OPINION MINING IN ALBANIAN LANGUAGE: A NEURAL NETWORK MODEL

Nelda Kote
Faculty of Informatics, New York University of Tirana, Tirana, Albania
Marenglen Biba
Faculty of Information Technology, Polytechnic University of Tirana, Albania

Abstract

Opinion Mining (OM) has become an interesting research area applicable in different fields, such as social media posts, product or service reviews, online blogs, etc. OM aims to analyze and gain information from online media opinions. The most popular techniques used for opinion analysis are machine learning. The present paper aims to evaluate the performance of an artificial neural network model to classify the text opinion based on the polarity of the expressed sentiment using a two-classes annotation schema: positive and negative. The dataset of the opinions used is in the Albanian language. A CNN model is implemented and trained using the Keras framework and Tensorflow as a backend. The proposed classification model for opinions in the Albanian languages has an accuracy of 73%.

***Keywords*:** Opinion mining, sentiment analysis, neural network, Albanian language

1. INTRODUCTION

Opinion Mining (OM) or Sentiment Analysis (SA) is a highly-interesting Artificial Intelligence (AI) field that aims to analyze and extract information from opinions expressed in online media. BrightLocal (2022) reported that 98% of the consumers read reviews about the local businesses. Analyzing the online media opinion can help people to decide whether to purchase a product, and even businesses to develop the best strategies to improve their services or products.

Opinion Mining (OM) aims to analyze and extract information from opinions expressed in online media. Opinion classification is one of the tasks in OM that aims to classify the opinions based on the polarity of the expressed

sentiment. The classification task can be performed in three levels, document level, sentence level, and aspect level. The classification schema can be two, three, or more classes (Liu 2015).

Many research studies in the field of OM have been devoted to languages such as English, Italian, German, Chinese or Japanese, but there are only some small research studies done in this field for the Albanian language. The Albanian language is considered a low-resources language for natural language processing tools.

Considered a separate branch of the Indo-European language family, the Albanian language is spoken by around 7 million native speakers all over the world. In addition, it is the official language in Albania and Kosovo, and the official regional language in Montenegro and the Republic of North Macedonia. Given the fact that the Albanian language is a separate branch of the Indo-European languages family, it is of high interest to be studied not only for linguistics purposes but also in other purposes or fields. The Albanian language has a complex grammar, and its alphabet has 36 letters.

The present paper reports about a CNN network model for opinion classification based on the polarity of the sentiment expressed by the opinion. We have used a two-classes annotation schema: positive and negative. As Albanian is a low-resource language in natural language processing tools and annotated datasets, a dataset has been created. The data set is an extended dataset of the datasets that we have used in (Kote and Biba 2018; Kote *et al*., 2018; Kote and Biba 2021). The data set is in Section 3 described.

The remainder of this paper is structured as follows: Section 2 reviews related works, Section 3 describes the dataset used, Section 4 informs about the architecture of the neural network model and the experimental results, and finally, Section 5 presents the conclusion of our work and some suggestion for future work.

2. RELATED WORKS

The recent years mark an increasing use of artificial neural networks (ANN) in OM tasks. An ANN is based on a collection of huge number of nodes called artificial neurons, which loosely model the neurons in a biological brain. In a neural network, the neurons are linked and organized in layers. A neuron is an information processing entity. The ANN learns to achieve a certain task, in our case opinion classification based on the polarity of the sentiment, through by readjusting the weights of the connections between the nodes. The typical architecture of a neural network model consists in three layers, the input layer, the hidden layer and the output layer. Based on the network topology, the neural network can be classified as feedforward and recurrent/recursive neural network. A convolutional neural

network (CNN) is a feedforward neural network and long short-term memory (LSTM) is a recurrent network. Deep learning is a multi-layer neural network that comparing to a typical three layers neural network has better learning capabilities (Zhang *et al*., 2018).

Schuller *et al*., (2015) studied the impact of different feature methods in the performance of classification techniques for sentiment classification. They perform an experimental evaluation of the performance of Naïve Bayes (NB), Maximum Entropy (ME), and Long Short-Term Memory (LSTM) model using four small-medium sized datasets and a large dataset using different features and n-gram configuration. They concluded that the performance is increased when 1-gram and 2-gram are used, and the best performing algorithm is ME. LSTM is the second-best performing with a difference of 1.5 points compared to ME.

To improve the accuracy of a classification model Liu *et al*., (2018) used a Bi-LSTM neural network that learns specific information by multiple domains and uses it in another domain. The used domain descriptors, memories, and adversarial training utilize better domain knowledge. The experimental evaluation of the method comparing it with existing and in-domain models indicated that the proposed model outperforms them. So, the performance of a model can be improved by using knowledge learned from other domains.

Moreas *et al*., (2013) experimentally evaluated a bag-of-words neural network model, NB and SMV algorithms to classify the opinion concluded that the bag-of-words neural network model outperforms the NB model and SMV model. Also, the experimental results in the paper of Barry (2017) show that the approach of a LSTM model in conjunction with GloVe embeddings and Word2vec has the best performance.

Tang *et al*., (2015) have proposed a neural network model for opinion classification based on points categories, 1 to 5. This approach takes into consideration not only the polarity of the opinion but even the person that has given it. The first component of the model changes the word's vector depending on the person who expressed the opinion and then the classifier is trained and used to predict and assign the points to a new opinion. The evaluation in two datasets indicates that the proposed model has a good performance.

The deep-learning approach proposed by Deriu and Cieliebak (2016) uses a two-layer CNN network to classify Italian tweets in nine possible labels combination. The proposed model first predicts if a tweet is a subjective or objective opinion and then for the subjective is predicted if it is positive or negative and if it is ironic. The authors have used a meta-classifier on top of the CNN to increase the robustness of the model. The model is best performing in the polarity classification task.

The approach proposed by *Minaee et al*., (2019) is based on a CNN network and a LSTM network, in a way to extract the local structure and to capture the temporal information of the data. The experimental evaluation using two binary polarity sentiment datasets, the IMDB dataset, and the SST2 show that the proposed ensemble approach outperforms the two individual models, CNN, and LSTM.

Over the last decade, there have been extensive research studies in opinion mining. and specifically in opinion classification almost for English, in addition to few research studies for low-resource languages like the Albanian language.

One of the first studies in opinion classification for the Albanian language is presented in the paper of Biba and Mane (2014). In this paper, the performance of different machine learning algorithms is evaluated by experiments in the task of opinion classification as positive and negative.

Kote and Biba (2018), Kote *et al*., (2018), Kote and Biba (2021) experimentally evaluated the performance of 50 machine learning algorithms to classify the opinion as positive and negative in the Albanian language in multi-domains and in-domain datasets. The authors concluded that the use of n-gram (values min=1 and max=2) and TF-IDF increase the performance of the algorithms and the best-performing algorithms in average terms are RBF Network and Naïve Bayes Multinomial.

Skenduli *et al*., (2018) have proposed a CNN network model for classifying user-emotion of Facebook posts in the Albanian language at the sentence level. The authors have evaluated the performance of the neural CNN model comparing it with the performance of tree machine learning algorithms Naive Bayes (NB) Support Vector Machines (SMO) and, Instance-based learning (IBK). The experimental results show that the CNN model outperforms compared to the other machine learning algorithms.

Vasili *et al*., (2021) evaluated different approaches used to analyze the sentiment of tweets in the Albanian languages. The authors have evaluated the performance of three main approaches techniques for sentiment classification: traditional machine learning, lexicon-based and deep learning approach. In the experimental evaluation are used feature extraction techniques such as bag-of-words, TF-IDF, Word2Vec, and Glove. The data used to train and evaluate the model are unbalanced. The experimental results indicate that LSTM based RNN with Glove as a feature extraction technique provides the best results with a F-score of 87.8%.

Haveriku and Frashëri (2021) presented a model to automatically process and extract information from tweets in the Albanian language. They have evaluated the performance of three machine learning algorithms, SVM, Logistic Regression and Naïve Bayes to analyze the sentiment of tweets.

3. THE DATASET

Considering that the Albanian language is a low-resource language in annotated corpora for natural language processing purposes, we have collected and created by ourselves an opinion annotated dataset. This dataset is an extension of the dataset used in our previous research papers (Kote and Biba 2018; Kote *et al*., 2018; Kote and Biba 2021).

We have collected in total 900 text-document opinions in the Albanian language from well-known Albanian online media. The chosen online media have a correct use of the Albanian language grammar in their articles. Language correctness is an important factor due to the fact that in a lot of Albanian online media the Albanian language is not correctly used, for example, the letters “e” and “c” are used instead of the letters “ë” and “ç”. The collected opinions are from five different domains: political, economic, higher education, tourism, and waste import.

First, each collected text-document opinion is manually cleaned from unnecessary information as the writer's name, publication date, and pictures. Once cleaned up, the text-document opinion is annotated with the label corresponding to its sentiment polarity. We have used a two-classes annotation schema: positive and negative. A text opinion is annotated as positive if the overall polarity of the sentiment expressed by it is positive and as negative if the overall polarity of the sentiment expressed by it is negative. The dataset was annotated only by one person. The dataset has a balanced number of positive and negative opinions.

Table 1 details the information about the corpus.

Table 1 The opinions dataset

Domains	**No. positive**	**No. negative**
Tourism	50	50
Higher education	50	50
Waste import	50	50
Economic	50	50
Political	250	250
TOTAL	450	450

Each text-document of the dataset before being used to train and to evaluate the classification model passes in a preprocess phase. First, we have here used the Albanian Language tool implemented in (Sadiku and Biba 2012). This tool consists of a word tokenizer, a stop-word, number and special character remover and lowercase transformation, and a stemmer. We have

used only the first two components. So, each text-document opinion is passed throw the word tokenizer component and then to the stop-word, number and special character remover and lowercase transformation component. The generated text-document opinion after the preprocessing phase is a bag-of-words text-document opinion.

To train a classification model implemented using CNN network and to evaluate its accuracy the input data need to be a sequence of data. So, the text of the bag-of-words text-document opinions of the dataset are preprocessed in a way to be converted as a sequence of data. We have discussed this in section 4.

4. THE ARTIFICIAL NEURAL NETWORK

This section discusses about the architecture of the used network for the binary opinion classification in the Albanian language based on the polarity of the sentiment expressed by the opinion as positive and negative. The neural network is implemented in Python using Keras (Chollet 2015) and Tensorflow (Abad *et al*., 2015).

The architecture of the network

The implemented model is based on a Convolution Neural Network. Figure 1 depicts the architecture of the network. Table 2 reports in details about the network.

We have used the CNN1 network because this network is very effective to derive features from segments with a fixed length. The difference between a CNN1 network and a CNN2 network relates to the way the feature detector moves over the data.

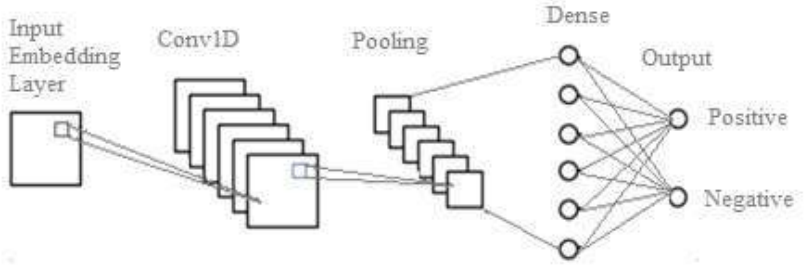


Fig. 1: The architecture of the network.

The input layer is an embedding layer where we have used the pre-trained word vector for the Albanian language from Fasttext (2021). The next layer is a Conv1D layer with 100 neurons and ReLu as activation. A regularizer l2

(0.0002) and SpatialDropout1D are applied to this layer to modify it for better generalizes. These techniques force the model to learn individual features and improve its performance. Then, a Glob-alMaxPooling1D layer is applied to select the most important features. The last layer is a Dense layer with softmax activation that outputs the probability over the two labels, positive and negative.

Table 2. The specification of the model's architecture

Model Architecture	
Input Layer	Embeddings
Hidden Layer	Conv1D
Nodes	100 nodes
Activation	ReLu
Activity regularizer	l2 (0.0002) and SpatialDropout1D
Pooling layer	GlobalMaxPooling1D
Output Layer	Dense
Nodes	2 (1 for positive & 1 for negative)
Activation	Softmax
Optimizer: Adam Loss Metrics	 lr=0.001 Categorical cross entropy Accuracy
Batch size	100
Verbose	1
Epochs	10
Validation slip (training)	0.1
Validation split (testing)	0.2

Experimental Evaluation

To train a model and test its performance the data used cannot be bag-of-words but must be a sequence. So, the bag-of-word data generated from the preprocessing are converted into a sequence of data. The vocabulary of the dataset is expanded using words from the embedding model and then the vectorized dictionary is built where each word is represented by a number. After that is created the embedding matrix.

The dataset is split into training data and testing data, and the ratio is 80:20. In training, we have used a batch size of 100, verbose of 1, a window size of 2, and validation split 0.1. We stopped the training after 10 epochs, because the model's performance is not improved anymore, and it goes in overfitting.

The performance of the model is evaluated in terms of accuracy and the evaluation phase are used unseen data before from the model. The model has an accuracy of 73%. Table 3 shows the experimental result.

Tab. 3 Experimental result

Model	Accuracy
CNN	73%

5. CONCLUSION AND FUTURE WORK

In this paper, we present the implementation of a neural network model for opinion classification in the Albanian language in a two-classes annotation schema: positive and negative. The neural network is implemented using Keras (Chollet 2015) and Tensorflow (Abad *et. al*., 2015). The network has 4 layers, an embedding layer, a Conv1D layer, a Glob-alMaxPooling1D, and a Dense layer. The model has an accuracy of 73%.

To train and test the classification model, we have created an opinion dataset in the Albanian language. The dataset contains 900 text-document opinions classified into positive or negative. An opinion is annotated as positive if the polarity of the sentiment expressed by it is positive and negative if the polarity of the sentiment expressed by it is negative. Prior to be used for training the model and then its accuracy, the dataset undergoes a preprocessing phase. Here, the stop-words, the numbers, the punctations, the special characters are removed, and all the words converted to lower case. The document generated is a bag-of-words. Then the bag-of-words data are converted as a sequence of data to be ready to be used in training and evaluating the model.

The accuracy of a neural model used for the classification task is affected by the size of the dataset and the annotation quality of the dataset. As future work, we propose enlarging the dataset and the dataset to be annotated by a second annotator to ensure the quality of the annotation. Also, we can suggest the use of deep learning for opinion classification tasks in the Albanian language.

REFERENCES

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Zh, Citro Z. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. HYPERLINK "https://arxiv.org/abs/1603.04467" [1603.04467] TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (arxiv.org) .

Barry J. 2017. Sentiment Analysis of Online Reviews Using Bag-of-Words and LSTM Approach-es. Proceedings of the 25th Irish Conference on Artificial Intelligence and Cognitive Science.

Biba M, Mane M. 2014. Sentiment analysis through machine learning: an experimental evaluation for Albanian. *Recent Advances in Intelligent Informatics. Advances in Intelligent Systems and Computing*, **235**: 195–203.

Chollet F. 2015. Keras. https://github.com/fchollet/keras.

Deriu J, Cieliebak M. 2016. Sentiment Analysis using Convolutional Neural Networks with Multi-Task. Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016). Napoli, Italy.

Haveriku, A, Frashëri N. 2021. Building asentiment analysis modelfor tweetsin Albanian language." Proceedings book of ICITEE 21. Enti Botues "Gjergj Fishta".

2021. https://fasttext.cc/docs/en/crawl-vectors.html.

2022. https://www.brightlocal.com/research/local-consumer-review-survey.

Kote N, Biba M. 2018. An Experimental Evaluation of Algorithms for Opinion Mining in Multi-Domain Corpus in Albanian. International Symposium on Methodologies for Intelligent Systems. Springer, Cham. 439-447.

Kote N, Biba M. 2021. Opinion Mining in Albanian: Evaluation of the Performance of Machine Learning Algorithms for Opinions Classification. *International Journal of Innovative Science and Research Technology*, **6 (6):** 964-973.

Kote N, Biba M, Trandafili E. 2018. A Thorough Experimental Evaluation of Algorithms for Opinion Mining in Albanian. International Conference on Emerging Internetworking, Data & Web Technologies. Springer, Cham. 525-536.

Liu B. 2015. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press.

Liu Q, Zhang Y, Liu J. 2018. Learning Domain Representation for Multi-Domain Sentiment Classification. Proceedings of NAACL-HLT 2018.

Minaee Sh, Azimi E, Abdolrashidi AA. 2019. Deep-Sentiment: Sentiment Analysis Using Ensemble of CNN and Bi-LSTM Models. arXiv:1904.04206 .

Moraes R, Valiati JF, WPG Neto. 2013. Document-level sentiment classification: an empirical comparison between SVM and ANN. ***Expert systems*** *with applications***, 40 (2)**: 621-633.

Sadiku J, Biba M. 2012. Automatic Stemming of Albanian Through a Rule-based Approach. J*ournal of International, Research Publications: Language, Individuals and Society*, 173-190.

Schuller B, Mousa A, Vryniotis V. 2015. Sentiment analysis and opinion mining: on optimal parameters and performances. *WIREs data mining knowledge discovery*, **5:** 255–263.

Skenduli MP, Biba M, Loglisci C, Ceci M. 2018. User-emotion detection through sentence-based classification using deep learning: A case-study with microblogs in Albanian. International Symposium on Methodologies for Intelligent System.

Tang D, Qin B, Liu T, Yang Y. 2015. User modelling with neural network for review rating prediction." IJCAI.

Vasili R, Xhina E, Ninka I, Terpo Dh. 2021. Sentiment analysis on social media for Albanian language. *Open Access Library Journal*, 1-31.

Zhang L, Wang Sh, Liu B. 2018. Deep learning for sentiment analysis: A survey. *WIREs Data Mining Knowledge Discovery*. **8 (4). https://doi.org/10.1002/widm.1253**

AJNTS No 57 / 2023 (XXVIII)

TWITTER SENTIMENT ANALYSIS FOR ALBANIAN LANGUAGE

Alba HAVERIKU
Faculty of Information Technology, Polytechnic University of Tirana, Albania
Neki FRASHËRI
Albanian Academy of Sciences, Tirana, Albania

ABSTRACT

The large amount of data generated in social media platforms has unavoidably become an important source of understanding human opinions and behaviors. Models of sentiment analysis and tools to process social media data have been developed in English, German, Italian etc., but studies related to the Albanian language remain limited. The present paper aims to: i) provide a better understanding of the steps needed in a sentiment analyzer and, ii) present a model to automatically process tweets in Albanian language. The process starts with the cleaning and classification are the first steps of this process, while the generation of meaningful results is the final one. Entity and text analysis are used to provide the best insights and a better understanding of humans' opinion about different trending topics. The comparison between three machine learning classifiers (Naïve Bayes, SVM, Logistic Regression) is here made to address the best classification method. The performance of these classifiers is evaluated based on statistical accuracy tests.

***Keywords*:** social media, classification methods, opinions

1. INTRODUCTION

Social media platforms are a key part of digital strategies of public and private institutions. Social media interaction is an umbrella term that encompasses all the two-way conversations and touchpoints that occur between the users. Twitter is one of the most used micro-blogging services nowadays, counting more than 330 million active users (Statista, 2021). The topology of the Twitter network is one directional, and each tweet is limited to 280 characters (Twitter Developer Platform, 2021). The use of Twitter data is

linked to the provision of the Twitter API, that is a well-structured API which provides a high level of accessibility towards the information that users accept to make public.

Social media analytics involves the extraction, analysis and interpretation of the information generated on social media platforms, converting data into meaningful insights. The process of building an application for 'Social Media Mining' is reported in (Bonzanini 2016) as following:

- Authentication - linked with the OAuth (Open Authorization) standard used in Twitter;
- Data collection - the obtaining the information related to a user or topic;
- Data cleaning and pre-processing - the data need to be transformed in a readable format for further processes;
- Modelling and analysing - depends on the results that need to be achieved

Sentiment Analysis (SA) is a subfield of natural language processing which analyses the sentiment and emotions expressed by users related to a product/ service/ topic or any other user in the network (Rusell and Klassen, 2019). Observing sentiment from social media platforms is possible through the combination of NLP and machine learning tools. The SA techniques provide an automatic way to analyze conversations in social media, allowing organizations to learn their customer's opinion and needs. Different algorithms can be used in a sentiment analysis model such as: rule-based, automatic and hybrid (Rusell and Klassen 2019). In this study, we are going to use machine learning techniques (which are part of the automatic category) to learn from the data in disposal. While using these techniques we consider two main processes: training and prediction (Liu 2015). A training dataset is used to teach the model how particular inputs can be associated with an appropriate tag. After the model is trained, it can be used to predict the perfect tag that matches each input given in the model.

The present paper aims to find an appropriate solution to influence different social groups via data mining techniques. Python has been chosen as a programming language with a rich ecosystem, easy syntax and semantic which together provide the best means for both beginners and advanced users. Using Python and its associated libraries provides an answer to another important question: "How can all SA steps be established?". The main Python libraries used in each process are Python Docs, (2021):

- Tweepy: Twitter REST API is used to access previously generated tweets;
- NumPy (Numerical Python): To process effectively data structures in an array like form;

- SciKit-learn: To have access to the main classification algorithms that will be used in this work (Naïve Bayes, Logistic Regression and SVM);
- NLTK: To process and classify textual representations;
- Jupyter Notebook: To code live and directly view graphical representations and textual input.

The structure of this work is as following: Section 2 presents research studies linked with Sentiment Analysis; Section 3 explains the methodology followed; Section 4 presents the experimental results achieved and Section 5 provides the main conclusions achieved and possible future work.

2. Related Work

Many papers provide information about Sentiment Analysis for languages such as English, German, Chinese etc., but studies about Albanian language remain quite limited. The first approach for SA in Albania could be found in (Biba and Mane 2013). After experimenting with different classifiers, they concluded that the best performing classifier for their corpus was Hyper Pipes, with an average accuracy of 87%. Trandafili *et al.,* (2018) compared the performance of text classification algorithms in a corpus with 20 classes (40 document each). Using the Weka software to test different algorithms (Naïve Bayes, Random Forest, SVM, Decision Tree, K-Nearest Neighbor, Simple Logistic, ANN), they concluded that the best performing algorithm in their corpus was Naïve Bayes and SVM (Trandafili *et al*., 2018). Skenduli *et. al*., (2018) made a comparative analysis between different classifiers using the collection of around 60000 posts from Facebook, which belong to 119 Albanian politicians, and the results showed deep learning techniques having a better performance than other classical machine learning classifiers. Kote *et al.,* (2018) created 5 corpuses each containing 50 text documents of positive and 50 of negative opinions. WEKA software was used to perform experiments and evaluate the performance of classification algorithms. The result achieved after the experiments is that for different corpuses, different algorithms give different performances. Hyper Pipes was evaluated as the algorithm with the best performance (83.62%) in (Kote *et al*., 2018). Additional interesting information about text and emotion classification in Albanian language could be found in (Skënduli and Biba 2013; Voca and Kadriu 2015; Kadriu and Abazi 2017; Vasili *et al*., 2018; Kadriu *et al*., 2019; Kote *et al*., 2019; Kote and Biba 2021; Vasili *et al*., 2021).

3. METHODOLOGY

Here, the steps needed to label posts in Twitter, assigning each post a positive or negative sentiment, is reported. The collection and processing of

the available data is part of an elaboration process. Figure 1 depicts the model used, including all the steps needed for the tweet's classification. While trying to build a proper model for collecting and processing the available information, the necessary details to be taken in consideration are directly linked with the characteristics of the Albanian language and the format of each post. The forthcoming paragraph provides detailed information about the process of cleaning, processing, classifying and visualizing.

Main objectives

The present paper aims to; i) present different methods that can be used to develop a sentiment analysis model in Twitter and, ii) present the achieved results. It presents the necessary steps that can be followed to label each post with a sentiment, providing an overview of the main NLP methods for students and other interested individuals that are new in this field. The dataset created by (Mozetic *et al*., 2016) is used as a training and testing dataset. It contains datasets for 15 different languages, including Albanian language. The authors concluded that the quality of the training data directly affects the quality of the classification model (Mozetic *et al*., 2016). By using the dataset from (Mozetic *et al.*, 2016) as training and testing dataset, an opinion analysis could be carried out. To proceed with a concrete example, the analysis is going to be focused on extracting people's opinions and feelings from their posts in Twitter. The following steps are followed to achieve the results:

1. Pre-processing of the training data, so that they can be ready for the mining process (data cleaning, the removal of hashtags and other users mentioning);
2. Training and testing the classification algorithms with the (Mozetic *et al*., 2016);
3. The evaluation of the result from different classifiers

Using the Pareto Principle, 80% of the data is used for training purposes and 20% of the data is used for testing (Bonzanini 2016). To achieve the expected results in the classification module, we the Naïve Bayes, SVM, and the Logistic Regression classifiers were used. The classification algorithms are accessed from the NLTK Python library and are fed with the training data and then tested accordingly.

The proposed architecture

The architecture in the Figure 1 represents all the steps followed. The process of pre-processing and cleaning are described below.

In the end, the main result to be achieved by using this model is the automatic classification of the posts into two categories (positive and negative). In addition, an overview about the important aspects could be developed.

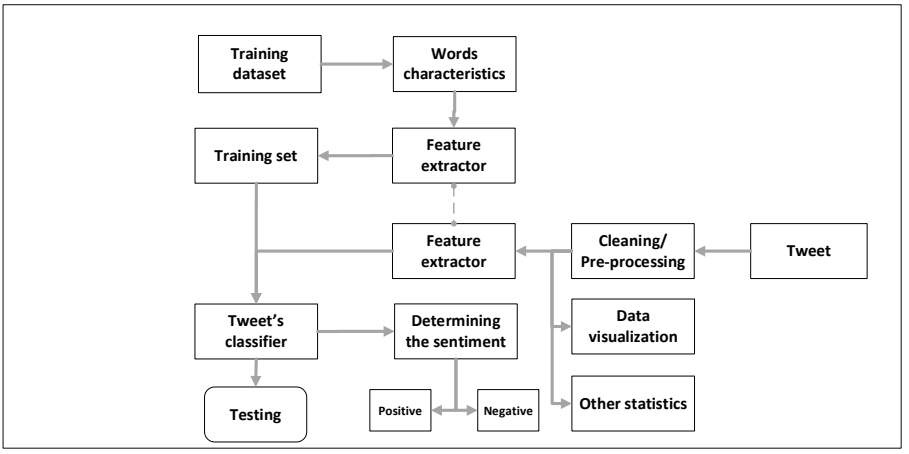


Fig. 7: The proposed architecture

Pre-processing

Having in disposal a previously created dataset (Mozetic *et al*., 2016), it is important to pre-process the data to give them a proper format before the analysis steps. The pre-processing phase is one of the most important phases in the sentiment analysis journey, since it captures the most important words in a sentence or document. Different rules are determined depending on the language of the posts that we are collecting. In this phase we need to clean all the collected tweets, to find the proper message. Specifically, the steps that could be included in the pre-processing or cleaning phase are:

1. Transform all text into lowercase;
2. Remove stop-words; for Albanian language: 'dhe', 'edhe', 'te', 'ne', 'sepse', 'por' etc.;
3. Remove blank spaces, punctuations;
4. If necessary, remove words with less than three letters;
5. Remove hashtags (#), URL (http://....), users tagged (@);
6. Stemming- Consists in removing the end of the words;
7. Lemmatization- Removal of inflectional ending, return the base (lemma) of the word.

An example of cleaned tweet is in the Table 1 reported:

Table 4. Pre-processing a tweet

Original tweet	Në takim me @EPP President @JosephDaul: Moshapja e negociatave ndëshkon Shqipërinë dhe ndihmon politikanët që kanë tradhëtuar qytetarët shqiptarë, duke bashkëpunuar me krimin për pushtetin dhe pasurimin e tyre. #EPPSummit#Wearefamily#AlbaniaintheEU
Pre-processed tweet	takim, president, moshapja, negociatave, ndëshkon, Shqipërinë, ndihmon, politikanët, tradhëtuar, qytetarët, shqiptarë, bashkëpunuar, krimin, pushtetin, pasurimin

Table 1 shows that cleaning a tweet is an important process, as unnecessary elements are removed. Pre-processing provides a way to group the most important words in a post and offer an easier classification process.

The pre-processing function is built in Python and leads the way to the following steps. From the mentioned pre-processing steps, we have considered the following: transformation of the words in lowercase, removal of some of the most common stop words in Albanian, removal of blank spaces and punctuations, removal of URLs, hashtags (#) and user mentions (@). The stop words list is composed of a group of 173 stop words in total. The word_tokenize function from the NLTK's library is further used to return a tokenized version of the initial text, which in this point contains the main words that are linked with the text meaning.

Stemming and Lemmatization are not in the present investigation included. The use of proper stemming algorithm and lemmatization would further optimize the usage of this model, leaving space for future experiments and optimization.

Training and testing dataset

Mozetic *et al*., (2016) said that dataset is used to train and test the classification algorithms. In addition, it is composed of around 53,005 posts, 8,106 of which express a negative sentiment, 18,768 neutral and 26,131 a positive sentiment. Since in this work only the positive and negative labels are taken into consideration, the total number of records is estimated to be 34,237. The dataset saves the following attributes for each tweet: Tweet_Id, Sentiment (Positive, Negative or Neutral) and the annotator's Id (Mozetic, 2016).

The majority of the techniques used to evaluate a model are rated according to the comparison between the label generated during the test and the proper label the input should have. The testing process has the same format as the training but the information differs.

Classification algorithms

To classify tweets in different classes (positive or negative) we need to build a classifier, which in this case is done with the help of the NLTK library. This library is one of the most powerful libraries in Python, at least for the classification algorithms included in it (Rusell and Klassen 2019). A script in Python is built to import these algorithms.

With the data in disposal in the training dataset, we can guarantee a satisfactory analysis of the Albanian posts, but there is a necessity to create a way for more detailed statistics with the help of a better constructed dataset in the future. The block diagram of this module is accordingly expressed in Figure 2.

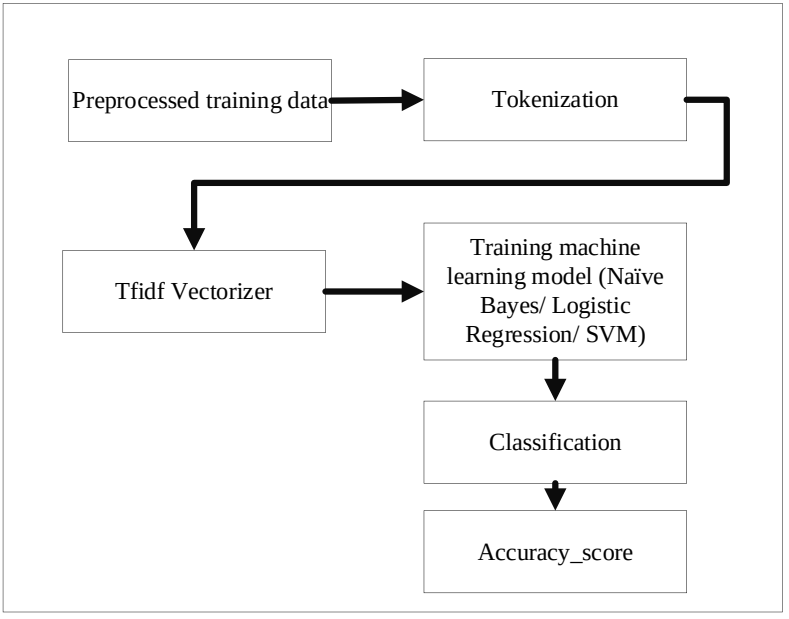


Fig. 8: Block diagram of the feature extraction module

The list of all the expression contained in the respective dataset is transformed into a list of pairs (word, sentiment). The fit and predict methods from the sklearn library are used respectively in the training and testing dataset (PythonProgramming, 2021). The fit function calculated the mean and variance of each of the features in the data in disposal, while the predict function uses the calculated weights to make predictions in the test data. The used classifiers are Naïve Bayes, Logistic Regression and SVM.

The initialization of the classifier provides the possibility to test it with different collected tweets. The training process provided the classifier with the

possibility to differentiate the presence of words and the sentiment related to them. In Jupyter Notebook we tried the following command:

```
tweet='Situata e sotme ishte e pakëndshme'
classifier.classify(extract_features(tweet.split()))
```

The classifying method (classify) takes as argument a group of tweet characteristics, which in this case contains the word 'pakëndshme' which is linked to a negative sentiment. The enlargement of the dataset would provide a better classification process. Once the training and testing process are performed, different data could be inputted in the model and produce a final targeting label.

4. Experimental results

The accuracy score function from the NLTK library is used to calculate the accuracy of each the classifiers. The evaluation of the classifier's accuracy is done by passing to the trained classifier the testing dataset. The results achieved are presented in Table 2.

Table 5. The accuracy of the classifiers

	Naïve Bayes	**Logistic Regression**	**SVM**
Accuracy	77.3%	79.7%	78.7%

Table 2 shows that the classifiers presented a good level of accuracy. The Logistic Regression classifiers present a higher accuracy rate for the dataset in disposal. For this reason, all the results achieved until now are relatively good to take into consideration. The classification and testing involved, provide a good model for the automatic classification of different posts regarding the sentiment they express.

By using this model, and the structure given, more advanced cases can be studied to achieve meaningful results for the future. We need to mention that by further checking the dataset and by using other pre-processing techniques, or even creating a more specific dataset, the classifiers accuracy in determining the sentiment may change and provide a more accurate final result.

5. CONCLUSIONS

This work provides the starting point towards sentiment analysis of text found in social media platforms. By providing all the steps that need to be followed the model represented is a good source of information for students and other interested individuals who are interesting in further developing their knowledge in natural language processing and opinion mining.

The built model can be used to determine a result for a specific topic or for a public person. The most important part is that by determining an Albanian language dataset, it is possible to classify the Albanian posts of different public personalities or Albanian companies. This model, even though is still in its first steps of implementation, can be further advanced to be helpful in different fields such as finance, marketing, innovation and so on.

Further on, a combination of the opinion and sentiment analyses linked with real time events, would provide a better possibility to understand what is happening around the world, or specifically in Albania. Political, artistic or sportive events can be processed in real time to gain a better insight about people's responsiveness.

Meanwhile, there are a lot of features to be fixed and updated so that to provide better results. From the review of literature and related works, we think that there is a necessity to create a well-structured public dataset in Albanian language for research and academic purposes. Further on, in this project only two sentiment groups were created (positive and negative). Adding other important categories such as neutral, angry, hope etc. would be beneficial in future cases. In the end, another important part, which would make the usage of this model and the part of visualization or classification more accessible, would be the development of a simple application where each person could search different topics or public personalities and find out the statistics related to them in graphical or statistical form.

REFERENCES

Biba M, Mane M. 2013. Sentiment Analysis through Machine Learning: An Experimental Evaluation for Albanian. *Advances in Intelligent Systems and Computing*.

Bonzanini M. 2016. *Masteing Social Media Mining with Python: Acquire and analyze data from all corners of the social web with Python.* Packt publishing.

Kadriu A, Abazi L. 2017. A comparison of algorithms for text classification of Albanian News Articles. *Entrenova*.

Kadriu A, Abazi L, Abazi H. 2019. Albanian Text Classification: Bag of Words Model and Word Analogies. *Business Systesm Research, 10*.

Kote N, Biba M. 2021. Evaluation of the Performance of Machine Learning Algorithms for Opinion Classification *International Journal of Innovative Science and Research Technology*, **6 (6)**.

Kote N, Biba M, Trandafili E. 2018. A thorough experimental evaluation of Algorithms for opinion mining in Albanian. *Advances in Internet, Data & Web Technologies*.

Kote N, Biba M, Kanerva J, Ronnqvist S, Ginter F. 2019. Morphological Tagging and Lemmatization of Albanian: A manually Annotated Corpus and Neural Models. ArXiv.

Liu B. 2015. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. doi:https://doi.org/10.1017/CBO9781139084789

Mozetic I, Grcar M, Smailovic J. 2016. Multilingual Twitter Sentiment Classification: The role of Human Annotators. doi:https://doi.org/10.1371/journal.pone.0155036

Python Docs. 2021. Retrieved 2021, from https://docs.python.org/3/library/

PythonProgramming. 2021. *Scikit-learn*. Retrieved from https://pythonprogramming.net/sklearn-scikit-learn-nltk-tutorial/

Rusell MA, Klassen M. 2019. *Miing the Social Web Data Mining Facebook, Twitter, LinkedIn, Instagram, Github, and more*. O'Reilly.

Skënduli MP, Biba M. 2013. A Named Entity Recognition Approach for Albanian. 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE.

Skenduli MP, Biba M, Loglisci C, Malerba D. 2018. User-Emotion Detection thorugh sentence-based classification using deep learning: A case-study with microblogs in Albanian. *International Symposium on Methodologies for Intelligent Systems, Limassol*, 258-267. doi:https://doi.org/10.1007/978-3-030-01851-1_25

Statista. 2021. Most popular social networks worldwide as of July 2021, ranked by number of active users, Retrieved 2021, from https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

Trandafili E, Kote N, Biba M. 2018. Performance Evaluation of Text Categorization Algorithms Using an Albanian Corpus. *Advances in Internet, Data & Web Technologies*.

Twitter Developer Platform. 2021. *The structure of a tweet*. Retrieved 2021, from https://developer.twitter.com/en

Twitter Developers. 2021. *Rate Limiting*. Retrieved 2018, from https://developer.twitter.com/en/docs/basics/rate-limiting.html

Vasili R, Xhina E, Terpo D. 2021. Sentiment Analysis on Social Media for Albanian Language. *Open Access Library Journal*, **8**: 1-21, doi: 10.4236/oalib.1107514.

Vasili R, Xhina E, Ninka I, Souliotis T. 2018. A comparative Review of Text Mining & Related Technologies. *RTA-CSIT, Tirana*.

Voca B, Kadriu A. 2015. A scheme for Albanian Language Processing. DSC2015. Thessaloniki, Greece.

TRANSMISSION OF TELEVISION CONTENT VIA INTERNET: AN ANALYSIS AND PRACTICAL EVALUATION

Aleksandër BIBERAJ and Elson AGASTRA
Faculty of Information Technology, Polytechnic University of Tirana, Albania
Eni HAXHIRAJ and Ardit SKËNDULI
Technical Department, Digitalb, Albania

ABSTRACT

The transmission of television content via internet, otherwise known over the top television (OTT) has expanded the television audience by reaching individuals who want to see their favourite content without a predetermined timetable or in a lot of different devices. It is an innovative technology, which is being used increasingly from OTT service providers that have started their activity in the recent years as well as traditional broadcast companies, which are adapting to the market requests. Although it is an important part of the industry, there is an insufficient amount of information regarding its technology and operation. In addition, this information is often unclear and unorganized. This paper gives a complete view of the OTT TV systems, including the technical challenges, transmission protocols and infrastructure. This explanation is provided using a 'demo' OTT Live platform with one channel and several clients. Building this system, revealed a lot of difficulties related to the creation of an OTT platform, which brings a lot of benefits to the service providers and users. However, there are a lot of aspects that can be improved through proper research like transmission latency, which is considerable due to the long path, which the content follows before reaching the client application, advertisement inclusion in a suitable way and maintaining a good quality of experience for each viewer in every device to ensure the longevity of the OTT television service.
***Keywords*:** over the top, television, live, transcoder, origin server, packager, CDN

1. INTRODUCTION

Over the top television is defined as the transmission of television content through the internet infrastructure and protocols. The last decade marked the introduction and development of this new technology, but in the first decade of the 2000s, television and the internet were completely separated from each

other. It was only in 2010 that OTT TV started to attract the attention of viewers and the media companies, considering Netflix's history of success and the technological advances which allowed television content to be viewed in a lot of different electronic devices (Lotz 2018). Over the top television has been successful in the distribution of live content and video on demand. One of the main reasons for this success has to do with the possibility to view the content in every device that has an internet connection such as smartphones, tablets, smart TVs etc. Given the role that internet has in people's everyday life, OTT TV is now accessible for most of the world's population and has attracted new audiences which were not satisfied with the traditional terrestrial, satellite or cable television (Blanc 2017; Sadana and Sharma 2021). However, OTT TV is still considered as a novelty in television industry. and as a result, there is little information regarding its technology, operation, and architecture. Also, this information can sometimes be unclear and difficult to understand. This fact brings the necessity for a detailed analysis of over-the-top TV service, which is the purpose of this paper (Taylor 2019). The following sections explain the full architecture of an OTT system as well as the new streaming protocols and technical aspects of the system. This explanation is accompanied with a demo platform, which is built to study more closely the constituent parts of it and to monitor all the content processing that is made before it is published on the internet and distributed to the client. The last section of the paper gives the conclusions and highlights the innovative aspects of over-the-top television and the challenges that it faces in order to achieve the same performance as broadcast TV.

2. VIDEO STREAMING PROTOCOLS

Protocols are often described as a set of rules that enable a successful communication and data transmission, but video streaming protocols have several characteristics related to their specific application, because most of the video content is not created for streaming purposes. This means that first it is necessary to convert the video into a suitable format. This includes breaking it up in small chunks, which are transmitted sequentially and are played the moment they are received. This is the core functionality of video streaming protocols, but they are more complex, because they usually use adaptive bitrate delivery. This technology is implemented by transcoding video content with multiple profiles, which means that the same content is transcoded with different resolution and bitrate. The Internet connection of different people does not have the same speed and even the connection of a specific user changes in time depending on circumstances. For this reason, OTT TV protocols offer different profiles of the same content, and the client application evaluates the Internet connection at the moment of viewing and chooses the

profile that is more suitable for the specific client, so every user gets the best quality that he can support. Some protocols focus on different aspects of the streaming process such as the latency or content encryption, but there are three most popular and used protocols HLS, MPEG-DASH and Microsoft Smooth Streaming. HLS was originally designed exclusively for Apple devices, but nowadays it is supported by a lot of devices and browsers, and it is in fact the most used streaming protocol. It ensures a good quality of video with low cost and high security. MPEG-DASH or Dynamic Adaptive Streaming over HTTP is a streaming protocol, which was developed by MPEG (Moving Pictures Expert Group), with the purpose of creating an alternative to Apple HLS. The crucial difference between these protocols is their ownership. Also, HLS supports only some specific video and audio formats whereas MPEG-DASH operates with all formats. Microsoft Smooth Streaming is a streaming protocol, like two protocols explained above, but unlike them, Smooth Streaming uses CPU usage as an indicator in choosing the right profile. This new indicator is specifically useful in mobile devices such as smartphones or tablets (Barz and Bassett 2016).

3. OTT TV PLATFORMS' ARCHITECTURE

OTT TV services use the internet network to distribute live or on demand video content to viewers. Although they use Internet attributes for the data transmission, there is a need for some specific processing steps in order for the content to be suitable for distribution to the client. The full architecture of an OTT platform is given in the Figure 1. Firstly, all television content (Live Streaming, Video on Demand) goes through the transcoder, which changes different characteristics so that this content is in the required format. Transcoding is the process that ensures adaptive bitrate delivery. The next step in an OTT platform is the packager that breaks up the content in fragments or chunks and creates the manifest files, both of which are published in the origin server, which is the server responsible for processing client requests and serving the content to the client application (Blanc 2017). However, an OTT platform with only one origin server that communicates with all the clients would not be efficient, because this server would become a one point of failure. In an effort to avoid this problem and also to improve the quality of viewing, OTT TV uses Content Delivery Networks, that are a group of servers placed in different geographical locations (Oliveira *et al*., 2018). These servers save the content in their memory so that when a client sends a request, it can be served to him directly from the closest server. After the CDN, the request is delivered to the client application, which is the software that has built the request for the content and when it receives this content is responsible for decoding and playing it. This is the transmission chain, in which the content

goes through, but there are two other important systems that complete the whole OTT platform which are the DRM and the Middleware. Digital Rights Management or DRM is a system that ensures that a specific content is available only for authorized clients and it does this through encryption, which actually happens in the packager, but it is the DRM system that authenticates the client and distributes the encryption and decryption keys towards the packager and the client application respectively. The middleware is a very important software for the operation of an OTT service with many clients and many channels because it manages the whole system. The middleware gives the client the opportunity to view his requested channel without the need for knowledge regarding manifest files and URLs. It guaranties a successful communication between all the components of the platform, which are designed by different vendors, and it communicates constantly with CRM (Customer Relationship Management) to exchange information about billing and information about the clients and their rights so that together with the DRM, they ensure that the OTT service is received only by authorized clients.

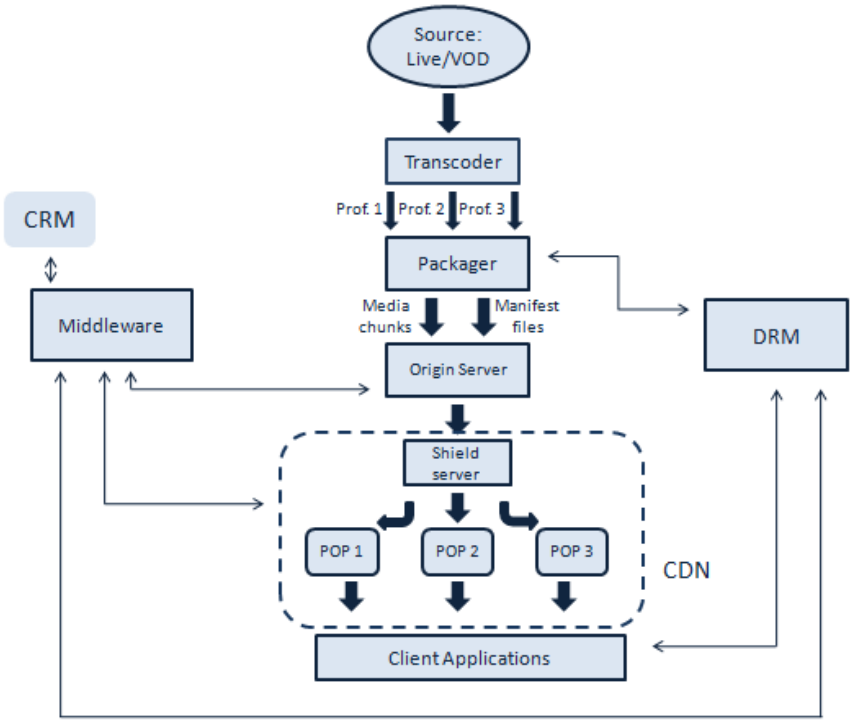


Fig. 9: Architecture of an OTT TV platform.

Transcoding

Transcoding is the process of transforming the media content in a different format to ensure that it is suitable for platforms and applications with different requests. Two of the most important components of transcoding are trans-sizing and trans-rating. Trans-sizing is the change of the video size which includes not only the place that this video occupies in the hardware memory, but also the resolution. Trans-rating is a similar process, but it has to do with the bitrate. It is necessary as the process that enables adaptive bitrate delivery.

Packaging

As aforementioned said, once the transcoding process is done, the content is transformed into a specific format, but an application responsible for playing the video requires more information than what can be included in an individual media file such as available resolutions and bitrates, available audio and video formats, audio languages, subtitles, and advertisement insertion points. HLS and MPEG-DASH protocols put all these information in a single file called a manifest file. In the HLS protocol this file is called a Master Playlist, which includes the bandwidth, resolution, and coding format for each profile and the URL where this format can be found. Media Presentation Descriptions give the same information in the MPEG-DASH protocol, but it is slightly more organized since it arranges media profiles into Adaption Sets. In an OTT TV platform, the packager does two essential services. It fragments the content in chunks, and it prepares the manifest file, as it is explained above.

Origin Server

An origin server is a computer with the main responsibility of processing and answering the clients' Internet requests. It can be the only part of the OTT platform responsible for delivering the content to an Internet entity such as a website as long as the traffic does not exceed the server's capabilities and short delay is not a priority. The physical distance between the origin server and the client increases the latency and as a result it increases the loading time of an Internet source. Using a Content Delivery Network (CDN) is the current solution for reducing round trip time and the number of requests that are handled by the origin server.

Content Delivery Network

A Content Delivery Network can be defined as a server platform located in strategic positions with the goal of decreasing the physical distance between a client and the server that responds to him. Usually, a CDN has three kinds of servers: the origin server, the shield, which is a server that protects the origin from overloading with requests and the points of presence (POP). Once a

request arrives, the shield subsequently checks the local memory and if the information is not found, the requests go to the origin. The third kind of servers which is the most common in a CDN is POP, which is a cache server located far from the origin. These POPs answer the clients request with the version of content that they have in their cache memory. If one POP does not find the requested files it searches them in other POPs of the network and only when content is unavailable or not updated, the client request goes to the origin server (Aljumaily 2016; Al-Abbasi Aggarwal *et al*., 2019) A Content Delivery Network has a lot of advantages such as availability, scalability, security, and a better performance for the platform (Held 2011; Zolfaghari *et al*., 2020).

4. BUILDING A DEMO OTT LIVE PLATFORM

Description of the demo platform building process

The following paragraph describes every step in the process of creating a demo version of a Live OTT platform, which is a small platform with one live channel, that does not include all the parts of the infrastructure but is very helpful in completing the OTT view and presenting a practical demonstration of the information explained previously. It also gives more insight into the whole processing flow of the signal and all the files that are created for specific needs.

The input signal of this platform is an IP stream that is generated based on DVB-T2 standard, which means that it is adapted for traditional terrestrial broadcast. According to the DVB-T2 standard the television signal is transmitted as an MPEG transport stream, which encapsulates several elementary streams, with different PIDs (Program Identifier) for different elements of the signal such as video, audio, subtitles etc. As it is mentioned earlier the OTT transmission requires TV signals with different characteristics that are suitable for the streaming protocols of the Internet. Therefore, after passing through the processing components of the demo platform, the output signal is presented as a set of files, whose type depends on the file-based protocol used for transmission such as HLS, MPEG-DASH, MSS etc. Each of these files contains the video, audio and metadata components of the signal. In the OTT platform each of these files is considered as a “chunk”, whose duration can be configured in the packager. However, for a specific time length we expect to have more than one file, considering the use of adaptive bitrate streaming, which means the same part of the TV content will be represented as different files with different bitrates and resolution. The necessary information regarding these files, their characteristics and order is included in the manifest file. In the final step of the demo platform, the player of the client app will request the proper files depending on the client device

screen resolution, network connectivity capacity and will play the content as a continuous stream.

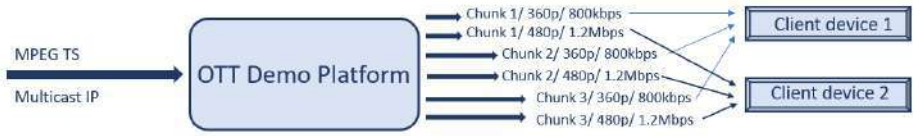


Fig. 10: Input and output signal of the OTT demo platform

In the demo platform, the two most important components are the transcoder and the packager, which are implemented together with the origin server. Although every part of the OTT architecture has a role in ensuring a good operation and quality of experience for the viewer, these two parts are crucial for the transmission. They enable the distribution of content via the Internet, while other components like DRM, CDN or Middleware offer services that facilitate the management and avoid problems that may rise in a platform with more content and more clients. Regarding the client application, the software chosen is VLC, which is an application that can be installed in every client's equipment.

Hardware equipment

In the creation of the demo OTT Live platform, there was a need for several computers and hardware equipment. First, it was used an Edge Probe Nano monitoring DVB-T/T2 signal equipment (TestTree 2020), which serves as a receiver of the terrestrial signal. It receives a DVB-T2 stream, chooses a specific channel and streams it in a multicast IP address. This multicast stream is captured by the transcoder, which is the HERO Live transcoder from Media Excel (MediaExcel 2021). This software is installed in a standard datacentre server running over a virtualized platform. The next device is a standard desktop computer with installed Ubuntu Server 21.04 operating system. This computer contains the Unified Streaming origin server (Unified 2021). A standard network and management computers are connected in a LAN (Local Area Network) for management purposes. Each build platform is accessed by the management computer either from their respective web interface, or from the PuTTy ssh client and WinSCP FTP client. In the same network, a standard computer is also used as OTT client with VLC application installed. Another client equipment used is an Android Box, which is connected to a TV and allows the installation of the VLC application. A switch is used to connect all of the mentioned equipment with each other in a small local network. OTT is the transmission of the signal via the Internet, but taking into consideration the

fact that the Internet is a very large network, using a small local network would not bring any difference.

Demonstration of the demo OTT platform

The following pictures demonstrate and explain the creation of the OTT Live platform. Figure 3 shows the web management interface of the HERO Transcoder, which has the option of configuring different groups of transcoders along with every channel, which has only one input and have one or more output. That means that different profiles can be created for the same content to enable the adaptive bitrate transmission. The channel configuration also includes pre-processing, logo overlay and video and audio pre-set for every output, as in the Figure 4 depicted.

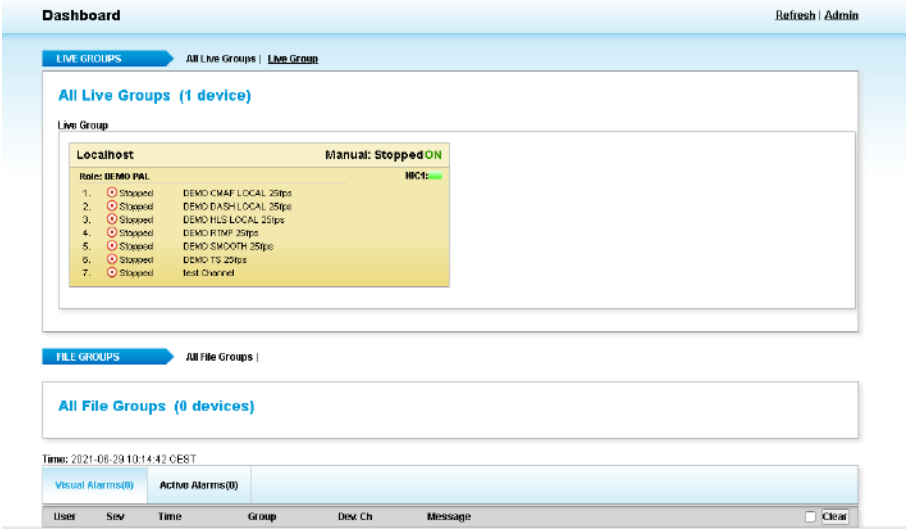


Fig. 3: The transcoder web management interface

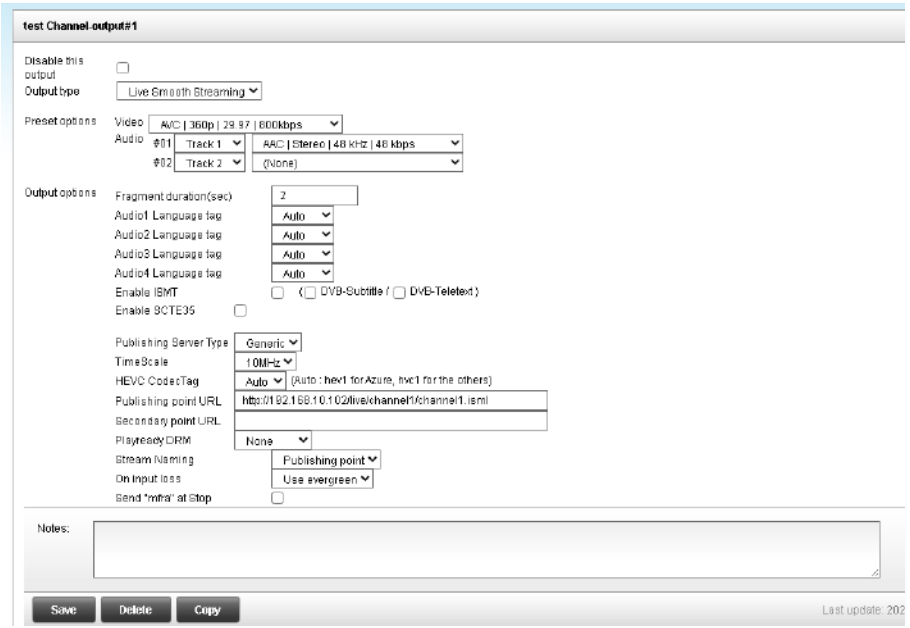


Fig. 4: The channel configuration in the transcoder.

As aforementioned said, the packager and origin server used in this platform in the Unified Streaming origin, which is installed in a computer that also contains the Apache software. This is an open-source software that enables the creation of several web servers in a single physical machine. A web server is called a virtual host in Apache, and it is set up by editing its configuration file. After configuring the web server, the next step is the creation of the publishing points, which are the folders, where the transcoder posts the content. In this platform, the transcoder and the origin server communicate through the Microsoft Smooth Streaming protocol, therefore the publishing point is created by writing the server manifest file (.isml). The script, executed to create the publishing point is shown in figure 5, while figure 6 shows the manifest file.

```
#/bin/bash

FOLDER="/var/www/html/live/demoservice"

MANIFEST="$FOLDER/demoservice.isml"

echo $FOLDER

mkdir -m 777 -p $FOLDER

mp4split --license-key=/var/www/usp-license.key -o $MANIFEST \
--archiving=true \
--archive_segment_length=3600 \
--dvr_window_length=120 \
--archive_length=86400 \
--restart_on_encoder_reconnect \
--hls.client_manifest_version=4 \

chmod 666 $MANIFEST

sudo chown -R www-data:www-data $FOLDER
```

Fig. 5: The executable script for creating the publishing point

```
channel1.isml
1  <?xml version='1.0' encoding="utf-8"?>
2  <!-- Created with Unified Streaming Platform (version=1.10.28-22802) -->
3  <smil xmlns='http://www.w3.org/2001/SMIL20/Language">
4    <head>
5      <meta name="clientManifestRelativePath" content='channel1.ismc' />
6      <meta name="creator' content="Unified Streaming Platform (USP)" />
7      <meta name="lookahead_fragments" content='2' />
8      <meta name="dvr_window_length" content='120' />
9      <meta name="archive_segment_length" content="3600" />
10     <meta name="archiving' content="true" />
11     <meta name="archive_length' content="86400" />
12     <meta name="restart_on_encoder_reconnect" content='true' />
13     <meta name="fixed_gop' content="48/25' />
14     <meta name="hls_client_manifest_version' content="4" />
15     <meta name="hls_no_multiplex' content="true" />
16   </head>
17   <body>
18     <switch>
19     </switch>
20   </body>
21 </smil>
```

Fig. 6: The manifest file (.isml) in the origin server.

Once the channel and its respective publishing point is created, the OTT distribution can begin. Figure 7 depicts the monitoring of the channel in the transcoder, while the state of the publishing point folder in the origin server, which now contains the manifest file (.isml) and the chunks for every profile of the channel (.ismv) is in the Figure 8 depicted.

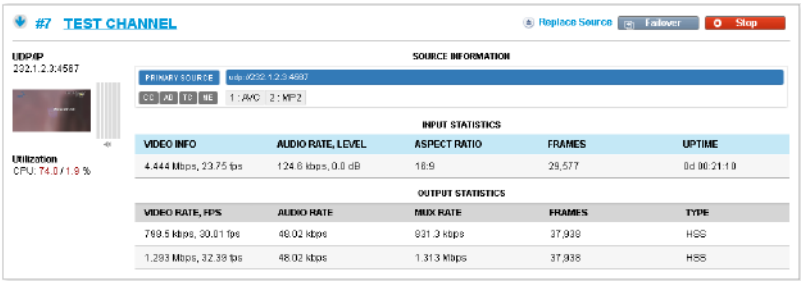


Fig. 7: Monitoring the channel after it is started.

It can be seen that since the moment the channel is started in the transcoder, in the publishing point there are two chunks created at the same time. As expected, there are two files with the same content, but with different resolution and bitrate.

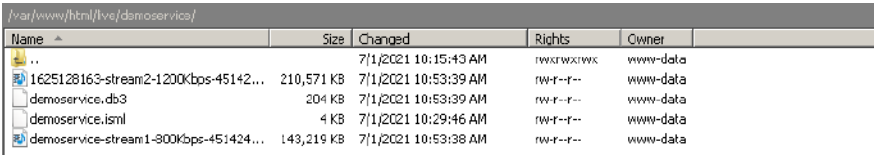


Fig. 8: The publishing point directory during the transmission.

The demo platform, configured with only two profiles, is sufficient to test the OTT transmission's capabilities and adaptiveness to the available resources of several clients. Table 1 shows the chosen profile of several client devices that differ regarding their screen resolution and the available bandwidth of their internet connection. In this experiment, the available bandwidth of the client devices has been controlled, so that the behaviour of the built OTT platform can be monitored, on presenting the right profile to each client that offers the best user experience with the available resources.

Table 6: OTT Profile chosen based on the client devices' screen resolution and available bandwidth

Client device	**Available bandwidth**	**Chosen profile**
Mobile phone	1024 kbps	360p / 800kbps
Mobile phone	2048 kbps	360p / 800kbps
Desktop computer	1024 kbps	360p / 800kbps
Desktop computer	2048 kbps	480p / 1200kbps
Android TV	1024 kbps	360p / 800kbps
Android TV	2048 kbps	480p / 1200kbps

Table 1 reports that the mobile phone, which is a device with a small screen resolution, chooses the 360p/800kbps profile even when the connection bandwidth can support the higher profile. The player of the device does not choose the highest profile available which is the 480p/1200kbps, because the screen resolution of the mobile phone does not support it and it would not bring any significant benefit to the user experience.

It clear that for the desktop computer and Android TV, which have a higher resolution and are considered as 'big screens', the player chooses the highest profile offered by the OTT platform that is permitted by the available bandwidth. In this case, when the available bandwidth is higher than 1200 kbps, the platform choses the 480p/1200kbps profile. When the Internet connection speed is reduced, it does not allow for the TV content to be streamed continuously from the platform to the client using the previous profile. As a result of that, the player chooses the next available profile (with a lower resolution and bitrate) so that the content can be transmitted without any error or missing packet, that would appear as an artifact in the video and would degrade the user experience. This ability to adapt to the client devices and their properties, ensured by the adaptive bitrate transcoding and packaging, is dynamic. This means that the connection of each client is monitored continuously so that the player automatically switches to different profiles when the network conditions change, and the best user experience is offered to all the clients.

Figure 9 shows the content played in the client software, which in this case is installed in a laptop. As explained previously, the computer screen is considered as a 'big screen'; therefore, the player of the client application chooses the higher profile, which is 480 pixels and 1.2 Mbps. Once the creating the demo version of a live OTT TV platform is created, the television signal starts to be played in the chosen client devices, and its profile is adapting to the capabilities of each device, which means that the demo platform is complete and the distribution of the content through the internet has been achieved.

Fig. 9: The VLC software playing the content

6. CONCLUSIONS

Over the top television is the transmission of television content through the Internet. In OTT TV, the viewer has to require the content in order for it to be transmitted. As an industry, the OTT television has developed a lot in the last 10 years, but it still faces many challenges, which have to be solved so that more audiences can be reached. OTT service providers have to analyse and offer flexibility in payment methods, marketing and advertisement inclusion. Moreover, the transmission latency is a performance parameter that appears to be not as good as in traditional TV due to the nature of the OTT protocols, that separate the content in chunks. This parameter needs to be improved, by using suitable algorithms, chunk duration and the exact moment when the video starts to be played (Bjelica *et al.*, 2015; Latkoski *et al*., 2016). The process of building a demo OTT platform showcased two major factors that contribute to the growing popularity of the OTT TV nowadays, which are the low cost and relative simplicity in creating an OTT platform, considering that is built on top of existing Internet infrastructure. Moreover, the demo version emphasized all the system components and their functions, which require further research in order to be developed so that a good quality of experience can be ensured for every client, despite his equipment, internet connection or the content he is watching (Goldstein *et al*., 2020). The build platform allows to monitor and test different video streaming profiles (different resolution and bandwidth) and different CDNs. These tests can be performed not only for the stream profiles as in this experiment, but also for more profiles and several channels with different type of content. The testing results can help to evaluate and fix

bottleneck problems or any other obstacles that may appear during streaming when the number of clients grows.

REFERENCES

Al-Abbasi A, Aggarwal V, Ra M. 2019. Multi-tier caching analysis in CDN-based over-the-top video streaming systems. *IEEE/ACM Transactions on Networking*, **27(2):** 835-847. DOI:10.1109/TNET.2019.2900434 Corpus ID: 60441459.

Aljumaily MS. 2016. Content delivery networks architecture, features, and benefits. Technical report.

Barz HW, Bassett GA. 2016. Multimedia networks: Protocols, design, and applications. First Edition, 223 – 244.

Bjelica MZ, Rikalovic D, Ilkic V. 2015. Minimizing impact of loading time and presentation to user experience in modern Over the Top television. 2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), pp. 228-231.

Blanc J. 2017. OTT TV for Broadcasters: Preserving broadcast-grade quality and services; The OTT encoder-packager point of view. Keepixo Seminar.

Goldstein AB, Belozertsev IA, Elagin VS, Spirkina AV. 2020 Providing QOS for OTT services in communication networks. Downloaded from IEEE Xplore.

Held G. 2011. A practical guide to Content Delivery Networks. Second Edition, pp. 1 – 40

Latkoski P, Porjazoski M, Popovski B. 2016. QoS control in OTT video distribution system, IEEE Eurocon

Lotz AD. 2018. We now disrupt this broadcast: How cable transformed television and the internet revolutionized it all. pp. 111 – 186

Oliveira T, Fiorese A, Sargento S. 2018. Forecasting over-the-top bandwidth consumption applied to network operators. IEEE Symposium on Computers and Communications (ISCC).

SadanaM, Sharma D. 2021. How over-the-top (OTT) platforms engage young consumers over traditional pay television service? An analysis of changing consumer preferences and gamification. *Young Consumers*, **22(3):** 348-367.

Taylor ChR. 2019. Over the top, connected, programmatic and addressable television! What does it all mean? Definitions and a call for research, *International Journal of Advertising*, **38 (3):** 343-344. https://doi.org/10.1080/02650487.2019.1599200.

Zolfaghari B, Srivastava G, Nemati H, Afghah F. 2020. Content Delivery Networks: State of the art, trends, and future roadmap. *ACM Computing Surveys*, **53(2):** 1-34. https://doi.org/10.1145/3380613.

TestTree ENENSYS Technologies. 2020. EdgeProbe Nano DVB-T/T2, compact monitoring probe, https://www.test-tree.com/product/dvb-tt2-compact-monitoring-probe/.

MediaExcel. 2021. HERO live multiscreen encoder/transcoder for live. Austin, Texas, USA https://www.mediaexcel.com/products/live.do.

Unified Streaming 2021. Unified origin stream any format, Amsterdam, The Netherlands https://www.unified-streaming.com/products/unified-origin.

AJNTS No 57 / 2023 (XXVIII)

AGILE IN DISTRIBUTED SOFTWARE DEVELOPMENT: A SYSTEMATIC LITERATURE REVIEW

Evis TRANDAFILI, Besjana MURAKU and Elinda MEÇE
Department of Computer Engineering, Faculty of Information Technology, Polytechnic University of Tirana, Albania

---

ABSTRACT

Distributed teams have long adapted their own versions of implementation the of Agile methods, and the use of many tools to facilitate the access to these methods. However, which is the most appropriate method and the factors impacting it remain unclear. A systematic literature review of the most recent publications to identify the challenges, trends, and the less explored area of Agile Methodologies for distributed software development (DSD) is here made for the years 2017-2021 by including here the pandemic years. The results showed a considerable interest in Agile for Distributed Software Development. Case studies and guidelines and proposal of hybrid models' implementation were considered for the review. Scrum and Extreme Programming are the most reviewed methods with focus on Peer Programming and Lean Programming. At the same time, an important attention is given to scaled agile and its adoption for distributed teams too, where most of the analysis is focused on Scrum of Scrums, Scaled Agile Framework and Disciplined Agile Delivery.
***Keywords*:** Agile, global software development, distributed software development, scurm

1. INTRODUCTION

Global software development (GSD) has an increasing interest both in industry and in academics. The decision to go global for many companies is considered cost-effective, time reducing thus faster development, access to global resources and increased flexibility. In addition, it faces several challenges such as geographical locations, cultures, time zones and languages all affected the team involved, the project structure and processes.

GSD-related topics have been in focus and discussed extensively in literature. However, understanding and examining all the aspects of GSD is

particularly challenging as it varies by the organization type and structure (how are teams are organized and distributed), the form of GSD implementation (whether the teams are independent or distributed), the team experience. It is noticed that each publication usually focuses on a particular challenge and project aspect.

Agile

Unlike the traditional software process development methods that are plan-driven and have more strict phases, agile methods have introduced a dynamic process with less documentation produced but more software delivered in less time, thus responding better to business requirements. In such a powerful procedure, prerequisites are profoundly unpredictable and consistent joint effort is basic to adapt to constantly changing necessities for chance alleviation because of conditions (Martakis and Deneva 2013). Engineer coordinated effort is subject to the correspondence of changes of new errands, just as on the consciousness of what others are doing and whether they are accessible to help (Damian *et al*., 2007).

Agile in distributed teams

There are several publications about Agile and global software development. In recent publications, authors have tried to give best practices also propose new models to minimize the challenges faced. However, there is a lack of information from real cases of industry which results could be found in (Vallon 2018).

Consequently, we will investigate the recent case studies or evidences coming from the industry and identify the trends in this area aiming to provide information about the evidences provided, factors impacting agile development in distributed software development, whether there are new models proposed or novel approaches for Agile in DSD, and the most studied methods in distributed Agile.

2. RESEARCH RESULTS

Information has been obtained from databases of ACM Digital Library, AIS Electronic Library (AISeL), IEEE Xplore, ScienceDirect and any open-source publication. The research query is composed of two main objectives: Agile and global software development. Taking into consideration that there may be different combinations and sub queries, the last version of the query is as follows:

(Agile OR scrum OR "extreme programming" OR "pair programming" OR "lean programming" OR dsdm OR kanban) AND ("global software

development" OR GSD OR "virtual team" OR "global team") AND ("empirical study" OR "case study").

The systematic literature review was implemented in two phases. First, all the papers resulting from the query were reviewed and independently selected by two authors. Second, the papers were categorized and reviewed.

There were 100 articles only for the period of 2020-2021. Once the manuscripts' content was read only 14 papers were selected, relevant to our systematic review. There were 46 papers selected in total for this review.

Following, we provide answers to the research questions by giving in depth details and explanation.

2.1. What evidence are provided (surveys/ case studies)?

The manuscripts could be divided into five categories. Three articles are included in the group of proposal papers for future research (could also considered as 'work in progress') where the authors proposed the field of study and the methods that will be using. Five papers are classified as 'Surveys'. Sixteen publications are classified as 'case studies,' where authors have provided insights from personal experience or real projects of Agile in DSD. There are also identified two papers included into the group of the 'New Model Proposal' papers.

Table 7 Papers categorisation. Total findings for each category.

Category	Papers	Total
Proposal for research	(Razzak, 2017), (Lunesu *et al.*, 2018), (Drechsler and Breth, 2019)	3
Surveys	(Boyer and Mili, 2011), (Werewka *et al*., 2017), (Seckin *et al.*, 2018), (Vithana *et al*., 2018), (Marinho, 2019), (Majdenbaum and Chaves, 2020) (Shameem *et al*, 2020)	7
Case Studies	(Awar *et al.*, 2017), (Inayat *et al.* 2017), (Santos and Nunes, 2017), (Bass *et al.*, 2018), (Costa *et al.*, 2018), (Kahya and Seneler, 2018 (a)), (Kahya and Seneler, 2018 (b)), (Rajpal, 2018), (Paramartha, 2018), (Aggarwal and Mani, 2019), (Bjørn *et al.*, 2019) (Gupta *et al.*, 2019 (a)), (Gupta *et al.*, 2019(b)), (Salameh and Bass, 2019), (Szabó and Steghöfer, 2019), (Uludağ *et al.*, 2019), (Qahtani, 2020), (Shafiq *et al*., 2020), (Moe *et al.*, 2020), (Britto *et al*., 2020), (McCarthy *et al.*, 2020), (Stray and Moe, 2020), (Khan *et al.*, 2021), (Beecham *et al*., 2021), (Geeling *et al*., 2020)	25
Findings from other case studies	(Khmelevsky *et al*., 2017), (Lous *et al*.,2017), (Humble, 2018), (Putta, 2018), (Calefato *et al*.,2020), (Camara *et al*., 2020), (Shafiq *et al*., 2020)	7

New models proposals	(Awar *et al*., 2017), (Kroll *et al*.,2017), (Beecham *et al*., 2021)	3

2.2. What factors impact Agile in Distributed Software Development?

Here the challenges and the drawbacks of Agile in global software development are identified along with the positive aspects that these methodologies combined with distributed software development might have brought.

Some of the main characteristics that global software development needs to address are related to temporal, geographical and sociocultural distances that pose challenges in communication, control, and coordination (Szabó and Steghöfer, 2019).

Agile practices require frequent communications in the team. In the research conducted by (Inayat et al. 2017) are investigated multiple-case study of four large, distributed companies. They found that in some companies, where the teams were involved in distributed projects, communication with the local colleagues was more frequent than with the remote ones and as a result, the members know less about the professional background and are less aware of the tasks that the remote colleague was working on. The work progress of the remote team was also not transparent. This resulted in people being more likely to communicate with someone they knew and that they knew they can help, thus communicating more with those locally than remotely. However, their study, despite the apparent believe, indicates that distance does not seem to matter to communication frequency, and the correlation results between communication, awareness and distance are indecisive. Further investigations need to be conducted for wider cases studies.

A particular case of GSD, where programming is appropriated over a twenty-four-hour working day, is Follow the Sun (FTS) (Carmel *et al.*, 2014). For projects like this, the lack of communication or interaction in real time, the time difference due to no overlapping of working hours or delays in response times for problems, were key challenges for the temporal distance (Kahya and Seneler, 2018). Also, the loss of concentration because of long meetings at late hours is another risk.

As suggested by (Khmelevsky *et al.*, 2017) the lack of face-to-face communication, because of the distribution of the team, should be compensated with rich communications using channels and as in real agile, at least weekly, or biweekly meetings between the teams should be hold. Brainstorming and frequent planed meetings are essential to overcome problems for distributed team.

In Agile there are specific roles that conduct specific activities. The Product Owner, that has knowledge about the system being developed, should be close to the developing team to interact with them (Kelly and Allan, 2019). In cases where the product owner is the client itself, in distributed environment it might not be possible. Thus, for distributed software development projects, it is recommended that the Product Owner should be a member of the team that has the feasibility to relocate near the customer to discuss the business requirements and translate them in user stories for the developing team (Paramartha, 2018). The role of the Product Owner may be found also with the name of Business Analyst. In case of frequent traveling and the utilization of tools for communication and implementing requests an adequate budged to be estimated (Rajpal, 2018).

On research conducted for DSD in India and US or Europe, is observed that the IT developers of India work under various transactional conditions that differ from those who work in US or Europe (Bjørn *et al.*, 2019). The idea of trans locality guides us to consider the accomplished work arrangements as a variety of word related governmental issues, infrastructural availability, and worldwide office, which reach past national fringes. By focusing on the manners by which procedure, work organization and technology shape the trans locality of the working environment we can thus understand their lived work experience in transnational work. Agile methodologies have an advantage of increasing transparency and coordination across team; however, this can be a risk for vendor companies offshore since it can disempower the developers by reducing their decisions, thus having a negative impact in their work (Bjørn *et al.*, 2019).

Technology diversity is another factor impacting distributed software development. Agile methodologies require tools for team communication and coordination. Sometimes, different teams work with different tools and the time of the adaptation is requested. In other occasions the remote team may pose resistance in moving from their own internal environment to something new.

Awar *et al*., (2017) identified the English language as the one of the most challenging issues. Different level of comprehension of the communication language led to misunderstanding or difficulties in understanding requirements or tasks assigned to the development team.

Using the Hofstede model, Lunesu *et al*., (2018) rises hypotheses with respect to effect of social foundation on rehearses appropriation of the teams involved. Ethnic, social, and cultural aspects determine the diversity in socio-cultural. Even though they are still hypothesis, whenever approved, would assist light-footed professionals with identifying early the potential difficulties that they will confront unavoidably (and in this manner to be increasingly arranged to this challenges)

Other challenges are related to correspondence, specialized competency, client commitment, information exchange, innovation, coordination, and control. However, training and coaching and community-oriented improvement were not found to have a noteworthy relationship with project success for distributed teams (Vithana *et al.,* 2018). They are close to personal attitude and building the consistent project success.

One of the most important aspects for organization is lowering development cost, and this can be achieved by going global and delegating the work in other countries (Humble, 2018). In Global Software Engineering (GSE), there are identified three success factors (competences, communication, and collaboration) and three benefits (flexibility, innovation, and efficiency) (Elbert *et al.,* 2016).

Agile methods are developed around communication and transparency that they imply and have a positive impact in global teams. These methods encourage frequent meetings and collaborations that help to reduce the gap between development teams. There are several tools that are used for collaboration (Jira, Cacoo, GenMyModel), for shared project workspace (like GitHub or Subversion) and communication in the team (Skype, GoogleHangouts) (Calefato and Ebert., 2019). Stray and Moe (2020) investigated the impact of instant messaging tool, Slack, in reducing the challenges of geographic distance. This tool supports frequent communication and fast responses within and between teams and their stakeholders, which in turn benefit GSE companies. However, even in mature agile GSE companies using new tools and coordinating with both scheduled and unscheduled meetings, faces the same old barriers – such as language, unbalanced activity, and difficulty with facilitating communication, the authors say.

2.3. Are there new models proposed or novel approaches for Agile in DSD?

Task scheduling is challenging to Global Software Development (GSD). Usually this is a process carried by the project managers who are responsible of creation and distribution of tasks. The process or task assignment becomes especially difficult considering the distribution, the multidisciplinary composition of the team and time zone differences. For FTS projects, the time zone difference is exploited as an advantage of GSD, however poor planning and a poor distribution of the tasks can increase the costs in an unacceptable manner (Penta *et al*., 2011).

Kroll *et al.,* (2017) have implemented a genetic algorithm-based assignment technique that uses a queue-based GSD simulator for fitness function evaluation. Their work contributes on task scheduling in conform to GSD context. Genetic Algorithm-based (GA) have been widely used in many optimizations, search, and machine learning (ML) problems (Camara *et al.,*

2020). This algorithm was also evaluated for task scheduling in three industrial project's data, and then the results have been confronted with the actual PMs (project managers).

Kroll *et al*., (2017) stated that in terms of reducing the project lifespan, the Genetic Algorithm performed as better as, and in cases even better than solution provided by the managers for task assignment to the development team.

Awar *et al*., (2017) proposed new model which is based on practices and state-of –the-art for the process of software development in distributed Agile team. This model divides the process into four phases: pre-implementation, where the goal here is to create a set a baseline for the cross-functional team; implementation, where the team should implement fully the Agile methodology; team-shared understanding, is the phase where issues are looked up by appropriated group. The last phase, post-Development, is expected to fortify the association by utilizing certain practices. However, this model is applied so far only in one case and further investigations need to be considered.

Sinha *et al*., (2020) suggested another model called SWOT model (strengths, weaknesses, opportunities, threats). What is important about this model is that it highlights some of the most principal factors which may influence the organization's future in a GSD system. The findings of this study reported 24 factors among which 13 identified as positive factors and 11 negative factors regarding their impact to scaling program. These factors are further categorized into the so-called SWOT matrix and based on this matrix successful strategies are identified for the organization. This model helps to examine the competitive position of the corporation by assessing the identified swots. This model will assist the GSD organizations to assess and measure their preparation preceding to the implementation of agile development.

2.4. What methods are mostly studied in distributed Agile?

From the resulted database, we identified a major contribution and discussion in regard of Scrum methodology. This method was studied in seven papers from a total of nine papers which had a focus on specific methodologies during the period of 2017-2019 and two papers from three during the period of 2020-2021. One paper was identified in regard of Lean Software Development and another paper was related to Pair Programming.

This major interest in Scrum relates to the fact that it is the most applied method in industry (Boyer and Mili, 2011). Initially considered for non-distributed projects, now Scrum needs to be adapted and customized based on global needs. Most proposals target changing components of the Scrum core procedures (Lous *et al.*, 2017).

Razzak (2017) reported about the Lean Software development. They inform about how leanness encourages adaptability in distributed programming to accelerate improvement process.

Beecham *et al.*, (2021) in their a GSD case study examined two scaling agile frameworks; the Disciplined Agile Delivery (DAD) and the Scaled Agile Framework (SAFe). Both these frameworks put a great emphasis on risk mitigation, so it seemed suitable to the authors to develop a GSD Risk Catalog of 63 risks and then to evaluate their efficacy at tackling global software development project risks by studying how well they covered the software project risks identified the GSD Risk Catalog. It is concluded in the end that the two mentioned scaling agile frameworks address the 63 software development risks in the GSD Risk Catalog, so they can potentially eliminate or mitigate software project risks in global software development. Scrum of scrums is another scaled agile framework suggested to be taken into consideration for further studies in GSD.

3. CONCLUSIONS

We presented a literature review on Agile in Global Software Development with the focus on the real case studies provided by the industry. The study is based on a dataset extracted from the publications of the recent years 2017-2021. We found 46 publications in regard, which is a considerable amount based on the brief period. In these publications we identified reported problems and challenges, but also best practices that resulted in successful stories.

It cannot be generalized in all scenarios if Agile should be the best solution for all cases of distributed software development. Many aspects should be taken in consideration. Results show that distribution its-self poses difficulties, like the lack of face-to-face communication, language misunderstanding, but if the basic guidelines of Agile about the frequent meetings using communication tools and brainstorming are correctly followed, these difficulties can be overcome.

Also, other aspects need to be delved into. Hypotheses are raised on the impact of ethic, socio- cultural aspect, but these need further investigation. In addition, the impact that GSD has on specific roles of Agile are another open point that need further investigation. The present review, only the definition of Technical Lead could be identified.

Furthermore, we identified that scrum is mostly considered by the research, with new proposals of further investigation on lean methodology and scaled agile SAFe and DAD.

Our approach leaves room for extension, as the focus of this review was to identify the state of the art in the industry of the global software development

using agile methodologies, and identify aspects need to be considered in future studies.

REFERENCES

Aggarwal A.K, Mani V.S, 2019. Using product line engineering in a globally distributed agile development team to shorten release cycles effectively. *ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE).* IEEE.

Awar K.B, Sameem M.SH.I, Hafeez Y, 2017. A model for applying Agile practices in Distributed environment: A case of local software industry. *International Conference on Communication, Computing and Digital Systems (C-CODE).* IEEE.

Bass J.M, Beecham S, Razzak M.A, Canna C.N, Noll J, 2018. An empirical study of the product owner role in scrum. In Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings (ICSE '18). Association for Computing Machinery, New York, NY, USA, 123–124.

Beecham S, Clear T, Lal R, Noll J, 2021. Do scaling agile frameworks address global software development risks? An empirical study. *Journal of Systems and Software*.

Bjørn P, Søderberg A-M, Krishna S, 2019. Translocality in global software development: The dark side of global agile. *Human–Computer Interaction 34.2*, (pp. 174-203).

Boyer J. A, Mili H, 2011. Agile business rule development. (pp. 49-71). Berlin: Springer.

Britto R, Smite D, Damm L-O, Börstler J, 2020. Evaluating and strategizing the onboarding of software developers in large-scale globally distributed projects. *Journal of Systems and Software*, Elsevier.

Calefato F, Ebert C, 2019. Agile Collaboration for Distributed Teams [Software Technology]. *IEEE Software 36.1* (pp. 72-78). IEEE.

Calefato F, Dubey A, Ebert C, Tell P, 2020. Global Software Engineering: Challenges and solutions. *Journal of Systems and Software*. Elsevier.

Camara R, Alves A, Monte I, Marinho M. L, 2020. Agile Global Software Development: A Systematic Literature Review. *SBES '20: Proceedings of the 34th Brazilian Symposium on Software Engineering*, (pp. 31-40).

Camara R, Alves A, Monte I, Marinho M, 2020. Agile Global Software Development: A Systematic Literature Review. *SBES '20: Proceedings of the 34th Brazilian Symposium on Software Engineering*, (pp. 31-40).

Carmel E, Espinosa J.A, Dubinsky Y, 2014. "Follow the Sun" Workflow in Global Software Development. *Journal of Management Information Systems 27.1*, (pp. 17-38).

Costa M.C.C, Lemos G.S, Beck F, 2018. Software engineering tools environment for outsourcing teams collaboration. *IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE).* IEEE.

Damian D.I, Izquierdo L., Singer J, Kwan I, 2007. Awareness in the wild: why communication breakdowns occur. *International Conference on Global Software Engineering*, (pp. 81–90). New Delhi, India.

Drechsler A, Breth S, 2019. How to go global: A transformative process model for the transition towards globally distributed software development projects. *International Journal of Project Management 37.8*, (pp. 941-955).

Ebert C, Kuhrmann M, Prikladnicki R, 2016. Global Software Engineering: Evolution and Trends. 11th International Conference on Global Software Engineering. Orange County, CA, USA, pp. 144-153, doi: 10.1109/ICGSE.2016.19.

Geeling S, Brown I, Weimann P, 2020. Cultural levels and emergent cultural contradictions in is development. *In Proceedings of the 28th European Conference on Information Systems (ECIS).* An Online AIS Conference.

Gupta R.K, Jain S, Singh B, Jha S.K, 2019, (b). Key Factors in Scaling up Agile Team in Matrix Organization. *Proceedings of the 12th Innovations on Software Engineering Conference (formerly known as India Software Engineering Conference).*

Gupta R.K, Venkatachalapathy M, Jeberla F. K, 2019, (a). Challenges in adopting continuous delivery and DevOps in a globally distributed product team: a case study of a healthcare organization. *ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE).* IEEE.

Humble J, 2018. Continuous delivery sounds great, but will it work here? *Communications of the ACM 61.4*, (pp. 34-39).

Inayat I, Marczak S, Salim S.S, Damian D, 2017. Patterns of collaboration driven by requirements in agile software development teams. *International Working Conference on Requirements Engineering: Foundation for Software Quality.* Springer.

Kahya M.D, Seneler Ç, 2018, (a). Geographical Distance Challenges in Distributed Agile Software Development: Case Study of a Global Company. *3rd International Conference on Computer Science and Engineering (UBMK).* IEEE.

Kahya M.D, Seneler C.O, 2018, (b) Temporal Distance Challenges in Distributed Agile Software Development: Case Study of a Global Company. 2nd International Symposium on Innovative Approaches in Scientific Studies

Kelly A, 2019. "Who Is the Product Owner?." The Art of Agile Product Ownership. (pp. 21-29). Apress, Berkeley, CA.

Khan A.A, Shameem M, Nadeem M, Akbar M.A, 2021. Agile trends in Chinese global software development industry: Fuzzy AHP based conceptual mapping. *Applied Soft Computing,Elsevier*.

Khmelevsky Y, Li X, Madnick S, 2017. Software development using agile and scrum in distributed teams. *Annual IEEE International Systems Conference (SysCon).* pp. 1-4, IEEE.

Kroll J, Friboim S, Hemmati H, 2017. An empirical study of search-based task scheduling in global software development. *IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, pp. 183-192.

Lous P, Kuhrmann M, Tell P. 2017. Is scrum fit for global software engineering?. *IEEE 12th International Conference on Global Software Engineering (ICGSE).*

Lunesu M.I, Münch J, Marchesi M, Kuhrmann M, 2018. Using simulation for understanding and reproducing distributed software development processes in the cloud. *Information and Software Technology 103*, (pp. 226-238).

Majdenbaum A, Chaves M, 2020. Social interaction promotion in distributed software development in agile projects. *In Proceedings of the 28th European Conference on Information Systems (ECIS).* An Online AIS Conference.

Marinho M, Noll J, Richardson I, Beecham S, 2019. Plan-driven approaches are alive and kicking in agile global software development. *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM).* IEEE.

Martakis A, Deneva M, 2013. "Handling requirements dependencies in agile projects: a focus group with agile software development practitioners. *International Conference on Research Challenges in Information Science*, (pp. 1–11). Paris.

McCarthy S, O'Raghallaigh P, Fitzgerald C, Adam F, 2020. Building Bridges, Burning Bridges: The Use of Boundary Objects in Agile Distributed ISD Teams.

Moe N.B, Stray V, Goplen M.R, 2020. Studying Onboarding in Distributed Software Teams: A Case Study and Guidelines. *EASE '20: Proceedings of the Evaluation and Assessment in Software Engineering*, (pp. 150–159).

Paramartha M.A, 2018. *Requirements Engineering Issues In Agile Distributed Software Development*.

Penta M.D, Harman M, Antoniol G, 2011. The use of search-based optimization techniques to schedule and staff software projects: an approach

and an empirical study. *In Software - Practice and Experience*, (pp. 495 – 519).

Putta A, 2018. Scaling agile software development to large and globally distributed large-scale organizations. *Proceedings of the 13th International Conference on Global Software Engineering.*

Qahtani A.M, 2020. Study of Agile Testing in A Distributed Software Development Project. *In Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis (ICGDA 2020).* Association for Computing Machinery, New York, NY, USA, 110–114.

Rajpal M, 2018. Effective distributed pair programming. *IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE).* IEEE.

Razzak M.A 2017. An Empirical Study on Leanness and Flexibility in Distributed Software Development. *arXiv preprint arXiv:1711.01097*.

Salameh A, Bass J, 2019. Spotify tailoring for B2B product development. *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE.* IEEE.

Santos E.W.d, Nunes I, 2017. Investigating the effectiveness of peer code review in distributed software development. *Proceedings of the 31st Brazilian Symposium on Software Engineering.*

Seckin I, Ovataman T, 2018. An empirical study on scrum application patterns in distributed teams. *Proceedings of the 13th International Conference on Global Software Engineering.*

Shafiq M, Zhang Q, Akbar M.A, Kamal T, Mehmood F, Riaz M.T, 2020. Towards successful global software development. *EASE '20: Proceedings of the Evaluation and Assessment in Software Engineering*, (pp. 445–450).

Shameem M, Kumar R.R, Nadeem R, Khan A.A, 2020. Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process. Applied Soft Computing. Elsevier.

Sinha R, Shameem M, Kumar C, 2020. SWOT: Strength, Weaknesses, and Threats for Scaling Agile Methods in Global Software Development. *Proceedings of the 13th Innovations in Software Engineering Conference on Formerly known as India Software Engineering Conference.*

Stray V, Moe N.B, 2020. Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack. *The Journal of Systems & Software, Elsevier*.

Szabó D.M, Steghöfer J-P, 2019. Coping strategies for temporal, geographical and sociocultural distances in agile GSD: A Case Study. *IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP).* IEEE.

Uludağ Ö, Kleehaus M, Dreymann N, Kabelin C, Matthes F, 2019. Investigating the adoption and application of large-scale scrum at a German automobile manufacturer. *ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE).* IEEE, (pp. 22-29).

Vallon R, Estácio B.J.d.S, Prikladnicki R, Grechenig T, 2018. Systematic literature review on agile practices in global software development. *Information and Software Technology 96,* (pp. 161-180).

Vithana, V.N, Asirvatham D, Johar M.G.M, 2018. An Empirical Study on Using Agile Methods in Global Software Development. *18th International Conference on Advances in ICT for Emerging Regions (ICTer),* IEEE, (pp. 150-156).

Werewka J, Wietecha M, Wolczyk K, 2017. Distinguishing and defining the role of a technical leader in outsourced teams developing IT solutions using Scrum. *FedCSIS Communication Papers.*

AJNTS No 57 / 2023 (XXVIII)

INCREASING THE NUMBER MINUTIAE EXTRACTED IN FINGERPRINT IMAGES BY IMPLEMENTING BIORTHOGONAL WAVELET FUSION BASED ON REGIONAL VARIANCE

Indrit ENESI, Alban RAKIPI and Desar SHAHU
Faculty of Information Technology, Polytechnic University of Tirana, Albania

---

ABSTRACT

Biometric identification systems usually use fingerprint images. The higher quality of fingerprint image, the more accurate identification of persons is. Finger prints provide complementary and redundant information Usually, two or more fingerprint images are captured from the person in different positions, and the fusion process of images usually increase the information quantity of the fused image compared to the original ones. Different fusion approaches exist. The present paper aims to increase the number of fingerprint minutiae by implementing biorthogonal wavelet fusion of fingerprint images based on the regional variance. The obtained results are analyzed in terms of minutiae extracted from the original and the fused fingerprint images. Simulations show that the number of minutiae detected in the fused fingerprint image is increased.

***Keywords*:** biometric, fingerprint, regional fusion, biorthogonal wavelet transform, minutiae

1. INTRODUCTION

Fingerprint recognition is used to **identify or confirm the identity of an individual based on the comparison of two fingerprints**. Its quality has a high importance in accurate extraction of minutiae requires images of high quality. Different fingerprint images obtained in various positions may have complementary information. These images taken from the same source are first aligned accorded to a base one (Xi and Geng, 2018), and then fusion techniques (Kaur *et al.*, 2021) are applied to increase information in it. Wavelet technology is widely used for image fusion (Patil, 2016; Kaur *et al.*,

2021) by employing the pixel-level method which separates connection between pixels and region-based method where different image regions are extracted and fused based on regional characteristics (Bikash *et al.*, 2019). Current studies in this field show that image fusion based on regional characteristics performs more fusion effects than pixel–level fusion without regional division (Rane *et al.*, 2017).

The minutiae are the points the fingerprint recognition system uses for comparison or identification as they are found on the surface tips of a human finger. The number of images is important for an accurate identification process. The present paper aims to fuse these fingerprint images left occasionally by an individual to increase the number of the minutiae. Usually, the image fusion is performed in the wavelet domain. Different approaches for image fusion exist as well as different wavelets used in these approaches. The approach of the chosen wavelet fusion is based on regional variance. In the present paper biorthogonal wavelets are used for image purposes. Every fused image undergoes to minutiae extraction. For every wavelet used in the fusion process, terminations and bifurcations are counted on fused fingerprint image and a comparison is done. The target is to increase number of minutiae.

Section two gives literature review, section three describes image fusion, section four briefly describes wavelets transform, section five informs about the proposed algorithm, experimental results are obtained and analyzed in section six. Finally, conclusions are done and future work is reported.

2. LITERATURE REVIEW

Biometric identification of individuals is widely used nowadays, and fingerprint-based identification is the most commonly used. Fingerprint recognition has been a proven method for personal identification, and the higher the image quality is, the better individual identification is. However, Hara (2009) says that there is still a lot of work to be done for a better quality of the fingerprint image. Fusion approach is widely used in image processing in different areas as computer vision, images taken from satellite, robot vision, in medical images, in vehicle guidance etc. (Kaur *et al.*, 2021) analyze different methods of implementing fusion techniques in spatial and transformed domains by presenting their positive and negative sides. Although there is a lot of work in the field of fusion techniques for image processing, there is relatively little work on fusion in fingerprint images.

Maltoni (2009) treat the increase in fingerprint image quality when biometric information is obtained from various sources.

Singh *et al.*, (2019) reviewed biometric fusion focusing on three questions: what to fuse, when to fuse, and how to fuse.

Leghari *et al*., (2021) a CNN based model for fusion of feature level of fingerprint image is proposed and implemented by implementing two schemes. The first scheme combines the features of fingerprint images and online signatures before the fully connected layers while the second scheme combines the features after the fully connected layers, comparing the obtained accuracy of each schema.

Gafurov *et al*., (2011) analyzed fingerprint fusion based on these three scenarios: a) two fingers captured by the same scanner; b) the same finger captured by two different scanners; and c) two fingers both captured by two different scanners. The authors concluded that fusion of different fingers both collected by different scanners are the best.

Alajlan *et al*., (2013) proposed a fuzzy adaptive genetic algorithm for the improvement of authentication performance of this multimodal biometrics computing the optimal weights required for fusion of matching scores from two modalities.

3. IMAGE FUSION

Images of the same scene obtained from single or heterogeneous sensors are processed using image fusion theory to increase information based on maximum combination of the obtained information yielding a complete and accurate description of the minutiae existing in the fingerprint images. The fused algorithm processes relevant information from two or more images usually taken in different positions into a single one, which contains most of the information from the source images (Gong *et al.*, 2020). Figure 1 depicts a general schema of image fusion.

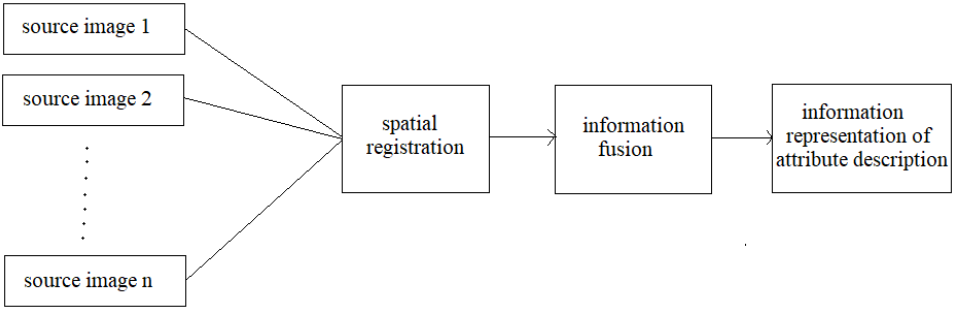


Fig. 11: Image fusion schema.

Information provided by multiple images modes is necessary to fuse the effective information in order to match the geometric positions of images analyzed in spatial domain (Gong *et al.*, 2020; Enesi *et al.*, 2021).

4. WAVELET DECOMPOSITION

Wavelets are oscillations which start at 0 up to some amplitude valueS, decreasing back to 0 as illustrated in Figure 2. Each scale component has a frequency range and carries a resolution matching its scale. Wavelets are convolved with the signal for analyzing it in time and frequency domain. A mother wavelet, shifted with the signal ranging from 0 to T, is multiplied with some portion of the signal. The result is integrated to obtain the wavelet coefficients (Bhataria and Shah 2018) as shown in Figure 3.

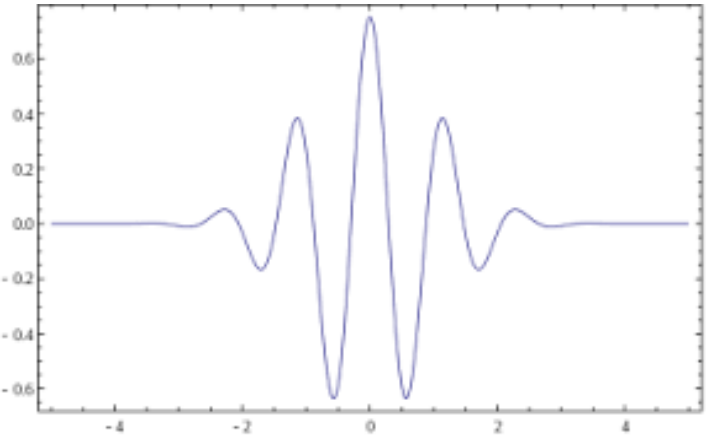


Fig. 12: Mother wavelet.

Wavelet transformation is a highly used for image processing purposes.

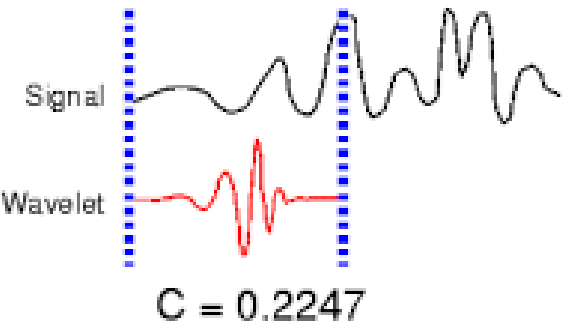


Fig. 13: Obtaining wavelet coefficients.

A mother wavelet has average 0. Usually, it is scaled by a factor 'a' and translated by a factor 'b', the general formula is presented in equation 1.

$$\psi_{a,T}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-T}{a}\right) \quad (1)$$

The wavelet transform of the signal y(t), denoted by $W_x(a, T)$ is presented in equation 2.

$$W_x(a,T) = \int_{-\infty}^{+\infty} y(t)\frac{1}{\sqrt{a}}\psi^*\left(\frac{t-T}{a}\right)dt \quad (2)$$

To design orthogonal wavelet basis, the conjugate mirror filter H(ω) is used and the relation is shown in equation 3.

$$\left|H_\phi(\omega)\right|^2 + \left|H_\phi(\omega+\pi)\right|^2 = 2 \quad (3)$$

where $H_\phi(0) = \sqrt{2}$

A conjugate mirror filter is used to construct orthogonal wavelet basis (Shaker, 2020). Scaling filter is used to determine the scaling function.

Regional image fusion based on wavelet transform

Regional feature refers to local or geometric features in an image (Gostashby and Nikolov, 2007). Correlation between neighborhoods pixels is considered for the regional fusion approach to evident the characteristics of the region and reducing the noise (Pajares and Cruz, 2004). The fusion process takes the appropriate approach for the two or more images to match as much it as can with each other to obtain local representation according to the characteristics of the image. Joint area representation is performed based on the two images' regional representations. Respective regions of the source images, image union region and rules fusion determine the key aspects of fusion schema. Figure 4 depicts fusion of target and background region based on image segmentation.

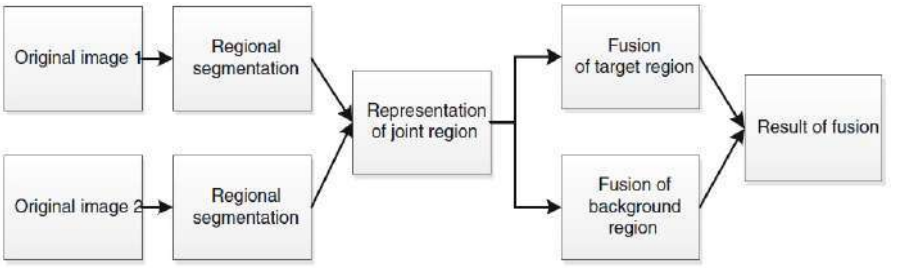


Fig. 14: Image fusion based on regional segmentation.

The block diagram of image fusion based on wavelet transform is in Figure 5 depicted.

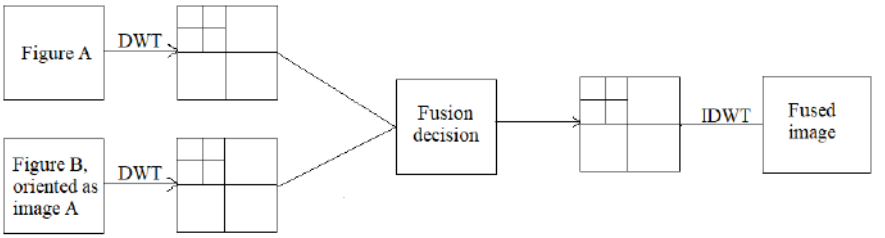


Fig. 15: Block diagram of image fusion based on DWT.

The wavelet transform of each image is performed to obtain low/high frequency components of the image. Decompositions layers are fused according to respective fusion algorithms and the fused wavelet pyramid is obtained. The reconstruction from the wavelet pyramid coefficients is the fused image (Gong *et al.*, 2020, Goshtasby and Nikolov, 2007).

The fusion process consists of: i) wavelet transform of the image in low/high frequency sub bands, ii) low frequency coefficients are obtained by applying the averaging method, iii) the variance of high frequency components is found applying a 3 x 3 sliding window and highest values are chosen as high frequency coefficients for image reconstruction, iv) IDWT is applied for fused image reconstruction (Gong *et al.*, 2020)(Naji, 2020).

5. PROPOSED IMAGE FUSION ALGORITHM

Fingerprint images which are to be synthesized are transformed in spatial domain first to match the geometric position as much as can (Agarwal and Bedi 2015).

STEP 1: Orientation and scaling of the second image according to the first one
Detect Features in both images
Extract Features descriptors from every image
Match Features descriptors
Select corresponding points in every image
Putative point matches are located
Transform second one according the first one

Algorithm 1 Detailed pseudo-code for aligning the images

Due to limited accuracy of the spatial transformation process, artifacts are found around the edges of the transformed image.

DWT is applied on both images to obtain low and high frequency coefficients (Amolins *et al.*, 2007).

STEP 2: Image fusion based on biorthogonal DWT
Input: Both fingerprint images are obtained
Decompose first image using DWT (level 1, different wavelets); vector C0 and matrix S0 are obtained
The other image is decomposed using multi-scale dyadic DWT (level 3, mother wavelet is biorthogonal 6.8); vector C1 and matrix S1 are obtained
Wavelet coefficients (c1 or c2) are obtained in the format:
C(1) = A(1) H(1) V(1) D(1)
C(2) = A(2) H(2) V(2) D(2)
A 3 x 3 sliding window filter is applied and new coefficients are obtained
Averaging of A(1) and A(2) values of each image yields in Low pass coefficients
Maximum value of respective H, V and D of each image yields in High pass coefficients IDWT is applied on the obtained coefficients

Algorithm 2 Detailed pseudo-code for image fusion

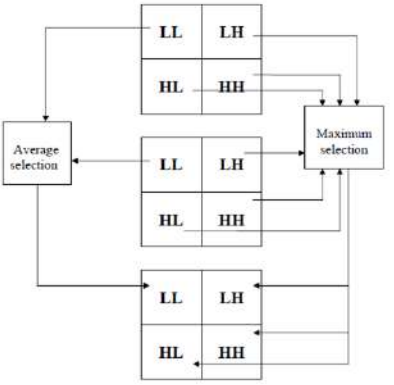


Fig. 16: Averaging and Maximum selections respectively on Low and High pass coefficients are used to obtain new coefficients.

Each decomposition layer is fused according to appropriate fusion algorithms, based on regional variance features of low/high frequency components. Fusion schema is in Figure 6 depicted. Different wavelets are implemented and the pyramid of fused wavelet is obtained for each one. IDWT is applied, and fused image are obtained. To calculate the number of terminations and bifurcations, an extraction minutiae algorithm is performed (Jiang *et al.*, 2015).

6. EXPERIMENTAL RESULTS

CASIA Fingerprint Image Database version 5.0 is a public database used for fingerprint images. Images are captured using URU4000 fingerprint sensor. Volunteers are asked to rotate the fingers with various levels of pressure. All fingerprint images are 8-bit gray level bmp files with resolution 328x356.

The three digits at the name of the image file compound the unique identifier of the person, L and R denotes the left and right hand, 0 stands for the thumb, 1 for the second finger, 2 for the third finger and 3 stands for the fourth finger. Fingerprint images are with damages and noises. The images considered in this paper are: 200_L0_1 and 200_L0_3, 220_L2_0 and 220_L2_1, 235_R2_1 and 235_R2_4, 257_R3_1 and 257_R3_4.

The second image is spatially transformed according to the first one based on the putative points (Figure 7).

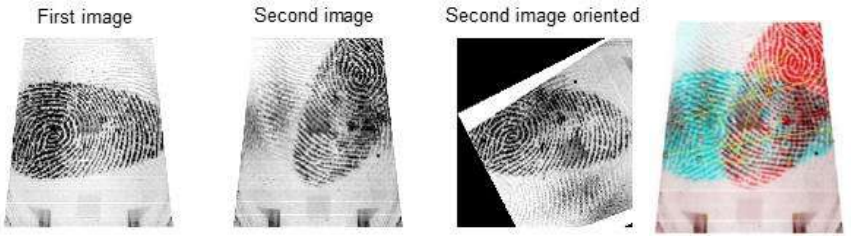


Fig. 17: Spatial transformation of 200_L0_3.bmp according to the 200_L0_1.bmp and putatively matched points

Wavelet transformation is used for fusion based on regional variance. The fused one for images 200_L0_1 and 200_L0_3 is in Figure 8 depicted.

Fig. 18: Fused image for 200_L0_1.bmp and the spatially transformed 200_L1_3.bmp.

Regarding the (Jiang *et al.*, 2015, Shaker, 2020) minutiae extraction is applied in fused fingerprint image, the number of terminations and bifurcations is counted. Figure 9 depicts minutiae extracted for fingerprint 200_L0_1.bmp and its fused version with wavelet bior2.4. To remove false minutiae, the considered Euclidian distance value is not less than 6.

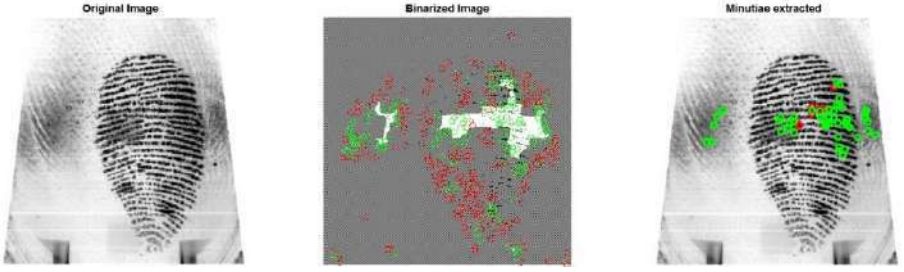


Fig. 19: Minutiae extracted for image 200_L0_0.bmp.

There are many for minutiae extraction, but Papillon 9.02 is the official version used by the Albanian police department for biometric person identification. Although it is fast, the manual technique for minutiae extraction used by a fingerprint expert is the most accurate one. Minutiae extraction is obtained automatically by Papillon 9.02 and manually by police expert, and the comparison between the two methods is in Figure 10 depicted.

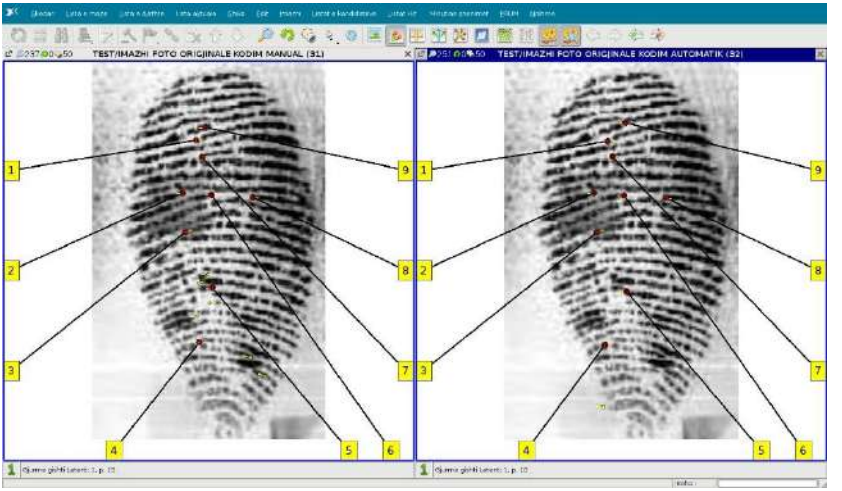


Fig. 20: Minutiae correspondences between the two methods.

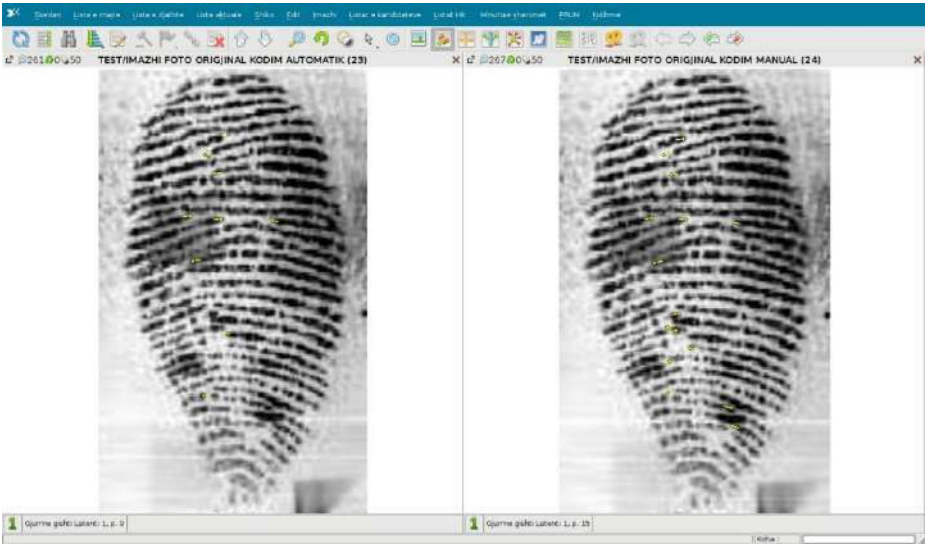


Fig. 21: Minutiae detected in original image by two technics.

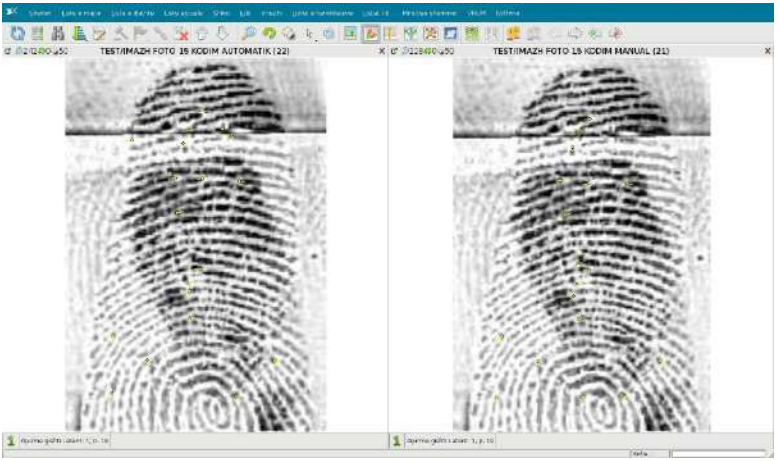


Fig. 22: Minutiae detected in the fused image with bior6.8.

Different wavelet can be used for image fusion. Biorthogonal wavelet transform has advantage comparing with other wavelet transformation regarding the reconstruction. In this paper is considered bior6.8 wavelet for image fusion. The automatic and manual minutiae extraction is performed in the original and the fused image, results are illustrated in figures 11 and 12.

Tab. 1 Number of terminations and bifurcations detected

	Automatic detection Papillon 9.2			Manual detection		
Image	Minutiae	Terminations	Bifurcations	Minutiae	Terminations	Bifurcations
First Image	9	6	3	15	15	0
Second Image	7	6	1	12	10	2
The fused one	19	10	9	19	15	4

Table 1 reports that the number of minutiae in the fused image is increased when compared to both original images.

The proposed algorithm will be tested on images taken randomly from the dataset CASIA-Fingerprint Image Database V5 (200 – 299): 220_L2_0 and 220_L2_1, 235_R2_1 and 235_R2_4, 257_R3_1 and 257_R3_4. Fingerprint images are randomly selected from left and right hands, in different positions relative to each other and with damages.

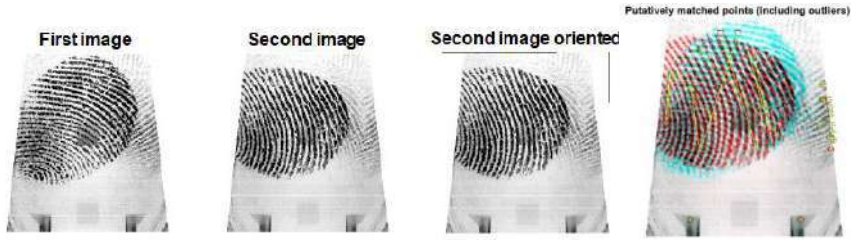


Fig. 13: Fingerprint images 220_L2_0 and 220_L2_1 and the oriented one

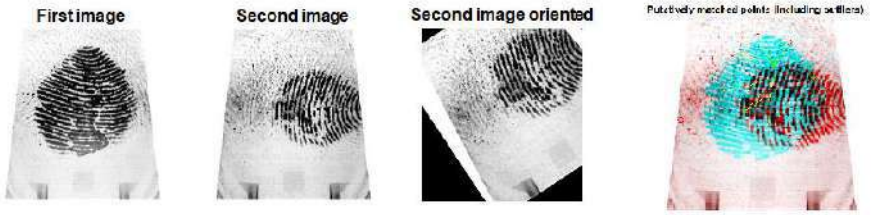


Fig. 14: Fingerprint images 235_R2_1 and 235_R2_4 and the oriented one

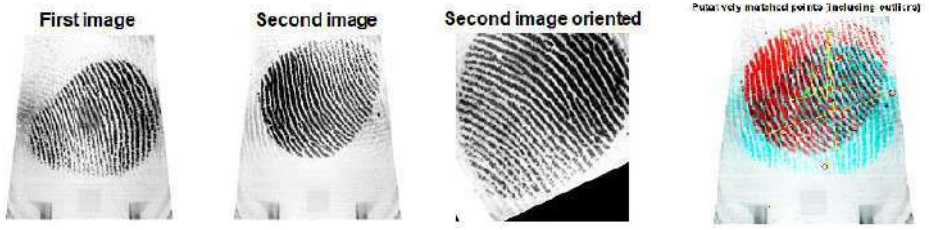


Fig. 15: Fingerprint images 257_R3_1 and 235_R3_4 and the oriented one

Fig. 16: Fingerprint images 220_L2_0 and 220_L2_1 and the fused one

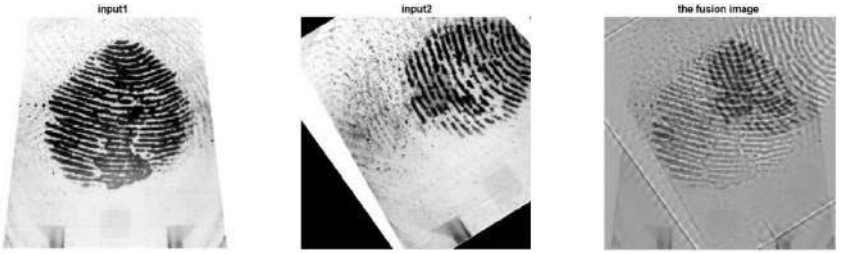


Fig. 17: Fingerprint images 235_R2_1 and 235_R2_4 and the fused one

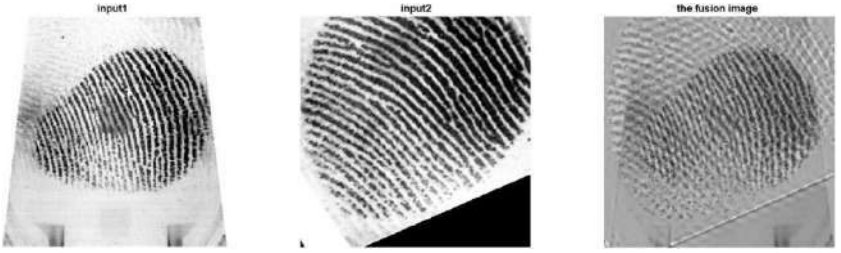


Fig. 18: Fingerprint images 257_R3_1 and 257_R3_4 and the fused one

In the following, minutiae must be extracted from fingerprint images, specifically from both fingerprints and the fused one. Then a comparison is performed. Papillon 9.02 software will be used to extract the minutiae.

Table 1. Number of terminations and bifurcations detected

	Automatic detection Papillon 9.2			Manual detection		
Image	Minutiae	Terminations	Bifurcations	Minutiae	Terminations	Bifurcations
220_L2_0.bmp	26	11	15	34	16	18
220_L2_1.bmp	29	10	19	35	13	22
220_L2_01_fused_bior6.8	37	12	25	50	23	27
235_R2_1.bmp	41	17	24	44	20	24
235_R2_4.bmp	6	5	1	10	7	3
235_R2_14_fused_bior6.8	44	19	25	56	26	30
257_R3_1	13	11	2	17	11	6
257_R3_4	16	0	16	19	5	14
257_R3_14_fused_bior6.8	28	12	16	29	15	14

7. CONCLUSIONS

The paper proposes and implements an image fusion transform based on regional variance to improve the accuracy of fingerprint recognition. Fingerprint images are spatially aligned according to each other and fused by

implementing average and maximum respectively for low and high frequency coefficients using different bior6.8 wavelet. Experimental results implemented on various fingerprint images from CASIA v5 database show that the number of minutiae detected is increased.

8. FUTURE WORKS

To give an accurate result of the proposed method more tests will be implemented on public FVC2004 and FVC2006 database of fingerprints, as well as more wavelets will clarify this result.

REFERENCES

Agarwal J, Bedi S. 2015. Implementation of hybrid image fusion technique for feature enhancement in medical diagnosis. *Human-centric Computing and Information Sciences*, **5(1)**: DOI:10.1186/s13673-014-0020-z.

Alajlan N, Islam MS, Ammour N. 2013. Fusion of fingerprint and heartbeat biometrics using fuzzy adaptive genetic algorithm. World congress on internet security (WorldCIS-2013). 76-81, doi: 10.1109/WorldCIS.2013.6751021.

Amolins K, Zhang Y, Dare P. 2007. Wavelet based image fusion techniques-A review and introduction. *ISPRS Journal of Photogrammetry & Remote Sensing*, **62:** 249–263**.**

Bhataria K. C, Shah B. K. 2018. A Review of image fusion techniques. Second international conference on Computing Methodologies and Communication (ICCMC). 114-123, doi: 10.1109/ICCMC.2018.8487686.

Bikash M, Sanjay A, Rutuparna P, Ajith A. 2019. A survey on region-based image fusion methods. *Information Fusion*, **48**: 119-132, ISSN 1566-2535.

Enesi I, Cico B, Harizaj M. 2021. Increasing quality in fingerprint images by Implementing wavelet transform based fusion technique. International conference on computer systems and technologies '21.

Gafurov D, Busch C, Bours P, Yang B. 2011. Fusion in fingerprint authentication: Two finger types vs. two scanner types. Conference: Proceedings of the 2011 ACM Symposium on Applied Computing (SAC), TaiChung, Taiwan.

Gong Sh, Liu C, Ji Y, Zhong B, Li Y, Dong H. 2020. Advanced image and video processing using Matlab. Springer.

Goshtasby AA, Nikolov S. 2007. Image fusion: advances in the state of the art. *Information Fusion*, 8(2): pp. 114–118.

Hara M. 2009. Fingerprint image enhancement. In: Li S.Z., Jain A. (eds) Encyclopedia of Biometrics. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-73003-5_49.

Jiang B, Zhang R, Zhang X. 2015. A Comparative Study of Wavelet-Based Image Fusion with a Novel Fusion Rule. *International Journal of Machine Learning and Computing*, **5(6):** 484-492. 10.18178/ijmlc.2015.5.6.558.

Jing D, Byron B, Frank D, Ranveer Ch, Sudipta NS. 2018. Learning to align images using weak geometric supervision. Computer Vision and Pattern Recognition.

Kaur H, Koundal D, Kadyan, V. 2021. Image Fusion Techniques: A Survey. *Archives of Computational Methods in Engineering* 28 (2021): 4425 - 4447.

Leghari M, Memon S, Dhomeja L, Jalbani A, Chandio A. 2021. Deep feature fusion of fingerprint and online signature for multimodal biometrics. computers. 10, 21. https://doi.org/10.3390/computers10020021.

Maltoni D. 2009. Biometric fusion. In: Handbook of fingerprint recognition. Springer, London. https://doi.org/10.1007/978-1-84882-254-2_7.

Naji ShA. 2020. Comparison between orthogonal and bi-orthogonal wavelets. *Journal of southwest Jiaotong university*, **55(2):** ISSN: 0258-2724.

Pajares G, Cruz JM. 2004. A wavelet-based image fusion tutorial. *Pattern Recognition*, **37(9):** 1855–1872.

Patil Sh M. 2016. Image Fusion using Wavelet Transform. *International Journal of Engineering and Advanced Technology* (IJEAT), **5(4)**: 66-69. ISSN: 2249 – 8958.

Rane ND, Kakde B, Jain M. 2017. Comparative study of image fusion methods: A Review. *International Journal of Engineering and Applied Sciences* (IJEAS), **4(10):** 67-72. ISSN: 2394-3661.

Singh M, Singh R, Ross A. 2019. A comprehensive overview of biometric fusion. *Information Fusion*, 52: 187-205, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2018.12.003.

Xi, S., Geng, W. 2018. Fast Algorithm of Fingerprint Singularity Region Enhancement. *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 400-404.

CONTINUOUS INTEGRATION / CONTINOUS DELIVERY: REVIEW OF CHALLENGES AND SOLUTIONS

Ina PAPADHOPULLI and Realb KUSHE
Faculty of Information Technology, Polytechnic University of Tirana, Albania

---

ABSTRACT

There is an increasing interest in literature on continuous practices (i.e., continuous integration, delivery, and deployment). For this reason, it is important to systematically review the approaches, tools and challenges related to these practices. In order to offer a "big view" of the continuous software engineering, this systematic literature review lists the strategies proposed to address and solve these challenges. We analyzed 46 relevant papers, filtered from four digital libraries. The implementation of Continuous Software Engineering practices is associated with challenges regarding the software builds, unit tests, integration tests and non-functional ones. Several strategies have been proposed by academics to address these challenges. In addition, tools have been developed to automate each stage of CI/CD pipeline. The progress in the optimization of continuous software engineering practices is inspired by the industrial needs. This Systematic Literature Review (SLR) emphasizes that as in many other engineering problems, there is no optimal continuous software engineering architecture to fulfil all the clients' needs. Selection of the right automation tool for continuous software development project depends on the sort of project.

***Keywords*:** continuous software engineering, continuous integration/continuous deployment, continuous software testing

1. INTRODUCTION

The Agile methodology defines the processes used to change software features and accelerate delivery. "DevOps culture" specifies roles and responsibilities of each human actor in a software development project; software developer or IT specialist, to increase their responsiveness. Continuous software engineering focuses on tools used for automation of software defined life-cycles (Doukoure and Mnkandla 2018). Thanks to these

practices, the gap between software development teams and operational ones has been steadily narrowed during the past years (Felidré *et al.*, 2019). The three development activities are continuous integration (CI), continuous delivery (CDE) and continuous deployment (CD). CDE means that the software **can** be deployed to production at any time, whereas CD means that the software is **automatically** deployed to production all the time. The relationship between these concepts is in Figure 1 depicted (Shahin *et al.*, 2017).

CI/CD has the following benefits: i) software of higher quality, ii) faster delivery of features to the customer, iii) easy to be used, simple and flexible to the needs of customers due to the ability to change things more quickly. Javed *et al.*, (2020) recommended to follow various principles and best practices to achieve these benefits:

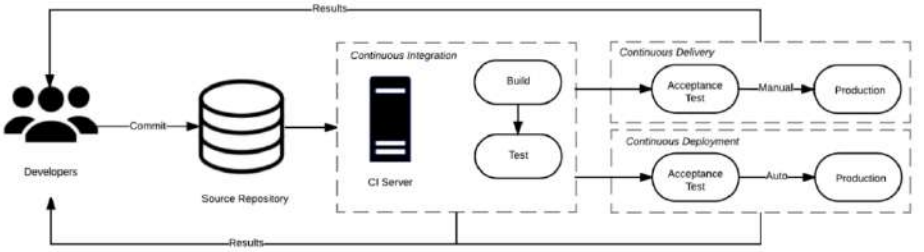


Fig. 23: Relationship between CI, CDE and CD (Shahin *et al*., 2017).

More and more strategies (mainly by academic researchers) and tools (mainly by industrial researchers) are being developed in support of each stage of CI/CD pipeline. There are difficulties to transition to CI/CD. Even when the team has successfully introduced the CI/CD culture, living up to its principles and improving the CI/CD practice is also challenging (Ciancarini and Missiroli 2020)

This SLR aims to provide a comprehensive analysis of CSE by balancing the benefits of its applications with their related limitations. Information about the latest research with regard could be found in (Shahin *et al*., 2017).

Research Questions (RQ): The study addresses three research questions:

RQ1: What are the limitations in the implementation of CI/CD stages?

RQ2: What strategies have been proposed to address CI/CD challenges?

RQ3: Which are the state-of-the-art tools for designing and implementing the deployment pipeline?

In summary, this paper makes the following contributions:

1. An analysis of the problems that arise during the implementation of CI/CD pipeline grouped by CI/CD stage
2. A list of solutions proposed in the last five years for the problems addressed in RQ1.
3. A comparison of the CI/CD state-of-the-art tools

Section 2 presents the used Review Protocol. Sections 3 - 5 addressed the three research questions. Conclusions are drawn in the end.

2 REVIEW PROTOCOL

2.1 Study Selection Process:

Metadata filtering: Research papers found in digital libraries were filtered based on metadata indicators: title of the paper ("Is this related to CSE?"), author names, date published with respect to inclusion and exclusion criteria as above defined.

Abstract filtering: Once the abstract was read, papers on DevOps culture or theoretical part of Agile methodologies were excluded from further reading.

Content-based filtering: Only papers answering to the research questions were taken into consideration for this SLR.

In the present investigation, 46 relevant papers were systematically identified and rigorously reviewed. In addition, synthetization of the data extracted from these papers to answer the research questions was made.

Table 8 The evaluation of research papers during study selection process.

Search Engine	**First-time collected (metadata filtering)**	**Included after filter application related to:**			**Total**
		Abstract	Content	Repetition of ideas	
IEEE Xplore	*49*	*43*	*40*	*29*	***29***
ACM	*23*	*21*	*16*	*9*	***9***
Science Direct	*8*	*6*	*5*	*5*	***5***
Scopus	*5*	*4*	*3*	*3*	***3***
Total	***85***	***74***	***64***	***46***	***46***

3. RQ1: WHAT ARE THE LIMITATIONS IN THE IMPLEMENTATION OF CI/CD STAGES?

The challenges are divided into groups in according to the correspondent CI/CD stage.

3.1 **Problems related to Continuous Integration**

Lack of frequent commits: (Pinto *et al.* 2018) emphasized that the most common CI problem reported by their survey group was infrequent commits due to time pressure. Felidré *et al.*, (2019) said that "2.36 commits/ weekday" is the lowest threshold value for a software development project to succeed, independently of the project size as based on (Cavalcanti *et al*., 2018).

Time-consuming builds: For a large-scale software project, the build can take hours as it includes compilation, unit and acceptance testing (Jin and Servant 2020). Continuous submission of code modification by developers and build latency time creates stalls at CI server build pipeline, and hence developers have to wait long time for the build outcome (Fan, 2019). These builds compete for system resources with other jobs waiting in the processing queue (Bezemer 2017).

Broken builds: Builds can be unsuccessful for a variety of reasons (Rebouc *et al*., 2017) points out the gap in the real-time addressing of problematical builds between commercial projects and open-source ones. Commercial projects tend to enter in a "fast-recovery" mode while open-source ones seem to offer a slower but more consolidated solution for the build failure (Avelino *et al*., 2016).

3.2 Problems related to "Continuous Testing" (CT)

3.2.1 Unit tests

Writing automated tests is time-consuming: DiffBlue survey (Zalozhnev, 2017) found that software engineers, among the most expensive talent in any company, spend 20% of their time writing unit tests and an additional 15% of their time writing all other types of tests.

Manual testing: DiffBlue survey (Camargo *et al*., 2016) found manual testing as a key bottleneck in a CI/CD pipeline. The reason is that resources are not invested in automated testing. When asked which stage of the DevOps pipeline respondents feel their organization places as its top priority, 51% chose developing, with deploying in second place (24%). Testing fell in last place (11%).

Non-deterministic automated tests and code: A considerable amount of disturbance to CI/CD pipeline is caused by non-deterministic (flaky) automated tests. These tests capriciously generate variable results even without changing the isolated tested code (Gallaba 2019). Maintaining flaky tests is costly, especially in large-scale software projects (Diffblue 2021b).

Poor test quality: Unreliable tests, high number of test cases, low test coverage and long running tests can impede the deployment pipeline and reduce the confidence of organizations to automatically deploy software on a

continuous basis (Shahin *et al*., 2017). Test-coverage is mainly limited by the use of an old-fashion testing metric: line coverage (Kim *et al*., 2017).

3.2.2 Integration Tests

System heterogeneity: The complexity of CT stands in its heterogeneity: distributed testing requires the participation of a lot of hardware resources shared between multiple platforms. Test results can be prone to errors while traversing the communication links especially in the master-slave architecture (Aghamohammadi *et al*., 2021).

Version Control (VI): A critical point for CI is the Version Control of the shared repository. Updates to the Version Control may trigger instability of the system: widespread file-locking and corrupted copies of the program files have been reported by CI processes in diverse software development projects (Xu *et al*., 2019).

3.2.3 Non-functional Testing

Difficulty to automate performance tests: Developing performance testing automation scripts is not a trivial task. Automating this process requires strong tool support. A lack of existing tools means that performance testing is normally left out of the scope of CI (Diffblue 2021a).

Skip security tests: Security tests in the CI stages are extremely important as they guarantee that none of weak dependencies between entities will process in the rest of the deployment pipeline.

3.3 Problems related to "Continuous Deployment"

Fulfill quality assurance step: Quality assurance (QA) is the final step before pushing the software to production. The fulfillment of both development and QA constraints is a difficult task (Parnin *et al*., 2017). The implementation of a unified framework for both teams is associated with additional costs of training project's members for its usage.

Duplicate production environment: Creating a duplicate production environment (shadow infrastructure) in order to enable software experimentation and accelerate software production is costly (Macho 2017).

Different Customer Environment: Shahin *et al*., (2017) said that continuously releasing software product to multiple customers with diverse environments is quite difficult as different deployment configurations for each customer's environment and component's version are needed to be established.

Maintenance window: Service deployment, including upgrading to new versions, rolling back to older ones, or introducing fix patches in case of a failed deployment, can be done during a maintenance window while reusing

the infrastructure resources due to the high cost of hardware and its maintenance (Pinto *et al*., 2018)

Smells in CD configuration files: Typical configuration files for specialized build tools which depend on the programming languages are usually too complex. This is the reason why there may be many smells in CD pipelines like 'Fake Success', 'Retry Failure', 'Manual Execution' and 'Fuzzy Version' (Javed *et al*., 2020).

4. RQ2: WHAT STRATEGIES HAVE BEEN PROPOSED TO ADDRESS CI/CD CHALLENGES?

Several strategies have been proposed for solving the CI/CD challenges. In order to answer to the RQ2, after analyzing the proposed solutions, we have made a mapping between the proposed solution and the challenge it addresses (Table 2).

Table 2. Mapping between challenges found in CI/CD pipeline and their proposed solutions. Problem Category (PC): ① (broken builds); ② (long running builds); ③ (Writing automated Unit tests is time-consuming); ④ (Long running unit tests); ⑤ (non-deterministic automated tests); ⑥ (Version Control); ⑦ (Difficulty to automate performance tests); ⑧ (Duplicate production environment); ⑨ (Maintenance window); ⑩ (Smells in CD configuration files);

Solution	PC	Description of the solution	Ref
"Filter and flush" architecture	①	Records metadata of the previous "failed builds" to reject builds that have crashed in the past.	(Hassan and Wang,2017)
"Taxonomy of broken builds"		Broken builds are classified based on their cause and their impact on the project. Most influential builds are tried to repair with highest priority.	(Konersman n *et al*., 2020)
"Exploration of dependencies between builds"	②	Proposal to use third-party tools to design annotated graphs that study dependencies between builds.	(Fan 2017)
"Benefit from local spatiality"		Builds with similar features are grouped together and a "build agent" examines and extracts their similarities. Builds from different clusters run concurrently.	(Melo and Rocio, 2017)

Build prediction models		Model that uses previous data to predict whether a build will be successful or not without attempting actual build so that developer can get early build outcome result.	(Yang *et al*., 2018) (Fan 2017)
Tool: SmartBuildSkip		Use ML to predict the first builds in a sequence of build failures	(Jin and Servant, 2020)
Skip commits		Automate the process of determining which commits can be CI skipped through the use of ML techniques	(Singh *et al*., 2019)
Tool: Evosuite	③	Search Based Software Testing tool used to automatically generate unit tests for Java applications	(Francalino *et al*., 2018)
Tool: DiffBlue Cover		Tool that uses AI to automatically write suites of unit tests for Java code	(Abdalkareem *et al*., 2021) (Diffblue 2021c)
Testing as a Service"	④	Automated Unit tests are executed parallelly in a distributed cloud infrastructure.	(King *et al*., 2018)
Choose a subset of test to execute		Use ML to predict which group of tests should be executed after each change submitted to the CI system.	(Islam and Zibran, 2017)
Postpone re-execution of flaky test	⑤	Non-deterministic tests are marked with a flag and re-executed after all other tests to identify the cause of ambiguity and dependencies that caused the failure	(Deepa *et al*., 2020) (Javed *et al.*, 2020)
Probabilistic approach to detect faulty tests		Fault localization can be achieved by creating a Bayesian network model that takes into consideration a broad range of tests metric: assertion count, test size, cyclomatic complexity etc.	(Diffblue 2021c)
Maintenance of VI System	⑥	VI updates should be scheduled at "off-hours". A trade-off should be found between the need for on-time VI updates and the need for the release of stable software products.	(Williams, 2021) (Javed *et al.*, 2020)
PerfCI Tool	⑦	PerfCI - helps developers to easily set up and carry out automated performance testing under CI	(Diffblue, 2021c)
Be fast to deploy but slower to release	⑧	Combination of 'dark launches' and 'feature flags'	(Macho, 2017)

Blue-Green Deployment Technique	⑨	Usage of Blue-Green Deployment Technique which targets to enable service updates with zero maintenance windows, and thus with no disruption to the end users	(Pinto *et al.*, 2018)
Tool: CD-Linter	⑩	A semantic linter that can automatically identify four different smells in pipeline configuration files on GitLab.	(Javed *et al.*, 2020)
Tool: Xeditor		Tool that extracts configuration couplings from Deployment Descriptors, and adopts the coupling rules to validate new / updated files	(Wen *et al.*, 2020)

5. Q3: WHICH ARE THE STATE-OF-THE-ART TOOLS FOR DESIGNING AND IMPLEMENTING THE CI/CD PIPELINE?

Choosing the right CI/CD tool is an essential part that can define the success rate of a software development project. In this SLR, we focused on the tools that automate the whole pipeline since they don't require extra synchronization with other tools.

The comparison is made based on two principles:

1. Research papers which describe the features of the CI/CD tools have made a transparent evaluation of the tool's metric.
2. Since there are different methods to evaluate the tools, we developed a numerical scale to compare the set of tools.

Figure 2 shows the **proposed numerical scale** for the evaluation procedure.

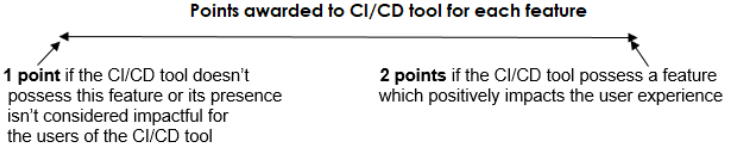


Fig. 2. Scoring points evaluation procedure

Features considered:

- Ease of install, ease of upgrade and backup
- If the CI/CD tool is an open-source tool or not
- If the hosting model includes both "On-premise" and "Cloud"
- If the CI/CD tool provides "test parallelization" in distributed environments
- Graphical pipeline view

- Re-usable pipelines

Once the features information was collected from a set of research papers on behalf of 7 CI/CD tools, the tools were compared by using the proposed numerical scale (Table 3).

This score-based evaluation shows that TravisCI tool is the most maturated tool for the implementation of CI/CD pipeline followed by TeamCity and GitLab.

Table 3. Comparison of CI/CD tool.

	Jenkins	TeamCity	Bamboo	CircleCI	GitLab	TravisCI	GoCD
Ease of install	2	2	2	1	2	2	1
Ease of upgrade	1	2	1	1	1	2	1
Ease of backup	1	2	2	2	2	2	2
License of tool	2	1	1	2	2	2	2
Hosting model	2	1	1	2	2	1	2
Test case parallelization	2	2	2	2	2	2	1
Reusable Pipelines for Microservices	2	2	1	2	1	2	2
Graphical Pipeline View	1	2	2	1	2	2	2
Total Points	**13**	**14**	**12**	**13**	**14**	**15**	**13**

The features were extracted from (Souza and Silvia, 2017; Hohl *et al*., 2018; Diffblue, 2021c).

6. CONCLUSIONS AND FUTURE WORK

CSE practices are being highly adopted by companies, which are transitioning their strategy from developing stand-alone programs to offering software as a service.

This SLR offers a bilateral analysis of both the problems that arise during the implementation of CI/CD pipeline and their counterpart solutions. Each phase of Continuous Development pipeline deals with implementation challenges (Section 3). There exist many proposed solutions to local and isolated problems, but it is hard to implement them in a vector of mixed CI/CD problems. This SLR emphasizes that, as in many other engineering problems, there is no optimal CSE architecture to fulfill all client's needs. Selection of the right automation tool is based in the nature of project.

In the present paper the tools that automate the whole CI/CD pipeline are compared, and our score-based evaluation shows that TravisCI is the most maturated tool.

Based on the number of papers on continuous practices that we found, we can conclude that in the last five years there has been a high interest in this field from academic and industrial researchers.

Several papers (i.e., 10 papers) proposed solutions based on AI techniques as an alternative to deterministic approaches to solve difficult problems related to CI/CD. It seems that the research is focused more on this direction.

Research papers reviewed in this systematic literature review were extensively focused on improving the CSE pipeline. However, we found that there is little work done for the investigation of how do bad DevOps practices actually interfere with Continuous Software Engineering ones.

REFERENCES

Abdalkareem R, Mujahid S, Shihab E. 2021. A Machine Learning Approach to Improve the Detection of CI Skip Commits. International Conference on Software Engineering.

Aghamohammadi A, Mirian-Hosseinabadi SH, Jalali S. 2021. Statement frequency coverage: A code coverage criterion for assessing test suite effectiveness. Information and Software Technology.

Avelino G, Passos P, Hora A, Valente MT. 2016. A novel approach for estimating truck factors. 24th IEEE International Conference on Program Comprehension, USA, 2016.

Bezemer C.-P, McIntosh S, Adams B, German D. M, Hassan. A. E. 2017. An empirical study of unspecified dependencies in make-based build systems. *Empirical Software Engineering* (EMSE), **22(6):** 317–324, 217.

Camargo A, Salvadori I, Mello S, Siqueira F. 2016. An architecture to automate performance tests on microservices. In Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services (iiWAS '16). USA.

Ciancarini P, Missiroli M. 2020. The Essence of Game Development. IEEE 32nd Conference on Software Engineering Education and Training, Germany, pp.11-14.

Deepa N, Prabadevi B, Krithika LB, Deepa B. 2020. An analysis on Version Control Systems. International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, Spain, pp. 5-9.

Diffblue. https://www.diffblue.com/ [Last accessed: 10 June 2021].

Diffblue.https://www.diffblue.com/DevOps/research_papers/2020-devops-and-testing-report/[Last accessed: 1 September 2021].

Diffblue.https://www.diffblue.com/Education/research_papers/2019-diffblue-developer-survey/[Last accessed: 10 June 2021].

Doukoure GAK, Mnkandla E. 2018. Facilitating management of agile and DevOps activities: Implementation of a data consolidator. International

Conference on Advances in Big Data, Computing and Data Communication Systems, United Kingdom.

Fan Z. 2019. A systematic evaluation of problematic tests generated by EvoSuite. In Proceedings of the 41st International Conference on Software Engineering.

Felidré W, Furtado LB, da Costa DA, Cartaxo B, Pinto G. 2019. Continuous integration theater. *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*: 1-10. Porto de Galinhas, Recife, Brazil.

Francalino W, Callado A, Matthews Jucá P. 2018. Defining and implementing a test automation strategy in an IT Company. In Proceedings of the Euro- American Conference on Telematics and Information Systems (EATIS'18).

Gallaba K. 2019. Improving the robustness and efficiency of continuous integration and deployment. International Conference on Software Maintenance and Evolution.

Hassan F, Wang X. 2017. Change-aware build prediction model for stall avoidance in continuous integration. In 2017 ACM/IEEE (ESEM), pages 157–162. IEEE.

Hohl P, Stupperich M, Munch J, Schneider K. 2018. Combining agile development and software product lines in automotive: Challenges and recommendations. IEEE (ICE/ITMC).

Islam MR, Zibran MF.2017. Insights into continuous integration build failures. IEEE/ACM 14th International Conference on Mining Software Repositories (MSR).

Javed O, Dawes JH, Han M, Franzoni G, Pfeiffer A, Reger G, Binder W. 2020. PerfCi: A toolchain for automated performance testing during continuous integration of Python projects (ASE).

Jin X, Servant F. 2020. A Cost-efficient Approach to Building in Continuous Integration. In Proceedings of the 43rd International Conference on Software Engineering, 2020.

Kim J, Jeong H, Lee E. 2017. Failure history data-based test case prioritization for effective regression test. Proceedings of the symposium on applied computing.

King TM, Santiago D, Phillips J, Clarke PJ. 2018. Towards a Bayesian network model for predicting cases of flaky automated tests. IEEE International conference on software quality, reliability and security companion.

Konersmann M, Fitzgerald B, Goedicke M, Olsson H, Bosch J, Krusche S. 2020. Rapid continuous software engineering - State of the practice and open research questions: SIGSOFT.

Macho C. 2017. Preventing and repairing build breakage. IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), pp. 471-475.

Melo S, Rocio S. 2017. How to test your concurrent software: an approach for the selection of testing techniques. In Proceedings of the 4th ACM SIGPLAN

International Workshop on Software Engineering for Parallel Systems (SEPS), New York USA, pp.42–43.

Parnin C, Helms E, Atlee C, Boughton H, Ghattas M, Glover A, Williams L. 2017. The top 10 adages in continuous deployment. *IEEE Software*, **34(3):** 86–95.

Pinto G, Castor F, Bonifacio R, Rebouc M. 2018. Work practices and challenges in continuous integration: A survey with travis CI users. Softw., Pract.Exper.

Rebouc M, Santos R, Pinto G, Castor F. 2017. How does contributors' involvement influence the build status of an open-source software project? 14th International Conference on Mining Software Repositories, pages 475–478, USA.

Shahin M, Ali Babar M, Zhu L. 2017. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices, April.

Singh C, Gaba S, Kaur M, Kaur B. 2019. Comparison of Different CI/cd tools integrated with cloud platform. *9th International Conference on Cloud Computing*, pp. 7-12.

Souza R, Silva B. 2017. Sentiment Analysis of Travis CI Builds. *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)* pp. 459-462.

Wen Ch, he X, Zhang Y, Meng N. 2020. Inferring and applying Def-use like configuration couplings in deployment descriptors. *35th IEEE/ACM International Conference on Automated Software Engineering (ASE):* 672-683.

Williams N. 202). Towards exhaustive branch coverage with PathCrawler. IEEE/ACM international workshop on automation of software test.

Xu, X, Cai, Q, Lin J, Pan S, Ren L. 2019. Enforcing access control in distributed version control systems, IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, pp. 772-777.

Yang B, Saller A, Jain S, Tomala-Reyes E, Singh M, Ramnath A. 2018. Service Discovery based Blue-Green Deployment Technique in Cloud Native Environments. *IEEE International Conference on Services Computing (SCC)*, San Francisco, CA, USA, *pp.* 108–121.

Zalozhnev AY. 2017. Big banks systems management software: Architecture. General requirements and functional components. 10th International Conference in Management of Large-Scale System Development, Moscow. 103-108.

SENTIMENT ANALYSIS USING DEEP LEARNING IN ALBANIAN: AN OVERVIEW

Muhamet KASTRATI and Marenglen BIBA
Department of Computer Science, University of New York Tirana, Albania

---

ABSTRACT

Sentiment analysis aims to automatically classify people's opinions, attitudes and emotions expressed towards a particular entity, topic, or event. Here, deep learning has been widely used, and proved useful to sentiment analysis problems. Efforts are mostly devoted to the rich-resource languages; there is only little research on sentiment analysis for the low-resource languages. Therefore, the present paper aims to provide an overview of prominent deep learning models successfully applied on sentiment analysis for low-resource languages such as Albanian language. It begins with a short introduction, motivation and some basic concepts on sentiment analysis, and most prominent deep learning models applied for sentiment analysis. Papers providing information about deep learning approaches and some of the current trends in the field are subsequently here highlighted. In the end, discussions are here made, and limitations of deep learning approach in sentiment analysis in Albanian language reported.
***Keywords*:** Albanian language, sentiment analysis, machine learning, deep learning

1. INTRODUCTION

Sentiment analysis or opinion mining aims to automatically classify people's opinions towards a particular topic or event (Liu 2012; Singh *et al*., 2020). Over the past 20 years, many researchers from different areas or disciplines are paying more and more attention to sentiment analysis, which is made possible thanks to the availability of the huge amounts of data on social media platforms such as Facebook, Twitter, WhatsApp, LinkedIn, Instagram, to name a few.

Sentiment analysis applications, among others, include domains such as business, politics, healthcare, communications, education, financial markets, etc.

During the last two decades, sentiment analysis has been widely studied in NLP, data mining, web mining, text mining, and information retrieval. It also has been studied in other areas from computer science to management sciences and social sciences. Over the last few years, many efforts have been made by machine learning research community to study sentiment analysis focusing on social networks.

There are several learning approaches for sentiment analysis to be grouped into five major categories including lexicon-based, machine learning, deep learning, a combination of lexicon and machine learning, concept-based learning approaches.

Deep learning is a kind of representation learning (hierarchical feature learning), which makes it easier to extract useful information (automatic feature engineering) when building classifiers or other predictors (Bengio 2013). Deep learning approaches have recently become the dominant methodology in sentiment classification and of great interest for the community and industry. Sentiment classifiers relying on deep learning (neural based) approaches have shown state-of-the-art results on various tasks (Zhang *et al*., 2018).

Research on sentiment analysis for rich-resource languages has already achieved great success especially for English due to the availability of large enough datasets, plenty of tools and opinion lexical resources. On the other hand, greater efforts ought to be made to develop and adopt existing tools and resources for similar performances in other low-resource languages, such as Albanian language.

Despite significant result achieved in the field of sentiment analysis, some challenges (limitations) regarding this area such as identification of sarcasm and irony, noisy texts, sentiment to objective statements, etc. still remain. In addition, one major limitation for sentiment analysis today is most tools and resources are available only for rich-resource languages; in contrast, many low-resource languages that are spoken and written by millions of people have no such resources or tools available.

The present paper reviews and assesses the performance of deep learning models that have achieved state-of-the-art results on sentiment analysis for low-resource languages.

The reminder of the paper is organized as follows: Section 2 overviews briefly the background theory through a general introduction of sentiment analysis, learning approaches used for sentiment analysis, especially from the context of sentiment analysis for the low-resource languages. Section 3 presents state-of-the-art review on sentiment analysis using machine learning and deep learning-based approach, focusing on low-resource languages. The paper concludes with some future directions presented in Section 4.

2. BACKGROUND

This section introduces shortly the basic concepts that are more than necessary to facilitate understanding of the sentiment analysis, learning approaches used for sentiment analysis and a brief overview of deep learning methods that are mainly used in this area.

Sentiment Analysis

Sentiment analysis, otherwise named opinion mining, is the field of study that analyses people's opinions, attitudes, and emotions expressed towards a particular topic or event (Liu 2012; Singh *et al*., 2020). The term sentiment analysis is mostly used in industry. Both sentiment analysis and opinion mining terms are used in the academic enterprises in this field. Here, the terms ''sentiment analysis'' and ''opinion mining'' are used interchangeably (Liu 2012).

Machine learning

Machine learning (ML) is a subfield of artificial intelligence that deals with building a computer system that learns from data without being explicitly programmed (Samuel 1959; Domingos 2012). It has made significant progress and received enormous attention in the research community and industry.

Over the last two decades, ML techniques have been used successfully to a wide range of applications ranging from speech recognition, document categorization, document segmentation, part-of-speech tagging, word-sense disambiguation, named entity recognition, parsing, machine translation, sentiment analysis and opinion mining and so forth.

In general, there are three major learning approaches in ML field: supervised, unsupervised and reinforcement learning. However, supervised learning dominates compared to other ML approaches, as almost all the significant results are achieved by using this approach.

Sentiment analysis using ML techniques deals with the classification of unstructured data and text into positive, neutral, and negative categories (Ahmad *et al.*, 2017).

Deep learning

Artificial neural networks (ANNs) or simply neural networks (NN) are well-known class of machine learning models that are loosely inspired by concepts from biological brains. On the other hand, deep learning is a subfield of machine learning that deals with deep learning in artificial neural networks (Schmidhuber 2015).

Despite the opportunities that offer the machine learning methods, these methods are struggling with their ability to deal with natural data in their raw form. In other words, these methods rely on manual feature engineering.

On the other hand, deep learning is a kind of representation learning that makes possible automatic feature engineering when building classifiers or other predictors (Zhang *et al*., 2018). Furthermore, deep learning benefits from the availability of huge amount of data and fast enough computers which make it possible to train large neural networks, which further benefit in their performance (Ng 2017).

Deep learning has made significant progress and received enormous attention in the research community and industry, due to their state-of-the-art result achieved in various applications including computer vision, speech recognition, natural language processing, machine translation, online advertising, web search, recommendation systems, etc. Further details about these models could be found in (LeCun *et al.*, 2015; Schmidhuber 2015; Kastrati and Biba 2021).

3. STATE-OF-THE-ART REVIEW

More recently, researchers have made considerable progress with regard to sentiment classification by employing several deep learning and machine learning concepts. In this section, we will briefly describe several studies concerning sentiment analysis of web contents regarding end user opinions, attitudes and emotions expressed towards a particular product, topic or event using deep learning approach.

Machine Learning for sentiment classification

Conventional machine learning models have been widely used for sentiment analysis tasks. Ahmad *et al*., (2017) through their comprehensive review on ML algorithms and techniques used for sentiment analysis, they provided a comparative study on some of the most popular machine learning models used in the area including SVM, Naïve Bayes, Logistic Regression, Maximum Entropy, Random Forest, and Decision Trees. They concluded that SVM outperforms the other machine learning algorithms regarding sentiment analysis tasks in terms of accuracy and efficiency.

Biba and Mane (2014) presented the first approach for sentiment analysis in Albanian. They developed a machine-learning model to classify text documents belonging to a negative or positive opinion regarding the given topic. To train their machine learning models they built a corpus of 400 documents containing political news consisting of five different topics. In this case, each topic was represented by 80 documents classified as positive or negative. The authors made an empirical comparison between 6 different

machine learning algorithms, including Bayesian Logistic Regression, Logistic Regression, SVM, Voted Perceptron, Naïve Bayes, and Hyper Pipes for classification of text documents, achieving an accuracy between 86% and 92% depending on the topic. The authors conclude that, in general, to achieve higher accuracy in sentiment analysis, a larger corpus in the Albanian language is needed.

Kote *et al*., (2018) presented a comprehensive experimental evaluation of machine learning algorithms applied for opinion mining in Albanian language. Among the algorithms tested that performed better include Logistic and Multi-Class Classifier, Hyper Pipes, RBF Classifier, and RBF Network with the classification accuracy ranging from 79% to 94%. Classification algorithms are trained on a corpus of 500 news articles in Albanian consisting of five different topics. Each topic was represented with a balanced set of articles classified as positive or negative. The experimental results were interpreted concerning several evaluation criteria for each algorithm showing interesting features on the performance of each algorithm.

Later, Kote *et al.,* (2018b) further extended their research from an in-domain to a multi-domain corpus (for the Albanian language). They created 11 multi-domains corpuses combining opinions from 5 different topics, which was later split for training and testing of 50 classification algorithms in Weka. The result obtained showed that from all evaluated algorithms seven of them performed better and three of them are based on Naïve Bayes. In addition, the authors concluded that the availability of a larger corpus in the Albanian language could further improve the performance of classification algorithms for opinion mining.

Kote and Biba (2021) presented their results of experimental evaluation about the performance of machine learning techniques on opinion classification tasks in the Albanian language. Their approach to opinion mining addresses the problem of classifying text document opinions as positive or negative opinions for Albanian language. Authors here conducted an experimental evaluation of fourteen techniques used for opinion mining. They tested several machine learning algorithm's performances by the help of Weka. Authors reported that Naïve Bayes Multinomial outperformed other machine learning algorithms tested, and the best result obtained in terms of correctly classified instances was 84.88%.

Deep Learning for sentiment classification

This section briefly overviews some of the most prominent deep neural networks employed in sentiment analysis tasks including Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and their extended version such as Bidirectional Recurrent Neural Network (BiRNN), Long-Short Term Memory (LSTM), Bidirectional Long-Short Term Memory

(BiLSTM) and Gated Recurrent Unit (GRU) and hybrid neural networks that takes the advantages of the two complementary deep neural networks.

BiRNNs are extended variant of RNNs architecture. While unidirectional RNNs are influenced from the previous inputs to make predictions about the current state, BiRNNs pull in future data to improve its accuracy. In other words, the output is influenced from previous inputs (backwards) and future states (forwards) simultaneously. LSTM is an RNN architecture specifically designed to handle the problem of long-term dependencies and performs better at storing and accessing information than standard RNNs. On the other hand, GRU was proposed with idea to make each recurrent unit to adaptively capture dependencies of different time scales, respectively, to reduce computational burden caused by additional parameters used in LSTM (Kastrati and Biba 2021).

In general, each of these neural networks is specialized to certain tasks, due to the different nature in their text modelling capabilities. For example, CNNs are specialized for using in computer vision, and RNNs are mainly used for sequential inputs such as time series or natural language. However, their extended versions have also achieved significant result, such as the case with 1D-CNN that can extract local features from the text. On the other hand, BiLSTM has proved to perform good at capturing contextual information from both direction as well as the long-range dependencies. Furthermore, the hybrid models benefit by taking the advantages of both complementary architectures (Kastrati *et al*., 2021).

There is a considerable number of research studies about sentiment analysis problems, mostly in English in the last decade, but only a few research studies on sentiment analysis for low-resource languages, such as Albanian language. Skënduli *et al*., (2018) presented an approach for user-emotion detection on microblogging texts and postings in the Albanian language. The authors studied user-emotion at the sentence level by using deep learning methods. Their approach was based on the idea of classifying a text fragment into a set of pre-defined emotion categories (based on Ekman's model), and therefore aims at detecting the emotional state of the writer conveyed through the text. To perform their experiments, they built a new dataset that contained manually annotated posts on Facebook belonging to some active Albanian politicians, which were classified using Ekman's model, and finally divided into six smaller datasets. A set of experiments on these datasets to evaluate deep learning and conventional machine learning models subsequently followed. In addition, during their analysis, they also adopted a domestic stemming tool for the Albanian language to pre-process the datasets, which showed a slight improvement in the classification accuracy. In general, the obtained results showed that deep learning outperformed the other conventional machine learning models, i.e., Naïve Bayes (NB), Instance-

based learner (IBK), and Support Vector Machines (SMO) with a given accuracy ranging from 70.2% to 91.2% for (correctly classified unstemmed instances), and 67.0% to 92.4% for (correctly classified stemmed instances).

Later, Skenduli and Biba (2020) proposed an analysis of micro-blogging content to characterize the users individually when writing posts with emotional content. First, the present investigation aimed at collecting textual units that allowed them to summarize the lexicon used by the user. They focused on sentence-based emotion detection problems. The main aim of this approach was to classify the textual units into a set of pre-defined emotion categories based on the Ekman model. Second, the analysis was carried out through a keyword extraction approach, which aimed at finding representative generic word sets in the form of prototypes of textual unit clusters. In addition, several experiments were carried out from different points of view, yet always focusing on the user. They provided useful information about classification accuracy, clustering, and other valuable information concerning user emotion profiling.

Kastrati *et al*., (2021) presented the sentiment analysis of people's opinions expressed on Facebook with regards to the pandemic situation in Albanian language. They developed a deep learning model to classify people's opinions belonging to a neutral, negative, or positive opinion regarding the given topic. To train the deep learning models they created a dataset containing 10.742 manually annotated Facebook comments in the Albanian language. The authors reported their efforts on the design and development of a sentiment analyser that relies in deep learning. The researchers reported their experimental results obtained using deep learning models with static and contextualized word embeddings, that is, fastText and BERT. The authors concluded that combining the BiLSTM with an attention mechanism outperformed the other approaches in their sentiment analysis task. The best results achieved by their proposed model given in terms of Precision Recall, and F1 score were 72.31%, 72.25%, and 72.09%, respectively.

Vasili *et al*., (2021) tested and reviewed several approaches in sentiment analysis on Twitter messages for the Albanian language. In addition, they compared the results among several methods and noted the challenges that arise when dealing with sentiment analysis for Albanian language, and finally made their suggestions for the future work. Authors here investigated the performance of sentiment classification techniques using three main approaches: traditional machine learning, lexicon-based and deep learning-based approach. Their experiments revealed that LSTM based RNN with Glove as a feature extraction technique provides the best results with F-score = 87.8%, followed by Logistic Regression.

Table. 9 A summary of the studies on sentiment analysis for low-resource languages (the case of Albanian language) using machine/deep learning classifiers

Ref no.	Motivation and aim of the work	Methods used in the work	Dataset used	Obtained Accuracy	Limitations
Biba and Mane (2014	To analyze polarity of documents containing political news in Albanian language consisting of five different topics	Six different machine-learning algorithms: Bayesian Logistic Regression, Logistic Regression, SVM, Voted Perceptron, Naïve Bayes, and Hyper Pipes.	400 documents containing political news consisting of five different topics.	Achieved accuracy was between 86% and 92% depending on the topic.	Small and in-domain corpus. Dataset is not publicly available.
Kote *et al*., (2018)	To analyse the sentiments of the news articles in Albanian language consisting of 5 different topics	Machine learning: Logistic and Multi-Class Classifier, Hyper Pipes, RBF Classifier, and RBF Network	500 news articles in Albanian consisting of five different topics.	Accuracy obtained was ranging from 79% to 94%.	Small and in-domain corpus.
Kote *et al*., (2018b)	To further extend their research from an in-domain to a multi-domain corpus	Machine Learning: classifiers. From 7 of them that performed better, 3 were based on Naïve Bayes.	11 multi-domains corpuses combining opinions from five different topics	Weka was used for analysis, and the result obtained were promising.	Small corpus. Datasets are not publicly available.
Kote and Bi (2021)	To evaluate the performance of machine learning techniques on opinion classification tasks in the Albanian language	Machine Learning: classifiers. Several algorithms evaluated: Naïve Bayes algorithm showed the best performance in terms of the average value.	500 news articles in Albanian language consisting of five different topics such as higher education law, waste import, VAT in small businesses, tourism, and politics	Naïve Bayes Multinomial outperformed others with 84.88% of correctly classified instances.	Small corpus. Datasets are not publicly available.
Kastrati *et a* (2021)	To analyze people's opinions expressed on Facebook regarding Covid-19 pandemic situation in Albanian language	Deep Learning: 1D-CNN, BiLSTM, and a hybrid 1D-CNN + BiLSTM model	10742 comments collected from the NIPHK's Facebook page between 13th March and 15th August, 2020	The best results achieved given in terms of P: 72.31%, R: 72.25%, and F1: 72.09%.	Small and in-domain corpus.
Skënduli *et al*., (2018)	To analyse user-emotion detection on microblogging texts and postings in the Albanian language	Deep Learning: Convolutional neural networks (CNNs)	6,358 Facebook posts belonging to Albanian politicians	Accuracy for: Unstemmed: 70.2% - 91.2% Stemmed: 67.0% - 92.4%	Small and in-domain corpus. Dataset is not publicly available.
Skënduli an Biba (2020)	To analyse micro-blogging content to	Deep Learning: Convolutional neural	6,358 Facebook posts belonging	Accuracy for: Unstemmed:	Small and in-domain

	characterize the users individually when writing posts with emotional content	networks	to Albanian politicians	70.2% - 91.2%	corpus. Dataset is not publicly available.
Vasili *et al.*, (2021)	To test and review different approaches in Sentiment Analysis on Twitter messages in the Albanian language	Deep Learning: Convolutional neural networks (CNNs)	Was used the dataset consisting of 15 languages from (Mozetič *et al.*, 2021)	By sing LSTM-RNN with Glove, was achieved the best results given in terms of F1: 87.8%	"low quality annotators which should be eliminated from further considerations "

4. CONCLUSIONS

Here papers about sentiment analysis and emotion detection in Albanian as summarized in Table 1 are reviewed. The performance of conventional machine learning and deep learning methods clearly depends on the pre-processing and size of the dataset. Furthermore, deep learning benefits from using representation learning, large amount of training data and higher computational power that make possible training large and complex deep neural networks for sentiment and emotion classifiers. As it can be noticed, in situation where the dataset is vast, deep learning methods such as 1D CNN, LSTM, BiLSTM and combined CNN with BiLSMT architectures generally outperform conventional machine learning methods (Naïve Bayes, SVM, Logistic Regression and Decision Trees). Overall, we can conclude that deep learning methods are the method of choice for sentiment analysis and emotion detection in Albanian.

REFERENCES

Ahmad M, Aftab S, Muhammad SS, Ahmad S. 2017. Machine learning techniques for sentiment analysis: A review. *International Journal of Multidisciplinary Sciences and Engineering*, **8(3):** 27.

Bengio Y, Courville A, Vincent P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, **35(8):** 1798-1828.

Biba M, Mane M. 2014. Sentiment analysis through machine learning: an experimental evaluation for Albanian. *Recent Advances in Intelligent Informatics*. Springer, 195–203.

Domingos P. 2012. A few useful things to know about machine learning. *Communications of the ACM*, **55 (10):** 78–87.

Kastrati M, Biba M. 2021. A State-of-the-art survey on deep learning methods and applications. *International Journal of Computer Science and Information Security* (IJCSIS), **19(7):** 53-63.

Kastrati Z, Ahmedi L, Kurti A, Kadriu F, Murtezaj D, Gashi F. 2021. A Deep learning sentiment analyser for social media comments in low-resource languages. *Electronics*, **10 (10):** 1133.

Kote N, Biba M, Trandafili E. 2018. A thorough experimental evaluation of algorithms for opinion mining in Albanian. International Conference on Emerging Internetworking, Data & Web Technologies. Springer, 525–536.

Kote N, Biba M, Trandafili E. 2018 (b). An experimental evaluation of algorithms for opinion mining in multi-domain corpus in Albanian. International Symposium on Methodologies for Intelligent Systems. Springer, 439–447.

Kote N, Biba M. 2021. Opinion Mining in Albanian: Evaluation of the Performance of Machine Learning Algorithms for Opinions Classification. *International Journal of Innovative Science and Research Technology*, **6 (6)**: 964-973. ISSN No-2456-2165.

LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. *Nature*, **521(7553):** 436-444.

Liu B. 2012. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, **5(1)**: 1-167. Morgan & Claypool Publishers, Vermont, Australia.

Mozetič I, Grčar M, Smailović J. 2016. Multilingual Twitter sentiment classification: The role of human annotators. PloS one, 11(5), e0155036.

Ng A. 2017. Machine learning yearning. URL: http://www.mlyearning.org/ (96), 139.

Samuel AL. 1959. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, **3 (3):** 210–229. doi: 10.1147/rd.33.0210.

Schmidhuber J. 2015. Deep learning in neural networks: An overview. *Neural networks*, vol. 61 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.

Singh NK, Tomar, DS, Sangaiah AK. 2020. Sentiment analysis: A review and comparative analysis over social media. *Journal of Ambient Intelligence and Humanized Computing*, **11**, 97–117.

Skënduli MP, Biba M, Loglisci C, Ceci M, Malerba D. 2018. User-emotion detection through sentence-based classification using deep learning: A case-study with microblogs in Albanian. *International Symposium on Methodologies for Intelligent Systems*. Springer, 258–267.

Skënduli MP, Biba M. 2020. Classification and clustering of emotive microblogs in Albanian: Two user-oriented tasks. *Complex Pattern Mining*. Springer, 153–171.

Vasili R, Xhina E, Ninka I, Terpo D. 2021. Sentiment analysis on social media for Albanian language. *Open Access Library Journal*, **8(6):** 1-31.

Zhang L, Wang S, Liu B. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **8(4):** e1253.

A STUDY OF QUANTIZATION NOISE EFFECT IN A DELTA-SIGMA ANALOG TO DIGITAL CONVERTER

Alban RAKIPI, Aleksandër BIBERAJ, Indrit ENESI
Faculty of Information Technology, Polytechnic University of Tirana, Albania

ABSTRACT

Digital technology increasingly pervades daily life, and the use of innovative techniques in analog-to-digital converter (ADC) schemes has become particularly important. Nowadays, analog to digital converters are crucial components in almost any electronic systems. Some of the most important aspects are related to the development and implementation of schemes that are energy efficient, that offer very high conversion accuracy, that suppress distortions, and are more immune to various noise sources. A problem that is widely encountered in ADC converters is quantization noise, which affects resolution, the correspondence between samples, induced delays, and limitations in developing higher order circuits. The quantization noise effect of a delta-sigma analog to digital converter is in the present paper investigated. Randomly generated quantization noise samples are used to evaluate the output performance of the converter, by measuring the signal to noise (SNR) ratio. In the simulated scenario are considered the following parameters: filter length, oversampling ratio, transition bandwidth and decimation factor. Optimal values of these parameters are suggested in order to reduce the noise effect and to improve SNR.

***Keywords*:** Analog-to-digital converter, delta-sigma, noise, quantization

1. INTRODUCTION

A delta-sigma ADC is used in various conversion systems to obtain a low-power, high-accuracy conversion using oversampling and noise shaping for audio, sensor, and wireless applications (Saikatsu and Yasuda, 2019). Successive approximation register (SAR) and delta-sigma (ΔΣ) architectures using oversampling techniques have recently gained popularity of ADC with high energy efficiency in the bandwidth range under tens of mega-hertz (Fukazawa *et al*., 2020).

In many modern electronic systems, analog to digital converters are essential components because they are responsible for converting a measured analog signal into a digital representation. The conversion of an analog signal to a digital one frequently restricts the overall system's speed and resolution. As a result, A/D converters that accomplish both high speed and resolution are required. Many image, communication, and instrumentation systems could benefit from such converters (Katara *et al*., 2012).

Pipeline, SAR, and delta- sigma modulators currently dominate the state of the art in analog-to-digital converters, which continue to push the boundaries of energy efficiency year after year (de La Rosa *et al*., 2015). The signal-to-noise ratio (SNR), measured in decibels, is a critical performance metric for different applications. Many external noise sources, such as clock noise, power supply noise, might affect the ADCs utilized in these systems (Reeder *et al*., 2005).

Fukazawa *et al.*, (2020) showed a quantization error extraction method that considers the quantizer delay and prevents the input leakage to the second stage to suppress distortion. The addition of a digital integrator reduces the amplitude of the quantizer input, preventing quantizer saturation.

Bajaj *et al*., (2020) presented a 12-bit successive approximation analog-to-digital converter embedded in a first-order noise shaping loop to obtain a 16-bit resolution while maintaining sample-by-sample correspondence. The need for high oversampling in ΔΣ A/D converters has limited their use to primarily low frequency applications (Katara *et al*., 2012).

A new method for extracting the VCO quantization noise is proposed in (Maghami *et al*., 2019). Using the proposed technique, in the time domain, the quantization noise of a VCO-based quantizer is extracted precisely as a PWM signal. By applying this pulse signal to another VCO-based quantizer, higher order structures could be implemented.

Increasing the gain coefficient of the integrator in the loop filter configuration of the delta-sigma ADC suppresses the quantization noise that occurs in the signal band. However, there is a trade-off relationship between the integrator gain coefficient and system stability (Saikatsu and Yasuda, 2019). Saikatsu and Yasuda (2019) proposed a SC integrator with the structure of a finite impulse response (FIR) filter in a loop filter configuration. This structure can realize broadband and high-precision performance, a high signal-to-quantization noise ratio, and increased integrator gain in the signal band.

The present paper evaluates the quantization noise effect of a ΔΣ A/D converter, by taking into consideration the length of the FIR filter, the oversampling ratio, the transition bandwidth and the decimation factor. The simulated results are compared with the theorical ones. The rest of the paper is outlined as follows. Section 2 describes the main architecture of a ΔΣ A/D

converter. Section 3 informs about simulation scenarios and reports the results. Section 4 makes conclusions.

2. THE DELTA – SIGMA ADC ARCHITECTURE

Delta-sigma (ΔΣ) ADCs with oversampling and noise-shaping are the most utilized ADCs for achieving high resolution among all categories (Tan *et al.*, 2020). Due to the requirement for high oversampling in A/D converters, their usage has been limited to mainly low frequency applications (Katara *et al.*, 2012).

Oversampling ADCs with a feedback loop have been created to further minimize noise in the low-frequency region; the most prevalent today is the A/D converter. The architecture of a such converter is in Figure 1 depicted. The serial bit stream coming out of the comparator contains the average value of the input voltage (1-bit ADC). The serial bit stream is processed by the digital filter and decimator, which produces the final output data (Kester 2009b).

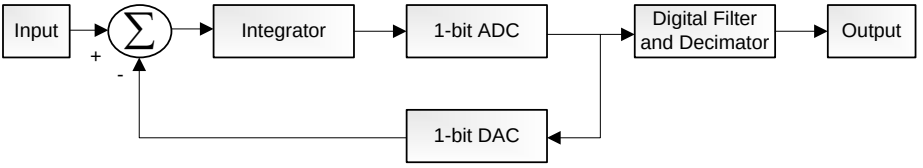


Fig. 1. The architecture of a first order delta-sigma ADC.

Given a signal power $E\left\{\left|x\right|^2\right\}$ and a noise power $E\left\{\left|n_q\right|^2\right\}=\sigma_q^2$ the output SNR is given by equation in (1) (Martin 2012):

$$\left(\frac{S}{N}\right)_{out}=\frac{E\left\{\left|y_x\right|^2\right\}}{E\left\{\left|y_n\right|^2\right\}}=\frac{E\left\{\left|x\right|^2\right\}}{2\sigma_q^2} \qquad (1)$$

However, it should be considered that the signal $x(t)$ can be highly oversampled, and low-pass filtering can reduce the final quantization noise.

Given an over-sampling ratio (OSR) in equation (2) (Martin 2012)

$$OSR=K=\frac{f_s}{2B_x} \qquad (2)$$

the actual noise power after decimation and down-sampling can be evaluated from the scheme in Figure 2 (Kester 2009a).

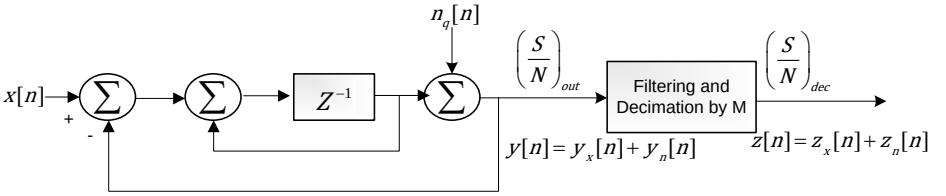


Fig. 2. The block-diagram for SNR evaluation after filtering and decimation.

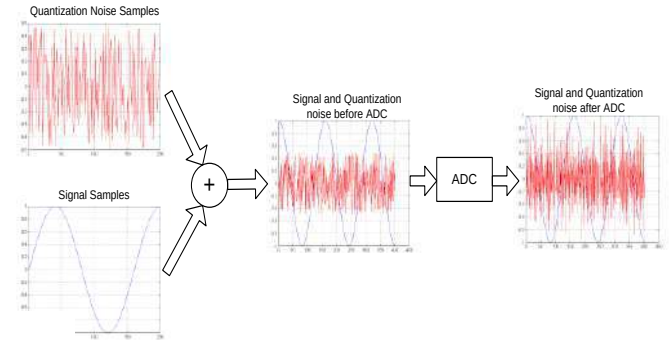
Once sampling occurs, filtering and decimation are carried out by the actual ADC to eliminate the quantization noise. For K much greater than one the signal to noise ratio after decimation is given by the formula in (3) as reported in (Martin 2012):

$$\left(\frac{S}{N}\right)_{dec} = \frac{E\left\{\left|Z_x\right|^2\right\}}{E\left\{\left|Z_n\right|^2\right\}} = \frac{E\left\{\left|x\right|^2\right\}}{2\sigma_q^2\left[\frac{1}{K} - \frac{1}{\pi}\sin\left(\pi\frac{1}{K}\right)\right]} = 3K^3\frac{E\left\{\left|x\right|^2\right\}}{2\sigma_q^2\pi^2} \qquad (2)$$

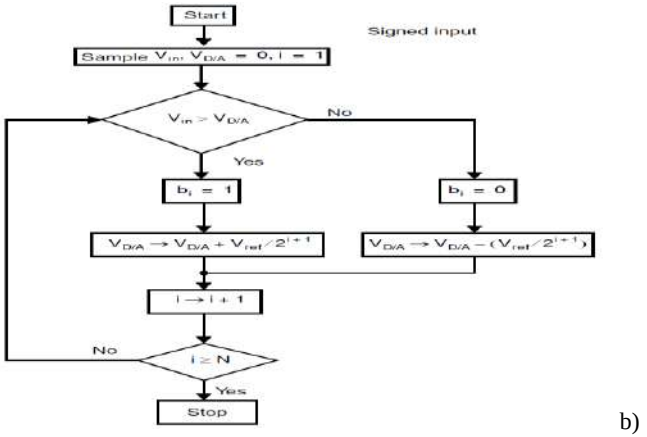
If good decimation filters can be designed, the SNR performance of Δ-Σ ADC improves with the oversampling ratio K.

3. SIMULATION SCENARIO AND RESULTS

The simulation of the delta-sigma ADC is modeled using MATLAB. The conceptual block-diagram of the simulation scenario is given in Fig. 3a. Randomly generated quantization noise samples (white noise) are added to the pure sinewave signal to be converted. The conversion algorithm used is based on the binary search algorithm which determines the closest digital word to match an input signal (Martin 2012). The flow-chart using a successive-approximation approach is shown in Figure 3b.



a)



b)

Fig. 3: a) Conceptual block-scheme for quantization noise evaluation, b) Flow chart for the successive-approximation algorithm (Martin 2012).

In our scenario four different parameters are considered. In the first run, the filter length parameter, LFIR is changed from 500 (the default value) to 400 and 600 with respect to the same oversampling ratio K, and the results are in Figure 4 and 5 depicted.

The Figure 4 and 5 show that as number of the filter coefficients (filter length LFIR) increases, a better SNR performance could be achieved. The initial SNR value is a little bit higher in a case of LFIR = 600 than in the case of LFIR = 400 and obtained SNR curve (the blue curve) almost follows the trend as the theoretical one (the red curve). With the lower filter length (400) the SNR values linearly decrease with the gradient proportional to a drop in LFIR.

In the second run, the oversampling factor K is considered to see whether the SNR performance can be improved. In the MATLAB routine the K value

is changed. A comparison for the chosen values of K = 80 and K = 140, are in Figure 6 and 7 depicted.

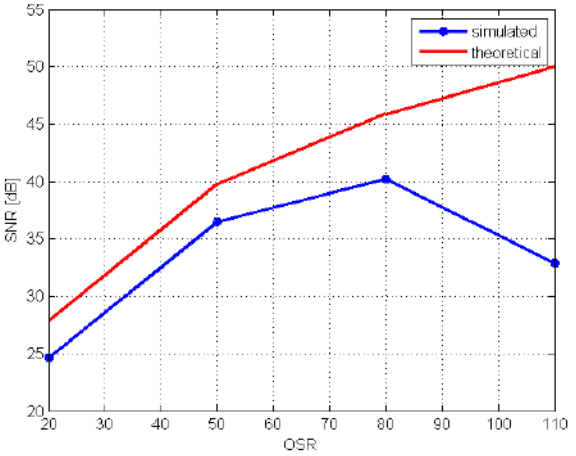


Fig. 4: SNR after filtering in Delta-Sigma ADC for LFIR=400.

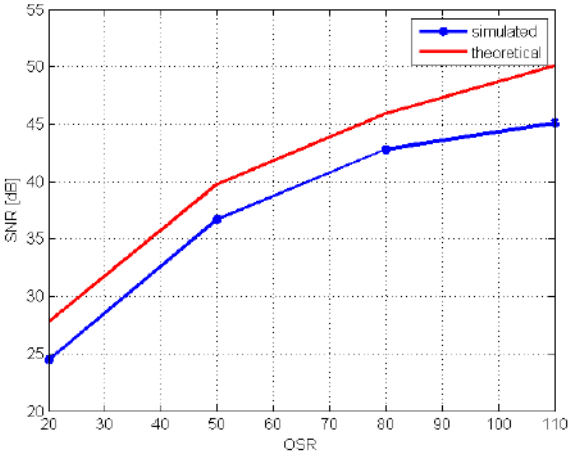


Fig. 5: SNR after filtering in Delta-Sigma ADC for LFIR=600.

The results show that SNR performance significantly change with the change of an oversampling ratio K. The SNR curve becomes almost linear as K decreases. In addition, it behaves as the theoretical with the lower SNR values. It is known that the signal can be highly oversampled and low-pass filtering can reduce the final quantization noise. With down-conversion of a

digital signal the oversampling factor decreases (K↓). Figure 6 and 7 show that for the lower K values (20-40) the SNR performance improves. A drop in SNR performance follows the rest of the graph.

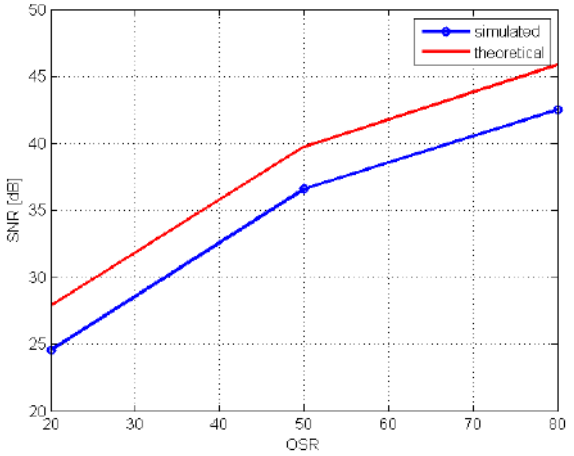


Fig. 6: SNR after filtering in Delta-Sigma ADC for K = 80.

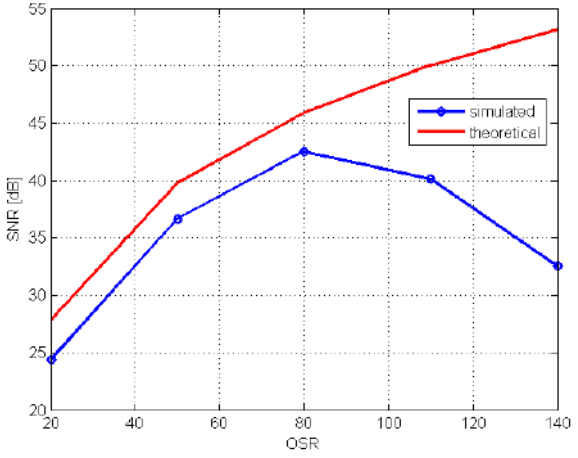


Fig. 7. SNR after filtering in Delta-Sigma ADC for K = 140.

In the third run, the impact of transition bandwidth in SNR is investigated. The performance is evaluated depending on the changes of the $band_1$ and $band_2$. $Band_1$ presents the filter passband and $band_2$ a lower limit of the filter stop-band. The filter transition band is the difference between $band_2$ and $band_1$. Both bands depend on the signal bandwidth f_0. The latter depends on

the normalized sampling frequency f_S and an oversampling factor K. The results are in Figure 8 and 9 shown.

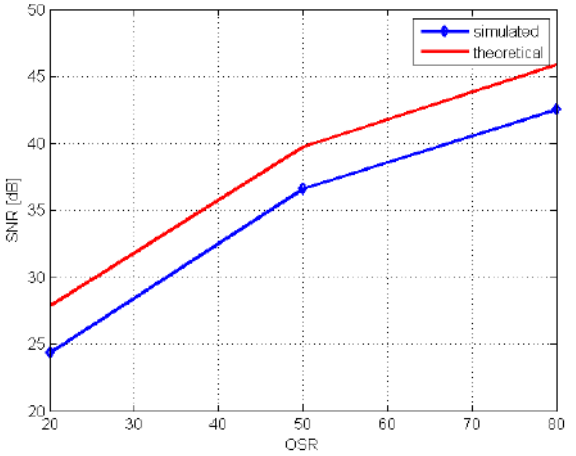


Fig. 8: SNR after filtering in Delta-Sigma ADC for the transition bandwidth where ($band_2=2.2f_0$ and $band_1=1.2f_0$)

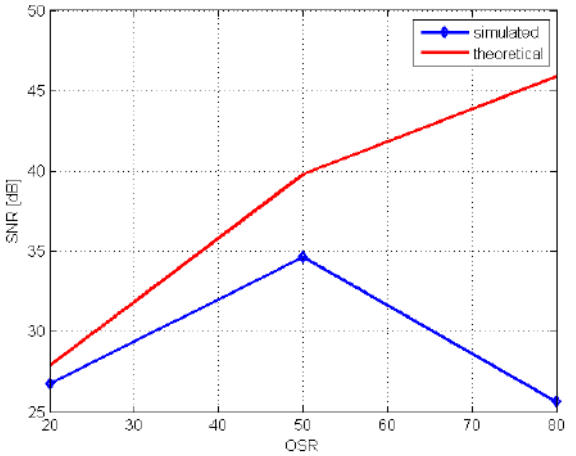


Fig. 9: SNR after filtering in Delta-Sigma ADC for the smaller transition bandwidth ($band_2=1.6f_0$ and $band_1=1.2f_0$).

Results show that the SNR performance drops as the difference between the filter passband and the lower limit of the filter stop-band becomes smaller. So, the smaller the transition bandwidth is, the lower the SNR values are (Figure 8). If the transition bandwidth is narrowed, the attenuation increases. Consequently, transition bandwidth is the critical parameter, not the

bandwidth. The SNR performance improves as increase the transition bandwidth increases. We changed the multiple rates of the signal bandwidth to get different values of $band_1$ and $band_2$, and therefore a larger value of the filter transition bandwidth. With a wider transition bandwidth, the measured SNR performance after filtering closely follows the theoretical one (Figure 11).

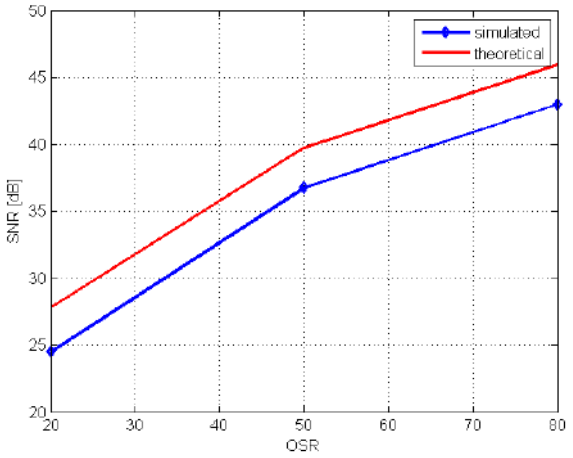


Fig. 10: SNR after filtering in Delta-Sigma ADC for the transition bandwidth ($band_2=2.4f_0$ and $band_1=1.0f_0$).

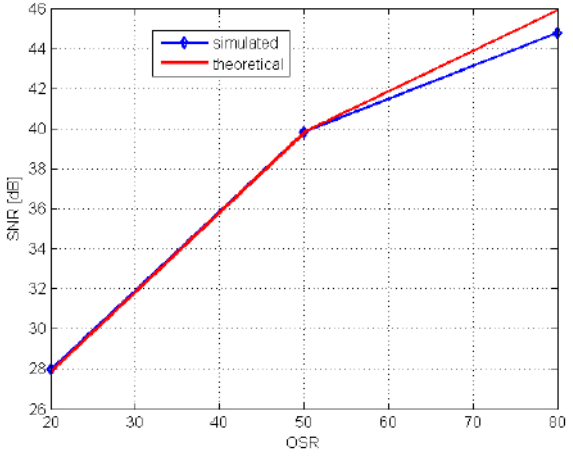


Fig. 11: SNR after filtering in Delta-Sigma ADC for the transition bandwidth ($band_2=1.8f_0$ and $band_1=0.8f_0$)

In the last run, the SNR behavior with respect to a decimation factor L is evaluated. When a digital signal is up converted, a decimation is performed by generating its replicas. With the decimation factor equal to 10, is obtained the SNR plot of Figure 12. Here we cold note that the initial SNR value is a little bit lower due to a lower limit of the filter stop-band inversely proportional to the decimation factor L. In the next step we modified L value to L = 6, L = 7 and L = 8, and the results are shown in Figures 13-15.

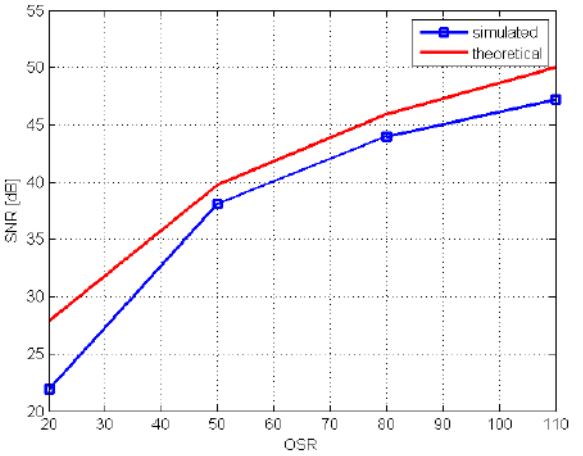


Fig. 12: SNR after filtering in Delta-Sigma ADC for L = 10.

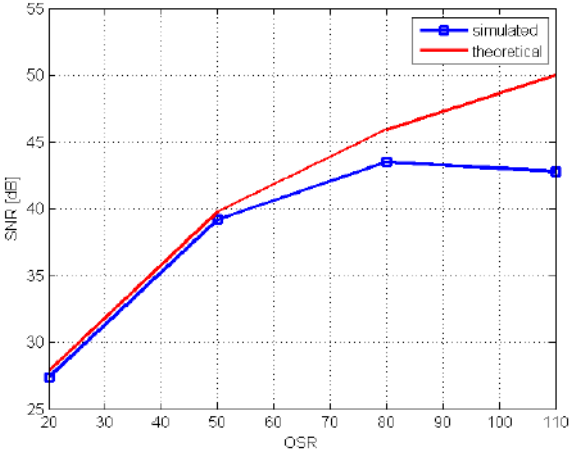


Fig. 13: SNR after filtering in Delta-Sigma ADC for L = 6.

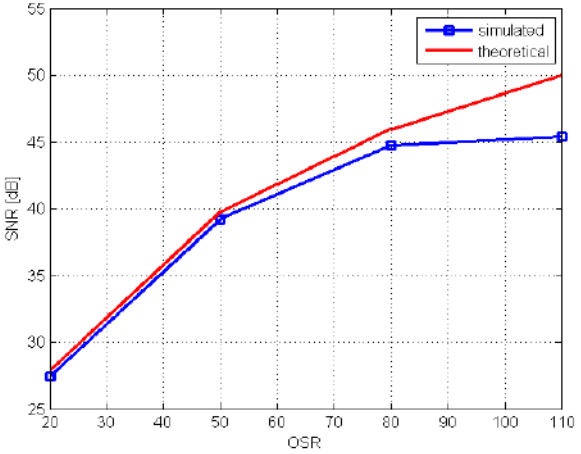


Fig. 14: SNR after filtering in Delta-Sigma ADC for L = 7.

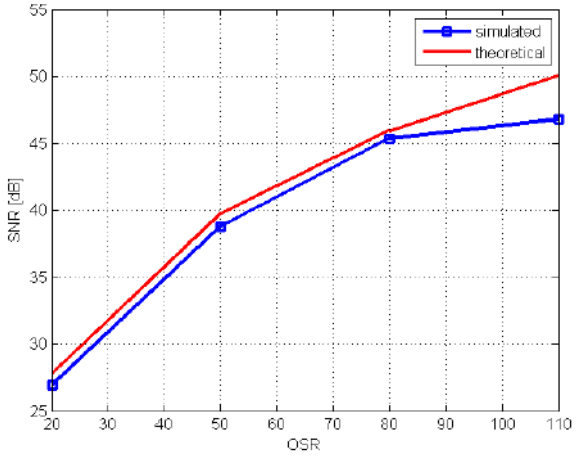


Fig. 15: SNR after filtering in Delta-Sigma ADC for L = 8.

Based on figures, it could be concluded that as the decimation factor increases, the initial SNR values get lower with respect to the theoretical ones. In the first step (K from 20 to 50) the measured SNR curve is becoming closer to the theoretical. In the second step (K from 50 to 80) they are increasing by the same gradient, and in the third step (K from 80 to 110) the measured diverges from the theoretical with a different increasing slope. By setting a lower decimation factor (L = 8) both SNR curves are approximately the same up to oversampling factor K = 80.

4. CONCLUSIONS

In present paper investigates the quantization noise effect and SNR performance of a delta-sigma ADC. The results show that the higher the number of the filter coefficients (filter length LFIR) is, the better SNR values with the higher oversampling factor K are.

The oversampling factor K should be high enough to ensure better SNR values after filtering in Delta-Sigma ADC, but with some limited value since with some high values of K the measured SNR performance deviate from the theoretical one.

The transition bandwidth, as a difference of passband and a lower limit of the stopband, should be larger and chosen in such a manner that SNR curve reaches optimal values; both bandwidths depend on the f_S and f_0 and should be chosen in an optimal proportion. If the transition bandwidth is increased, the attenuation increases. Consequently, the transition bandwidth is the critical parameter, not the bandwidth.

Both filtering and decimation could be carried out at a lower speed, and by properly choosing decimation factor it is possible to reduce the noise effect and consequently improve SNR.

REFERENCES

Bajaj V, Kannan A, Paul ME, Krishnapura N. 2020. Noise shaping techniques for SNR enhancement in SAR analog to digital converters. *Proceedings - IEEE International Symposium on Circuits and Systems*, *2020-October*. https://doi.org/10.1109/iscas45731.2020.9180536.

De La Rosa JM, Schreier R, Pun KP, Pavan S. 2015. Next-Generation Delta-Sigma Converters: Trends and Perspectives. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *5*(4). https://doi.org/10.1109/JETCAS.2015.2502164.

Fukazawa M, Fujiwara M, Ochi A, Alsubaie R, Matsui T. 2020. An Input Insensitive Quantization Error Extraction Circuit for 8-MHz-BW 79-dB-DR CT MASH Delta-Sigma ADC with Multi-rate LMS-Based Background Calibration. *IEEE Solid-State Circuits Letters*, *3*. https://doi.org/10.1109/LSSC.2020.3024061.

Katara A, Bapat AV, Chalse R, Selokar A, Ramteke S. 2012. Development of one bit delta-sigma analog to digital converter. *Proceedings - 4th International Conference on Computational Intelligence and Communication Networks, CICN 2012*. https://doi.org/10.1109/CICN.2012.98.

Kester W. 2009a. ADC Architectures IV: Sigma-Delta ADC Advanced Concepts and Applications. *Imid 2009*.

Kester W. 2009b. MT 022: ADC Arquitectures III: Sigma Delta ADC Basics. *Delta*.

Maghami H, Mirzaie H, Payandehnia P, Zanbaghi R, Fiez T. 2019. A novel time-domain phase quantization noise extraction for a VCO-based quantizer. *Midwest Symposium on Circuits and Systems*, *2018-August*. https://doi.org/10.1109/MWSCAS.2018.8623894.

Martin JC. 2012. Analog Integrated Circuits Design 2nd. In *John Wiley & Sons, Inc.*

Reeder B, Looney M, Hand J. 2005. Pushing the State of the Art with Multichannel A/D Converters. *Analog Dialogue*, **39(05):** 1. https://www.analog.com/media/en/analog-dialogue/volume-39/number-2/articles/the-right-adc-architecture.pdf.

Saikatsu S, Yasuda A. 2019. Delta-sigma ADC based on switched-capacitor integrator with FIR filter structure. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *E102A*(3). https://doi.org/10.1587/transfun.E102.A.498.

Tan Z, Chen CH, Chae Y, Temes GC. 2020. Incremental Delta-Sigma ADCs: A Tutorial Review. In *IEEE Transactions on Circuits and Systems I: Regular Papers*, **67 (12):** https://doi.org/10.1109/TCSI.2020.3033458.

INTRUSION DETECTION SYSTEM IN SOFTWARE-DEFINED NETWORKS USING ML'S XGBOOST ALGORITHM AND OPEN-SOURCE IPS

Olimpjon SHURDI, Ergest ALITE, Aleksandër BIBERAJ and Genci MESI
Faculty of Information Technology, Polytechnic University of Tirana, Albania

---

ABSTRACT

Public cloud has nowadays become one of the most required and used IT platforms—especially during the outbreak of the pandemic caused by the coronavirus. The pandemic period was unfortunately characterised by an increase of cyber-attacks activity. Given the situation, studying and implementing an end-to-end security solution that is based on machine learning's algorithm, called XGBoost, and open-source IPS called snort would have been considered important. The present paper informs about the implementation of an end-to-end cyber security solution to address the protection of public cloud platforms against some different types of security attacks like Distributed Denial of Service (DDoS) attack, Denial of Service (DoS) attack, User to Root (U2R) attack and Remote to Local (R2L) attack, and the results showed its effectiveness when compared to the other methodologies which are using Support Vector Machines (SVM) algorithm.

***Keywords*:** Public Cloud, XGBoost, Distributed Denial of Service (DDoS), Support Vector Machines (SVM)

1. INTRODUCTION

The present paper provides information about cloud computing security due to its great importance and the benefits—especially during the outbreak of the pandemic caused by the coronavirus while labor force working from home with the same efficiency as from the office.

The security of using these platforms has come to the fore due to the high number of users and the high number of already essential services they offer (such as business communication applications Teams, Zoom, etc.). The proposed analysis is based on the study of SECURE (Podlodowski *et al*.,

2019) technique extending it to work for both UDP flood and NTP amplification security attacks— making it appropriate for a comprehensive use. It will also enable us to see changes in the execution performance of this method as a result of its modification by adding protection from additional security attacks.

The present paper reports that the modified technique is one which achieves a defense against more than one type of cyber security attack with the same effectiveness. Here, a network level intrusion detection system called SNORT (Snort 2020) is used. An anomaly detector based on the decision tree and the extreme gradient boosting algorithm (XGBoost) will also be used to analyze abnormal activities (unknown attacks). SNORT is the most effective Intrusion Detection System (IDS). Various Machine Learning (ML) techniques are used for anomaly-based IDSs, but XGBoost is the most widely used anomaly detector based on recent studies. Both of these mechanisms will guarantee the performance and reliability of this method and will ensure the desired result. To extend this method, the attack generator is modified by adding a UDP (UDP Flood) and NTP (NTP Amplification) attack generator to the attack generator. A UDP flood and NTP amplification attack module is also be added to the security attack module. It is noted while analyzing the SECURE technique that this technique protects systems from executing five different types of security attacks, including Denial of Service DoS, Probing, Remote to Local Distance (R2L), User to Root attacks (U2R) and Distributed Denial of Service (DDoS) attacks.

Efforts to use this technique as a defensive infrastructure against other types of malwares such as ransomware attacks, slowloris, etc. will be made in a near future. Here, strategic procedures would be necessary to maintain the effectiveness and reliability.

2. RELATED WORKS

According to a Forbes' report published in 2015, cloud-based security spending is expected to increase by 42%. According to another study, IT security spending had increased to 79.1% by 2015, showing an annual increase of more than 10%. The International Data Corporation (IDC) in 2011 showed that 74.6% of corporate clients rated security as a major challenge. This paper summarizes a number of peer-reviewed articles on Security Threats in Cloud Computing and Preventive Methods. The objective of the research is to understand Cloud components, security issues and risks, along with emerging solutions that can potentially mitigate vulnerabilities in the Cloud. It is a widely accepted fact that Cloud Computing has been a valuable hosting platform since 2008. However, the perception regarding security in the Cloud

is that it needs significant improvements realized at higher levels of adaptation at the company level (Barse *et al*., 2020).

As identified by another research, many of the issues faced by Cloud Computing need to be addressed urgently. The industry has made significant progress in combating threats to Cloud Computing, but there is still more to be done to reach the level of maturity that currently exists with traditional on-premise hosting.

Recent studies on cloud computing security propose ways to protect, but all solutions require a lot of computer resources (Buchanan *et al*., 2016; Khan *et al*., 2016 Gaur *et al*., 2017; Ramachandra *et al*., 2017). Cloud Computing due to its distributed nature, complex architecture and utilized resources poses a unique and serious risk to all actors. Understanding the risk and mitigating the risk appropriately is of critical importance for all the stakeholders. Security must be built at every layer on a Cloud Computing platform incorporating best practices and new technologies to effectively mitigate risk.

Cloud computing allows firms to outsource their entire information technology (IT) process, giving them the opportunity to focus more on their core business to increase their productivity and innovation in customer service. This allows businesses to reduce the heavy cost incurred on IT infrastructure without losing focus on customer needs (Priya *et al*., 2019).

3. **SECURE+, the protection technique in cloud computing**

The present paper aims to create a self-defense and autonomous mechanism against intrusions and cyber attacks (known and unknown) carried out on cloud computing platforms by proposing a technique with self protection and automatic interaction in cloud computing platform called SECURE+. It is an improved version of the SECURE technique by affecting important metrics such as intrusion detection rate (IDR), false positive rate (FPR) and utilization of computer resources (CPU, RAM and bandwidth). SECURE+ will create automatic signatures and provide security against DoS, DDoS, Probing, U2R and R2L security attacks (Kasongo *et al*., 2020). This study is based on the eXtreme Gradient Boosting method (XGBoost), a machine learning algorithm with a set of tree structure-based decisions, which uses a gradient reinforcement framework. The main motivation is the construction of a powerful classification model, which in cooperation with the SNORT system can classify the data entered into the network as accurately as possible, as quickly as possible and with the lowest possible use of available computer resources.

The present paper paper proves that XGBoost is the most suitable method to be used for defensive purposes as a robust classification model could be

built. This will lead to a more accurate intrusion detection system, and a more secure environment to share information on cloud platforms.

The SECURE+ technique helps to detect attacks by a combination of an attack detection system called SNORT and the XGBoost algorithm. SNORT is used to record known attacks on the database that this technique possesses (known attacks). While to detect abnormal activities (unknown attacks) will be used one of the newest techniques based on decision-making tree' algorithm of automatic learning called extreme gradient boosting (XGBoost) (Devan *et al*., 2020). This algorithm allows for the setup of a database which is called the training database, and designs XGBoost to identify and diagnose attacks from incoming network traffic data. The SECURE+ technique will automatically create a new signature and provide security against DoS, DDoS (UDP Flooding and NTP Amplification), probing, U2R and R2L type attacks.

This technique provides a system for detecting intrusion and avoiding computer attacks by way of improving the gradient according to the tree decision technique as in the Figure 1 depicted, automatically, performing an intelligent analysis of packet flow in the network, and being followed by avoidance actions, which are consistent with the decision-making components of intervention detection. The point-to-point detection of intrusion and evasion process is based on three basic applications called Creation of Characteristics, Gradient Enhancement, and Attack Avoidance.

We have chosen to use the gradient boosting algorithm as it is considered the most effective and valuable method in the case of structured data tasks (as is the case with our study).

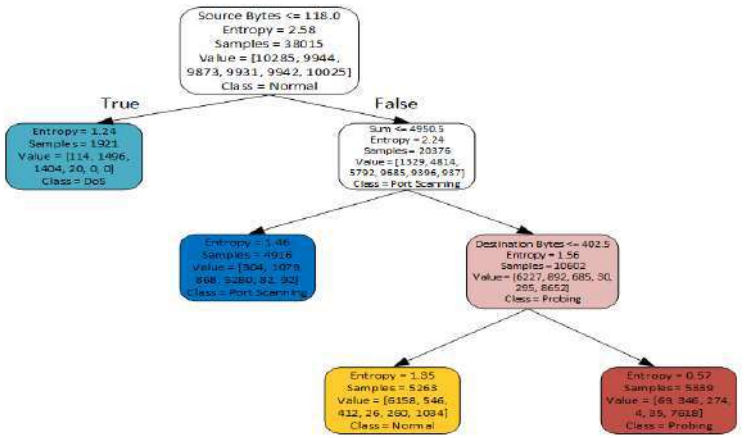


Fig. 16: Decision Tree Algorithm Example for gradient improvement during the task of Intrusion Detection on the network level.

4. **SECURE+ Architecture and Functionality**

The SECURE + technique architecture is in Figure 2 depicted, and its main components are as following:

- Cloud platform users submit their execution requests.
- All user requests are stored by a buffer called the service request handler. This buffer then forwards the workload to the Workload Administrator.
- The workload administrator distributes the workload along with their quality of service (QoS) requests to the Detection Node.

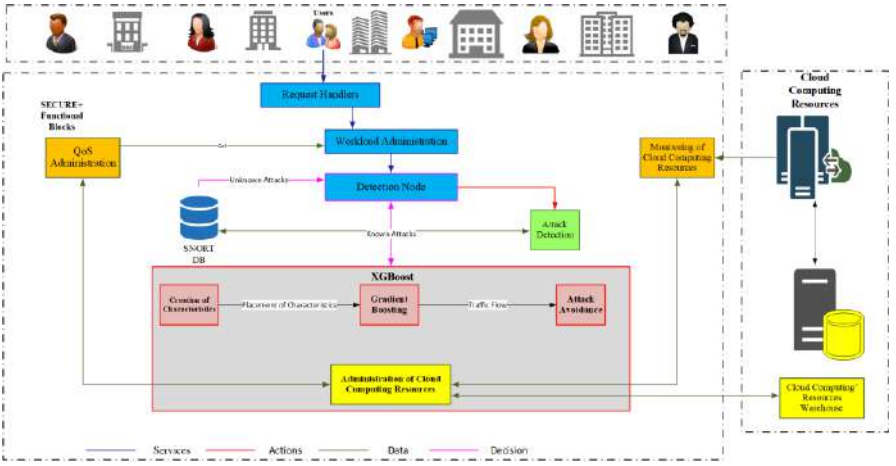


Fig. 2: Architecture of the SECURE+ technique.

- The detection node carries out a two-level defense for protection purposes. Performed by the SNORT intrusion detection application, the first level addresses the known attacks. Performed by the tree decision making (ML) machine learning algorithm, based on the latest XGBoost technique, the Second Level addresses the unknown attacks.
- The Resource Administrator keeps the resource information including the number of CPUs used, the RAM capacity used and the resource numbers. In addition, it stores information about available and reserved resources with respective descriptions (source name, source type, configuration, availability information, and usage information) according to the cloud service provider.
- The autonomous level consists of these three elements: i) creation of characteristics, ii) gradient improvement and, iii) attack avoidance.

- Resource Usage Monitoring which measures the resource usage value during workload execution.
- Cloud Resource Warehouse stores cloud resource configuration.
- Autonomous Attack Detection System Via Machine Learning Technique

The SECURE+ technique considers three steps regarding the autonomous detection and interaction system. The interaction of these subunits such as creation of characteristics, gradient enhancement / increase, and attack avoidance is in Figure 3 described.

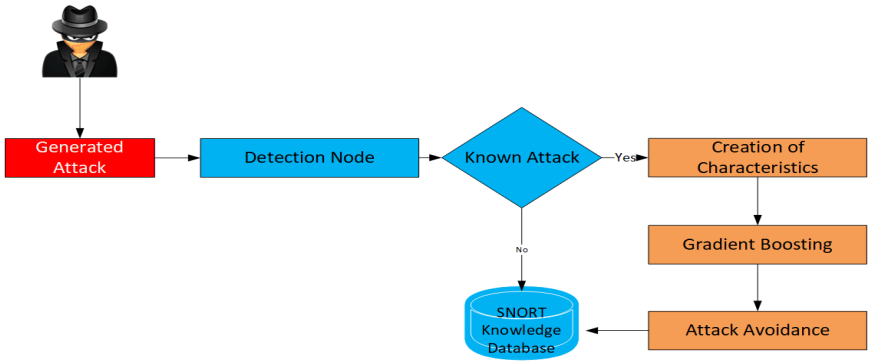


Fig. 3: Interaction of autonomous system's subunits.

The feature creator collects network traffic from the intrusion detection node in case the attack is unknown to snort and calculates the value required by the gradient enhancer for the respective data's flux.

The gradient enhancer applies its pre-built intervention detection models to the flow pattern and passes the result towards the Attack Avoider.

The attack avoider then determines the action to be taken, based on the classification result, and installs the flow rules in the attack detector to prevent the attack if necessary.

Figure 4 depicts the block diagram of the point-to-point function of the proposed security solution.

The detection node collects network traffic data and if they belong to unknown attacks, they are taken over by the feature creator. After acquisition, characteristics are created for each flow as summarized in the block diagram in Figure 5.

In this way common characteristics are generated for each stream, enclosing each incoming stream in the detection node, and using the block diagram shown in Figure 6, e.g., the average flow time and the total number of packets in a transaction are created with an initial switch overflow input.

Consequently, specific flow characteristics, e.g., the duration of the flow and the calculation of packets from source to destination are obtained by passing overflow inputs.

As we look at the stream inputs, the feature vector created for one stream is immediately sent to the Gradient Enhancer, without waiting for the feature creation for the other flows inputs to finish.

- The characteristics creator also draws flows matching fields, such as the source IP and MAC addresses, and the physical gateway from which the packet comes from. Attack avoidance uses these features as a conclusion. Common features include Mean, Stddev, Sum, TnP_PSrcIP, TnP_PDstIP, and TnP_Per_Dport. Hash groups are used to store unique source and destination IPs and destination port numbers. A list is implemented to save the duration of the input stream. As it surrounds the input stream, the number of packets in the input stream is added to the total packet calculation. The input stream duration is added to the duration list. Source IPs, destination IPs and destination port numbers are added to the respective hash group. The flow bit count is added to a hash map. The keys in this map consist of the source IP, the source port, the destination IP, and the destination port for the TCP and UDP packets.
- For ICMP packets, the keys consist of the source IP and the destination IP, unless in this case they do not have port numbers. This map is later used to obtain reverse flow statistics. Once the detection node is enclosed, the main characteristics are calculated using the total packet count, hash groups and duration list.
- The gradient enhancer takes feature vectors from the characteristics creator one by one and classifies them using its prefabricated intrusion detection model. If the result of the classification is any type of attack, then attack avoidant takes the type of detected attack and source identifiers such as. Source IP and source MAC address. The automated learning model used is dynamically updated by including new data learned about the same existing types of attacks as soon as they are detected. The gradient-enhanced model construction is a multi-class classification model, which is formed using learned data that exists in different types of attacks in addition to normal traffic.

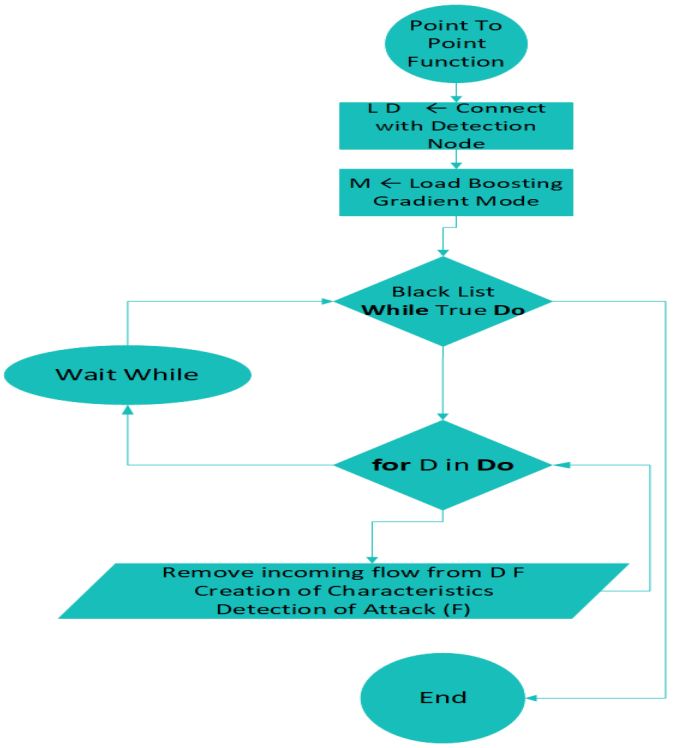


Fig. 4: Block Diagram of Point-to-Point' Function.

4.1. Description of Research Methodology

We designed the experiment scenarios to observe and obtain the measurement results. In the present investigation the performance between two rigorously repetitive and quantitative techniques is compared. The experiments are based on a test environment which compares the SECURE technique with the SECURE+technique.

This environment will have a 100Gbps bandwidth network and several different types of harmful traffic.

Harmful traffic types are selected because the default rules can be applied simultaneously to SECURE and SECURE+. Moreover, these are the most common types and cover a much larger number of attack's types.

The performed experiments will compare the performance of the two techniques by measuring the False Positive Rate (FPR), the Intrusion Detection Rate (IDR), the execution time, as well as the percentage of CPU, the use of RAM memory and the percentage of lost packets on the network.

Normal network traffic for conducting experiments is realized using an open-source network traffic generator called Hping3 (Hping, 2020). This application will generate network traffic up to 20 Gbps. Malicious traffic is generated using the following applications:

- Metasploit for DoS attacks,
- NMAP for Probing attacks,
- Hydra for R2L attacks,
- NetCat for L2R attacks
- DDoSIM for DDoS attacks.

Legitimate and malicious network traffic is generated as combined traffic and then entered into the detection node.

4.2 Realization of Experimental Work

Experiments were carried out to show the advantages of the SECURE+ technique versus the SECURE one using virtual machines with the same computer resources.

First, the experiment was carried out so that real-time observes the performance of the SECURE+ and SECURE techniques by processing a legitimate 10Gbps traffic from the legitimate traffic generator (Hping3) for comparison purposes.

1,470-byte packets for TCP, UDP and ICMP protocols were used for the accurate results of the experiment. We injected these packets in both techniques with a network speed of 10Gbps. The experiment is based on the logic grid diagram as in Figure 5 depicted.

We have installed each technique separately in identical virtual machines with the same parameters of computer resources and the same rules for the snort application. We used an application called Network Performance Monitor by Solarwinds (SolarWinds 2020), which records and measures CPU, memory, and network usage. In addition to this application, we have used several other applications such as Metasploit structure, Snort logs, nmap etc.to record and measure the features.

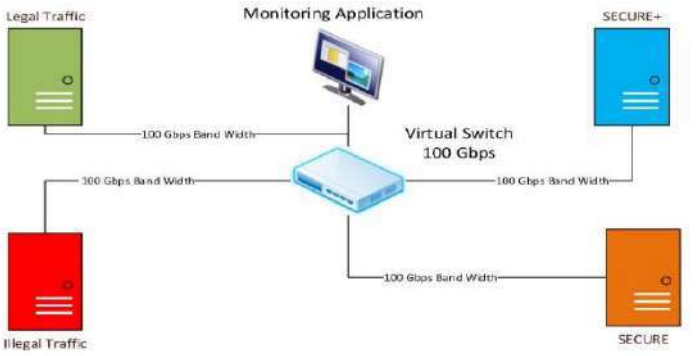


Fig. 5: Presentation of the experimental environment network.

5. EXPERIMENTAL RESULTS

The experiments were carried out in cycles lasting 8 hours each. In total we have performed 10 cycles of tests, and we have achieved a total duration of 80 hours of experiments, this to increase the reliability of our results. The following packets are injected as basic traffic ranging from 1Gbps to 10Gbps as follows:

- 1,000,000 UDP packets at a speed of 500 packets / second, each packet size consists of 1,470 bytes.
- 1,000,000 TCP packets at a speed of 500 packets / second, each packet size consists of 1,470 bytes.
- 1,000,000 ICMP packets at a speed of 500 packets / second, each packet size consists of 1,470 bytes.

Result analysis

The way we chose to inject the packets was the one with normal traffic, specifying the number of packets per second and the total number of packets.

The results showed that the use of CPUs in the SECURE+ technique was lower compared to that of the SECURE technique measured during the processing of the same network traffic of 10Gbps.

The Figure 6 gives the average CPU usage score for both techniques during the 80 hours of testing.

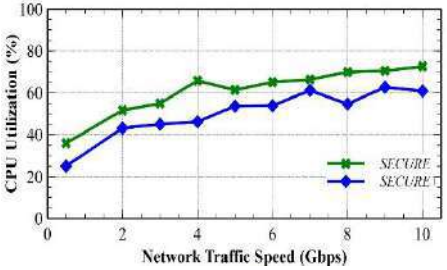


Fig. 6: Average CPU usage.

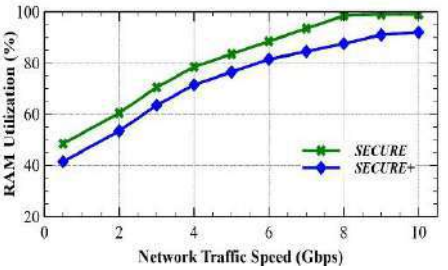


Fig. 7: Average RAM usage.

The collected performance data tells us that the memory usage in the SECURE+ technique is lower than that of the SECURE technique.

It is clear that the average memory usage in the case of the SECURE+ technique increases from 19GB when performing tests at 1Gbps and continues to increase at a variable rate up to a maximum of 30GB when performing tests with 10Gbps network speed. Graphically this usage is in Figure 7 depicted.

Processing packages in the SECURE+ technique are faster than processing packages in the SECURE technique as per Figure 8.

In other words, for the same amount of UDP, TCP and ICMP packets (1,000,000 packets) injected in both techniques over a period of 80 hours (10 test cycles of 8 hours each) we noticed that the SECURE+ technique showed an improved performance versus that of SECURE.

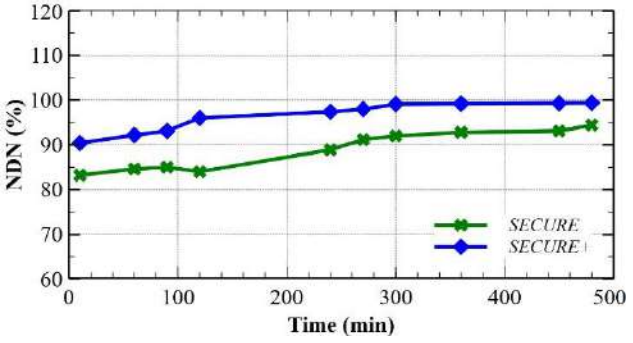


Fig. 8: Average processing speed (number of packets per unit time) for both techniques.

To compare the performance of SECURE and SECURE+ techniques, two most important metrics such as intrusion detection rate (IDR) and false positive rate (FPR) were investigated as they are also the most important criteria for evaluating algorithms and techniques.

The results obtained for these metrics were realized by generating attacks through Metasploit applications for DoS attacks, NMAP for probing attacks,

Hydra for R2L attacks, NetCat for L2R attacks and DDoSIM for DDoS attacks.

Figure 9 graphically shows the results obtained during the experiments performed for the mean values of false positive rate (FPR) for all categories of attacks (4 types of attacks, DoS, R2L, Probing and DDoS).

Mostly, the algorithms that have a low value of the IDR metric are not considered at all, and are not used, no matter how high the value of the FPR.

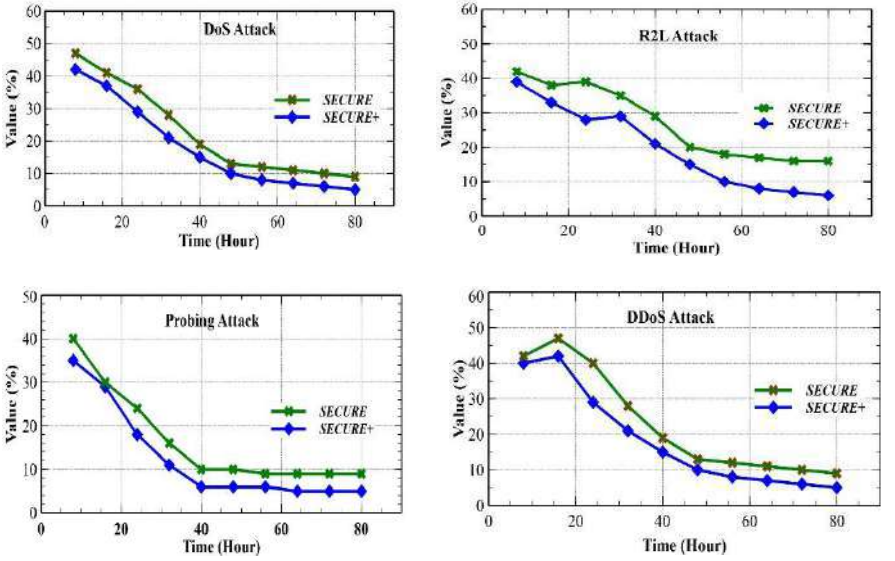


Fig. 9: Results of experiments for FPR for all types of attacks.

The Figure 10 depicts that the intrusion detection rate (IDR) increases with time. In this case we performed the same experiment divided into 10 cycles of 8 hours each.

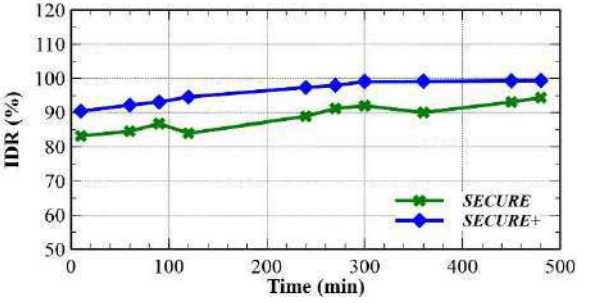


Fig. 10: Results of experiments for FPR for all types of attacks first and last cycles.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a technique with self-protection and automatic interaction in cloud computing platforms called SECURE+ is proposed. This self-defence and automatic technique detect intrusions and attacks (known and unknown) carried out on Cloud Computing platforms. The SECURE+ technique is built to detect attacks by a combination of an attack detection system called SNORT and a machine learning algorithm, called XGBoost. It functions as a single system of interventions where a series of snort rules operate in parallel with the logic applied by the tree-based automated learning algorithm XGBoost.

SECURE+ protects cloud computing platforms from five different types of security attacks including DoS, DDoS (also UDP Flooding and NTP amplification), Probing, U2R, and R2L attacks. Furthermore, we have tested the performance of the SECURE+ technique in terms of intrusion detection rate, execution time, false positive rate and use of computer resources.

The present paper compares two intervention detection techniques, the SECURE+ and SECURE are. Both techniques turned out to be efficient and high-performance detection systems. The results showed that the SECURE+ technique is more effective and uses less computer resources compared to the SECURE technique as it uses approximately 10% less processing resources (CPU) and about 7% less RAM. In addition, the SECURE+ technique processes a larger number of packets about 30% more packages per second compared to the SECURE technique.

The SECURE+ technique has a lower packet loss rate than the SECURE technique. In both techniques it was observed that the operating system was responsible for packet losses in case of increased traffic from 2Gbps to 10Gbps. At these network speeds, memory buffers were completely occupied by not being able to read packets within these buffers, so both techniques require more RAM in the case of high network capacities ranging from 2Gbps to 10Gbps. This phenomenon does not exist for low network speeds from 100Mbps to 1Gbps.

The results of the experiments showed that the SECURE+ technique has a false positive rate about 4% lower than the SECURE technique.

The other very important metric investigated was the intrusion detection rate, as it determines the efficiency of a detection system. Results showed that the value of IDR increased with increasing time reaching the maximum value of 98.8% during the 40th hour.

Considering the proposed and studied technique, realization and results obtained, we conclude that the following works can be based on the future for such technique:

- Realization of the SECURE+ technique in a real Cloud environment.

- Improved SECURE+ for identifying and preventing day-zero attacks.
- Improvement of SECURE+ to identify the rate of breach of service level agreement (SLA).
- Improved SECURE+ to work with some other parameters such as energy efficiency, scalability, etc.

REFERENCES

Barse Y, Agrawal D.2020. BotNet Detection for Network Traffic using Ensemble Machine Learning Method. *International Journal of Innovative Technology and Exploring Engineering* (IJITEE), **10(1):** 201-204. ISSN: 2278-3075. 100.1/ijitee.A81221110120 DOI: 10.35940/ijitee.A8122.1110120.

Buchanan J, Flandrin B, Macfarlane F, Graves R. 2016. A methodology to evaluate rate-based intrusion prevention system against distributed denial of service.

Devan P, Khare N. 2020. An efficient XGBoost–DNN-based classification model for network intrusion detection system. *Neural Computing and Applications*, **32**:12499–12514. SpringerLink. https://doi.org/10.1007/s00521-020-04708-x.

Hping. 2020. http://www.hping.org/.

Kasongo S.M, Sun Y.2020. Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. *Journal of Big Data*, **7:** 105 https://doi.org/10.1186/s40537-020-00379-6.

Khan N, Al-Yasiri A .2016. Identifying cloud security threats to strengthen cloud computing adoption framework. *Procedia Computer Science*, **94**: 485-490.

Podlodowski L, Kozlowski M. 2019. Application of XGBoost to the cyber-security problem of detecting suspicious network traffic events. 2019 IEEE International Conference on Big Data (Big Data), 10.1109/BigData47090.2019.9006586.

Priya MSS, Sahu BK, Kumar B, Yadav M. 2019. Network intrusion detection system using XG Boost. *International Journal of Engineering and Advanced Technology* (IJEAT), **9(1):** 4070-4073. ISSN: 2249 – 8958**.** 10.35940/ijeat.A1307.109119.

Ramachandra G, Iftikhar M, Khan FA. 2017. A comprehensive survey on security in Cloud Computing. *Procedia Computer Science*, **110**: 465-472. https://doi.org/10.1016/j.procs.2017.06.124. Elsevier B.V.

Roundup of Cloud Computing Forecasts and Market Estimates. 2015. http://www.forbes.com/sites/louiscolumbus/2015/01/24/roundup-of-cloud-computing-forecasts-and-market-estimates-2015/#56c0b0f0740.

Snort. 2020. https://www.snort.org/

SolarWinds 2020. Network Performance Monitor, https://www.solarwinds.com/network-performance-monitor