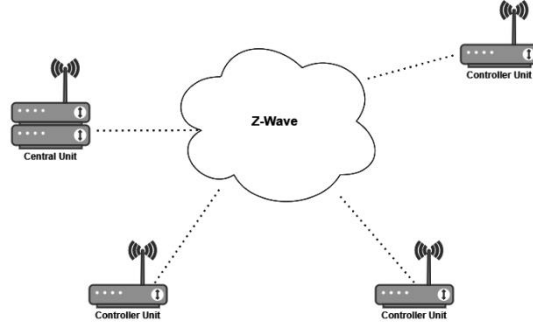


Figure 2. Internal topology

The CEU is periodically tracking the data from appropriate CUs using Z-Wave technology. Periodic tracking allows us to use less energy during power signal transmission. Sub-gigahertz Z-Wave technology gives the devices the ability to communicate over longer distances inside the buildings with much better signal penetration and less interference than other technologies. It provides a secure full-mesh communication network [6] and because of the high energy efficiency of radio devices is also reliable and allows us to create long-lasting solutions. CEU units are equipped with a cellular data modem from Quectel company. This is the main communication link to the remote system, the Data Collection System.

4 Communication

There are two main areas where different communication principles were implemented. On one side we have an external network as shown in Figure 1 and on the other side is the internal network shown in Figure 2. The first area needs sophisticated solutions, which allow scalability, distribution, and also diagnostics data transfer. This communication protocol uses a well-known Rest API over HTTPS protocol. Communication on this side uses JSON-rich body structure to separate all transferred key parts, like measurements and diagnostics data. According to the given example in the Code 1, there are two identification parts.

```
{
  "requestNumber": 1,
  "centralUnit": 1,
  "measurement": {
    "controllers": [
      {
        "controllerUnit": 1,
        "measurements": [
          {
```

```

        "sensorIO": "I1",
        "count": 2
    }
]
},
"diagnostics": {
    "battery": 3.8,
    "localTime": "2020-03-01T00:00:00.000Z"
}
}

```

Code 1. JSON body example

The request number is used for synchronization and the central unit points to a concrete source of measurements. Measurement object provides all necessary data to collect from devices and also can provide optional diagnostics data.

The second network area covers the interior of buildings using Z-Wave radio technology. For this purpose, we were inspired by the widely used STX/ETX protocol better known as Binary Synchronous Communication (BSC or Bisync). According to the Table 1 internal protocol allows multiple message types to be transferred between Z-Wave nodes using a file separator (FS) in the data part.

Table 1. Internal communication packet structure

STX	LENGTH		DATA					ETX	CRC
	Hi	Lo	TYPE	MSG	FS	TYPE	MSG		
1 B	1 B	1 B	1 B	...	1 B	1 B	...	1 B	1 B

The whole data part is surrounded by the start of text (STX) and end of text (ETX) characters followed by a cyclic redundancy check (CRC). These packets are transmitted securely over periodically formed full-mesh topology of nodes delivered to the CEU.

5 Front-End and Back-End of server

Expected customers of our solution are developers that pay for utilities in flats that they rent or waterworks, power plants, etc., that provide their services to several households and then bill them respectively. This process requires that developers or authorized employees have their account created, hence allowing them to create projects, details of which they can then consult with a legitimate person from our team

that provides them with an ability to add flats, central units, controller units, and actual sensors that are being used to do those measurements to the system.

The view that is provided to the developers then shows them all their projects, the hierarchy of addresses corresponding to that project, and also possibly residing flats at those addresses. The developer can also view the utility usage statistics and some interesting (mainly relevant) information collected from all utility meters. Individual residents that rent the flat that the developer owns/own their flat but are subsidiary to the powerplant can only view information collected from their utility meters, in the same manner they can now. The access is provided based on a number that they are given during the installation of the solution.

When it comes to the data stored, the solution has a database model that is managed by Glassfish Datasource which provides the requesting application with a connection, not storing any kind of connection passwords, etc., on the machine. Important data such as measurements, users, addresses, logs, etc., are stored, which can be shown in Figure 3.

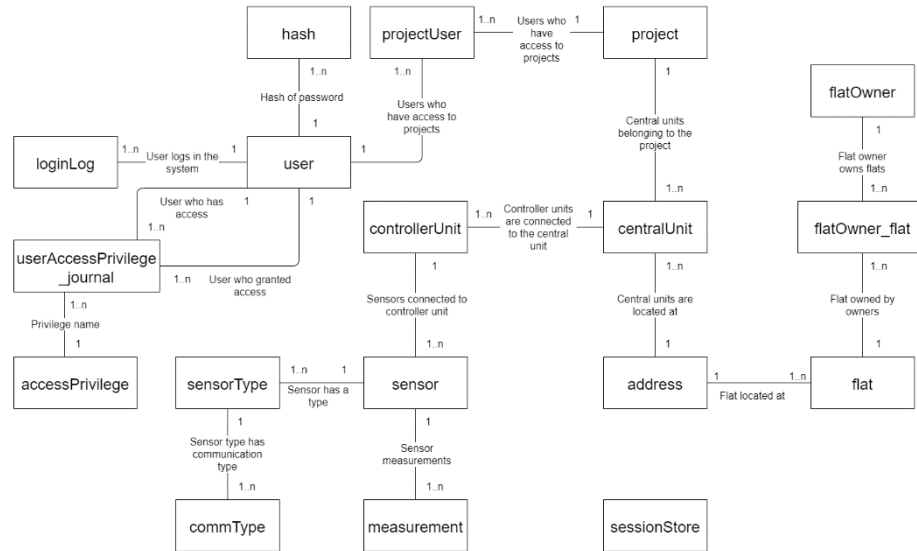


Figure 3. Database model

The server is run on a machine with Ubuntu 20.04, which provides access via SSH, all users have their RSA 2048 public keys stored on the server and their private keys at their machines. Only after the verification is done, can the user access the Glass-Fish configuration window, deploy WAR file on the server, etc., not exposing critical configuration access risk to the untrusted internet.

6 Future of the project

With the development of smart home systems and their growing popularity, this system has good potential. The project took into account the possibilities of expansion and scalability. In the future, this system can become an assistant for large commercial organizations such as waterworks, etc.

Humans are already using smartphones, smartwatches are also beginning to arise, so interaction with the computer to watch your consumption will no longer be needed, hence the repetitive writing of the same data to the web will be erased. When it comes to the future of consumption measurement we believe that the technology we use, Z-Wave, will emerge to be used widely, that is why we are starting to utilize this technology, as a form of experiment and despite the lack of precise documentation examples, we believe that it is more suitable for our needs.

There is still room for improvement in this project such as implementing better statistical algorithms for the water, heat, and energy consumption in households and adjusting the intervals of sending messages about the collected data to save more energy and extend the time the batteries used in our device will last. These improvements can be made only after our system is fully tested in the real-world environment since many factors have to be considered.

7 Conclusion

Intelligent households are supposed to become prevalent in the future, yet various partial innovations can be integrated into our daily life. Such an example is the Data Collection System - an intelligent solution to track the energy consumption of households. In this paper, we have demonstrated how the collection system can be designed, along with implementation details. Our work can be used as a base for continuous research, and/or as a guide to the practical application of the system as well.

8 References

1. Kim, W. H., Lee, S., & Hwang, J.: Real-time energy monitoring and controlling system based on Zigbee sensor networks. In: The 8th International Conference on Mobile Web Information Systems 2011, vol. 5, pp. 794-797. Procedia Computer Science (2011).
2. Fuentes, H., & Mauricio, D.: Smart water consumption measurement system for houses using IoT and cloud computing. In: Environmental Monitoring and Assessment, vol. 192, pp. 602. Springer (2020).
3. Osman, N. I., and Abbas, E. B.: Simulation and Modelling of LoRa and Sigfox Low Power Wide Area Network Technologies. In: International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE) 2018, pp. 1-5. Khartoum (2018).
4. Tao G. J., Wu T. C., Jun L. L., Ling Z.: A New Monitoring System of Portable Microcomputer Injection Pumps Based on Z-Wave. In: Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) 2016, Macau, China, pp. 19-21. IEEE (2016).

5. Sameera, N. H. K, Jayawardana, O., Palayangoda N. A., Sanjeewa, A. D., Abeygunawardane S. K.: Smart energy tracking system for domestic consumers. In: Moratuwa Engineering Research Conference 2016, pp. 254-259. IEEE (2016).
6. Z-Wave Wireless Solutions for the Smart Home, <https://www.silabs.com/wireless/z-wave>, last accessed 2021/3/14.

VR Lab

Ľubomír Kurčák*, Erik Paľa*, Patrik Tománek*, Viktor Beňo*

Slovak university of technology in Bratislava, Faculty of Informatics and Informative Technologies, Ilkovičova 2, SK-842 16 Bratislava 4
tp16.radiant@gmail.com

Abstract. The pandemic situation in 2020 introduced us to new problems in our educational system. Mainly the education in specialized laboratories with hands-on exercise requiring multiple people in one room, which was not an option during the pandemic restrictions. The goal of this project is to overcome this problem using modern and affordable technologies in virtual reality and provide an online form of practical exercises in virtual environment, which would be as effective and entertaining as a real on-site exercise with a teacher. We propose an alternative solution to practical education and hands-on experience for which we have demonstrated its plausibility by building an example environment in which students can practice electric circuit construction and simulation. We also provide our future intentions and possible improvements to this project.

Keywords: Virtual Reality, Circuit, Simulation

1 Introduction

Because of the current coronavirus events the education system has become fragile. Mainly children are not able to go to their classrooms and continue to educate themselves, which is the most crucial part of childhood. Coronavirus pandemic has shown the world, that we are not able to educate children in the form of online teaching [1]. Especially practical exercises are completely left out because children need to attend specialized laboratories with the presence of professional supervisor [2]. We aim to overcome these coronavirus limitations by creating an online form of teaching in Virtual Reality. Our goal is to create a specialized laboratory for practical exercises in the field of electrical circuits. In this laboratory, students will be able to construct and simulate their own electrical circuits and teachers will be able to share their presentations on an interactive in-game board. We have also decided to create a lecture about constructing simple electrical circuits for an electrical go-kart, which will also be provided for all the participants. They will be able to construct various electrical circuits and put them on their go-karts. After that they can go and test their skills in electrical circuits

* Master study programme in field: Informatics
Supervisor: Ing. Juraj Vincúr, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

and driving in a fun race competition with other students.

2 Related works

The authors of *Perspective Electrical Circuit Simulation with Virtual Reality* [3] created a simple 3D room as their environment for electrical circuit simulations. During the evaluation of their work, they concluded that virtual reality was determined to be the highest effective media for studying electrical circuits. In another project called *Primary School STEM Education: Using 3D Computer-based Virtual Reality and Experimental Laboratory Simulation in a Physics Case Study* [4], authors used an educational application *Water Cycle in Nature*, that focuses on physics phenomena such as vaporization and condensation. During their evaluation, they found out that 90% of participants would like to take part in more lessons, because the application eased the understanding of the described concepts.

3 Schooling from home

Current coronavirus pandemic disrupted the way of education around the world. The younger generation has suffered the most, because teachers, parents and students were not prepared for the drastic transition into an online education. The most vulnerable, and at the same time one of the most important fields of education are practical exercises. Practical exercises require special equipment and professional supervisor, and thus the physical presence of several persons in specialized laboratories. Because of this, an enormous number of children are not being educated properly, which will result in a serious problem in the future. That is why we have decided to overcome these coronavirus limitations and use the most modern and affordable technologies in the field of Virtual Reality, hereinafter referred as VR, to provide an online form of educating children. Thanks to the infinite opportunities of VR, we aim to create an experience, which is at least as good and effective as teaching from inside of a specialized school laboratory [5].

As our main goal, we have decided to create an online laboratory for practical exercises in the field of electrical circuits. Users will be located in specialized laboratory, in which every user will have their own working space. This working space will consist of a grid-board and a set of electrical components, which can be inserted into the mentioned grid-board. After constructing their electrical circuits, users will be able to test their functionality through a real-time simulation of this circuit.

Our project will consist of two ways of teaching. The first way is teaching with a teacher, and the second one is teaching without a teacher. Both ways will be described herein.

3.1 Teaching with a teacher

Teachers and their students will be able to join their own dedicated online laboratory. Every participant will obtain their own working space. As for teachers, they will also obtain a fully interactive board. On this board they will be able to share their own presentations from their devices, highlight important parts of the presentation or draw whatever they need.

3.2 Teaching without a teacher

It's important for us to create an opportunity of education to every user that has the desire to learn. We have decided to create one lecture, in which we will provide information and tutorial on how to build electrical circuits for a simple electrical go-kart. Users will be able to understand and create such circuits without a need for a teacher.

With the opportunity of creating a VR game, we think that it is important to provide quality education with something extra. That is why we have decided to provide a simple electric go-kart 3D model to every participant in this lecture. After finishing the education part of the lecture, they will be able to test the functionality of their circuits in a fun race. Prior to that, they will first need to put all their electrical circuits on the go-kart. During the lecture, they will be able to create various electrical circuits such as simple electric motor, lights, horn, and so on. With every correctly build circuit their go-kart will be upgraded.

When their go-kart meets the minimal requirements to join the race, all the participants will be situated on a pre-created race map. At this race they will be able to compare their skills in electrical circuits as well as driving. We believe that this way of education, with something extra, will be even better and more effective as teaching from inside of a specialized laboratory with no fun parts involved during the process.

4 Current results

The current state of the project consists of following parts:

- Interactive working station – working station consists of 3 main part:
 - Gridboard part allows users to construct their own electrical circuits on a futuristic table. All connections are displayed on a standalone console screen. Each gridboard table also has its own terminal, from which player can operate this simulation. Player can start simulation, remove all components from gridboard table, generate spice code of currently simulated electrical circuit or change components values during the simulation. See Figure 1 for visual details.
 - Components modifier part allows users to pick up all of the available electrical components, as well as create their own specifications. Players are able to pick up a component and put it into the tuning station. From this station players can permanently change components values, as well as models. They can create and insert their own model

specifications in spice language. After that the terminal will parse this specification and create a new available model, which can be used on given components. Players can also save all of their modified components, so they won't have to create them again in the future. See Figure 2 for visual details.

- Vehicle part allows users to see the current progress of their go-kart. Go-kart parts as motor, lights, horn, etc. will need their own electrical circuit to be functioning. Players have to build these circuits and insert them into the given parts. These parts then can be inserted back to the go-kart. After their go-kart is finished, they can take it for a drive.
- Integration of Spice# [6] and Spice#Parser [7] projects for electrical circuits simulation – the process of creating electrical circuits in-game and running simulation is described in Figure 3. For this simulation to be running in real time without affecting performance of our game, it is mandatory to establish standalone thread for Spice# calculations. Every working station can have only 1 running simulation at a time. The process of creating this instance is described in Figure 4.
- Integration of go-kart physics – player is able to sit in his go-kart and take it for a drive. This go-kart has fully interactive steering wheel as well as lever, with which player can change the speed.
- Interactive board – player is able to write whatever he needs on an interactive board. Player is also presented with several differently colored marker pens.



Figure 1 - Gridboard station



Figure 2 - Tuning station

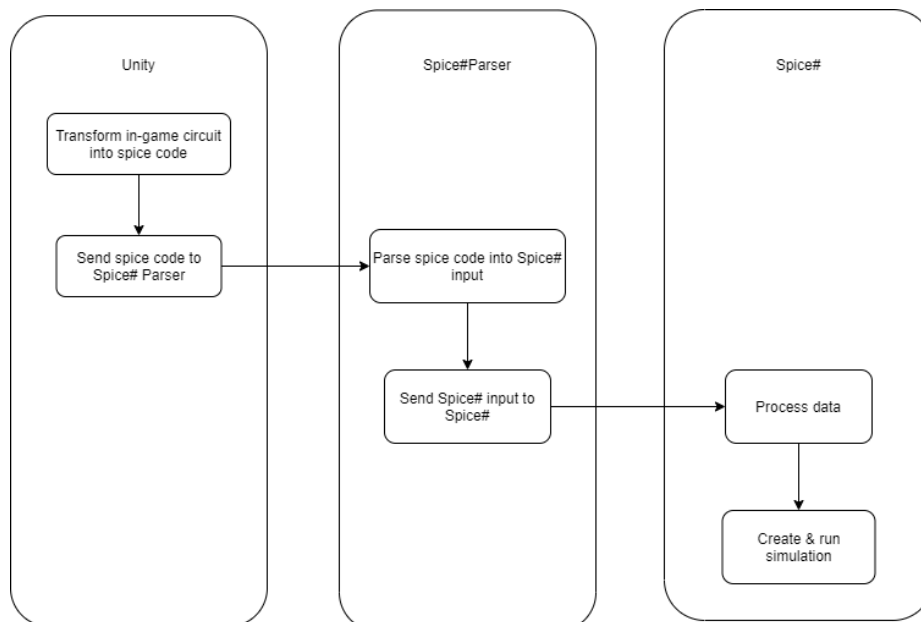


Figure 3 - Process of running simulation

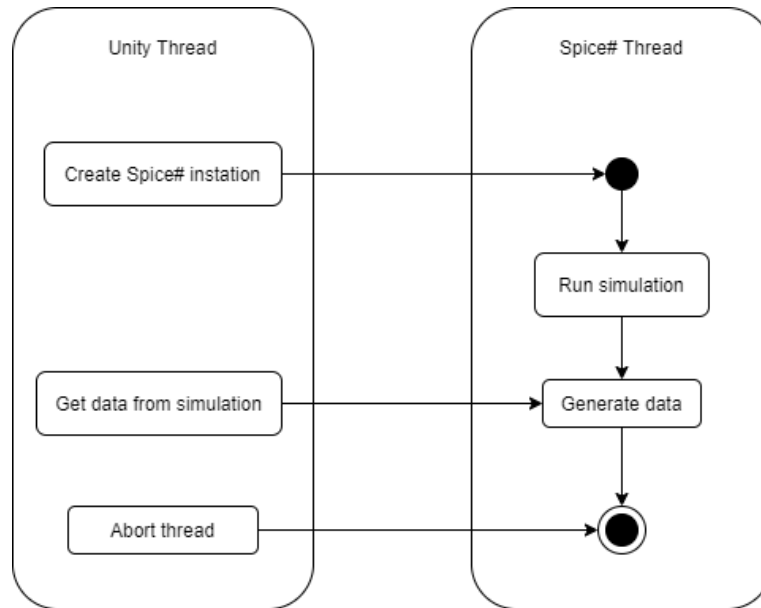


Figure 4 - Creating Spice# instance

5 Technologies used

In our project we work with several technologies:

1. Unity – game engine
2. VR Interaction Framework – collection of scripts and prefabs that aim to help with development of interactions in VR
3. Spice# - spice-based circuit simulator
4. Spice#Parser – parses spice netlists and simulates them using Spice#

6 Future work

- Currently our prototype only supports one client who controls the application from his local machine. One important aspect of future development is support for multiplayer collaboration for designing and creating the vehicle and its circuits.
- For now, we do not have the means to test our work with real participants. Testing would help us make improvements regarding our teaching material or help us solve some technical problems that could arise during the teaching process.

- We would also like to implement more functionalities for our go-kart model that could provide additional electrical components as learning material. With this comes the need for new electric component models and some new animations. Another aspect to consider is VR headset performance optimization for smooth experience.

7 Conclusion

In response to the current international pandemic, we propose an alternative solution to provide practical education and hands-on experience that substitutes physical in-person experience. We have demonstrated the plausibility of our approach with the aim to mimic the interactivity of real-world laboratory work leveraging virtual reality while keeping the students and teachers physically distanced and protected. We believe virtual reality has applications in education as it makes experiments safer, more accessible and without need for any physical resources other than the headset itself.

References

1. NICOLA, Maria, et al. The socio-economic implications of the coronavirus and COVID-19 pandemic: a review. *International journal of surgery*, 2020.
2. BOGUSEVSCHI, Diana; MUNTEAN, Cristina; MUNTEAN, Gabriel-Miro. Teaching and learning physics using 3D virtual learning environment: A case study of combined virtual reality and virtual laboratory in secondary school. *Journal of Computers in Mathematics and Science Teaching*, 2020, 39.1: 5-18.
3. NUANMEESRI, Sumitra; POOMHIRAN, Lap. Perspective Electrical Circuit Simulation with Virtual Reality. *International Journal of Online & Biomedical Engineering*, 2019, 15.5.
4. BOGUSEVSCHI, Diana, et al. Primary school STEM education: using 3D computer-based virtual reality and experimental laboratory simulation in a physics case study. In: *Ireland International Conference on Education, IPeTEL workshop*, Dublin. 2018.
5. CALVERT, James; ABADIA, Rhodora. Impact of immersing university and high school students in educational linear narratives using virtual reality technology. *Computers & Education*, 2020, 159: 104005.
6. Sven Boulanger, *SpiceSharp*, 2017.
<https://github.com/SpiceSharp/SpiceSharp>
7. Marcin Golebiowski, *SpiceSharpParser*, 2018.
<https://github.com/SpiceSharp/SpiceSharpParser>

Web-based teamwork support for the SystemVerilog language

Matúš Pilňan*

Slovak University of Technology Bratislava
Faculty of Informatics and Information Technologies
xpilnan@stuba.sk

Abstract. In this paper we deal with extending ASICDE, a web development environment for the SystemVerilog language with collaborative editing support. We analyzed the techniques of cooperation in the development of software projects and the possibility of their application to the design of digital systems. We discussed the availability of various tools that would be designed for the SystemVerilog language and at the same time support team collaboration in real time, but we did not find any product that would meet both of these requirements without any plugins or additions. We also analyzed approaches to concurrent document editing, specifically operational transformation. We have found several existing libraries that implement this technique at various levels. Finally, we analyzed ASICDE, into which we plan to implement the possibility of team cooperation. In doing so, we will deal not only with the concurrent editing of the source code, but also with concurrent editing of the diagrams of digital systems, which can be created in this environment and used for generating source code. We managed to implement this using an operational transformation library.

Keywords: Real-time collaboration · Web applications · SystemVerilog.

1 Introduction

Digital system designers often work on their projects in plain text editors, at best in Notepad ++, which supports only basic functionality such as syntax highlighting or command completion. However, these do not provide as wide a range of functionalities as modern development environments for common programming languages have to offer. Lack of such support can contribute to lower efficiency of digital system designers [1].

Another problem that can cause reduced work efficiency in this sector, and especially in the case of multi-developer projects, is the lack of solutions for teamwork. Developers of digital systems often work together on one file, but only with the help of video conferencing or screen sharing. However, even when

* Master study programme in field: Intelligent software systems Supervisor: Ing. Lukáš Kohútka, PhD., Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

using this method of real-time collaboration, their effectiveness may be reduced, as in the end, the file is modified by only one developer at a time. We therefore consider the lack of tools to enable the designers of real-time system designers to work together to be the main problem.

Due to the lack of support tools for team collaboration in the design and development of digital systems, we want to create a tool that will allow teams of digital designers to collaborate without the need to install add-ons in existing editors. We will deal mainly with the synchronization of source code in the SystemVerilog language, but also with the synchronization of diagrams for this language, from which the source code can be generated [2][3].

In the following section, we summarize the techniques used in collaborative applications. We also discuss libraries and frameworks relevant for our implementation. In the third and fourth sections, we outline the implementation of text and diagram synchronization respectively. In the last section, we summarize the results we achieved.

2 Related work

Collaboration in software development is not a new idea. Techniques such as pair programming, in which two developers work on the same code at the same time, have been known for more than forty years [4] and appear to be able to increase development efficiency [5] as well as assist in learning [6], or to help spread knowledge of the developed product [7].

Pair programming is an example of real-time collaboration [4]. The disadvantage of pairwise programming is the fact that programmers must be physically in one place [8]. This problem can be circumvented by the aforementioned screen-sharing video conferences, but it turns out that this approach may in fact have a negative effect on the performance of programmers [8][9]. However, using advanced technologies, such as sharing access to a computer instead of just sharing a screen, increases productivity [10], suggesting that allowing multiple developers to edit a single document can also have a positive effect on digital systems development.

2.1 Synchronized document editing

WebSocket is used for synchronized online document editing [11][12][13][14][15], due to its two-way communication capability [16]. The basic principle is that with this two-way communication, every change made to the file is sent to the server, which then sends it to other clients, and they make the change on their side [15]. The most important task when editing a single document at the same time by multiple users is to maintain consistency, so all copies must look the same after each edit. The image 1 shows a typical example of a consistency violation during concurrent editing. Both users start with the same text and edit at the same time. User A inserts the character "y" in position 1. User B deletes 2 characters from position 1, i.e. deletes "ab". Both parties shall inform the counterparty of

the execution of the operation. At this time, however, the delete (1, 2) operation on A's side no longer deletes the text "ab", but "ya", because adding a character changes the indexes of the following characters. Because these two operations are not commutative, each user has a different text after they are performed.

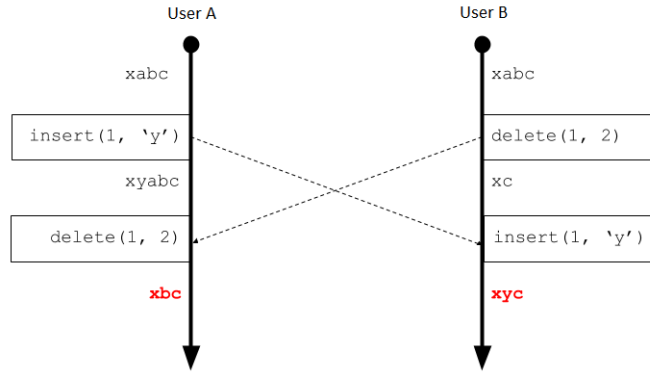


Fig. 1. Disruption of consistency when editing a document by two users at the same time.

The basis of the concurrent text editing and the solution of not only the above-mentioned problem is the technology of operational transformation [17][18]. The essence of this technology is a transformation function, which is used to modify any following operations so that the result is consistent. Thus, in the example above, the delete (1, 2) operation would be changed to delete (2, 2), thus taking into account the addition of the "y" character. This technology was originally designed to address concurrent document editing, especially for add and delete operations, but over the years, support has been developed for various other operations using various algorithms [19][20]. However, sharing source code is in some ways more complicated than sharing plain text because the program must have the correct syntax. Examples of this are the names of functions and variables that may need to be changed in several places at once, or constructions such as cycles and conditions. In these cases, we need to consider whether it's

appropriate to submit code modifications that cause errors elsewhere in the file. [21].

2.2 Libraries and frameworks

Because operational transformation is a relatively long-evolving field, and many collaborative web applications are emerging that use it, there are various libraries and frameworks that implement operational transformation. In addition to those mentioned below, there are various other libraries, the development of which, however, has been stopped and is therefore not relevant to us. We researched the libraries TogetherJS¹, ShareDB², Google Diff-Match-Patch³ and formic [22].

The above libraries are all open-source. The first two mentioned libraries offer the most functionality specially focused on collaborative editing in the web environment. As in the above-mentioned works, these libraries also use the WebSocket protocol. The Diff-Match-Patch library provides basic functions for comparison and searching in the text, it does not provide network communication between the editing parties and it would therefore be necessary to implement it separately.

System performance is also important for real-time collaboration. The key is the response time, so that the other parties see the changes made as soon as possible. This is mainly influenced by the number of users editing the document, as well as the frequency of edits. The formic project also compared itself with other tools [22]. When measuring performance, it is compared with ShareDB and Google Docs. In the experiment, this library achieved comparable performance to Google Docs and ShareDB, with higher (more than 40) modifications per second, achieving better results than Google Docs, but worse than ShareDB. However, we do not anticipate the need for group editing in teams so large that the response time would deteriorate significantly.

Of these four tools, the TogetherJS and ShareDB libraries seem to be the best options. While TogetherJS is easier to integrate into an existing application, using it would probably require intervention in its source code to work exactly as it would be appropriate in our application. However, its advantage is the built-in possibility of communication. On the other hand, ShareDB is intended for the simultaneous editing of any text or JSON documents, so it should not be necessary to interfere with its inner operation. This project is also more active, which is definitely an advantage. ShareDB also performed well in performance tests [22], which is another reason to use this library. However, if we decided to implement the possibility of communication, whether text or voice, we would have to deal with it separately. Overall, however, taking into account project activity, community, documentation quality and functionality offered, ShareDB seems to be the best option.

¹ <https://github.com/jsfiddle/togetherjs/graphs/code-frequency>

² <https://github.com/share/sharedb>

³ <https://github.com/google/diff-match-patch>

2.3 ShareDB

ShareDB is a JavaScript library for sharing and editing JSON documents⁴. It can utilize various database servers for storing documents. It has no built-in communication that could be useful for collaboration. However, according to the project repository on GitHub, it appears to be the most up-to-date of these libraries. It is also used by the DerbyJS⁵ web application development framework.

Unlike TogetherJS, the ShareDB library itself has no pre-built implementation for integration into an existing web application. It only takes care of editing and saving documents on the backend of the application, while exposing the interface to which the client part of the application connects via WebSocket. Each document saved using ShareDB is assigned a version number, the type of operational transformation to be used, and the document content itself. It also supports various types of documents, for example text or JSON documents. In addition to the content of the documents themselves, it also has a mechanism for sharing user metadata, which does not have to be stored and is only relevant at a given time. An example of such data may be the cursor position of individual users. This concept is called Presence.

The types of transformation used are provided by the ottypes library⁶ and differ in the synchronization algorithm used. The offered algorithms have different time complexity, which results in different synchronization speeds. The basic type for text synchronization has a time complexity of $O(M + N)$, where N is the complexity of the operation that will be modified after applying an operation of complexity M . This type of transformation does not satisfy Transformation Property 2 (TP2). This property is intended to ensure consistency when adjustments are made to different states of the document [19]. If necessary, it is also possible to use an implementation that meets this property, but according to the documentation, it is approximately 20 times slower. In addition, ShareDB does not allow modifications for which this property would be relevant, and therefore in most cases it is sufficient to use the basic implementation. Another type of transformation offered is the transformation of JSON objects. However, its implementation uses a slower algorithm than the one for plain text, with complexity $O(M * N)$. Therefore, it is recommended to use this type only when it is really necessary to modify JSON objects.

2.4 The ASICDE development environment

ASICDE is an integrated development environment (IDE) designed for the SystemVerilog language. It is a web application developed at STU as a subject of bachelor's [23][2] and diploma projects [24][3][25]. This IDE provides all the functionality known from commonly used IDEs, such as checking and highlighting syntax, adding, or working with Git. In addition, it provides the ability to design models of digital systems using diagrams and the conversion between these

⁴ <https://github.com/share/sharedb>

⁵ <https://derbyjs.com/>

⁶ <https://github.com/ottypes/docs>

diagrams and the SystemVerilog code. To edit the code, it uses Monaco Editor, an open-source text editor from Microsoft, which is also the basis of their popular product Visual Studio Code. The JointJS library⁷ is used to visualize SystemVerilog code using diagrams [26].

3 Text synchronization

When opening the editor, the application will work with the ID of the document selected from the file tree. A document with this ID will be subscribed to using ShareDB. Upon opening the editor, a connection to server-side ShareDB is established and the current copy of the document is obtained. The contents of the file from this document are displayed in the Monaco Editor. From this point until the file is closed, the application will listen for changes to the document from other users using ShareDB. At the same time, a presence connection is established via a channel whose identifier will be the ID of the open file. The presence connection is used to share the editing users' metadata.

3.1 Local change processing

A local change is a change that a user makes by writing to a file in the Monaco Editor. Monaco Editor provides the ability to listen to various events. One possible event is a change in the text. When this event occurs, we capture it and find out where the file was changed, what characters were added, and how many characters were deleted. This information can be provided directly by Monaco Editor. We then convert this data into the form of an operation that ShareDB can process and apply it to the document. ShareDB will take care of processing the operation and making the necessary adjustments in the database, as well as distributing changes to other connected clients. Because a cursor position often changes when you make a change to a document, we also use presence to send the new cursor position of the user who made the change. This information is also provided by Monaco Editor.

3.2 Remote change processing

When a remote change is accepted, the change is converted from a ShareDB operation format to a format suitable for Monaco Editor. Monaco Editor provides the ability to edit the contents of the editor directly from the code by defining a range of positions and text to replace that range. Adding text and deleting text are special cases of replacing a range of text. When adding, the text replaces a range of zero length, and when deleting, the text in the range is replaced with an empty string. However, when applying remote edits, care will be taken to ensure that these edits are not recaptured when listening to local edits and not sent to ShareDB, as this could create an endless loop of edits. Therefore, we will temporarily disable the posting of local edits while applying remote edits.

⁷ <https://www.jointjs.com/>

It's also possible that other users' cursors will change when edits are applied. However, the library for processing ShareDB operations on text documents provides the possibility to calculate a new position based on the specified ShareDB operation and the cursor position. Monaco Editor moves the user's cursor automatically when the changes are applied.

3.3 Sending metadata

Accompanying data, such as cursor position or selected text, will be sent using ShareDB presence. After establishing the presence connection when opening a file, a request is sent to obtain the presence of other participants. This request will display cursors and text markers for other users (fig. 2).

Another type of Monaco Editor event we can listen to is changing the cursor position. This will be very helpful when sending cursor position information. As we listen to these events, Monaco Editor will provide us with the row and column number of the new position of our cursor. This applies to all ways of changing the cursor position, whether with the keyboard, mouse or otherwise. When this event occurs, we send the new position through the presence to other participants working with the same file.

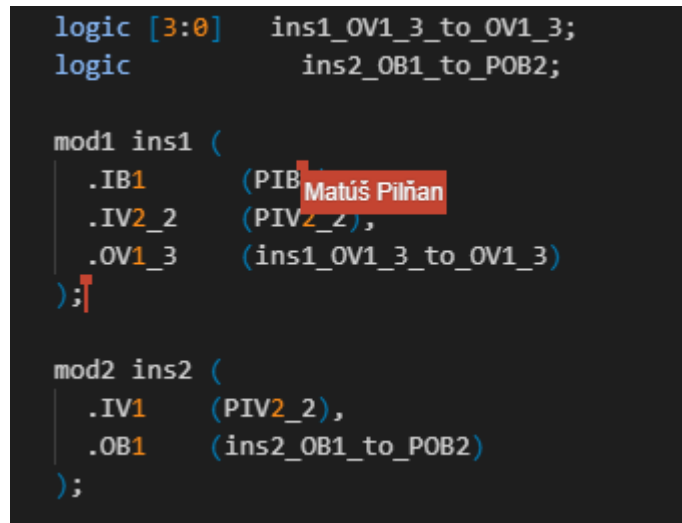


Fig. 2. Remote cursor with and without hover effect enabled.

As with the cursor position, the Monaco Editor sends an event when the text selection is changed. This event contains the start and end position of the selected text. Both will be sent to the other participants.

3.4 Processing metadata

When receiving a presence request, the application does two things: first, it sends its cursor position and highlighted text through the presence, and at the same time, because participants send this request only after opening the file, it displays a notification that a new user has joined the file. Since user names will also be sent in the presence, it will be possible to directly display the name of the newly connected user.

We will use a similar approach to synchronizing cursors as the Monaco Collaborative Extensions library uses. This library focuses directly on working with remote cursors and text markers in the Monaco Editor, but is not well compatible with the ngx-monaco-editor library used in the application. To display the cursors, we will use the so-called widgets provided by Monaco Editor. This is the way to display an HTML element in a given place in the Monaco Editor. The cursors of different users will be color-coded, but if one user has multiple instances of the application open, each of his cursors in the file should have the same color. This can be achieved by assigning a random color to the cursor, with the username being the basis for the random color generator.

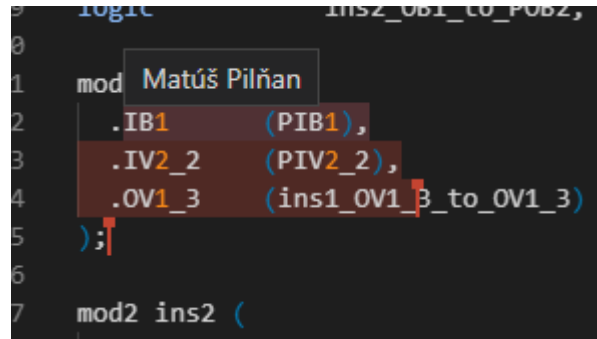


Fig. 3. Remote users' text selection displayed on local side.

4 Diagram synchronization

Diagrams will be synchronized via ShareDB as JSON objects. The structure of these objects will be defined according to the JointJS library. JointJS, like the Monaco Editor, offers the ability to listen to various events, which we will use to synchronize changes. The equivalent of displaying cursors and selected text when editing text in diagrams will be highlighting where the user clicks and marking the diagram elements they will work with.

4.1 Local change processing

A local change can be an addition, movement or removal of an element in a diagram. JointJS provides an API to subscribe to each of these events. Therefore, when any of these events occur, we use ShareDB to apply the appropriate change to the document with the visualizations. We apply the change to the diagram object, which will be stored under the keys of the current Git branch and the ID of the current visualization board. The diagram is represented by a list of cells. The change takes place as a cell is added, removed or replaced in the list. We will always append cells to the beginning of the list to ensure that we do not try to access to an invalid index. In case of removing and changing the cell, we will have to specify the index in the list that we want to access. We find this index by the unique ID that each cell of the JointJS diagram has. Changing a cell will be done as deleting and replacing it with a new version.

4.2 Remote change processing

When listening to changes in a ShareDB document, we have the operation that caused the change, as well as the affected diagram cell. Thus, depending on the cell ID, we will always know which cell to add, delete, or change in our diagram view. We will be able to make these changes in the diagram using JointJS. As with text synchronization, in this case we will have to be careful that the application of remote changes is not captured as other local changes, which could create an infinite loop.

4.3 Sending metadata

When collaborating on diagrams, we will share the position of the cursor between users when clicking on the diagram and the diagram elements they work with. Click position sharing will serve as a kind of pointer to facilitate communication between team members.

After clicking on the diagram area, the click coordinates within the diagram area are sent through the presence. On mouse click on an element, the grabbed element's ID is sent. On release, the same ID is sent with a release flag.

Also for visualizations, the request for the presence of other participants editing the same diagram will work analogously to the presence request when editing text.

4.4 Processing metadata

When a click position is received, the application will temporarily mark that position with a colorful "ripple" effect that will catch the user's attention and disappear after a few seconds. The color of the effect will depend on the user name, as will the colors of the cursors in the text.

When an ID of a grabbed element is received, that element is highlighted in the user's color. In addition, the ability to move that element is blocked to

prevent multiple users from moving the same element at the same time. On mouse hover over an element locked in this way, the name of the user who works with it will be displayed (fig. 4). It will not be necessary to lock other elements, as the position of the connections that connect it to other elements is automatically recalculated as the element moves.

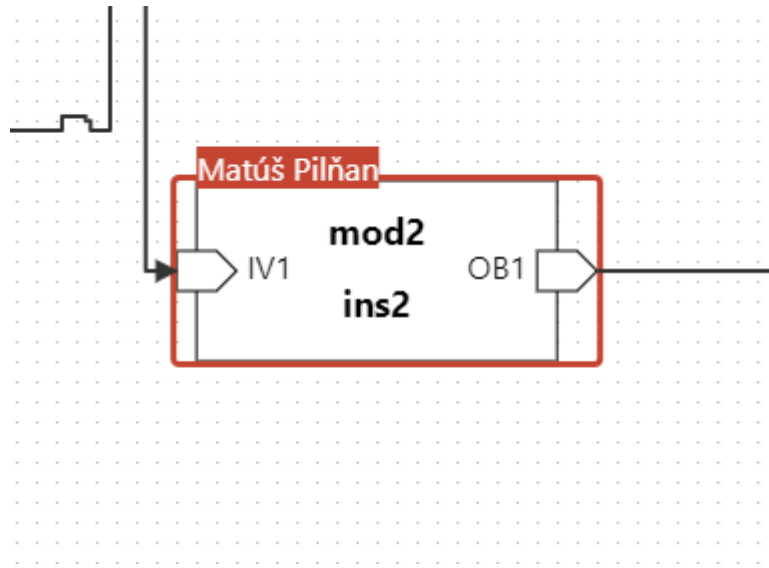


Fig. 4. Diagram element being moved by a remote user. The user's name shows on mouse hover.

5 Conclusions and future work

The aim of the work was to add support for synchronized editing of source code and diagrams, which can be used to express code in the SystemVerilog language, to the existing web IDE for the SystemVerilog language, ASICDE. As part of the project, we explored several tools that support the development of applications with real-time content synchronization.

We have succeeded in this goal and the ASICDE application now supports simultaneous editing of text files and diagrams. In addition, we've improved the user experience by using the application by eliminating user-burdensome obligations, such as having to manually save files before closing them and adding autosave to avoid possible data loss due to misuse or error. One known flaw of this solution could be a failure to send a local change if it happens exactly when a remote change is applied to local copy. This would be due to the Monaco Editor change subscription mechanism.

We tested the implemented solution using the Protractor⁸ end-to-end testing framework. We used it to simulate multiple writers editing a single document simultaneously. The expected result was that after writing sample text by all writers, each of those writers should have the same content in the editor. We have tested this behavior with up to 5 writers, since we are not expecting teams of more than five members. The tests were successful, which leads us to believe that the solution is correct, and usable in the context of our application.

The next part of this project is to polish some minor details, as well as get some feedback from the users. In the future, teamwork in ASICDE will be further improved by adding text and voice chat.

References

1. Kohutka, L., Stopjaková, V.: Chipde - a development environment for system verilog-based digital ic design. 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA) pp. 273–278 (2018)
2. Orlovský, K.: Podporný nástroj pre vývoj číslicovej elektroniky. Bachelor’s thesis, Slovenská technická univerzita v Bratislave, FIIT UPAT (2018)
3. Orlovský, K.: Podpora vývoja číslicových systémov pomocou modelovania a vizualizácie. Diploma thesis, Slovenská technická univ. v Bratislave FIIT UPAT (FIIT) (2020)
4. Jensen, R.: A pair programming experience (2003)
5. Vanhanen, J., Lassenius, C.: Effects of pair programming at the development team level: An experiment. p. 10 pp. (Dec 2005). <https://doi.org/10.1109/ISESE.2005.1541842>
6. Williams, L.: But, isn’t that cheating? [collaborative programming]. In: FIE’99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education. Conference Proceedings (IEEE Cat. No.99CH37011. vol. 2, pp. 12B9/26–12B9/27 vol.2 (Nov 1999). <https://doi.org/10.1109/FIE.1999.841619>, iSSN: 0190-5848
7. Zieris, F., Prechelt, L.: Does Pair Programming Pay Off? pp. 251–259 (May 2019)
8. Hanks, B.: Empirical evaluation of distributed pair programming. International Journal of Human-Computer Studies **66**, 530–544 (Jul 2008). <https://doi.org/10.1016/j.ijhcs.2007.10.003>
9. Parrish, A., Smith, R., Hale, D., Hale, J.: A field study of developer pairs: productivity impacts and implications. IEEE Software **21**(5), 76–79 (Sep 2004). <https://doi.org/10.1109/MS.2004.1331306>
10. Rajpal, M.: Effective distributed pair programming. pp. 6–10 (May 2018). <https://doi.org/10.1145/3196369.3196388>
11. Gao, L., Gao, D., Xiong, N., Lee, C.: CoWebDraw: a real-time collaborative graphical editing system supporting multi-clients based on HTML5. Multimedia Tools and Applications **77** (Nov 2017). <https://doi.org/10.1007/s11042-017-5242-4>
12. Kurniawan, A., Soesanto, C., Wijaya, J.: CodeR: Real-time Code Editor Application for Collaborative Programming. Procedia Computer Science **59**, 510–519 (Dec 2015). <https://doi.org/10.1016/j.procs.2015.07.531>

⁸ <https://www.protractortest.org/>

13. Katayama, S.Y., Goda, T., Shiramatsu, S., Ozono, T., Shintani, T.: A Fast Synchronization Mechanism for Collaborative Web Applications Based on HTML5. In: 2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. pp. 663–668 (Jul 2013). <https://doi.org/10.1109/SNPD.2013.13>, iISSN: null
14. Inoue, R., Kato, Y., Goda, T., Ozono, T., Shiramatsu, S., Shintani, T.: A Real-Time Collaborative Mechanism for Editing a Web Page and Its Applications. In: 2012 Fifth International Symposium on Parallel Architectures, Algorithms and Programming. pp. 186–193 (Dec 2012). <https://doi.org/10.1109/PAAP.2012.35>, iISSN: 2168-3042
15. Fan, X., Li, G., Song, X., Zhang, Shi, du, d.: CoVSCode: A Novel Real-Time Collaborative Programming Environment for Lightweight IDE. *Applied Sciences* **9**, 4642 (Oct 2019). <https://doi.org/10.3390/app9214642>
16. WebSocket: RFC 6455: The WebSocket Protocol, <http://www.hjp.at/doc/rfc/rfc6455.html>
17. Ellis, C.A., Gibbs, S.J.: Concurrency control in groupware systems. In: *Acm Sigmod Record*. vol. 18, pp. 399–407. ACM (1989)
18. Overleaf: Overleaf, <https://www.overleaf.com/>
19. Ellis, C.A., Sun, C.: Operational transformation in real-time group editors: issues, algorithms, and achievements. In: *Proceedings of the 1998 ACM conference on Computer supported cooperative work*. pp. 59–68. Citeseer (1998)
20. Sun, C.: Undo As Concurrent Inverse in Group Editors. *ACM Trans. Comput.-Hum. Interact.* **9**(4), 309–361 (Dec 2002). <https://doi.org/10.1145/586081.586085>, <http://doi.acm.org/10.1145/586081.586085>
21. Levin, S., Yehudai, A.: Collaborative Real Time Coding or How to Avoid the Dreaded Merge (Apr 2015)
22. Jungnickel, T., Bräunlich, R.: formic: Building Collaborative Applications with Operational Transformation. In: Chen, L.Y., Reiser, H.P. (eds.) *Distributed Applications and Interoperable Systems*. pp. 138–145. *Lecture Notes in Computer Science*, Springer International Publishing, Cham (2017)
23. Starý, J.: Podporný nástroj pre vývoj číslicovej elektroniky. Bachelor’s thesis, Slovenská technická univerzita v Bratislave, FEI ÚIM (2018)
24. Starý, J.: Integrované vývojové prostredie pre jazyk SystemVerilog. Diploma thesis, Slovenská technická univ. v Bratislave FIIT UPAI (FIIT) (2020)
25. Sokolík, M.: Integrované vývojové prostredie pre jazyk SystemVerilog. Diploma thesis, Slovenská technická univ. v Bratislave FIIT UPAI (FIIT) (2020)
26. Orlovský, K.: Support for development of digital systems using modeling and visualization. In: *Proceedings in Informatics and Information Technologies Student Research Conference 2020*. pp. 126–136. SPEKTRUM STU Publishing, Bratislava (2020)

Web system intrusion detection based on computational intelligence methods

Jakub HUBERT*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovicova 2, 842 16 Bratislava, Slovakia
`xhubert@stuba.sk`

Abstract. This paper documents our research in field of web system intrusion detection with focus on anomaly detection methods. Our aim was to design and implement usable node IDS system for detecting HTTP-based attacks on application layer of RM OSI. We decided to develop a detection module that will be based purely on unsupervised learning methods. Every web system can provide normal data profile of HTTP communication for module's learning phase, so we believe that strict unsupervised approach can significantly boost usability of our IDS system in a real environment.

We evaluated most of the available computational intelligence methods and picked the ones that are efficient under strict unsupervised learning conditions. Detection model was trained for each URL segment individually. Classifiers based on LOF, RDA and Damerau-Levenshtein distance methods provided sufficient average detection rates and low false alarm rates. Experiments showed that most deceiving were URL segments containing authentication parameters in their payload.

As an input to our classifiers we used statistical features calculated from HTTP request payloads and Damerau-Levenshtein distance obtained by comparing tokenized payloads with expected tokenized payload pattern, specific for each URL segment.

Top layer of our proposed IDS system's detection module was composed of RDA metaclassifier. This Stacking ensemble architecture is employed to reduce false alarm rates and improve overall detection rate.

Keywords: Intrusion detection · Anomaly detection · Web attacks · HTTP · Unsupervised learning · IDS

1 Introduction

From the beginning of the public internet era was the protection of web systems against hacker attacks always a big matter. New types of attacks are emerging

* Master study programme in field: Intelligent Software Systems, Supervisor: Assoc. Professor Ladislav Hudec, Institute of Computer Engineering and Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

over time and we need to employ protective mechanisms to address them accordingly. One of the most important protective autonomous mechanisms are Intrusion Detection Systems (IDS). By the methodology of attack detection, we divide them into signature based, anomaly based and those using stateful protocol analysis. Anomaly based IDS systems are the only capable choice for detecting a *Zero Day*¹ attacks.

IDS systems use so called agents to collect various data for security analysis. Useful data for anomaly detection can be network traffic between web client and web server, web server's hardware usage, database server logs and more. Location of agent also plays a role. Application layer agents can collect unencrypted HTTPS payloads, while network layer agents can not. On the other hand, the network layer agents can provide more information about routing and lower level protocols.

We decided to pursue the anomaly detection approach of IDS systems. Signature based and hybrid IDS systems are still the most used in a real environment, because anomaly detection often leads to higher false alarm rates, lower detection rates and often complicated detection module training and deployment process. We believe there is still a lot to be discovered to improve likeability for anomaly detection systems to be employed in real scenarios. In next chapter, we describe existing studies and state of the art methods to detect anomalies in HTTP traffic between client and server of web systems.

2 Related work

Anomaly based intrusion detection is a subject of many research works. Wang, et. al. constructed effective IDS based on SVM² with usage of LDMRT³ with 99.31 % detection rate and 0.60 % false alarm rate on NSL-KDD dataset [1]. SVM method was also used in work of Parhizkar, E. and Abadi, M. [2], where they used more SVM classifiers reduced by BeeSnips algorithm. Their detection solution was evaluated on CSIC2010 dataset and reached 95.90 % detection rate and 2.82 % false alarm rate.

Mac, H. et. al. analysed usage of autoencoders in intrusion detection by comparing effectiveness of 3 variants of autoencoders in this use case. RDA - regularized deep autoencoder proved to be the most effective with 94.6 % detection rate on CSIC 2010 dataset [3]. Wang, J., et. al. evaluated usage of LSTM⁴ and CNN⁵ in web attack detection on CSIC 2010 dataset. CNN achieved 98.3 % and LSTM achieved 97.9 % detection rate.

Vartouni, M. et. al. experimented with anomaly detection in HTTP traffic n-gram model using Stacked Autoencoder (SAE) as feature extraction method and Isolation Forest as a final classifier. Their solution was evaluated on CSIC2010

¹ Zero Day attacks are new, undiscovered types of attacks

² Support Vector Machines

³ Logarithm marginal density ratios transformation

⁴ Long Short Term Memory

⁵ Convolutional Neural Network

dataset with more SAE configurations the best configuration achieved 88.34 % detection rate. Best performing configuration was using sigmoid decision function and Adam’s optimizer. [5]

Nguyen, H. et. al. experimented with application of the Generic Feature Selection in detection of web attacks. Feature selection method was used to reduce feature set with minimal effect on classifier’s performance. Experiments showed, that even after the removal of 63 % of the features, detection accuracy was lowered by only 0.12 %. CFS⁶ proved to be superior feature reduction method for CSIC 2010 dataset. Random Forest classifier achieved on CFS reduced feature set 93.68 % detection rate with 7.2 % false alarm rate. [6]

Kozik, R. et. al. proposed a method [7] for modelling the normal behaviour of web application based on HTTP requests. Method was segregating incoming HTTP requests by URL and HTTP method. Unknown URL and HTTP method combinations were automatically classified as anomalies. If request had parameters, histogram feature vector was constructed and used as an input to classifier. Proposed method was evaluated on CSIC 20120 dataset with Naive Bayes, AdaBoost, PART and J48 Tree classifiers. Best detection rate - 95.97 % and 3.54 % false alarm rate was achieved by J48 Tree algorithm.

3 Proposed intrusion detection method

We propose a new stacking ensemble model for intrusion detection in HTTP traffic of web systems, based on stacked autoencoder. Our main goal is to support fully unsupervised approach, where optimization, training and deployment of such model and its subcomponents doesn’t require any anomalous data to be present. We think, that this aspect is the key of success of the method in a real environment. Classifiers of the first ensemble layer were chosen by separate evaluation of their performance under strict unsupervised conditions. The final proposed ensemble is showed on the figure 1.

3.1 Model segmentation

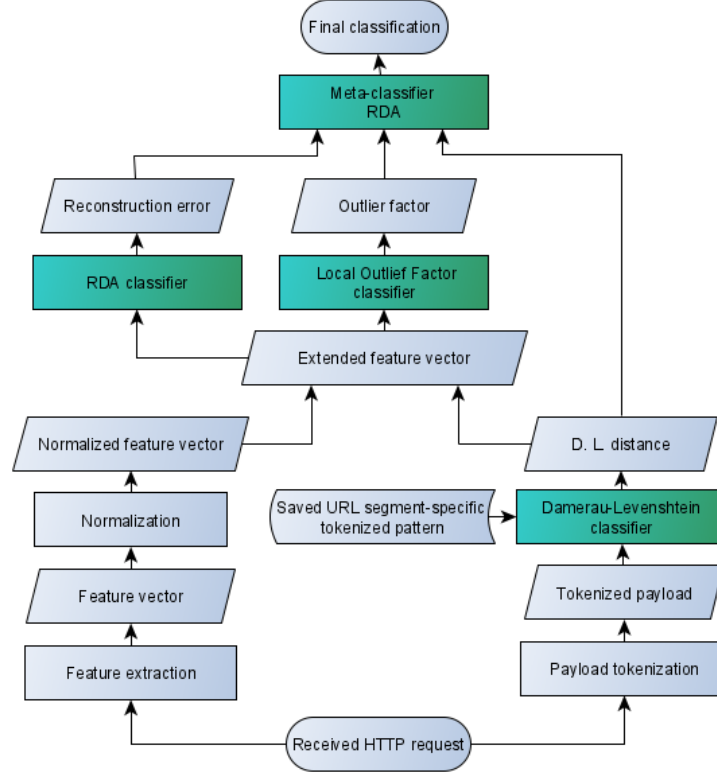
Similarly as in Kozik’s work [7], model training and optimization is done for each URL segment and HTTP method combination separately. This approach leads to higher dimension of simpler models that learn normal behaviour profile of specific web system routes.

3.2 Process of choosing first layer classifiers

Classifier’s ensemble requires well performing components based on various methods to be efficient. We decided to select first layer of classifier’s ensemble by separately evaluating more classifiers based on different anomaly detection methods to select the best ones. We used CSIC 2010 dataset. The table 1 shows results of evaluation of individual classifiers.

⁶ Correlation Feature Selection

Fig. 1. Proposed classifier's ensemble



3.3 Feature extraction methodology

IDS agent is located on application's layer of RM OSI has an access to decrypted HTTP communication between client and the web server. HTTP headers and payloads can be processed in it's plain text form in a various ways. We identified two groups of usable feature extraction methods:

- *statistic methods*: computing counts of certain character group occurrences and optionally applying statistical functions (average, eg.) to them
- *text methods*: extracted with language processing, or string similarity methods like TF-IDF⁷, Damerau-Levenshtein distance and other methods

Feature set was derived from existing research paper[6]. We picked 5 best performing statistical features and added 3 new features, which were URL segment specific. Damerau-Levenshtein distance feature was extracted using string

⁷ Term Frequency-Inverse Document Frequency

Table 1. Evaluation of level 1 classifiers

Classifier	Recall	FAR	Notes and observations
One Class SVM	0.8750	0.4381	Proved to be very hyperparameter sensitive. Higher FAR rate was caused by inability to build and optimize model for certain URL segments without attack samples. Strict unsupervised approach failed.
Reg. Deep Autoencoder	0.8765	0.0594	Model's hyperparameters can be optimized with just normal data profile. Recall and FAR tradeoff can be modified in runtime by changing the α constant.
Damerau-Levenshtein	0.8156	0.0643	Requires segment-specific pattern list for each protected URL. α constant can be used to moderate Recall and FAR tradeoff similarly as in RDA.
Isolation Forest	0.4507	0.2263	Seems to be efficient only if is optimized and trained on combination of normal and attack samples. Strict unsupervised approach failed.
Local Outlier Factor	0.8471	0.0181	Efficient in strict unsupervised approach when used default implementation's hyperparameters. We should evaluate model optimization with just normal data profile in the future.

similarity method, but we used it as an statistical feature set extension in this context. The final feature set is shown in the table 2.

Table 2. Extended set of features

Nr.	Feature name (abbrev.)
1.	Special characters count (SCC)
2.	Request length (RL)
3.	Digits characters count (DCC)
4.	Letter characters count (LCC)
5.	Total parameter length (PL)
6.	Parameter values average length (PVAL)
7.	Parameter count (PC)
8.	Damerau-Levenshtein distance (LEV_DST)

3.4 URL segment-specific payload pattern and Damerau-Levenshtein distance classifier

Computation of Damerau-Levenshtein distance requires a reference value to compare the incoming HTTP requests with. It is natural to assume, that each URL segment serves a specific purpose and often contains different parameters and their specific value patterns. Therefore we propose explicit definition of URL segment patterns via tokenized notation.

Tokenized notation maps legal parameter names to symbol P , string of numeric characters as N and string of alphabetic characters as S . Special characters are left unmapped.

Damerau-Levenshtein classifier's training phase will consist of collecting HTTP request data and computing their Damerau-Levenshtein distances. Average Damerau-Levenshtein distance \bar{D} , standard deviation s and tradeoff constant α are then used to compute anomaly threshold θ as stated in the equation 3.2.1.

$$\theta = \bar{D} + \alpha.s \quad (3.2.1)$$

In the protection mode without ensemble, HTTP requests that have Damerau-Levenshtein distance greater or equal than computed anomaly threshold θ are considered as web attacks.

In an ensemble mode, there is no training phase and the observed Damerau-Levenshtein distance is passed right to the meta-classifier as it's input.

4 Evaluation

Proposed classifier's ensemble for web system intrusion detection was evaluated on CSIC 2010 dataset. Dataset contains XSS attacks, various injection attacks, buffer overflows, information gathering, files disclosure, parameter tampering and other. For evaluation purposes, we picked URL segments from this dataset, that contained at least 500 normal and 500 anomalous samples. For these segments we optimized and trained separate classifier ensembles.

Level 1 classifiers RDA and LOF were optimized using *Grid Search* algorithm. Optimization process used only normal data profile to simulate fully unsupervised learning conditions. Selection of best hyperparameter setting was done using *10 fold Cross Validation*. Key validation metric was *accuracy* - we did not have access to *false alarm rate* and *detection rate* metrics. Training process of level 1 classifiers used 80 % of the available normal profile instances. Remaining 20 % were left out for test purposes.

RDA meta-classifier was optimized and trained on the outputs of level 1 classifiers. Afterwards we tried it's intrusion detection efficiency on left out normal profile instances combined with all anomalous instances and compared the average results with existing research papers.

Table 3. Evaluation of proposed classifier’s ensemble

Classifier	Recall	FAR
Parhizkar - SVM ens. with Beesnips [2]	95.90 %	2.82 %
Nguyen - RF [6]	93.68 %	7.20 %
Nguyen - CART [6]	93.71 %	6.80 %
Kozik - J48 [7]	95.97 %	3.54 %
Kozik - Naïve Bayes [7]	88.89 %	6.26 %
Kozik - Ada Boost [7]	83.23 %	15.45 %
Our RDA classifier’s ensemble	86.01 %	1.13 %

5 Conclusion and future work

Ensemble architecture fulfilled it’s purpose and meta-classifier’s false alarm rate was lower than any individual classifier false alarm rate. Our proposal method achieved lowest false alarm rates of all compared existing methods in the table 3. Proposed intrusion detection model is capable of optimization, training and deployment in fully unsupervised manner. There is no need to simulate anomalous data by penetration testing tools, or generating synthetic data to use this model in a real environment.

Evaluation process showed that URL segments that contain authentication parameters are significantly reducing overall detection rate. We suppose that this phenomenon is caused by the fact that password parameters include various special characters that can deceive model’s perception. Average detection rates were above 99 % if we do not include these URL segments in the evaluation. Future work on this model should be aimed on designing better feature processing methods that will reduce the negative effect of authentication parameters on detection rate.

References

1. Wang, H., Gu, J., Wang, S. An effective intrusion detection framework based on SVM with feature augmentation. In *Knowledge-Based Systems*. ISSN 0950-7051, 2017, vol. 136, p. 130–139.
2. Parhizkar, E., Abadi, M. OC-Wad: A One-Class Classifier Ensemble Approach for Anomaly Detection in Web Traffic. In *23rd Iranian Conference on Electrical Engineering (ICEE)*. Tehran : 2015.
3. Mac, H., Truong, D., Nguyen, L. et. al. Detecting Attacks on Web Applications using Autoencoder. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, 2018, p. 416 - 421.
4. Wang, J., et. al. Evaluating CNN and LSTM for Web Attack Detection. In *ICMLC 2018: Proceedings of the 2018 10th International Conference on Machine Learning and Computing*. New York : 2018, p. 283–287.
5. Teshnehlab, M. et. al. An Anomaly Detecion Method to Detect Web Attacks Using Stacked Auto-Encoder. In *6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*. Tehran : 2018.

6. Nguyen, H., et. al. Application of the Generic Feature Selection Measure in Detection of Web Attacks. In Proc. of International Workshop in Computational Intelligence in Security for Information Systems (CISIS 11), LNCS 6694, 2011, p. 25-32.
7. Kozik, R., et. al. A Proposal of Algorithm for Web Applications Cyber Attack Detection. In *Computer Information Systems and Industrial Management, Berlin, Heidelberg..* Springer Berlin Heidelberg, 2014, p. 680–687.

Controlled generation of realistic human faces

Róbert Belanec^{*}[0000–0002–8340–9922]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
belanecrobert22@gmail.com

Abstract. In this work we present a method of controllable image synthesis for the state-of-the-art generative model - StyleGAN2 [12]. We describe the process how we managed to control generated outputs by finding a feature axis in generator’s latent space for a single feature, and how it is applicable to bigger number of features. Using this method to generate human faces with selected attributes, we were able to see modified results that differed from its unmodified version mostly by this selected attributes. Due to low accuracy of our trained classifier, we were unable to find the exact feature axis, but we were able to increase occurrence of images with selected attribute and therefore we were successful in controlling StyleGAN2 generation.

Keywords: Image synthesis · Controlled generation · Generative adversarial networks · StyleGAN2 · Neural networks.

1 Introduction

Drawing is one of the first activities used by human, that helped our ancient predecessors to express information and communicate to other peers. Today, we can use variety of drawing software to express our thoughts, or to capture and store visual information, so we can recall to it later, or share it with other people.

This information-to-image transformation often requires time, depending on the quality of result and amount of information, we would like to transform. In some cases, there is not enough time to properly capture every detail, or to deliver results in high quality. But what if we need to produce a high quality, detailed drawing within a short period of time? Primary examples are missing persons descriptions and perpetrator descriptions, which are then transformed by a police draftsman. That is why we decided to use state-of-the-art generative model, to generate human faces. To simulate a police drafts drawing from person description, we need to control the output.

Over the last decade, we are witnessing a breakthrough in generative algorithms, mostly because of the invention of generative adversarial networks (GAN) [8].

^{*} Bachelor study programme in field: Informatics Supervisor: Assoc. Professor Peter Lacko, SAP Labs Slovakia, SAP Slovensko s.r.o.

GAN model consists of two main parts a generator and a discriminator. A generator synthesises a fake samples, while the discriminator tries to distinguish between real and fake samples. Both generator and discriminator are performing unsupervised learning, where generator is trying to fool discriminator and discriminator is trying to outsmart generator. As stated in [8] by playing this adversarial min-max game, generator learns how to produce realistic images from random noise which is often represented by Z . This primary generative architecture has gone through a series of improvements, and in the year of writing this paper, we know about architectures, that are capable of generating more realistic images, and in some cases, we cannot easily determine, if the image was generate by a computer, or shot by a human using camera. Among current image generating, state-of-the-art models, which are based on GAN architecture are StyleGAN [11], StyleGAN2 [12] or BigGAN [4].

Each one of these models is capable to generate realistic images, which is important, because higher similarity with the real world helps people to recall memories. We also need to control features based on the description, therefore we assume that StyleGAN models are better for our work, because of their latent space disentanglement improvement.

2 Related work

Currently existing GAN works are mostly focused on quality of synthesis [4, 10–12, 16] rather than controlling its generation. Many works also proposed to reverse the mapping from the latent space to the image space [3, 9, 18], which is a good application for realistic image generation.

Some studies were focused on researching GAN latent space and its properties [19], which we can use to better understand what is happening inside GAN and understand its learning process [2].

There have been several attempts to control GAN generated samples [1, 5, 15, 17, 21]. One of the most popular methods to control generated samples is conditional learning. In conditional learning dataset labels are encoded as an additional image channel. This modified images are passed to discriminator, while training. Downside of this method is that, we need a labeled dataset to train GAN and we cannot simply swap or combine multiple features after training.

Few articles are also focusing on how to take an advantage of known latent space properties and control the output of generation [13, 20]. Knowledge from this articles turned out to be important part in our work.

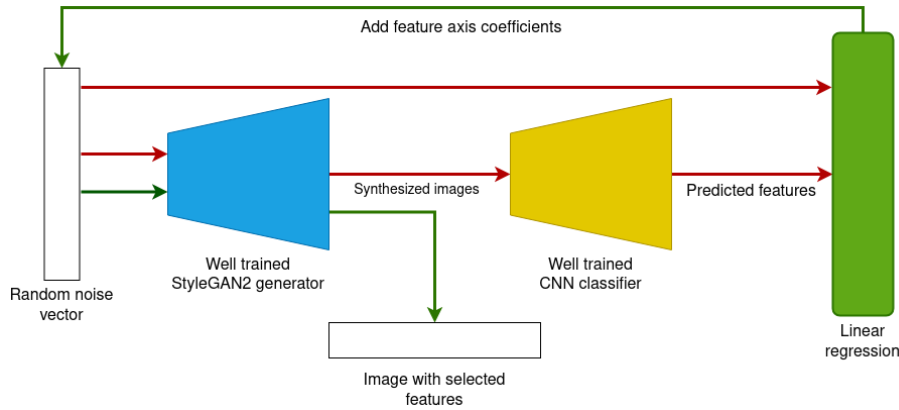


Fig. 1. Flowchart representing our method of controlling StyleGAN2 generator

3 Generating controlled images

We are presenting a method, that we used to control StyleGAN2 generation of samples, by finding a feature axis in the latent space of pretrained StyleGAN2 generator as seen in figure 1. We can simply divide this method into following steps¹:

1. Finding a latent space distribution by training a generator
2. Training a convolutional neural network classifier with the features that we want to control
3. Classifying samples generated by our generator with our trained classifier
4. Using linear regression to regress feature axis

In this section, we will describe these steps in more detail.

3.1 StyleGAN2 generator

As previously described, the very first step to control generation is to train a generator, to find a good latent space representation. For that we have used official StyleGAN2 implementation created by Nvidia Labs.²

This official implementation includes a manual for training and generating images. If we wanted our results to be in high quality and easily recognizable to humans, we chose to use pretrained weights, also offered by official implementation. Training this architecture for a high quality results, on a high quality

¹ Steps 1 and 2 can be skipped, if using pretrained models

² Official StyleGAN2 implementation taken from <https://github.com/NVLabs/stylegan2>

dataset would also require a graphics card with at least 16GB of global memory and a total time of 69 days and 23 hours on a single GPU. Therefore we used *config-f* configuration trained on the Flickr-Faces-HQ dataset.³ StyleGAN2 model using this weights is capable to generate detailed images of human faces in 1024x1024 resolution as can be seen in figure 2.



Fig. 2. Samples generated by StyleGAN2 with config-f-ffhq pretrained configuration

3.2 Image classification

Next in order, we need to classify our generated images for features that we would like to control. For this purpose, we chose to classify if the person in the image is wearing eyeglasses. Therefore we have created a simple convolutional neural network based on Xception network [6], pretrained on Imagenet dataset [7] and added some custom layers, like layers for average pooling, dropout layer and a sigmoid activation output layer. We have used Adam optimizer and a binary cross-entropy, to calculate loss.

We have trained our model on CelebA dataset [14], which contains 202,599 number of face images of 178x218 resolution and with 40 binary attributes annotations per image. For proof-of-concept purposes, we have only choose mentioned *Eyeglasses* attribute. While training, we were monitoring training loss and AUC, as well as validation loss and AUC. At the end of training (10th epoch), the best validation loss was 0.0604 with validation AUC of 0.4187. Test AUC was exactly 0.5 and we have also calculated test accuracy of 93.542 and test F1 of 0.

After training of our model, we have generated 48,695 images of 1024x1024 resolution and resized them to images of 178x218 resolution using cubic inter-

³ Full dataset can be found on <https://github.com/NVlabs/ffhq-dataset>

polation. After that, we have used our trained classifier to predict labels Y for every image to create pairs with latent vectors Z .

3.3 Feature axis regression

Our last step for determining the axis of selected feature is using correlation between noise vectors Z and predicted labels Y in linear regression algorithm. For input vectors $Z = z_1, z_2, \dots, z_n$ and labels vector $Y = y_1, y_2, \dots, y_n$ found relationship could be represented by a mathematical function with model of form $Y = X\beta + \epsilon$. Where β is dimensional parameter vector which contains a coefficient for each dimension and ϵ is error. Fitting linear regression model therefore means estimating regression coefficients β , such that error $\epsilon = Y - X\beta$ is minimized.

After we fit a linear regression predictive model for a selected feature, where its slope β is representing a feature axis of this selected feature. Now we can use coefficients of this linear regression model to move input noise vector Z along the feature axis.

With found feature axis, we can now add its transposed coefficients β^T to vector Z , and input this modified latent vector to our StyleGAN2 generator to generate modified image, with our *Eyeglasses* feature.

Linear regression can also be used on a set of multiple features, generalizing linear regression. If we have a set of feature labels i.e. $\{Y_1, Y_2, Y_3, \dots, Y_m\}$, linear regression then returns a set of functions $\{f_1, f_2, f_3, \dots, f_m\}$, by applying linear regression for each feature. Generalized linear regression can be used to find axis for multiple features.



Fig. 3. Examples of classified samples. Numbers over the images represent the probability of image with a person that is wearing glasses (1 - this is an image with person wearing glasses, 0 - this is not an image with a person wearing glasses)

4 Results

In this section, we will present and evaluate our generated results. Our evaluation will be mostly manual, because we are comparing primarily qualitative visual changes in images. Firstly, we have generated 25 of modified images and compared them to 25 not modified images, both of them were generated from a different input noise vectors. This allows us to see if we found the right axis. If we did, all images from modified batch should contain a person wearing glasses. Unfortunately, not all of the images generated from modified noise vectors contained a person with glasses, but we can say, that the probability of generating an image of a person with glasses has raised from average 3 of 25 images (not modified) to average 8 out of 25 (modified) images. From this, we cannot clearly determine if we were successful in controlling StyleGAN2 generation.

Therefore we tried a different testing approach, where we generated multiple pairs of images generated from the same latent vector, but one of them was modified with our feature axis coefficients. With this approach we found out that we were able to control generated output by finding a feature axis. As the first test has shown, we didn't really find exact feature axis. We think that, this misdirection happened by a poorly trained classifier (as seen from evaluation metrics at chapter 3.2) because as we can see on the figure 3, a person wearing eyeglasses was predicted with really low score, which created a misdirection later in our linear regression.

As we can see on figure 4, adding eyeglasses feature into the image also slightly affects other features like gender, ears or hair as can be seen on figure 5. This effect is caused by feature entanglement, so while modifying one feature, we can end up with modifying more than one. This effect is more notable in different models than StyleGAN or StyleGAN2, because they introduced a mapping system, which consist out of fully connected layers, that maps latent vector Z to intermediate vector W and therefore easing feature entanglement.

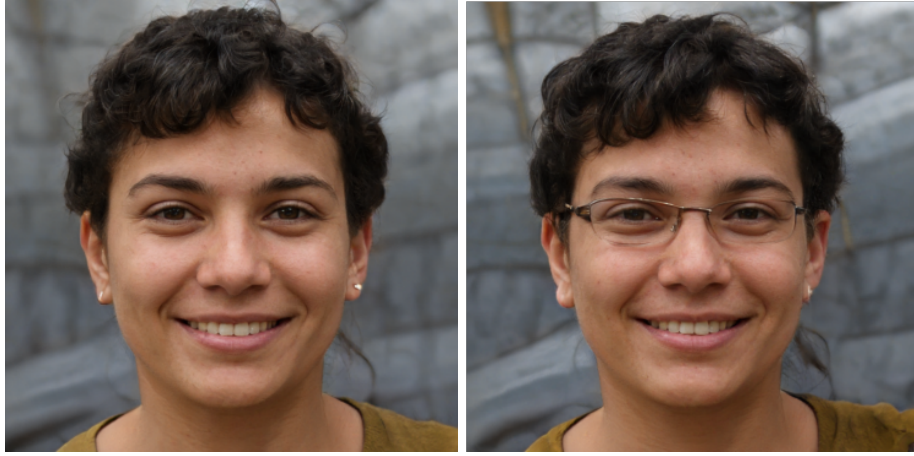


Fig. 4. Comparison of images generated from unmodified and modified latent vector sample



Fig. 5. Comparison of images generated from unmodified and modified latent vector sample, with lesser gender and hair modification as an unwanted effect

5 Conclusion and future work

In this paper, we present a method of finding a feature axis in latent space using linear regression. We use coefficients of found feature axis to move a latent vector from which we generate an image of human face with selected feature.

We conducted a series of manual visual tests, to see if we are able to control StyleGAN2 generation. We successfully found a feature axis, but due to lack of classifier training or wrong convolutional model selection, we have ended up also generating images without our selected feature.

In the near future, we would like to improve the classification of human faces with selected feature and apply this method to more or all features from CelebA dataset. After we successfully find multiple feature axis, we will be able to synthesise image from a set of features or a facial person description, simulating a police draftsman.

References

1. Antipov, G., Baccouche, M., Dugelay, J.L.: Face aging with conditional generative adversarial networks (2017), <http://arxiv.org/abs/1702.01983>
2. Bau, D., Zhu, J.Y., Strobelt, H., Zhou, B., and William T. Freeman, J.B.T., Torralba, A.: Gan dissection: Visualizing and understanding generative adversarial networks (2018), <http://arxiv.org/abs/1811.10597>
3. Bau, D., Zhu, J.Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., Torralba, A.: Seeing what a gan cannot generate (2019), <http://arxiv.org/abs/1910.11626>
4. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis (2019), <http://arxiv.org/abs/1809.11096>
5. Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., Abbeel, P.: Info-gan: Interpretable representation learning by information maximizing generative adversarial nets (2016), <http://arxiv.org/abs/1606.03657>
6. Chollet, F.: Xception: Deep learning with depthwise separable convolutions (2017), <http://arxiv.org/abs/1610.02357>
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
8. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2014), <http://arxiv.org/abs/1406.2661>
9. Gu, J., Shen, Y., Zhou, B.: Image processing using multi-code gan prior (2020), <http://arxiv.org/abs/1912.07116>
10. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation (2018), <http://arxiv.org/abs/1710.10196>
11. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks (2019), <http://arxiv.org/abs/1812.04948>
12. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan (2020), <http://arxiv.org/abs/1912.04958>
13. Lee, M., Seok, J.: Controllable generative adversarial network (2019), <http://arxiv.org/abs/1708.00598>
14. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
15. Mirza, M., Osindero, S.: Conditional generative adversarial nets (2014), <http://arxiv.org/abs/1411.1784>
16. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks (2018), <http://arxiv.org/abs/1802.05957>

17. van den Oord, A., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K.: Conditional image generation with pixelcnn decoders (2016), <http://arxiv.org/abs/1606.05328>
18. Perarnau, G., van de Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional gans for image editing (2016), <http://arxiv.org/abs/1611.06355>
19. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks (2016), <http://arxiv.org/abs/1511.06434>
20. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing (2020), <http://arxiv.org/abs/1907.10786>
21. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes (2016), <http://arxiv.org/abs/1512.00570>

Scheduling Tool for Deterministic Communication in Distributed Real-Time Systems

Veronika Búcsiová *

Slovak university of technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
veronika.bucsiova@gmail.com

Abstract. A network with critical data streams, where the timing of incoming and outgoing data is a necessity, is called a deterministic network. These networks are mostly used in association with real-time systems that use periodic hard-real-time data which need to be scheduled. The algorithm proposed in this paper is focused on achieving the lowest possible latency within the specified network topology depending on the physical parameters of the network and the periodic requirements of the communicating nodes. The algorithm allows to add priorities to the data flow between nodes and also the option to specify if preemption is allowed within the network, since not all devices support it. The result is a schedule of hard-real-time data within a specified network topology where all requirements of communicating nodes are met if such a solution is possible.

Keywords: Deterministic Scheduling · Time-sensitive Networking · Scheduling Algorithm · Ethernet · Deadline

1 Introduction

Every day more and more devices get connected to the Internet and other networks. These devices all have different requirements, regarding quality and timing of data transmission. Some of them need to transfer time-sensitive data, which means that not only does it matter that the data was received, it also matter when it was received by the destination. While we are perfectly fine if a media we stream is lagging a bit, may that be a bit troublesome, but if data about e.g. an overheating component of a machine gets delayed, that may have some more serious consequences [1]. Time critical data can be categorized into soft and hard real time data, the difference lies in the abiding of the deadline, in which a task must be completed. In hard real-time systems, missing a deadline is considered a failure of the whole system [16]. In large networks with

* Master study programme in field: Information Systems Supervisor: Dr. Lukáš Kohútka, Institute of Computer Engineering and Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

many endpoints scheduling time sensitive traffic can be problematic, because of conflicts that can arise if more than one endpoint wants to send data over the same path. To solve these conflicts and create a schedule for communicating endpoints the need for an algorithm has arisen. In this paper we will discuss communication planning procedures and propose a solution for scheduling data in a deterministic network.

2 Related Work

Publicly available tools for planning deterministic communication do not exist at the time of writing this paper. However, there is a large number of communication planning and shaping procedures that we will introduce in this section [19].

2.1 Quality of Service and priorities

The definition of Quality of Service (QoS) is not entirely unambiguous, as each technology or service defines it according to its needs. The most widely known standards in this area are the IEEE 802.1p and 802.1Q standards and their definition of QoS and their assignment of priorities [2] [20]. According to these standards IEEE proposed eight priority levels where 0 is the lowest priority and 7 is the highest. A recommended type of traffic was also assigned to these priorities. Zero priority traffic is recommended to be Best Effort traffic and the highest priority should be assigned to Network control traffic. Using these priorities, traffic can be shaped. The priority information is in the four-byte 802.1Q header, which is inserted into the Ethernet frame between the source MAC address and Ethertype fields [3].

2.2 Strict priorities

The algorithm for strict priorities is supported by CISCO routers. These routers support priorities. Each interface supports multiple streams which correspond to these priorities. According to the strict priority algorithm the higher priority packets get transmitted first. Packages from lower priority streams are not processed until packages from a higher priority stream are sent [4].

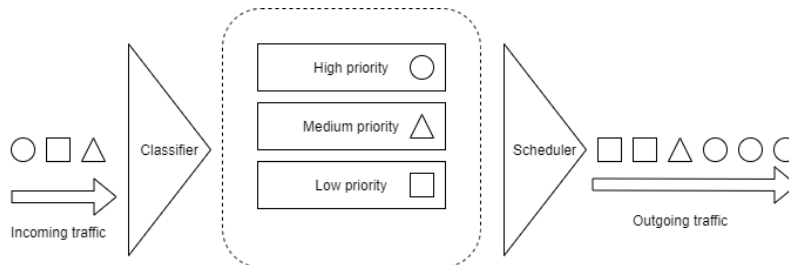


Fig. 1: Example of streams with strict priorities [5].

2.3 Weighted fair queuing

There is a number of variations of round-robin scheduling algorithms for data streams, which allow each stream to have a fair share of resources. It is possible to fairly distribute transmissions from all streams, but in reality, however, fairness does not mean that all streams can send the same number of packages [6]. This depends on what the algorithm deems as the weight (e.g. earliest finish time of processing). Priorities could be added to ensure that higher priority streams get to send more, in which case these higher priority streams will be served for longer than lower priority streams [7].

2.4 Hierarchical scheduling

Hierarchical scheduling is a framework that is supported and described by CISCO. It allow service providers to manage their QoS at three or four levels of the hierarchy. Instead of allocating an implicit resource guarantee to each queue, the three-level scheduling parameter uses the ratio of the remaining resources to allocate unused resources (such as bandwidth) to each logical queue [8].

The three-level hierarchy would look like this [8] :

- Physical Layer - Used to shape a physical interface, such as an OC-3 port.
- Logical Layer - Used to schedule child interfaces, such as VLAN or PPP sessions.
- Class layer - Used for series classes defined by QoS policy

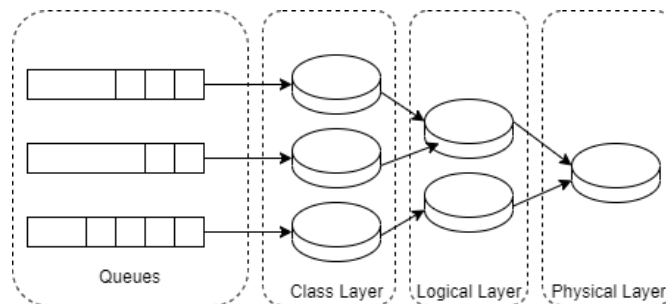


Fig. 2: Visualization of three-level hierarchy [8].

2.5 Credit based scheduling

Credit Based Traffic Shaping is used primarily for forwarding audio and video frames. The individual frames are organized into streams according to their priorities. Each stream is then assigned with it's own credit score. Frames are only sent if this credit is greater than or equal to zero. This traffic shaping is a great solution for achieving smooth video and audio transmissions. It adds flexibility as new data can be dynamically added to streams [9].

2.6 Time aware scheduling and preemption

Time aware scheduling is used primarily in industrial networks, where we can specify the reference time (defined according to IEEE 802.1AS) and the list of events. The event list consists of a time interval and the event itself. The event may be, for example, opening a port [10]. Of course, there may be a situation where a lower priority frame is still transmitted at the time of the event. For this case, a so-called guard zone is introduced. This guard zone is as large as the largest possible frame that could be transmitted and interfere with the planned communication. In case this band is too large, preemption is going to be used on the frame with lower priority. It is going to be divided into smaller fragments and the guard zone must therefore only be the size of the largest possible fragment [11] [12].

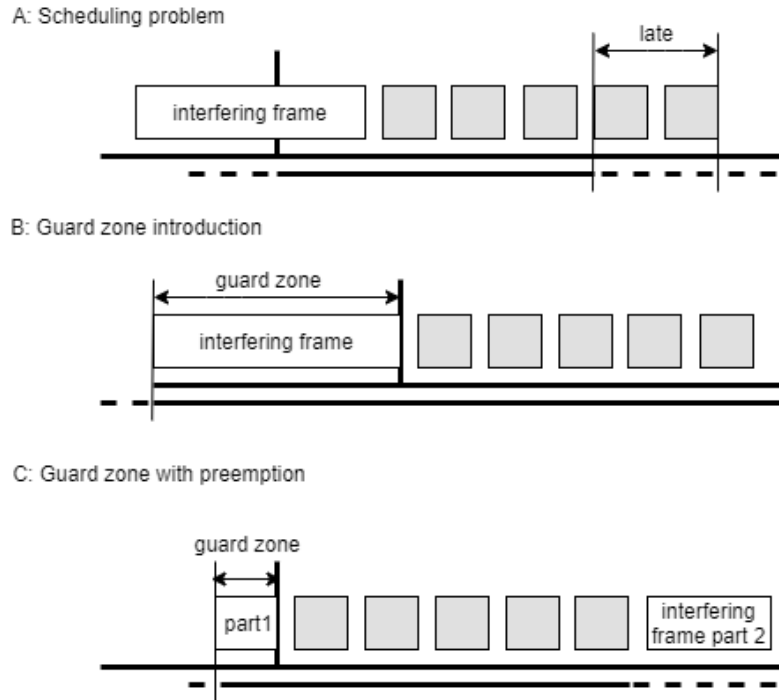


Fig. 3: Visualization of Time aware scheduling [11].

2.7 Scheduling in real-time distributed systems

There are multiple real-time systems that need hardware task scheduling and are using various approaches to achieve this. One approach is implementing more complex task cells, but this is not always possible, hence another approach should be used such as using a semaphore that solves conflicts whenever they occur [13] [15].

3 Proposed Solution

The previously mentioned methods focused primarily on prioritizing and scheduling data, in a way that they could determine, if the scheduling is possible. However they would be either not suitable for larger networks or they did not guarantee that all data, even the lowest priorities, would get transmitted in time.

The input for the algorithm contains data about the allowance of preemption, the topology of the network, such as the throughput, and latency as well as data about the communication pairs (period in which data is being sent, the data size and the priority of it).

The program then creates a weighted graph, where the weight of the edges is equal to the latency between nodes. For each communication pair the shortest path is then found and saved. Taking into consideration these shortest paths, the number of conflicts on the edges is then computed. If two or more communication pairs have the same edge in their shortest path, a schedule has to be made [14].

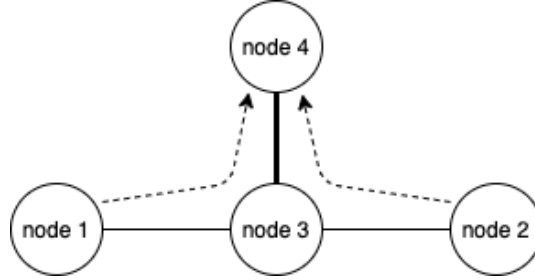


Fig. 4: Visualization of a path with conflict where both node 1 and 2 want to send data to node 4.

The algorithm labels all edges with conflicts on them as unsolved. It then starts with the edge that has the most communication pairs wanting to send data through them. At first a period has to be set, which is the lowest common multiple of all the periods of communication pairs that use that edge. Next the algorithm calculated the time that is needed to send all the data from all of the communication pairs. If this time is lower than the calculated period, then offsets for each communication pair that uses the edge are calculated, starting with the ones that have the highest priority.

If the time is greater than the calculated period, then a different route than the shortest has to be found, with emphasis on the lowest possible overall latency of the network. The algorithm uses hill-climbing to search for this solution.

In case that while solving a collision edge where one of the communication pairs already has an offset that has been set in earlier iterations, the option of preemption plays a role. If preemption is allowed in the topology, then data can be split into more packets to accommodate the already set offset. If preemption is not an option, then the already set offset is altered to be the last offset on the collision edge and a "backward control" is triggered. This is to see if the change has had impact on any previously solved conflict edges. If yes then the offset is

once again set anew to be the last offset on that already set edge and the control is triggered anew. This control is running until either an offset is found that doesn't interfere with any other data offsets, or until a solution is found. The proposed offset plus the time to send data has to be greater than the period on that conflict edge in which case a new path using hill-climbing has to be found.

The algorithm either finds a schedule solution where all communication pair prerequisites have been met with the lowest possible network latency, or informs the user that no solution is possible.

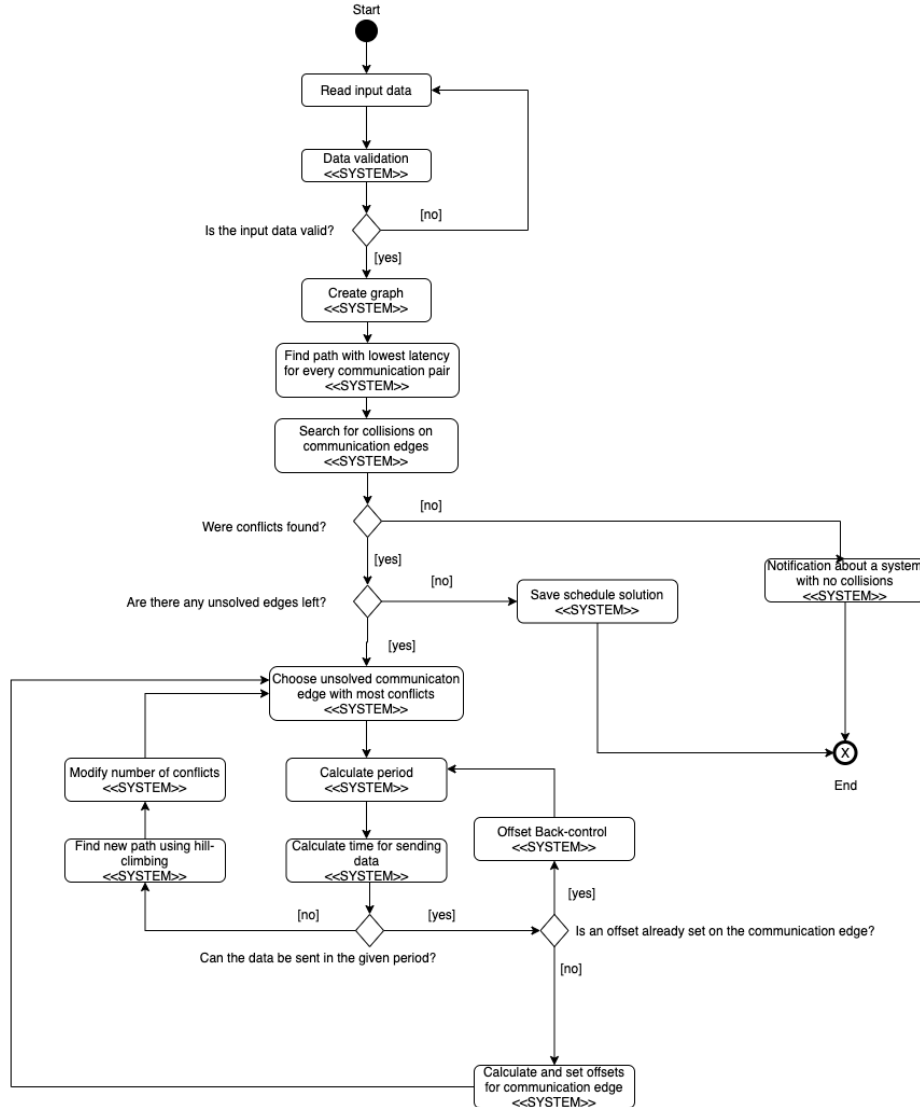


Fig. 5: Activity diagram of the proposed solution

3.1 Schedule example

To demonstrate the algorithm on an example, let's propose a simple topology as illustrated in Fig. 6. Both node 1 and node 2 want to send data to node 4 and node 3 wants to send data to node 5. The size of the data and time period of the communication are displayed in the table.

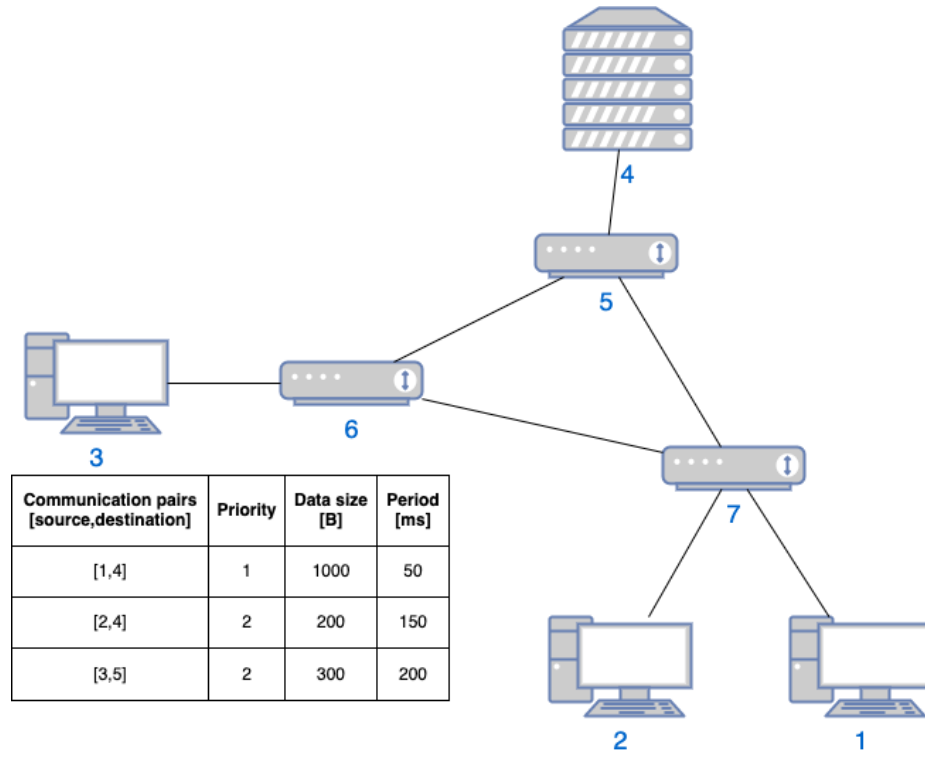


Fig. 6: Topology example with data about communication pairs.

The algorithm then creates a weighted graph of the topology as shown in Fig. 7. It then calculates the shortest paths from the source to the destination using Dijkstra's algorithm, where the weight of the edges is equal to the latency on that edge. In the next step it looks for conflicts on the edges. In this example both communication pairs [1,4] and [2,4] have the edge C5_7 in their shortest path. This means that this edge is marked as unsolved and the algorithm tries to create a schedule based on the period of both communication pairs.

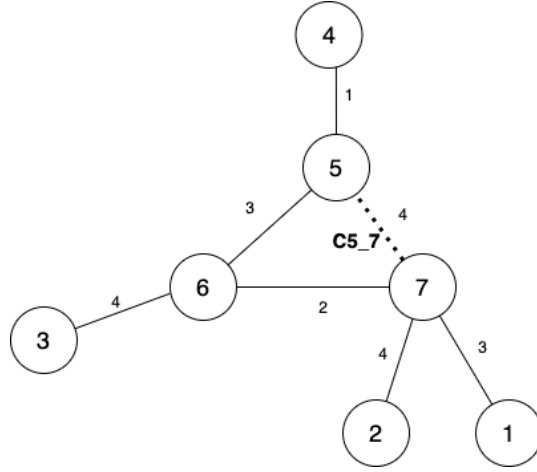


Fig. 7: Weighted graph of the topology.

Since the communication pair $[1,4]$ wants to send a big amount of data in a short time frame, it is not possible to also schedule the other communication pair. The pair $[2,4]$ would still be transmitting data and a conflict would arise. (Fig. 8, left). Since the communication pair $[1,4]$ has a higher priority, the algorithm tries to reschedule the pair $[2,4]$.

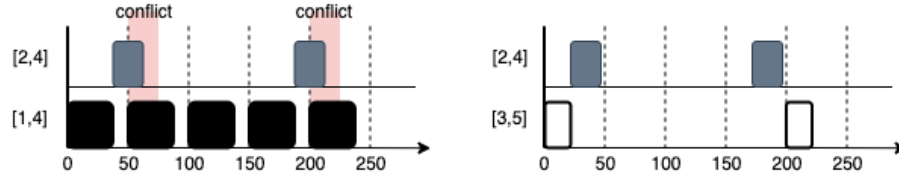


Fig. 8: Examples of schedules on different edges.

The algorithm finds a new path for the communication pair $[2,4]$, if such a path exists, with emphasis on the lowest possible latency. This time the new path would create a conflict on the edge between node 6 and node 5. This new edge is then marked as unsolved and a schedule based on the period of both communication pairs is calculated. In this example both pairs could fit into the period (Fig. 8., right). If a schedule is possible, the algorithm will notify the user of the offsets of this schedule otherwise it would inform the user that a solution does not exist.

To demonstrate the algorithm further, we present the results to the topology present in Fig. 9. We created a communication pair with a higher priority that would take the whole period to transmit. This way, the other communication pairs would have to be transferred through a different route. The algorithm is able to find new paths that ensure that the latency as a whole is the lowest possible and creates an output schedule, where there are no collisions (Fig.10.).

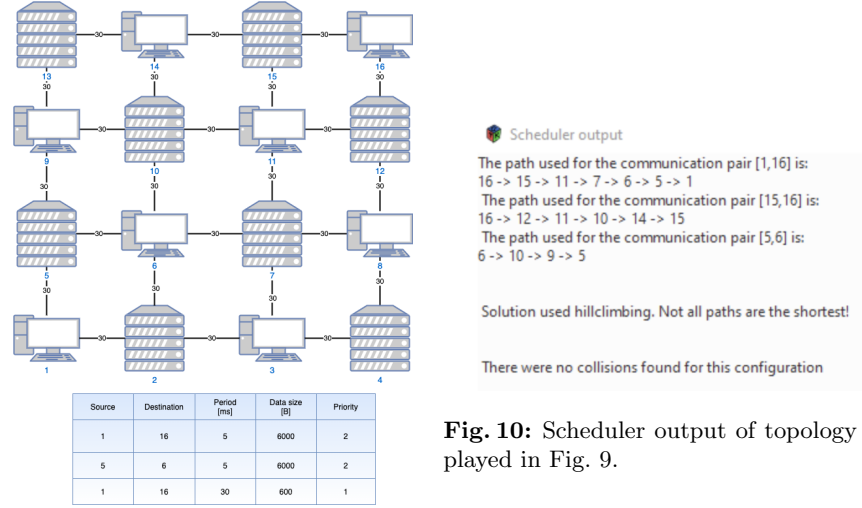


Fig. 9: Example of topology without collisions after finding a new path.

3.2 Comparison to other approaches

The algorithms and approaches mentioned in Section 2 also support non-scheduled traffic, which our solution doesn't support. To best compare our solution we also needed a topology that had its throughput and latency described. Chen et al. [21] described their credit based low latency packet scheduler (CBLPS) on the bottleneck topology shown in Fig. 11. Nodes marked S are source nodes, D are destination nodes and Rs and Rd are nodes that represent routers. Fig. 12 illustrates queuing delays in their CBLPS compared to other approaches (LLEPS = Low Latency and Efficient Packet Scheduling, SFQ = Start-time Fair Queuing, WF2Q+ = Worst-case fair weighted fair queueing, SCFQ = Self-clocked fair queueing, NDRR = Nested deficit round robin). Using three 1Mbps flows CBLPS is capable of sharing bandwidth.

We recreated the topology where nodes 1-10 are source nodes, 11-20 are destination nodes and nodes 21-22 represent the routers. For our solution to create a schedule, the bottleneck resource needs to be evenly distributed between all communication pairs. This means that if we were to transmit 1 MB data from all sources to all destinations a schedule would not be possible. The credit based scheduler on the other hand is capable of scheduling all the data, but there would be delays, which is not allowed for hard-real time data. To send all more data, either the allowed period should be greater, or a cable with a bigger throughput should be used.

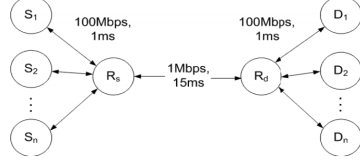


Fig. 11: Topology of comparison [21].

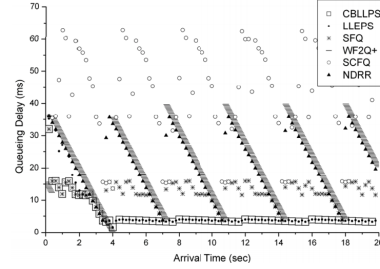


Fig. 12: Queuing delays on the conflict edge using different approaches [21].

```

The overall period on the path between node 21 and node 22 is: 1000
The offsets are :
For the communication pair [1,11] : 0
For the communication pair [2,12] : 100
For the communication pair [3,13] : 200
For the communication pair [4,14] : 300
For the communication pair [5,15] : 400
For the communication pair [6,16] : 500
For the communication pair [7,17] : 600
For the communication pair [8,18] : 700
For the communication pair [9,19] : 800
For the communication pair [10,20] : 900
  
```

Fig. 13: Proposed schedule output.

In Table 1. we compared both solutions. Our first point of comparison was the time to create a schedule for the data, in the given topology. For the given topology our algorithm was able to create an output in less than a second. The credit based scheduler doesn't create any schedule, it just assigns credits to every communication pair. The second point of comparison was the time in which data is transmitted. Our solution creates a predictable schedule, where every communication pair is given an offset and every period data is send at time of the given offset. The Credit based scheduler sends data of the communication pairs when there are credits available to the pair, this means data transmission time cannot be predicted at all. We also measured the success rate of the data transmission schedule. If every communication pair wants to send more than 0,1MB/s our solution is not able to create a schedule, while the credit based scheduler sends data even with a delay. If the data to send for every communication pair is 0,1 MB/s or less, the proposed solution is able to schedule all pairs. The credit based scheduler is also able to send all data, but the periods of the communication pair aren't fulfilled. Regarding the delay, the proposed solution is an all-or-nothing approach, it either can schedule everything without a delay or is not able to create a schedule. The credit based scheduler has an average of 7s of delay. The next point of comparison is predictability or the ability to send data periodically. As mentioned above, the proposed solution supports this kind of data while the credit based scheduler does not. The credit based scheduler

also doesn't take any network latency into consideration, while the proposed solution would search for the lowest latency path for every communication pair.

Table 1: Comparison of proposed solution vs. credit based scheduler for given topology.

Comparison in:	Proposed solution	Credit based scheduler
Time to create schedule	less than 1s	none
Data transmission time	according to schedule offset and period	when there are credits
Success rate (1Mb/s per pair)	couldn't create schedule	transmitting data with delay
Success rate (0,1Mb/s per pair)	scheduled all pairs	transmitting data without predictability
Delay	none if schedule was possible	on average 7ms
Periodicity/predictability	according to schedule	not possible to plan
Network latency	lowest possible	isn't taken into consideration

In conclusion our solution is not suited for systems with best effort traffic, or with non periodical data, as we can see, it would not be able to create a schedule at all if even one communication pair wants to send more data. For these cases other solutions such as the above described credit based scheduler may be more useful. For real-time networks or for systems with hard-real time data where no delays can be allowed, our solution is more suited, since it creates hard offsets that don't allow any delays and all data is going to be sent on time, so no queuing is needed.

4 Future work

Scheduling of communication between endpoints over Ethernet in the domain of real-time distributed systems is closely linked to task scheduling in individual endpoints and therefore it should be seen as one complex distributed real-time system where there are coordinated task plans in individual endpoints as also communication between these points using this solution. The aim for the future is to combine this proposed solution with real-time distributed systems. [17] [18]

5 Conclusion

In this paper we proposed an improved scheduling algorithm for deterministic networks that creates a schedule with emphasis on the overall lowest latency of the physical network.

References

1. Buttazzo, G.C.: Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications (2011).
2. Cisco Systems, Inc: Internetworking Technology Handbook, chap. QoS, <http://index-of.co.uk/Tutorials/Networking%20set%20of%20books.pdf>.
3. Huawei, Inc: IEEE 802.1Q Frame Format, <https://support.huawei.com/enterprise/en/doc/EDOC1100088104>.
4. Cisco Systems, Inc: Configuring QoS Queues, https://www.cisco.com/assets/sol/sb/SG220_Emulators/SG220_Emulator_v100-18_201406.
5. Kamayseh, Y.: Strict Priority Scheduler (2011), https://www.researchgate.net/figure/StrictPriorityScheduler_fig2_51953677.
6. Yang, L., ChengSheng P., Zhang, E., HaiYan, L.: A New Class of Priority-based Weighted Fair Scheduling Algorithm. *Physics Procedia*. 33, (2012)
7. Qadeer, M. A. Javaid, A. Y., Habib S.: Apportioning Bandwidth to Clients as per Privileges. *International Journal of Future Generation Communication and Networking* (2009)
8. Cisco Systems, Inc: Hierarchical Scheduling and Queuing https://www.cisco.com/c/en/us/td/docs/ios/12_2sb/feature/guide/3levelsch.html.
9. Thangamuthu, S.: Analysis of automotive Traffic shapers in Ethernet In-Vehicular Networks (2015) <https://pdfs.semanticscholar.org/2307/00933e09df00fadaaf812ff3ad4fc97a087d.pdf>.
10. Boiger, Ch.: Time Aware Shaper (2012), <http://www.ieee802.org/1/files/public/docs2012/bvboigertimeawareshaper0712-v01.pdf>.
11. Stanton, K. B.: Time Sensitive Networking (TSN) Also known as “Ethernet”, (2016), http://www.telecom-sync.com/files/pdfs/itsf/2016/day3/1000_Time%20Sensitive%20Networking.pdf
12. Houtan, B., Ashjaei, M., Daneshmand, M., Sjödin, M., Mubeen, S.: Work in Progress: Investigating the Effects of High Priority Traffic on the Best Effort Traffic in TSN Networks. 2019 IEEE Real-Time Systems Symposium (RTSS), (2019).
13. Kohutka, L., Stopjaková, V.: Task scheduler for dual-core real-time systems, 2016 MIXDES - 23rd International Conference Mixed Design of Integrated Circuits and Systems, (2016).
14. Bucsióva, V.: A new scheduling algorithm for Real-time networks, IIT.SRC 2017
15. Kohútka, L., Stopjaková, V.: Real-Time Task Scheduler for Quad-Core CPUs, 5th International Conference on Advances in Electronic and Photonic Technologies (ADEPT), (2017)
16. Kohútka, L., Stopjaková, V.: Extension of hardware-accelerated real-time task schedulers for support of quad-core processors, 2017 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Riga, (2017)
17. Kohútka, L., Nagy, L., Stopjaková, V.: A novel on-chip task scheduler for mixed-criticality real-time systems, *J. Circuits Syst. Comput.* 28 (2019)
18. Kohútka, L., Stopjaková, V.: Novel efficient on-chip task scheduler for multi-core hard real-time systems, *Microprocessors and Microsystems* vol. 76, (2020)
19. Serna Oliver, R., Craciunas, S. S., Stoger, G.: Analysis of Deterministic Ethernet scheduling for the Industrial Internet of Things. 2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, (2014)

20. Craciunas, S. S., Oliver, R. S., Chmélík, M., Steiner, W.: Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks. Proceedings of the 24th International Conference on Real-Time Networks and Systems - RTNS '16., (2016)
21. Chen, L.-H., Wu, E. H.-K., Hsieh, M.-I., Horng, J.-T., Chen, G.-H.: Credit-based low latency packet scheduling algorithm for real-time applications. 2012 IEEE International Conference on Communication, Networks and Satellite (ComNet-Sat), (2012)

Visualization and Manipulation of a Volumetric Representation for the Needs of 3D Scan Processing

Štefan Schindler*

Faculty of Informatics and Information Technologies STU Bratislava, Slovakia
xschindler@stuba.sk

Abstract. The paper deals with problems related to the process of 3D model acquisition from real-world objects through 3D scanning – namely unwanted artifacts that occur during the 3D scanning process on resulting 3D models. It discusses their visualization and correction. The output format of 3D scanning is usually point cloud, that is converted to polygonal mesh. Some algorithms, which repair certain errors, e.g. holes in the output model, work best on a volumetric representation, therefore the first step is to convert the mesh to a volumetric representation. Next it needs to be visualized in order to make sure the correction algorithm works correctly. In the paper, we present an algorithm for correction of holes and gaps in the 3D scans, that happened because of glossy surfaces or thin parts of the scanned objects. We also present a module for visualization of a volumetric representation implemented into existing graphical library called HIRO. In the end is introduced an algorithm for simulation of imperfections on correct 3D scans. It serves the purpose of erroneous models generation – so called datasets for development and verification of correction algorithms.

Keywords: voxelization · 3D scan · visualization · repair · error simulation · volumetric representation · voxel · algorithm

1 Introduction

Currently, the acquisition of 3D models through 3D scanning is becoming an increasingly used method [6]. In the design industry, it represents a more accurate and often faster form of obtaining a 3D model compared to manual modeling. In other domains, it may even be the only option, e.g. autonomous vehicles, computer vision, production lines - here we need to reconstruct objects automatically and mostly in real time.

However, this task is not easy. For technical reasons, 3D scanners have their limitations of the accuracy with which they can reconstruct objects. 3D models

* Bachelor study programme in field: Informatics. Supervisor: Dr. Martin Madaras, Institute of Computer Engineering and Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

created from such imperfect data may contain inaccuracies and errors that we need to eliminate during the processing so that we can subsequently use the cleaned data in other applications. Various correction algorithms are used for this purpose, which focus on eliminating errors, such as holes, gaps, outliers and so on.

In this paper, we focus on removing unwanted holes and gaps. The algorithm we present works on a volumetric representation of the 3D model. In order to verify the correct operation of this algorithm, or other processing algorithms, we need to see how they behave. The basis is therefore the visualization of this volumetric representation - before and after the application of the algorithm. We also need test data - 3D models, on which we can test the correction algorithm. Ideally, these erroneous 3D models shall come directly from 3D scanners, but if a large amount of test data is needed, then it is a time-consuming issue. We therefore present an additional algorithm that can automatically generate a broken model from a correct 3D model according to the specified parameters. This algorithm is destructive - it is not reversible - so it is impossible to construct a correction algorithm for it, which would simply do the opposite steps. These are synthetically generated datasets, which have the advantage that for each generated broken dataset we also have the ground truth model from which it was generated.

2 Visualization of a volumetric representation

The algorithms presented in this paper work with 3D models in volumetric representation. The basis for developing and verifying such algorithms is to see how the given 3D scan model changes after their application. The first step is therefore to find or create a tool that will provide us with such visualization.

2.1 Volumetric representation

First, we will briefly introduce volumetric representation [3], and what are its pros and cons. In this representation, the 3D model is made up of so-called voxels. They are the basic building block in 3D space analogous to pixels in 2D space. We can imagine it as a set of equally-sized cubes distributed in an uniform 3D grid. Or as a three-dimensional array of zeros and ones (or other attributes), which describe where in the grid the voxels are present (ones), or what color they are and so on (other attributes). The advantage is that it can also represent the volume, not just the surface as in the case of polygonal mesh, the disadvantage is the memory inefficiency, respectively the detail of the representation depends on the resolution of the grid. The higher the resolution, the higher are the details of the represented object at the cost of higher memory consumption.

BINVOX. ¹ Our algorithms work with a volumetric representation stored in *BINVOX* format, due to its openness, simplicity and efficiency as well as its use

¹ <https://www.patrickmin.com/binvox/binvox.html>

in the existing 3D scan processing pipeline. The format starts with an ASCII header with metadata. For the actual data a simple data structure is used - an array, sequence of zeros and ones that are additionally compressed using RLE compression [4]. It does not store any information about the color or material of individual voxels. Because we don't need to store such information about the voxels, this format is sufficient.

2.2 Existing solutions

Since the *BINVOX* format is open and used by several applications, there are available solutions for its loading and visualization. One of the main visualization applications is the *Viewvox*² application. We tested it and have to say that it works reliably, but does not have very good user experience regarding the controls and runs very slowly. It cannot load complex models and maintain a stable FPS. Therefore, we decided to try developing more efficient implementation, which would be able to provide more pleasant work with the *BINVOX* format.

2.3 Our solution

We built our volumetric representation visualization application on the *HIRO* graphical library [5]. A screenshot of the app can be found in Fig. 1. This graphics library provides a ready-made environment for real-time rendering of 3D models. By default it only supports mesh formats, however. We had to extend the library using so-called data module for the visualization of the *BINVOX* format.

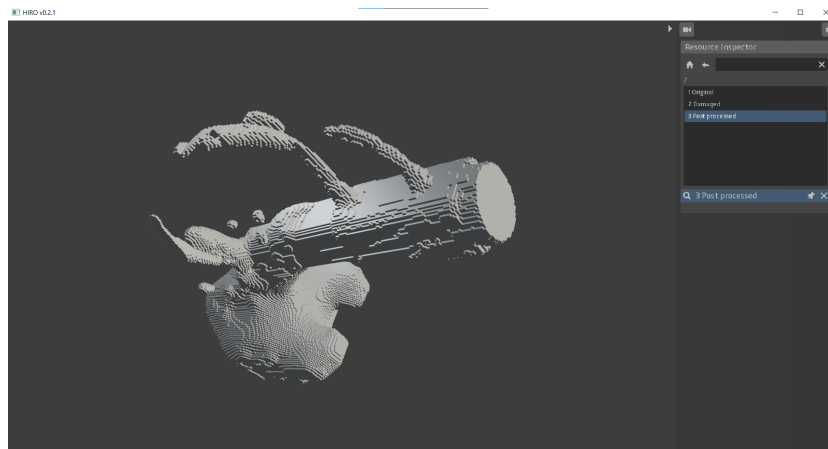


Fig. 1. A screenshot of the visualization application

The data module consists of two parts - Resource and Gadget. The Resource's task is to load the data and prepare it for the visualization. The Gadget is

² <https://www.patrickmin.com/viewvox/>

instantiated of a an existing Resource instance and stores additional data for the visualization customization. The data module we have implemented works the following way:

1. A *BINVOX* file is loaded into its own data structure - an array of bits, which is sufficient for the complexity of the scans we deal with, and also effective for the needs of the correction algorithm.
2. A mesh is created from the volumetric representation and rendered using an existing functionality in the *HIRO* library. The mesh creation consists of replacing each voxel with a cube shaped mesh with the size of a single cell of the volumetric grid. To make rendering more efficient, the sides of cubes adjacent to other cubes are removed. They would not be visible during visualization either.

Once launched, the application reads the paths to the *BINVOX* files for visualization from command line parameters in the following form:

```
visualizer.exe <[name ::] filepath> [<[name ::] filepath> ...]
```

Example:

```
visualizer.exe Original::original.binvox Apple::original2.binvox
```

For each path, a new Resource is created, which is provided the name of the object and the path to the *BINVOX* file. We use custom *VoxelModel* structure to store the volumetric representation. In addition to the data itself, the structure also stores the transformation of the model, the dimensions of the volumetric grid and the number of voxels - the number of ones in the voxels array.

3 3D scan correction algorithm

3D scanners have their limits, and the scanned data often contains errors and unwanted artifacts that we must remove in order for the data to be usable in other applications [2]. The errors we want to remove are quite specific, so we need to develop custom correction algorithm tailored specifically for this purpose. Existing correction algorithms are usually more generally oriented or require user-interaction during the correction process, so they did not seem to be able to fix our errors.

3.1 Problem

The pipeline, which this algorithm will be part of, uses non-contact 3D scanning methods that have a common disadvantage of not being able to accurately capture parts of the object that are not clearly visible or visible only from certain angles. The problem is also caused by shiny or transparent surfaces that do not completely reflect light rays, but some amount of refraction takes place too. For these reasons, the resulting 3D models may contain various errors - we focus on holes and gaps. These gaps arise mainly in the case of thin parts that are perpendicular to a certain surface (see Fig. 2). The algorithm will need to identify such locations and fill the gaps.

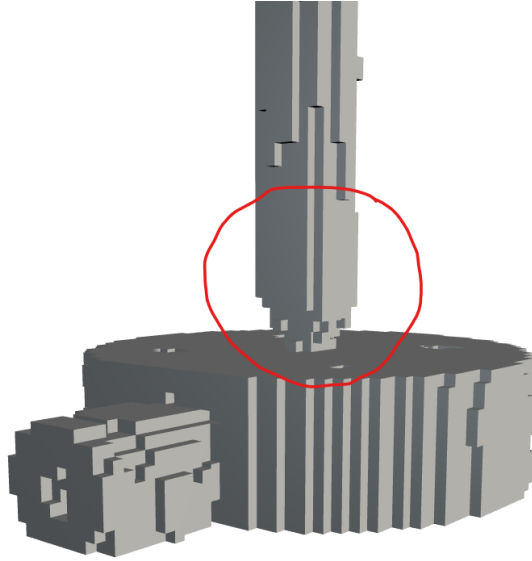


Fig. 2. Unwanted gap at the contact point of a thin part perpendicular to a surface

3.2 Pipeline overview

This correction algorithm is only one part in the overall 3D model processing pipeline - beginning with the scanner output and ending with the final processed 3D model. For better illustration, we present an overview of the processing pipeline. The pipeline consists of following steps:

1. The 3D scanner captures the scanned object from different angles, what results into multiple point clouds [1].
2. These point clouds are aligned, filtered and merged into single unstructured point cloud.
3. A mesh is reconstructed from the point cloud using Poisson reconstruction. This step removes some errors like holes, however other errors like unwanted gaps may still persist.
4. The mesh is converted to a volumetric representation.
5. Correction algorithms are applied to the volumetric representation and it is visualized.
6. The volumetric representation is converted back to mesh.
7. The mesh can be used in other applications.

A graph of the processing pipeline is shown in Fig. 3.

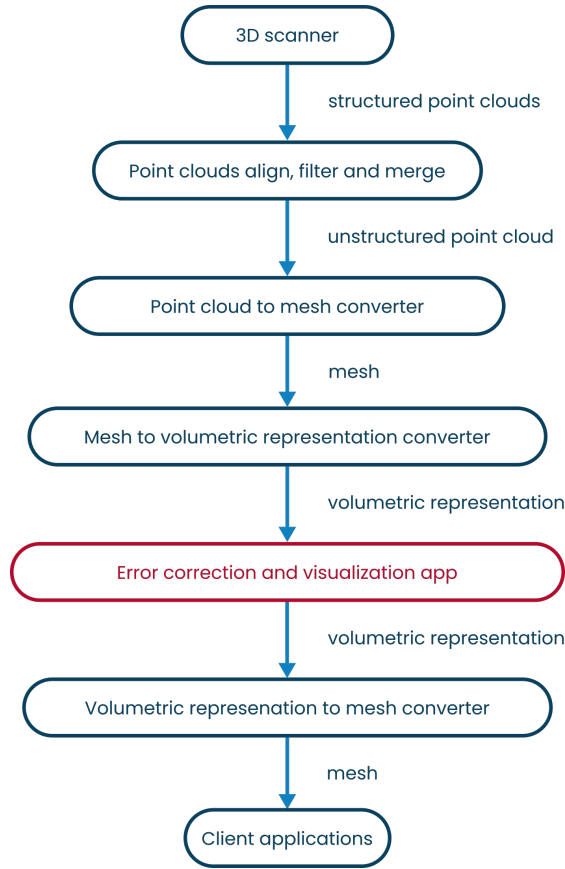


Fig. 3. 3D scan processing pipeline

3.3 Implementation

The correction algorithm works on the principle of sweep line algorithms - it gradually passes through the layers of the volumetric grid and in case it encounters a hole or gap, it evaluates whether or not it should be filled. The main idea is the concept of inertia - if a part of the model is continuous and suddenly there is a gap after which continues another sufficiently long part, then these two parts are joined - the gap between them is filled. A detailed description is in the form of a flowchart in Fig. 4. To explain the terms used in the flowchart please refer to Fig. 5.

The application is implemented in a console way, without a graphical user interface. The application reads the input *BINVOX* file and parameters from the command line in the following format:

```
fixer.exe [-a <x|y|z>] [-f <float>] <input> <output>
```

Example:


```
fixer.exe -a z -f 0.7 original.binvoy pp.binvoy
```

Where the `-a` parameter expresses the axis/-es in which the algorithm operates and the `-f` parameter defines the hole and gap fill factor. The resulting model is saved in a file that is defined by the `<output>` parameter.

4 Error simulation algorithm

4.1 Motivation

To generate enough erroneous datasets to verify and tune the correction algorithm, we have created an algorithm that simulates error in good 3D scans by adding unwanted gaps to places where thin parts perpendicularly touch a surface (see Fig. 2).

4.2 Implementation

The application is implemented in a console way. The input is a *BINVOX* file and parameters set by the user, the output is a *BINVOX* file which contains the model with the error simulation applied. After the correct setting of the parameters, gaps will appear on the output model in places where the thin parts perpendicularly touch a larger surface (see Fig. 2). The realism of the gap is achieved by cutting the end of the thin part and shaping its end with a conical shape displaced by a random noise (see Fig. 6).

The following properties of the damage are parameterizable:

- Axes in which the algorithm works (X, Y and Z). As with the correction algorithm we can be sure that the input model is aligned according to the global axes.
- The ratio between the size of the contacting surfaces from which the given contact point is considered the meeting point of a thin part and larger surface.
- Reach, cone cutoff and the gap from the surface (see Fig. 7).
- The intensity of the surface displacement by random noise.

Algorithm. The algorithm also works on the principle of sweep line algorithms. As it gradually iterates over the layers of the volumetric representation, it compares the ratio of areas of adjacent layers. If this ratio exceeds the value entered via the parameter, then the given place is identified as a touch point of a thin part and an area, and a defect is created - the end of the thin part is cut off and replaced by a cone. The shape of the cone is also parameterizable (see 7). The surface of the cone is displaced with random noise. This displacement is applied based on a generated black and white noise texture. This texture has a square dimensions approximately corresponding to the size of the volumetric layer being replaced by the cone. Each pixel of the noise texture is defined by a value in a zero-to-one interval. Higher value means a stronger displacement

is applied to the cone surface at the given location. The process of generating a noise texture is achieved by gradually dividing the texture into equally large square cells, which are assigned a random value within -0.5 to 0.5 interval. The texture division works in such a way that at the beginning the whole texture is one cell, in the following iteration this cell is divided into four equal square cells, which in the next iteration are each recursively divided again into four cells. This division stops when the size of one cell corresponds to the size of one pixel. After assigning a random value to a cell, the value is added up to the current values of all pixels, that are covered by the cell. After each addition, the pixel value is truncated to be within the zero-to-one interval. An illustrative example of noise texture generation can be seen in Fig. 8. As mentioned in the introduction, this algorithm is destructive, so it is not reversible. This irreversibility lies in the fact that after error simulation, we cannot retrospectively identify the contact point of a thin part perpendicular to the surface by gradual iteration through layers, as the error simulation removed these layers and this information is not present in the model anymore. A more detailed description of the algorithm is in form of a flowchart in Fig. 9.

5 Results

5.1 Volumetric representation visualization app

We tested the performance of our visualization application by measuring the frames per second in both - our application and in the Viewvox application for BINVOX files with different resolutions of the volumetric grid. The measured results are shown in Tab. 1.

Table 1. FPS comparison of volumetric representation visualization apps

Volumetric grid resolution	Our visualization app	Viewvox
128 x 128 x 128 (see Fig. 10 top)	60 fps	30 fps
256 x 256 x 256 (see Fig. 10 middle)	60 fps	5 fps
512 x 512 x 512 (see Fig. 10 bottom)	60 fps	<1 fps

The measured data shows, that our app offers faster and more effective rendering of the BINVOX files.

5.2 Correction and error simulation algorithms

The 3D scan error correction application has been tested on an erroneous dataset provided by the supervisor, the result of the correction is shown in Fig. 11. Additionally, the error simulation application has been used to generate more erroneous datasets that have also been run through the correction algorithm. The results of these tests can be seen in Fig. 12, in each row three images are shown, from left - original model, model after error simulation, model after application of the correction algorithm.

6 Conclusion

This paper dealt with processing of 3D scans - their correction and visualization. Visualization of the volumetric representation was successfully implemented using graphical HIRO library, with significant speed-up compared to the existing solution. We also implemented an algorithm for correcting errors - holes and gaps - in 3D scans. The algorithm works quite reliably, but as can be seen in the results showcase, it is sometimes impossible to prevent it from filling the holes or gaps that are not erroneous, but intentional. In the future, this algorithm could be replaced by a machine learning solution trained on datasets generated by the error simulation algorithm. Lastly, we implemented the error simulation algorithm. After some time of parameter tuning, we managed to simulate error on each dataset provided by the supervisor - an error that could occur during 3D scanning. The disadvantage of the algorithm is that it works only in three global axes (X, Y and Z) and works best when the model does not contain two or more parallel thin parts perpendicular to a larger surface, but only one such part.

References

1. Funkhouser, T.: Overview of 3d object representations. Princeton University, COS D **597** (2003)
2. Macháček, P.: Problems of 3d scanning and scanned data processing. Problems of 3D Scanning and Scanned Data Processing (nd): n. pag. Web **1** (2016)
3. Nooruddin, F.S., Turk, G.: Simplification and repair of polygonal models using volumetric techniques. IEEE Transactions on Visualization and Computer Graphics **9**(2), 191–205 (2003)
4. Patil, S., Ravi, B.: Voxel-based representation, display and thickness analysis of intricate shapes. In: Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05). pp. 6–pp. IEEE (2005)
5. Skeletex Research: Hiro. <http://hiro.skeletex.xyz/0.4.0/index.html> (2020), knižnica
6. Weyrich, T., Pauly, M., Keiser, R., Heinzle, S., Scandella, S., Gross, M.H.: Post-processing of scanned 3d surface data. SPBG **4**, 85–94 (2004)

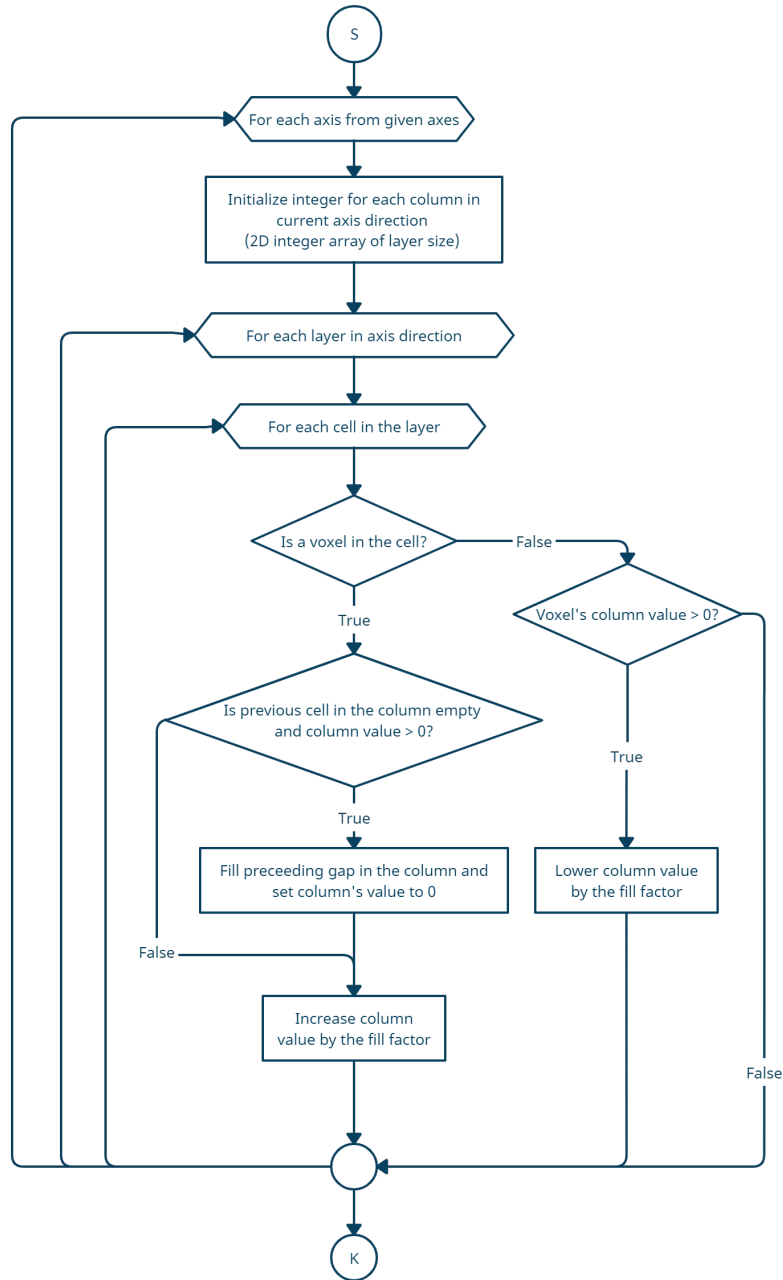


Fig. 4. Flowchart of the correction algorithm

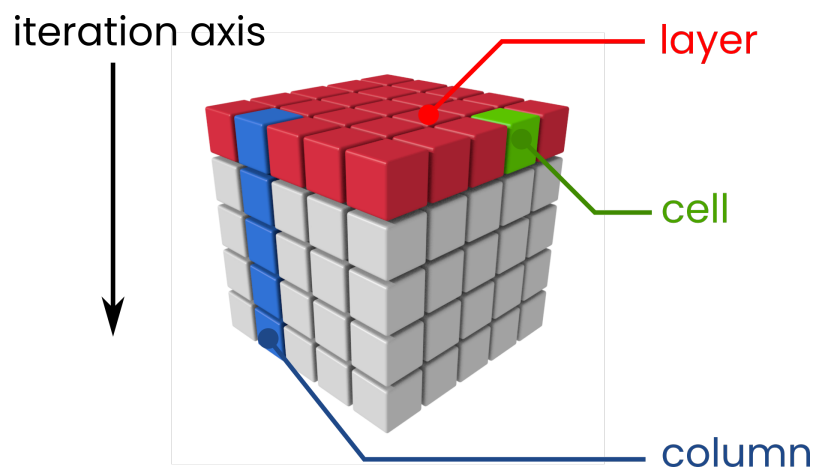


Fig. 5. Iteration over the voxel grid layers

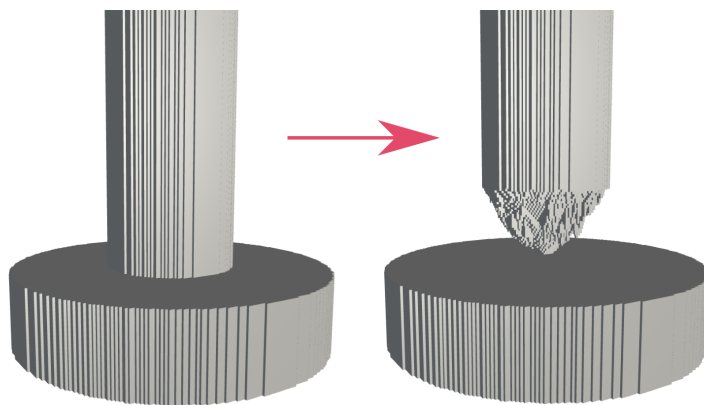


Fig. 6. The end of a thin part replaced by a conical shape

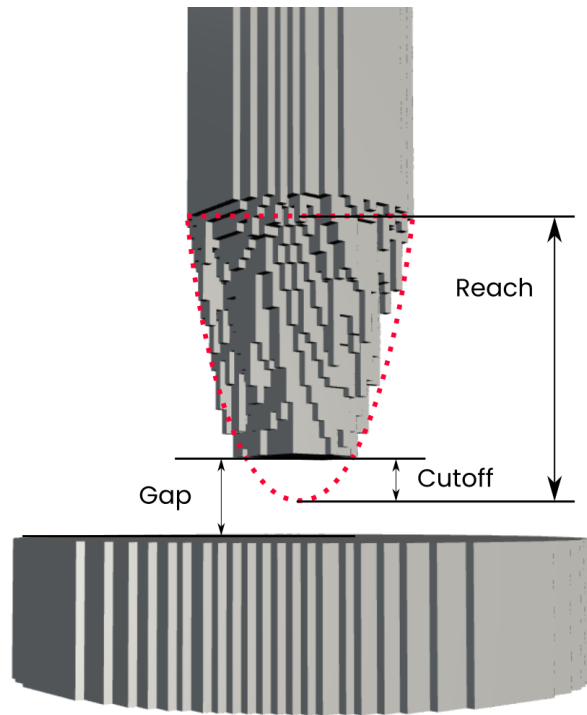


Fig. 7. Reach, cutoff and gap of the cone

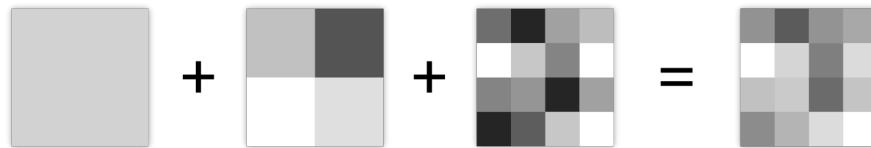


Fig. 8. Example of generating a 4 x 4 noise texture

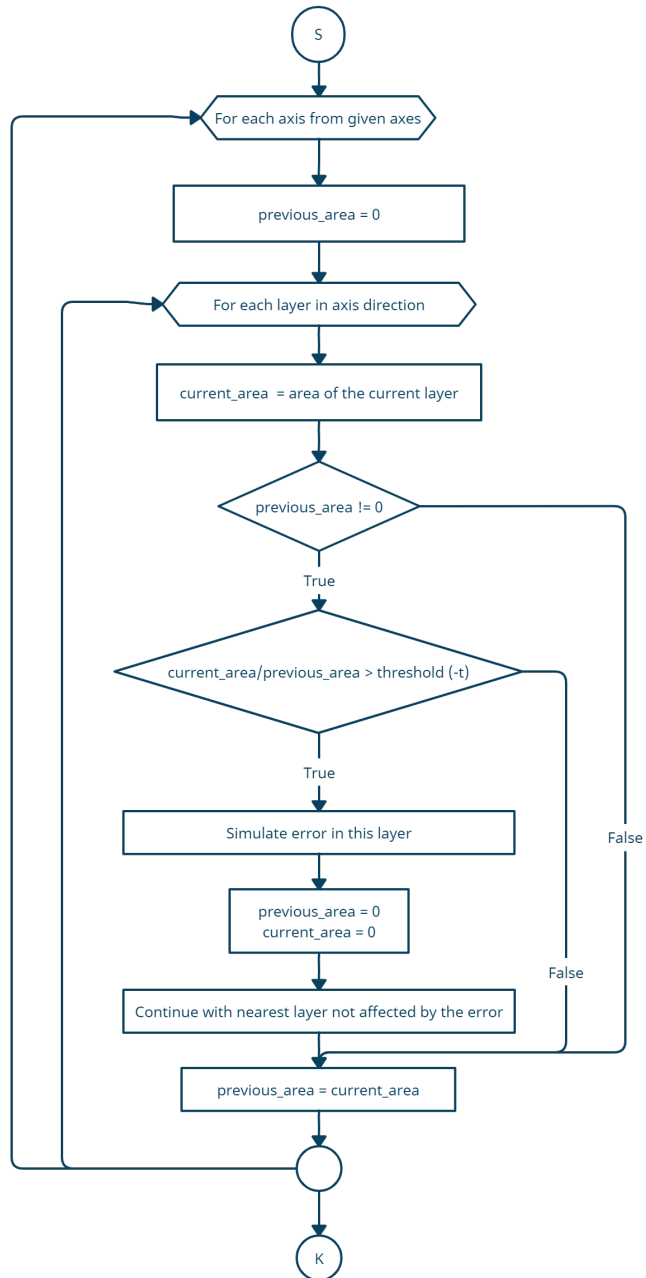


Fig. 9. Flowchart of the error simulation algorithm

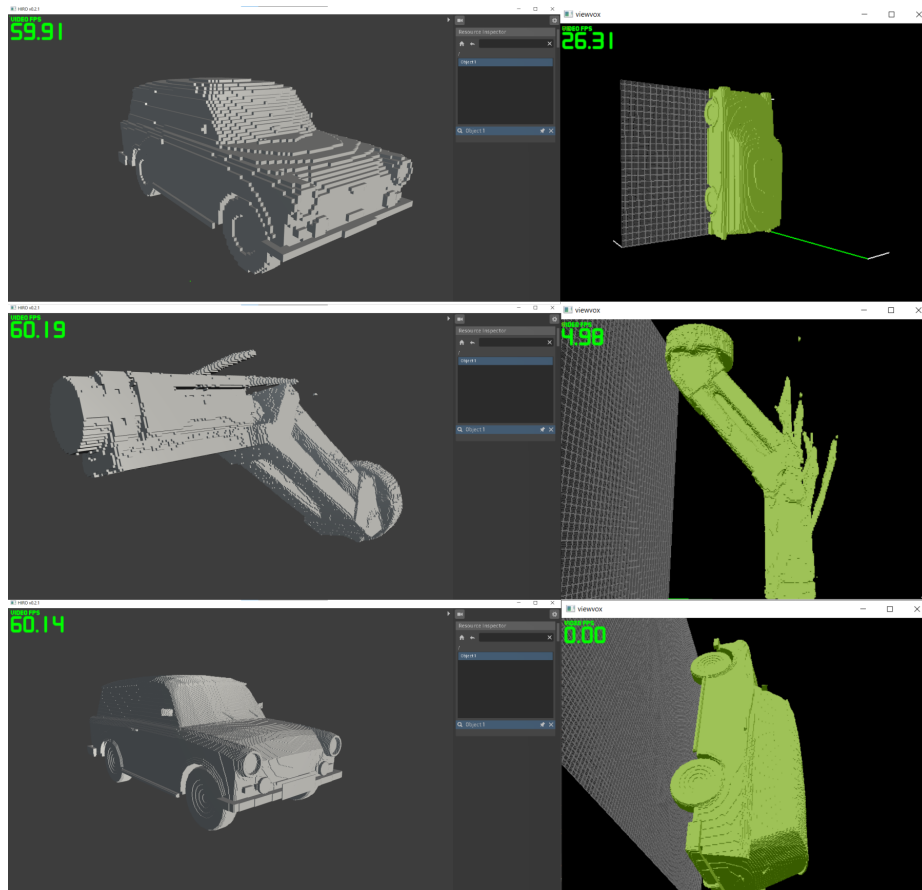


Fig. 10. Measuring FPS - left: our visualization app, right: Viewvox app

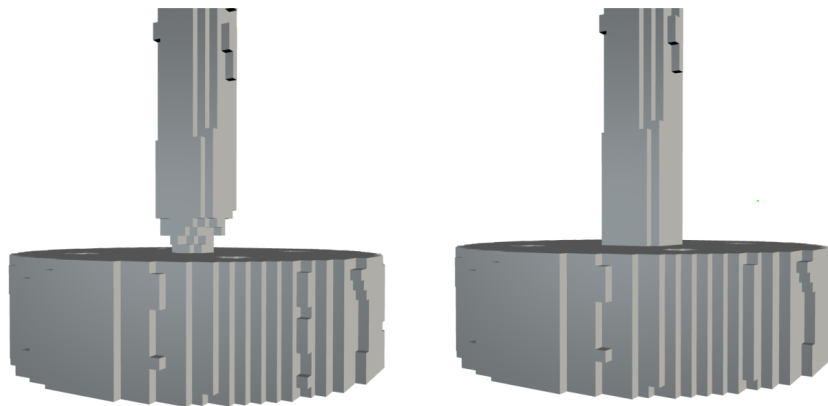


Fig. 11. Verification of the error correction algorithm on an erroneous dataset

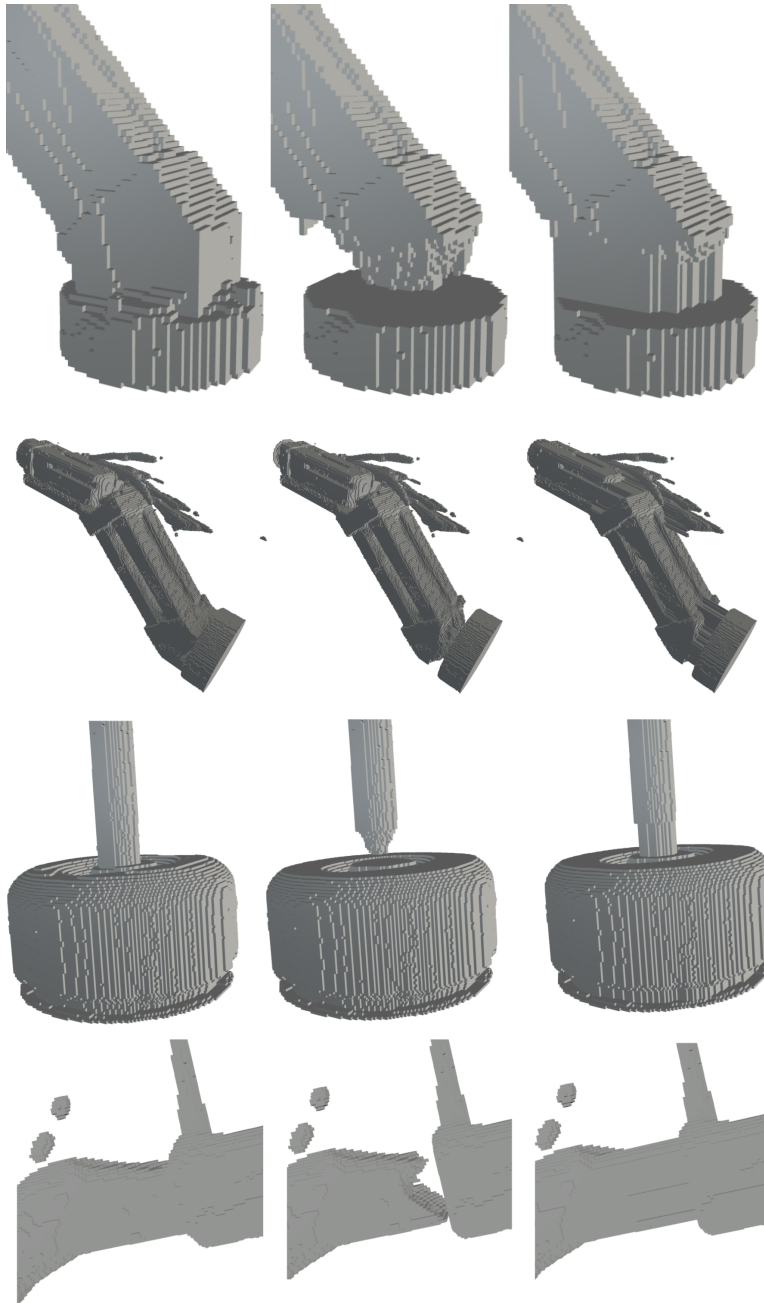


Fig. 12. Showcase of correction and error simulation algorithms (from left - original, with simulated error, corrected)

Lines detection via Hough Transform

Kamil-Martin Somrak

Faculty of Informatics and Information Technologies STU, Ilkovičova 2, 842 16
Karlova Ves, Slovakia
xsomrak@stuba.sk

Abstract. In this age of autonomous and intelligent devices car manufacturers were forced to develop smart cars. The purpose of these cars is to diminish the physical and mental strain exerted upon the driver. One of such technologies is the lane departure system, which helps the driver maintain the car inside the lane in case of inattention or other reasons. In our work we focused on creating such system, that would be able to detect the lines defining the lane and issue a warning in case it was needed. In order to detect the line that define the lane we used multiple image processing techniques, including image filtering, edge detection algorithm and the Hough transform algorithm, which allowed us to process images into palpable data, that we can easily evaluate.

Keywords: Hough Transform · Edge detection · LDWS system.

1 LDWS systems

Nowadays the attention of the drivers is lower than ever and lane departure becomes a very common mistake that leads to accidents that could have been avoided. The most common source of these accidents is either exhaustion, inattention or bad visibility conditions.

In order to lower the number of accidents caused by previously mentioned reasons car manufacturers developed a technologies that would issue a warning when the driver was about to exit their lane without using the turn signal. These types of technologies are called lane departure warning systems (LDWS).

Most of these systems use information captured by a camera, which is usually mounted on the windshield of the car. The information is processed by the car's computer and if the car is going to depart the lane an warning is issued.

2 Human information processing

During this phase the environment information(road conditions, other vehicles, weather conditions) and information from the driver's car are being analyzed and evaluated by the brain. The brain decides whether the current information represent a priority or not. We can split human behaviour into three categories(see Fig. 1). Most of the times the decision is made based on previous knowledge. This means that an experienced driver is less prone to make a decision that would

lead to an accident. Another big factor when it comes to information processing is exhaustion. If a driver is exhausted the decisions he takes might not be the correct ones. Most of the accidents are caused by exhaustion or inattention[6]

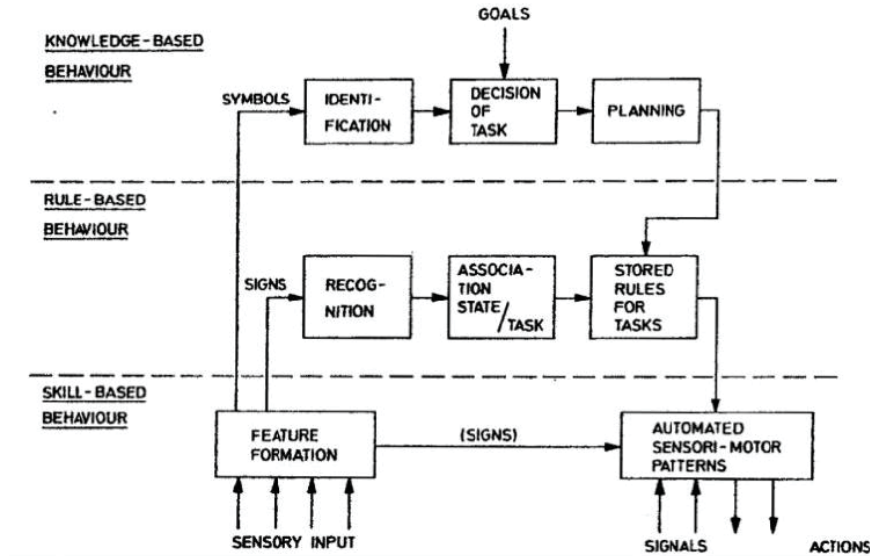


Fig. 1: Rasmussen's three-level model of human behaviour

3 Techniques and algorithms used in computer graphics

3.1 Edge detection

Image processing within computer graphics refers to digital image processing[3], it implies the use of the computational capacity of a computer in order to alter an existing image based on an algorithm[5]

In order to extract useful information from an image we need to convert it. The easiest way to analyze the image is to analyze the objects separately. In order to segment the image into objects we need to reduce the objects to simple shapes, reduce them to simple lines. This process is called edge detection[1].

Edges are represented by high contrasted zones of an image. These high contrast zones usually represent the border between two objects. Edge detection is a technique used to highlight only the borders of an object inside an image. There are multiple such algorithms, some of the most popular ones are: Sobel, Canny, Prewitt, Roberts. The majority of them work based on a convolution mask (see Fig. 2), it allows a pixel to be influenced by the surrounding pixels, in other words if the pixels around are very different the center pixel will stand out.

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Fig. 2: Sobel's operator for edge detection

3.2 Hough Transform

Initially it was a technique used to detect simple shapes(lines) in an image. It was invented by Paul V. Hough[4] in 1962. Although the original algorithm could detect only lines it was generalized and perfected and can be used to detect any kind of shape, usually circles. The Hough transform algorithm that we know today was invented in 1972 by Richard Duda and Peter Hart[2]

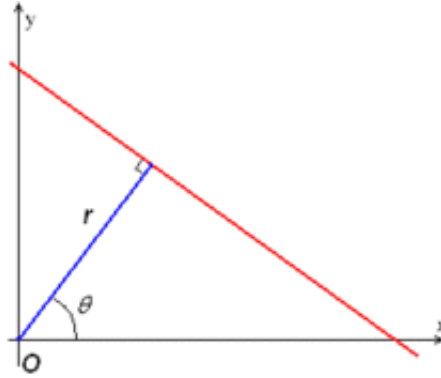
Theoretical Foundations The simplest case of the Hough transform[7] is the detection of a line. The classic definition of a line is the following:

$$y = mx + b \quad (1)$$

Is pretty easy to define a line if we know it's slope and it's intercept point with the y axis. The problem appears when the line we are trying to define a line perpendicular to the x axis, because we would get a division by zero.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (2)$$

This is why Hough transform uses Hesse's normal form to define lines(see Fig. 3).



$$r = \cos \theta + y \sin \theta \quad (3)$$

Fig. 3: Hesse's normal form

Thanks to this form we can define a line by using the following pair of parameters r , which represent the distance from the origin point of the axis system to the closest point on the line and the angle θ formed between the line r and the x axis

Implementation The Hough transform for detecting lines uses a 2d array as an accumulator for detection of lines that are defined by an equation of Hesse's normal form type, ex. $r = \cos \theta + y \sin \theta$. The accumulator is initialized with null values. The algorithm iterates through the accumulator, each cell in the

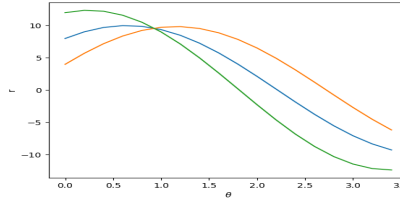


Fig. 4: Hough space example

accumulator represents a point in a graph called Hough space (see Fig. 4). If on the current point in the Hough space we detect something we increment the value of the respective cell in our accumulator. After traversing the Hough space we search for maximum values in our accumulator, those most likely represent the lines we were trying to find.

4 Our application

The final product of our research and work will be an application(see Fig. 5) that would be able to detect lane departure using Hough transform algorithm. The user would be able to use a recorded file as the input for the algorithm or the live feed from a camera. The application would split the video in individual frames, analyze each one of them and in case lane departure was detected issue a warning.

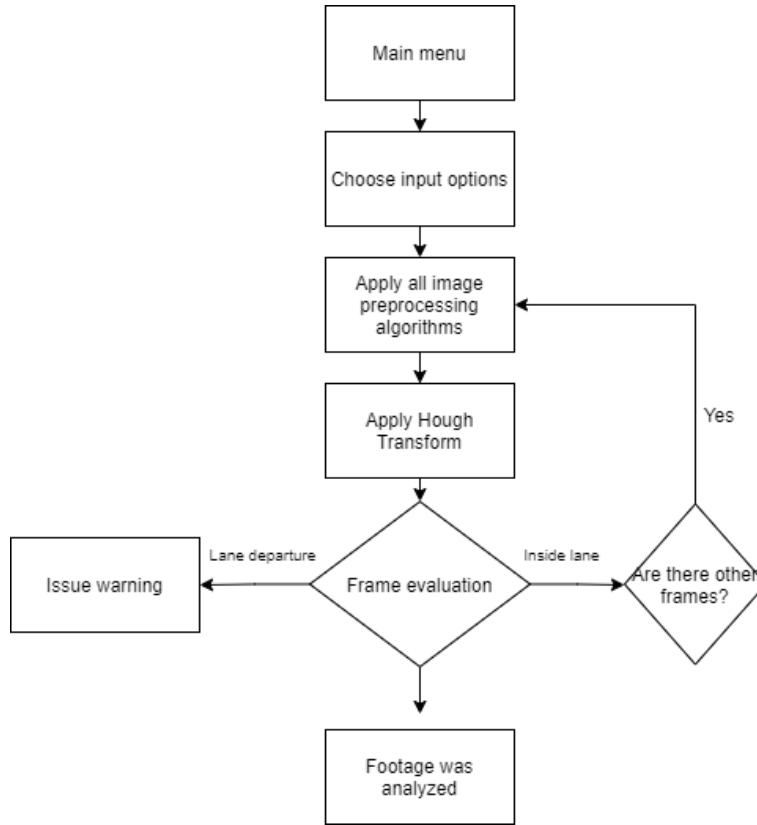


Fig. 5: Workflow of our application

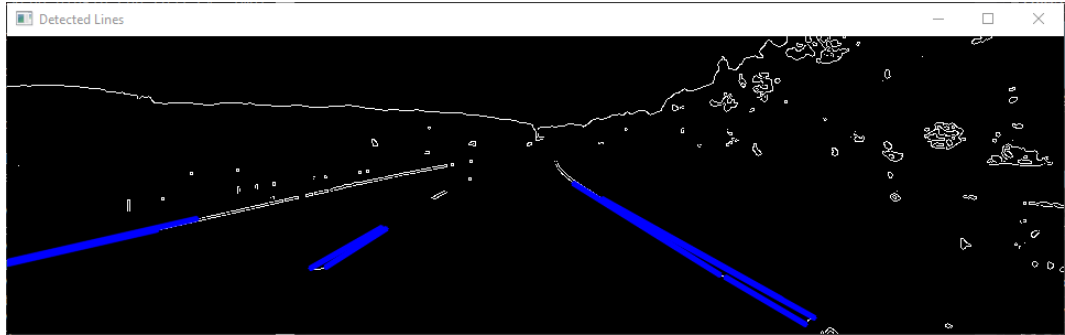
4.1 Testing methods

We will use a existing dataset of videos containing footage from moving vehicles. Each frame from these dataset has an attribute that specifies if lane departure is present in the image. We would compare this attribute to the result of our algorithm and calculate the accuracy percentage of our algorithm.

Preliminary results So far we have managed to implement the line detection feature of our application, the detection of lane departure is yet to be finished. The following image is an example of how the detected lines are being displayed after processing the original image (see Fig. 6). The accuracy of line detection reaches 76%, which represent the percentage of frames in which the desired lines were detected successfully.



(a) Original image



(b) Result of the line detection algorithm

Fig. 6: Example of an image before and after processing

5 Conclusion

In conclusion we focused on creating an application that would help diminish the strain created upon the driver using image processing. The main algorithm that we used to achieve this goal is the Hough transform algorithm which allowed us to detect the lines defining the lane and evaluate if the car is moving properly. We expect certain results based on visibility conditions: above 95% for perfect visibility conditions, above 60% for rainy weather and in case of snowstorms the system would warn the driver of its unavailability.

References

1. Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8, 6 (1986), 679–698.
2. Duda, R. O., and Hart, P. E. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM* 15, 1 (Jan. 1972), 11–15.
3. Gonzalez, R. C., and Woods, R. E. *Digital image processing*. Prentice Hall, Upper Saddle River, N.J., 2008.
4. Hough, P. V. Method and means for recognizing complex patterns.
5. Hughes, J. F., van Dam, A., McGuire, M., Sklar, D. F., Foley, J. D., Feiner, S., and Akeley, K. *Computer Graphics: Principles and Practice*, 3 ed. Addison-Wesley, Upper Saddle River, NJ, 2013.

6. Treat, J. Tri-level study of the causes of traffic accidents: An overview of final results. Proceedings: American Association for Automotive Medicine Annual Conference 21(1977), 391–403.
7. Towards Data Science, Lines Detection with Hough Transform 2020, Accessed 5 April, <https://towardsdatascience.com/lines-detection-with-hough-transform-84020b3b1549>

Tool for testing web application security

Patrik Bobocký*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava
pbobocky@gmail.com

Abstract. This paper focuses on security of web applications. It analyzes the most common security issues like SQL injection and cross-site scripting. It also contains examples of unsecured implementations and examples explaining how to secure them. It also contains the analysis of existing tools which check the security of web applications and evaluate their benefits and negatives for practical usage.

The knowledge acquired during the analysis of existing tools was used to specify the goals and requirements for the final tool which can effectively test the security of web applications. We selected one of the analyzed tools, designed and implemented an upgrade for this tool.

Keywords: Security, Testing, Web applications, Vulnerabilities, SQL injection, XSS, Tool

1 Introduction

The continuous development of web technology from the first web page to nowadays is the reason why internet services and web applications are available for almost everyone with any smart device and connection to the Internet. Web applications are becoming more and more popular, complex, helpful and complicated. Development of new features also brings new vulnerabilities. Some of them are being continuously fixed, but others aren't. In some cases, the ones that haven't been fixed have been partially mitigated. With the vast amount of web pages and web applications in existence, there is a concerning amount of unpatched vulnerabilities. Considering this fact, we will focus on security and vulnerabilities of web applications, tools for detection of vulnerable source code and we will upgrade one of the tools. Such tool for vulnerability searching can improve the efficiency of security testing of web applications.

* Bachelor study programme in field: Informatics Supervisor: Ing. Viktor Lančarič, Institute of Computer Engineering and Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

2 Vulnerabilities

In 2019, Accunetix Team ran a vulnerability test with 10 000 scan subjects [1]. 32% of these targets were vulnerable to Cross-site scripting and 14% were vulnerable to SQL injection. These two vulnerabilities can be detected by scanning the source code of the web application.

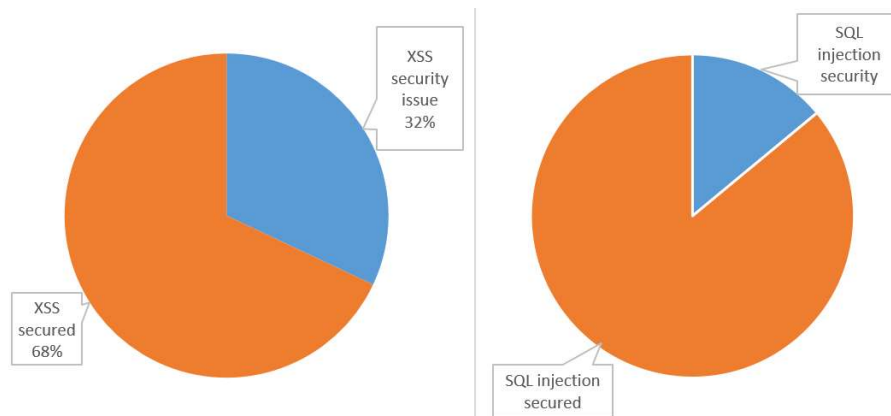


Fig. 1 - Detected security issues

2.1 Cross-site scripting (XSS)

To execute certain tasks, web applications use script languages like JavaScript on the client-side. JavaScript can change the content of a website. The potential attacker can try inserting a dangerous script into the website. When the user loads the website with the inserted script, script is executed in the security context of the user's browser. In Figure 1, we have provided a script that sends cookies to an attacker's server.

```

```

Fig. 2 XSS script for sending cookies to attacker server

There are three types of XSS: persistent, reflected and DOM-based. Reflected and persistent XSS are server side injection issues. DOM based XSS is a client side injection issue.[2]

Persistent XSS. An application that is vulnerable to persistent XSS allows the attacker to permanently store his payload (for example into a database). This script will be repeatedly loaded from the database and it will be executed for every user which visits the webpage. One of the possible solutions is the validation of inserted content for every user.

Reflected XSS. Applications vulnerable to reflected XSS allow the attacker to trick a victim to open a link with XSS payload. When the payload is sent to the server, the server includes the payload as a part of the HTTP response.

DOM-based. This is a specific type of XSS when an attacker creates a payload. This payload changes the Document Object Model (DOM) in a victim's browser.

2.2 SQL injection

Most web applications use databases to save data. These databases are running on database servers and applications use SQL to communicate with these servers. When the application inserts non-sanitized user's input into a query, some specific user inputs can change this query, and send the changed query to the server. [3]

```
Vulnerable code implementation:
    mysqli_query($connection, "SELECT * FROM users WHERE name = '". $user_input. "'");

Attacker Input: a'; DELETE * FROM users;--

SQL query:
    SELECT * FROM users WHERE name = 'a'; DELETE * FROM users;--'
```

Fig. 3 – An example of source code with the SQL injection security issue

The typical example of a SQL injection attack could be a web application which allows to change some important attributes and values, for example a user password. The user of this web application wants to change his password and uses the provided web interface to do so. The interface will show the text fields that need to be filled, and the remaining attributes remain hidden, for example the user ID. The attacker could modify this hidden value, for example by changing the ID to an administrator ID. If the ID verification isn't secured on the back-end, the attacker could change the administrator password this way.

3 Methods for static analysis of source code

Web application security testing tools use several different methods to analyze the source code. The input of these methods is the source code of the targeted web application and the output is the analysis report of the given application. The tool can be implemented as an extension into a development environment, a command in command line, a file upload or a number of different ways. Analysis can process one or more files or a folder. The result of the analysis contains warnings regarding security issues, which should be fixed. The analysis can be set to automatically scan the specified file, thus making the security issues reparable directly during the development process.

3.1 Data flow diagram (DFD)

DFD is a graphic representation of data flow in a web application. At the start, unvalidated input values (like the user's input or client-side stored values) are tagged as untrusted. The next step tags all values in the graph that are affected by the untrusted values as untrusted. At the end, we can see the flow of potentially non-sanitized values.

3.2 Abstract syntax tree (AST)

AST represents source code in the form of a tree structure. Nodes in tree represent an abstract view of the structure of the source code. Using machine learning techniques, typical patterns can be discovered. These patterns allow to "extrapolate" the known vulnerabilities by identifying the potentially vulnerable code with similar characteristics. [4]

3.3 Regular Expression (Regex)

Regex is a character string that represents a searched pattern. This pattern is a matching text and we can see the matches in the source code. [5]

Some functions of programming languages can contain a vulnerability, which is usually discovered as the time goes on. The usage of these functions is dangerous and these functions are usually replaced by other, safer functions. Regex is a good solution to this problem. We can define a name of a deprecated function as a regex expression, and the developer can be warned when he uses such a function in his source code.

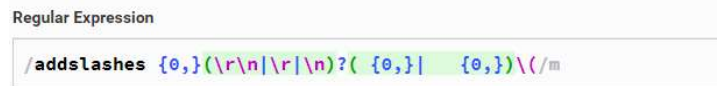


Fig. 4 - Regular expression to match "addslashes" function

4 State of the art

Manual testing is a time-consuming way of testing, which can be inefficient. Because of this inefficiency, we may use testing tools that make this process simpler and more effective. Some of the more popular testing tools are shown in Table 1.

Table 1. Table of existing tools.

Name	Open-Source	Analyze technique	Documentation	Source
In-sightAppSec	No		No	https://www.rapid7.com/products/in-sightappsec/
DevSkim	Yes	Regex	Yes	https://github.com/microsoft/devskim
	No	Regex	No	https://npo-echelon.ru/en/solutions/appchecker.php

Fortify	No	DFD	Yes	https://www.microfocus.com/en-us/products/static-code-analysis-sast/overview
IronWASP	No	Dynamic testing on browser's side	Simply and not updated	http://blog.ironwasp.org/
Burp Suite	No		No	https://portswigger.net/burp
Grabber	No		No	http://rgaucher.info/beta/grabber/
Sonar Source	No		Yes	https://www.sonarsource.com/
Acunetix	No		No	https://www.acunetix.com/solutions/security-engineer/
SonarQube	Yes	AST	Yes	https://www.sonarqube.org/

During our detailed analysis of the tools in Table 1, we mainly focused on real usability of the tool and we verified the functionality of the tool on applications containing SQL injection and XSS security issues. During our analysis we skipped the tools that couldn't be tested because of varying reasons. In most such cases, the tool was not open-source or a proper documentation wasn't available.

4.1 IronWAPS / Sboxr

Ironwasp was originally an open source tool which resulted into a commercial tool called Sboxr. This tool is focused on vulnerabilities detectable on the client side from the point of view of the DOM. Due to this fact it cannot detect vulnerabilities executable on the server side and is not suitable for comprehensive application control. We followed the documentation during our testing and, despite our best efforts, we were unable to run it.

4.2 Fortify

The Fortify tool from Micro Focus is a commercial tool with a 15-day trial license. This tool has several analysis options and runs as a web application on the company's server. Due to the various possibilities of using the tool, it is necessary to set the test parameters during testing in an interface that is not easy to manipulate for a new user. The main disadvantage of this tool is the fact that the tests need to be run manually. During our testing, this tool detected selected vulnerabilities.

4.3 DevSkim

DevSkim is an open-source tool developed by Microsoft. This tool focuses on the analysis of a source code, and works with multiple programming languages, including PHP. The documentation provided for this tool lacks the examples of rule definitions and other configuration options. A library of detectable security issues is included. The

analysis of the source code uses regular expressions to search for the security issues. The library is in JSON format, which allows the user to easily add new rules. Each rule must contain the following parameters: name, description, programming language specifications, regular expressions and others. There is an option to specify suggested solutions for the issue, by clicking on the specified part of the source code to remove it.

There are many different options which allow us to use DevSkim, one of them being a web interface “DevSkim Blazor”, but unfortunately this option doesn’t meet the needs of our analysis, as it can’t process the source code as a whole. This option only allows the user to input one source file, which is insufficient when analyzing a modern web application that uses multiple source files.

The next tested options were using the command line approach, and installing the IDE plug-in “Visual Studio Code”. To install this plug-in, we used “Marketplace”, which is a part of the “Visual Studio Code” IDE and the DevSkim plug-in is already integrated. After the installation process, we verified that the plug-in settings are correct and then proceeded to create a non-sanitized query for a database server using the SQL injection method and “addslashes” function. The tool did not detect these security issues.

The absent rules for this programming language caused the security issue to go undetected. A possible solution for this problem could be the creation of a set of rules, which would detect this issue. This particular solution could effectively detect the “addslashes” function. However, this approach would not be sufficient to detect an SQL security issue, because there are many different ways an SQL query could be rigged, and regular expressions alone cannot reliably detect these issues. The DevSkim tool doesn’t allow the user to create a logical evaluation of multiple search queries.

5 Our draft concerning DevSkim upgrade

We designed a solution that allows the user to input a specific logical operator when defining the rules, by using the strings “and”, “or” and “!” to represent the logical operators “AND”, “OR” and “NOT”, respectively. A possible improvement could be achieved by adding the option to use embedded logical expressions, by using parentheses. The logical operators will be detected when processing the regular expression and the regular expression will be divided into simpler regular expressions. The divided regular expressions will be searched for in the source code, and then the results will be put together by applying the logical operators.

After the implementation of the aforementioned rules with logical branching, the tool should be able to detect more complicated security issues, like SQL injection, which can be patched, but the current version of DevSkim is not able to detect them.

6 DevSkim changes implementation

We implemented a functionality that analyzes a regular expression before its searched for. We added a mechanism that checks if the regular expression contains

strings "AND" and "OR". If these strings are present, the regular expression needs to be divided into partial regular expressions. Our mechanism separates these expressions, searches for the expressions separately, and then adds the results together. Our mechanism also checks whether the string "NOT" is present, signaling a negated expressions. If the negation is present, we search for a regular expression without the specified string. If a string was found, we set the result as "null", and if a string was not found, we searched for any character. If this search method is used, the first character in the specified source code will be flagged. This method also works if the user inputs an empty source code. This analysis of the regular expression is implemented in a recursive function "analyzePattern" in class "DevSkimWorker". The recursion allows for nesting of the regular expressions. The levels of nesting are separated by two-character delimiters "~{" and "}~". These strings were chosen, because they don't have any meaning in any programming language, and they are not used in regular expressions either.

7 Conclusion and Future work

This paper explained some of the most commonly exploited web application security issues, as well as the analysis of source code in this context. From the tools in Table 1, we chose the "DevSkim" tool, for which we designed an upgrade. The testing of the upgraded tool was separated into different phases. After the testing, we identified that our upgrade improved the ability of the tool to search for vulnerabilities in code, and reduced "false negative" occurrences. Even with our maximum effort, our upgrade cannot prevent the occurrence of "false positive" vulnerabilities. These false positive vulnerabilities also occur in the original version of the tool, and our upgrade does not prevent them. To prevent these vulnerabilities, a more complex analysis of the source code would be needed. The method of using regular expressions is not sufficient for this task.

The method of analysis of source code with the usage of regular expressions is not sufficient for vulnerabilities which are dependent on a specific implementation.

References

1. Accunetix Team, "Accunetix web application vulnerability Report 2019, 2019, pp 14,19,23,26,
2. Rama Koteswara Rao, G., R. Satya Prasad a M. Ramesh. *Neutralizing Cross-Site Scripting Attacks Using Open Source Technologies*. Udaipur, India: ACM New York, NY, 2016 ISBN 978-1-4503-3962-9.
3. Fang Yu a Yi-Yang Tung. Patcher: An Online Service for Detecting, Viewing and Patching Web Application Vulnerabilities. In: *2014 47th Hawaii International Conference on System Sciences*. IEEE, 2014, s. 4878-4886 DOI: 10.1109/HICSS.2014.598. ISBN 978-1-4799-2504-9.
4. YAMAGUCHI, Fabian, Felix LINDNER, Konrad RIECK, 2011. Assisted Discovery of Vulnerabilities using Machine Learning

5. L. Sharmila, U. Sakthi, A. Geethanjali and S. Sagadevan, "Regular Expression Based Pattern Matching for Gene Expression Data to Identify the Abnormality Gnome," 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), Tindivanam, India, 2017, pp. 301-305, doi: 10.1109/ICRTCCM.2017.71.