In the Name of God

Journal of
Information Systems & Telecommunication
Vol. 11, No.3, July-September 2023, Serial Number 43

Research Institute for Information and Communication Technology
Iranian Association of Information and Communication Technology
Affiliated to: Academic Center for Education, Culture and Research (ACECR)

Print ISSN: 2322-1437
Online ISSN: 2345-2773
Publication License: 91/13216
Editorial Office Address: No.5, Saeedi Alley, Kalej Intersection., Enghelab Ave., Tehran, Iran,
P.O.Box: 13145-799 Tel: (+9821) 88930150 Fax: (+9821) 88930157
E-mail: info@jist.ir , infojist@gmail.com
URL: www.jist.ir

Indexed by:
- SCOPUS www. Scopus.com
- Index Copernicus International www.indexcopernicus.com
- Islamic World Science Citation Center (ISC) www.isc.gov.ir
- Directory of open Access Journals www.Doaj.org
- Scientific Information Database (SID) www.sid.ir
- Regional Information Center for Science and Technology (RICeST) www.ricest.ac.ir
- Magiran www.magiran.com

Publisher:
Iranian Academic Center for Education, Culture and Research **(ACECR)**

This Journal is published under scientific support of
Advanced Information Systems (AIS) Research Group and
Telecommunication Research Group, ICTRC

Acknowledgement

JIST Editorial-Board would like to gratefully appreciate the following distinguished referees for spending their valuable time and expertise in reviewing the manuscripts and their constructive suggestions, which had a great impact on the enhancement of this issue of the JIST Journal.

(A-Z)

- Afsharirad, Majid, Kharazmi University,Tehran, Iran
- Azizi, Sadoon, University of Kurdistan, Sanandaj, Iran
- Badie, Kambiz, Tehran University, Iran
- Fadaeieslam, Mohammad Javad, Semnan University, Iran
- Fekri Ershad, Shervan, Islamic Azad University Najafabad, Isfahan, Iran
- Ghaffari, Hamidreza, Ferdous Azad University, South Khorasan Province, Iran
- Ghazalian, Reza, Babol Noshirvani University of Technology, Mazandaran, Iran
- Izadkhah, Habib, Tabriz University, Tabriz, Iran
- Khazaei, Mehdi, Kermanshah University of Technology, Kermanshah, Iran
- Kolahkaj, Maral, Islamic Azad University, Karaj Branch, Iran
- Kheirkhah, Fatemeh, ACECR, Tehran, Iran
- Kasaei, Shohreh, Sharif University, Tehran, Iran
- Marvi, Hossein, Shahrood University of Technology, Semnan Province, Iran
- Mirzaei, Abbas, Islamic Azad University, Ardabil, Iran
- Omid Mahdi, Ebadati, Kharazmi University, Tehran, Iran
- Qhasemzadeh Mohammad, Yazd University, Tehran, Iran
- Rasi, Habib, Shiraz University of Technology, Shiraz, Iran
- Razavi, Seyyed Mohammad, University of Birjand, South Khorasan Province, Iran
- Roshani, Saeid, Islamic Azad University, Kermanshah Branch, Iran
- Sable, Nilesh, Vishwakarma Institute of Technology, Pune, Maharashtra, India
- Shakibian, Hadi, Alzahra University, Tehran, Iran
- Soleimanian Gharehchopogh, Farhad, Islamic Azad University Urmia, Iran
- Tanha, Jafar, Tabriz University, Tabriz, Iran
- Wellem Anselmus Teniwut, Politeknik Perikanan Negeri Tual, Indonesia
- Yaghoobi, Kaebeh, Ale Taha Institute of Higher Education, Tehran, Iran

Table of Contents

- **Deep Transformer-based Representation for Text Chunking 176**

Parsa Kavehzadeh, Mohammad Mahdi Abdollah Pour and Saeedeh Momtazi

- **Deep Learning-based Educational User Profile and User Rating Recommendation System for E-Learning 185**

Pradnya V.Kulkarni, Rajneeshkaur Sachdeo ,Sunil Rai and Rohini Kale

- **Implementation of Machine Learning Algorithms for Customer Churn Prediction 196**

Manal Loukili, Fayçal Messaoudi, Mohammed El Ghazi and Raouya El Youbi

- **Recognition of Facial and Vocal Emotional Expressions by SOAR Model 209**

Matin Ramzani Shahrestani, Sara Motamed and Mohammadreza Yamaghani

- **Long-Term Software Fault Prediction Model with Linear Regression and Data Transformation 222**

Momotaz Begum, Jahid Hasan Rony, Md. Rashedul Islam and Jia Uddin

- **A survey on NFC Payment: Applications, Research Challenges, and Future Directions 232**

Mehdi Sattarivand, Shahram Babaie and Amir Masoud Rahmani

- **Content-based Retrieval of Tiles and Ceramics Images based on Grouping of Images and Minimal Feature Extraction 241**

Simin RajaeeNejad and Farahnaz Mohanna

- **Spectrum Sensing of OFDM Signals Utilizing Higher Order Statistics under Noise Uncertainty Environments in Cognitive Radio Systems 251**

Mousumi Haque and Tetsuya Shimamura

- **Trip Timing Algorithm for GTFS Data with Redis Model to Improve the Performance 260**

Mustafa Alzaidi and Aniko Vagner

Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 176-184
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Deep Transformer-based Representation for Text Chunking

Parsa Kavehzadeh[1], Mohammad Mahdi Abdollah Pour[1], Saeedeh Momtazi[1*]

Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran

Received: 27 Jul 2021/ Revised: 20 May 2022/ Accepted: 18 Jun 2022

Abstract

Text chunking is one of the basic tasks in natural language processing. Most proposed models in recent years were employed on chunking and other sequence labeling tasks simultaneously and they were mostly based on Recurrent Neural Networks (RNN) and Conditional Random Field (CRF). In this article, we use state-of-the-art transformer-based models in combination with CRF, Long Short-Term Memory (LSTM)-CRF as well as a simple dense layer to study the impact of different pre-trained models on the overall performance in text chunking. To this aim, we evaluate BERT, RoBERTa, Funnel Transformer, XLM, XLM-RoBERTa, BART, and GPT2 as candidates of contextualized models. Our experiments exhibit that all transformer-based models except GPT2 achieved close and high scores on text chunking. Due to the unique unidirectional architecture of GPT2, it shows a relatively poor performance on text chunking in comparison to other bidirectional transformer-based architectures. Our experiments also revealed that adding a LSTM layer to transformer-based models does not significantly improve the results since LSTM does not add additional features to assist the model to achieve more information from the input compared to the deep contextualized models.

Keywords: Text Chunking; Sequence Labeling; Contextualized Word Representation, Deep Learning; Transformers.

1- Introduction

Assigning appropriate labels to each particular token in text has always been a critical issue in Natural Language Processing (NLP). One of the most challenging tasks in sequence labeling is text chunking which entails detecting different phrases in unlabeled data. Finding distinct phrases in text could play an important role in various semantic and contextual analysis since each chunk usually contains a precious piece of information. This means that extracting the phrases from a particular corpus would endow the main idea and purpose of it. However, in spite of other sequence labeling tasks, detecting phrases in text is sometimes so sophisticated that it requires language experts' assistance.

Generally, phrases follow special semantic and syntactic patterns in text, which enables models to predict them automatically in large corpuses. To be more specific, linear statistical models like Hidden Markov Models (HMM) [10] or CRF [16] have been used frequently in sequence labeling tasks. There are many papers in previous years that combined CRF with more complicated models such as Long short-term memory (LSTM) [14] and Bidirectional LSTM (BiLSTM) [11]. Over the last couple of years, the emergence of transformer-based models has assisted researchers to take a huge step in handling various complicated NLP tasks more accurately and achieve state-of-the-art results on challenging datasets.

In this article, we employed state-of-the-art pretrained transformer-based models for sequence chunking. As these models are trained on a huge amount of textual data, they offer valuable contextual information that could be useful for various NLP tasks. We also use CRF and LSTM-CRF, a recent state-of-the-art model for sequence labeling tasks, after getting output from transformer-based models to evaluate the effect of classification module together with transformers. This research provides a comprehensive comparative study on the impact of transformer-based representations on chunking. Although different studies focused on text chunking, we still suffers from the lack of information from different perspectives: (1) the available models have not provided information about the differences between different representation models in text chunking, (2) in case of any difference, which type of representation performs the best is not clear, (3) in case of using any transformer-based representation model, it is not studied well if we still need the well-known LSTM-CRF architecture or we can benefit from a simpler module for classification.

In the rest of the paper, first, we introduce related works about sequence chunking in Section 2. Our model will be specified in Section 3. In Sections 4 and 5, we will

✉ **Saeedeh Momtazi**
momtazi@aut.ac.ir

elaborate the training procedure and experimental results. Finally, we will draw the conclusion in Section 6.

2- Related Works

Considering the sequential behavior of textual data, various NLP tasks, such as parts of speech tagging, named entity recognition and chunking work based on sequence labeling models.

Most traditional sequence labeling approaches use rudimentary language-specific methods [13].

Probabilistic graphical models, such as HMM and CRF have been widely used for these tasks. Different types of Recurrent Neural Networks (RNN) are more recent approaches that have played an important role in sequence labeling including chunking tasks. RNN was first designed by Rumelhard et al. [31] to carry the information of a sentence by passing through it. Hochreiter et al. [14] and Cho et al. [4] introduced more complicated versions of RNN, called LSTM and GRU, which enables the model to keep crucial information of certain tokens in sentence without the problem of vanishing or exploiting gradients occurred in previous RNNs. To benefit from the information of both sides of a sentence, BiLSTM was introduced by Graves et al. [11], enabling models to capture more information during training which achieved better results in the chunking task.

Ma et al. [23] and Huang et al. [15] used BiLSTM to obtain word representations with respect to both right and left context and a subsequent CRF layer to consider sentence level tag information. Huang et al. [15] also used SENNA [6] pre-trained embeddings. Moreover, in the proposed model of Ma et al. [23], a max-pooling and a convolutional layer were used to obtain character embeddings for each word. They also used the concatenation of character representations, linguistic features like POS and NER labels, and word embeddings to create a general embedding before feeding to BiLSTM.

An attention-based RNN was used for chunking by Li et al [19]. A context window was defined to generate embeddings and this assisted RNN to consider more local dependency. Attention component c_t helps to selectively obtain information from encoding layers instead of totally relying on a particular hidden state.

Attention segmental recurrent neural networks (ASRNN) was used by Lin et al. [20] for text chunking. The hierarchical architecture employed in their model helps the model to capture both character and word-level information from the text and also separate important and less information while building the segmental representation. Wei et al. [39] used a novel attention-based model called position-aware self-attention to extract both successive and discrete dependencies of each word in sequences.

There are many works employing semi-supervised and unsupervised learning to enhance their results. In the proposed model of Rei et al [30], a secondary unsupervised section for language modeling was used to enhance performance in sequence labeling tasks including chunking by learning more complicated features. Peters et al. [25] designed a bidirectional language model TagLM and trained it on unlabeled data. Then, they used pre-trained embeddings achieved from TagLM for the chunking task. Clark et al. [5] also proposed a semi-supervised approach working on both unlabeled and labeled data.

Wang et al. [38] employed a meta self-training approach for sequence labeling tasks in order to overcome the lack of annotated data challenge. Using meta self-training, they only needed to use a small amount of labeled data along a large unlabeled corpus, which helped their model to benefit from the information within the huge amount of unlabeled data.

Multi-task learning was another approach used by Liu et al. [21], Sogaard et al. [34], and Hashimoto et al. [13] to improve the performance of the model in chunking. Liu et al. [21] fine-tuned word-level pre-trained GLOVE [1] embeddings and also used character-level embeddings to build a language model alongside handling chunking. Sogaard et al. [34] used BiRNNs [33] for different sequence labeling tasks including shallow parsing by employing SENNA embeddings. Hashimoto et al. [13] devised a Joint Many-Task (JMT) model whose goal is handling different NLP tasks in a deep neural architecture. They also used two types of embeddings, Skip-gram [24] and character embeddings.

Zhai et al. [41] designed three models to handle segmentation and labeling tasks simultaneously. They also concatenated two embeddings: SENNA and embeddings gathered by adopting CNN on character embeddings of words to achieve final embeddings. Xin et al. [40] designed IntNet to learn the internal structure of words by their composing characters. Afterwards, in order to capture context information and handle sequence labeling and chunking, they feed these embeddings to LSTM-CRF.

Character-level language models have been also used to obtain highly contextualized word embeddings for sequence labeling tasks including chunking. After training the character-level language model, Akbik et al. [1] concatenated their own word embeddings to pre-trained GLOVE embeddings and passed them to a BiLSTM-CRF network.

Akhundov et al [2] combined byte embeddings extracted by Byte BiLSTM with word embeddings and fed the result to another BiLSTM to get word-level scores and use a CRF layer to handle sequence tags.

[1] http://nlp.stanford.edu/projects/glove/

Obviously, most well-known works on the chunking tasks are based on statistical models like CRF beside various kinds of recurrent neural networks such as LSTM, BiLSTM, as well as pre-trained static word embeddings which are appropriate to be fed into the aforementioned models.

Transformer-based models have also been studied in recent works in sequence labeling tasks. Although the proposed models by Tsai et al. [36] and Chawla et al. [3] benefit from state-of-the-art representation models, they only focused on one specific representation and a specific classification model. They did not compare the impact of different available models for each of the representation and classification components of text chunking. %However, the novel contextualized transformer-based models started with Vawsani et al. [37] which have revolutionized many NLP tasks in recent years have not been explored for chunking.

In the next section, we analyze the performance of the newest transformer-based pre-trained models in the chunking task by providing a comparative analysis on different models.

3- Models

Our approach consists of two major parts: (1) a pre-trained transformer-based model to capture contextual features of tokens, (2) a probabilistic graphical model and a neural network model, such as CRF and LSTM-CRF, which receives the output of pre-trained models to learn the labels and their dependency. In this part, we first briefly introduce the pre-trained models which build the former part of our architecture. Then we continue with the latter part.

3-1- Transformer-based Models

With the emergence of transformers [37], a great step has been taken in the NLP area for achieving outstanding results in different tasks. The unique architecture of transformers assists models to learn much more sophisticated contextual information from text and outperform embedding models like Word2Vec and ELMo [26], and other RNN based architectures like LSTM. Over the last couple of years, some special transformer-based architectures have been developed, which enhanced experimental results in various NLP tasks. We used those versions of transformer-based models that have between 300 to 500 million parameters in order to compare their performance in a relatively equal situation. Here, we describe the models that are used in our proposed architecture for shallow parsing.

3-1-1 BERT

Devlin et al. [9] designed a deep bidirectional transformers architecture pre-trained on a tremendous amount of unlabeled text, BookCorpus [42] and Wikipedia, for two specific tasks of masked language modeling and next sentence prediction. At the beginning stage, BERT sums three embeddings: token, segment, and position embeddings to create final input embeddings. It helps the model to consider the position of input tokens in sentence and the segmentation part of them before feeding them into the bidirectional transformer layers. In the masked language modeling pre-training task, the model is forced to predict some masked tokens in input sequence by the context of other inputs. Another pre-training task for BERT is next sentence prediction in which two sequential sentences were fed to the model and the model is expected to predict whether two sentences are contextually related or not. Pre-trained parameters could be fine-tuned for various NLP tasks such as sequence labeling by giving labeled data to the model.

3-1-2 XLM

Lample et al. [17] proposed a bidirectional transformer-based model trained on both supervised and unsupervised tasks. Like BERT, masked language modeling is the unsupervised objective for XLM which forces the model to predict some masked input tokens by considering other words in the sentence. Predicting the next token is another unsupervised task designed for XLM to be trained on monolingual data. By considering the previous input tokens in a sentence, the model is forced to predict the next token. They also trained the model by translation language modeling, a more flexible version of masked language modeling that uses multilingual parallel sentences in two different languages to predict masked tokens. For instance, to handle translation language modeling, the model could use the tokens in the French sentence to predict masked inputs in the English sentence. In this way, the model was forced to learn how to use translations for predicting masked words. XLM is another transformer-based model that encouraged us to be used for the shallow parsing task.

3-1-3 GPT2

GPT2 is a unidirectional transformer-based model first introduced by Radford et al. [27]. The purpose of GPT2 is predicting the next word which is called causal language modeling. Due to the special unidirectional architecture and causal language modeling task, GPT2 is an ideal model for text generation and predicting next words of a sentence by considering previous ones. The original GPT2 model is trained on 8 million web pages and contains 1.5 billion parameters. We chose the medium version of GPT2 as one of our transformer-based models to create

appropriate embedding for each word. It should be mentioned that the medium version of GPT2 consists of 345 million parameters, which has roughly the same number of parameters that large versions of other transformer-based models have.

3-1-4 RoBERTa

RoBERTa is introduced by Liu et al. [22] to improve some aspects of BERT. The RoBERTa architecture is completely similar to BERT. However, Liu et al. [22] designed some different scenarios during the pre-training phase of RoBERTa. The next sentence prediction objective is removed from pre-training targets since they proved that eliminating next sentence prediction would result in improving the model downstream task performance. Masked language modeling is kept as a major pre-training objective for the model. RoBERTa was trained on a larger amount of data rather than BERT including CC-News[1], OpenWebText[2], and STORIES [35] in addition to Book Corpus [42] and English Wikipedia data. RoBERTa also has some differences in implementation. Liu et al. [22] used larger batches for RoBERTa than BERT. RoBERTa uses a different tokenizing scheme than BERT called byte-pair encoding which is similar to GPT2's tokenizer. Similarities and differences between RoBERTa and BERT motivated us to opt RoBERTa as one of candidate transformer-based models in order to compare its performance on the chunking task with other pre-trained models specially BERT.

3-1-5 BART

Lewis et al. [18] proposed an architecture composed of a bidirectional transformer-based encoder (like BERT) and a unidirectional decoder (like GPT2). BART is pre-trained by five major tasks including masked language modeling, token deletion, token infilling, sentence permutation, and document rotation. In the masked language modeling task, like other models such as BERT, BART is forced to predict the masked words in the sentence. In the token deletion task, in contrast to masked language modeling, some tokens are deleted in a sentence and the model duty is to determine the positions of deleted tokens. Another innovative task, token infilling, is replacing spans of tokens with one single mask token and obliging the model to learn how many words are in that single masked token. Sequence permutation shuffles the sentences in the document and feeds them to the model. In the document rotation task, each time, a single token is randomly selected and the document is rotated in a way to set that particular token as the beginning token of the document. This task teaches the model to predict the starting of each document. Although the main goal of the BART model is text generation and its related tasks, other NLP tasks like sequence labeling and shallow parsing could be handled by employing BART as well. We decided to use the large BART as a candidate of state-of-the-art transformer-based models for our chunking task.

3-1-6 XLM-RoBERTa

XLM-RoBERTa was proposed by Conneau et al. [7]. They set masked language modeling for their model to predict masked tokens of the input. Lample et al. [17]'s model is employed to enhance the model in some particular issues. To be more specific, they designed the model to be multilingual and they trained it on 100 different languages, requiring a great amount of data. For this, they also built a huge dataset, CommonCrawl, containing 2 terabytes of text data from 100 languages. The similarities between XLM and XLM-RoBERTa and the huge multilingual data that it was trained on motivated us to include XLM-RoBERTa in our transformer-based models for text chunking.

3-1-7 Funnel Transformer

Funnel Transformer is a new-brand bidirectional transformer-based model proposed by Dai et al. [8]. It consists of two major parts, an encoder and a decoder. In encoder, there are pooling layers between transformer layers, reducing the size of the initial input and endowing lower computation cost to the whole model. In tasks like sentence summarization or text classification, using just the encoder part of the Funnel Transformer model could be sufficient. To handle token classification and sequence labeling tasks like chunking, designers added a decoder module to the model in order to resize the reduced input by upsampling from encoded layers. Similar to BERT, the objective of the Funnel Transformer is masked language modeling; i.e., the purpose of pre-training is predicting the masked tokens in input sequence. Consequently, this unique architecture makes Funnel Transformer an appropriate option for sequence labeling tasks.

3-2- Learning with LSTM and CRF

After receiving the outputs from transformer-based models, we fed them to a sole CRF layer or a sequence of LSTM-CRF layers. CRF considers past and future labels in a sequence to predict the label of a particular token. CRF computes the best possible tag sequence between all possible sequences by minimizing the objective function presented in Equation 1.

$$E = -s(y) + \sum_{\tilde{y} \in Y} e^{s(\tilde{y})} \quad (1)$$

[1] http://commoncrawl.org/2016/10/newsdataset-available
[2] http://Skylion007.github.io/OpenWebTextCorpus

where Y represents all possible tag sequences and the goal is to find the sequence that minimizes the formula for a given input sequence.

CRF has two matrix parameters $A^{\{k \times k\}}$, $P^{\{n \times k\}}$, where k represents the number of tags and n is the length of each sequence. A is transition scores between different tags and P determines the probability of each tag in a position. The score of a tag sequence $y = \{y_1, \dots, y_n\}$, $y_i \in \{1, \dots, K\}$ and a given input sequence $x = \{x_1, \dots, x_n\}$ is calculated based on Equation 2.

$$s(x, y) = \sum_{i=1}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \qquad (2)$$

We also study another architecture, LSTM-CRF, which has been a great success story in sequence labeling tasks, such as parts of speech tagging, named entity recognition, and text chunking for several years [15,40,23,2].

We passed the outputs of transformer-based models to LSTM to observe the impact of combining recurrent neural models with state-of-the-art transformers.

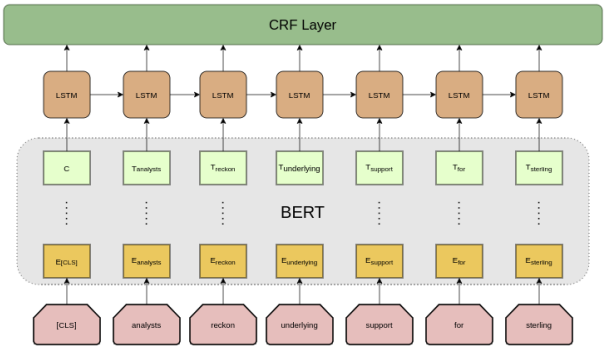


Fig. 1 BERT LSTM-CRF model. After passing through the BERT transformer-based model, token embeddings are given to a LSTM-CRF layer to detect their chunk labels.

Figure 1 shows the combination of one of our transformer-based models, BERT, with a LSTM-CRF layer. In the first stage, a pre-trained BERT model with multi-layer bidirectional transformer-based architecture converted tokens to fine input embeddings and passed them through the transformer layers. We feed the outputs of BERT to an LSTM layer to evaluate how the combination of LSTM and BERT could affect the quantity and the quality of information extracted from input. At the last layer, CRF is utilized to enhance the model performance in predicting output chunk labels by considering the dependency between chunk tags.

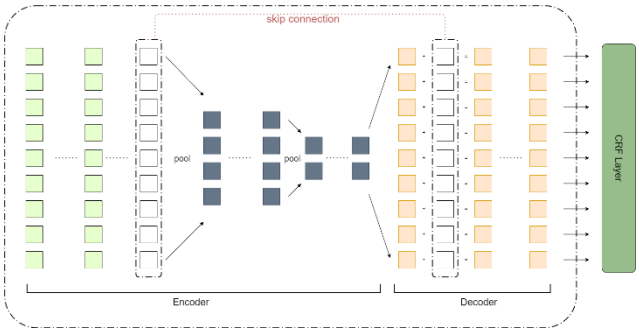


Fig. 2 Funnel Transformer CRF model. After giving tokens to an encoder-decoder transformer-based architecture, outputs will be given to a CRF layer in order to predict chunk labels.

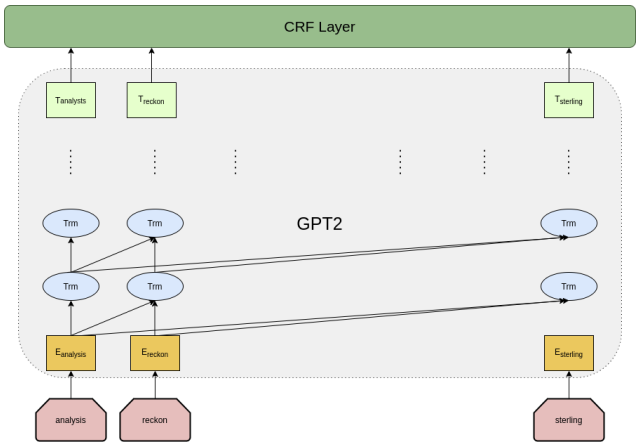


Fig. 3 GPT2 CRF model. Unidirectional transformer-based architecture of the GPT2 model caused a relatively poor performance against other transformer-based models BERT, XLM, and Funnel Transformer.

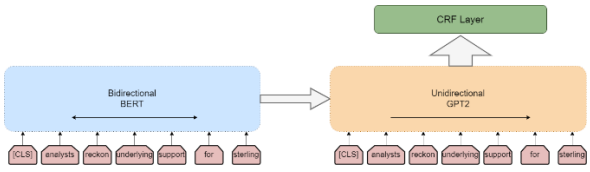


Fig. 4 Bart CRF model. Bart is composed of two parts, a bidirectional encoder (BERT) and an unidirectional decoder (GPT2). For fine-tuning, uncorrupted inputs should be given to both encoder and decoder and final hidden states of the decoder should be considered as output.

In Figure 2, the encoder-decoder architecture of Funnel Transformer is shown. By passing through the encoder section, pooling layers gradually reduce the dimension of transformer layers to encode the information in relatively smaller size. The decoder part resizes the encoded information to their original size, which makes the model appropriate for token classification tasks such as chunking

and named entity recognition. At the final stage, we use a CRF layer for predicting target labels.
Figure 3 shows the combination of Open AI GPT2 model with a CRF layer to predict chunk sequences. The unidirectional transformer-based architecture of GPT2 is also shown in the diagram. Although this characteristic has made GPT2 a great smart in text generation, we will see in the next section how this architecture affects the performance of the model in chunking.
In Figure 4, BART-CRF model is shown. Two major parts of BART, bidirectional encoder and unidirectional decoder, are very important during pre-training of the model. To handle the span masking objective, corrupted inputs were fed to the encoder (BERT) and uncorrupted to the decoder (GPT2). Then, BART had to learn masked spans by the information that bidirectional and unidirectional architectures of BERT and GPT2 represent.

4- Experiments

4-1- Dataset

We evaluate our approaches on the standard chunking dataset from CoNLL2000 [32] which contains 11 distinct phrase types. The detailed statistics of the dataset is presented in Table 1. Chunk labels in CoNLL2000 are based on the IOB scheme introduced by Ratnaparkhi et al. [29]. In the next section, we explain how we changed the IOB scheme to IOBES [28]. In Table 2, we present one example of each type of phrase existing in CoNLL2000.

Table 1: Size of sentences, tokens, and labels in CoNLL2000 datasets

train	# of sentences	8936
	# of tokens	211727
test	# of sentences	2012
	# of tokens	47377
	# of labels	22

Table 2: One phrase example for every label existing in the CoNLL2000 dataset.

Phrase	***Label***
their current 15% level	B-NP I-NP I-NP I-NP
has been eroded	B-VP I-VP I-VP
because of	B-PP I-PP
even though	B-SBAR I-SBAR
similar and conservative	B-ADJP I-ADJP I-ADJP
at least	B-ADVP I-ADVP
have to serve	B-VP I-VP I-VP
Good morning	B-INTJ I-INTJ
as well as	B-CONJP I-CONJP I-CONJP

4-2- Setup of Experiments

Due to the many reported experiments, replacing the IOB scheme with IOBES would yield better final accuracy [25,19,2]. As a result, we decided to use the IOBES scheme to evaluate the impact of transformer-based models for sequence chunking. In addition to regular Outside (O), Inside (I), and Beginning (B) labels, we added two End (E) and Single (S) tags, which contain more detailed information in terms of the labeling scheme. To this aim, in the IOB format, in case of having B label with no I label, we converted the B label to S. Also the last I label in each chunk is converted to E label.
For all experiments, we trained our model on the train set and reported the final results on the test set. Since CoNLL2000 does not give an explicit validation set, we randomly selected 10% of the sentences from train data as our validation set.
To employ pre-trained models and build CRF and LSTM-CRF layers, we used the transformers package [1] and Pytorch[2]. Due to the purpose of BERT, XLM and Funnel Transformer pre-trained models, TokenClassification module has been already implemented for these models and we used it to generate embeddings of input tokens. We also manually add the TokenClassification module to GPT2 and BART models to see their performance on the chunking task.
We set the probability of the dropout layer to 0.3, 0.4, 0.5 and injected it between the pre-trained model and target CRF or LSTM-CRF layer. Adam optimizer was used for training parameters of CRF and LSTM-CRF layers and fine-tuning the parameters of pre-trained transformer-based models. Learning rate was fixed in 1e-5 during the training process. Max length of sentences was fixed to 110 and batch-sizes were set to 4, 8, 16 regarding the pre-trained model used to split data during training.

[1] https://huggingface.co/
[2] https://pytorch.org/

4-3- Experimental Results

Table 3 presents the results of all described models. Our BERT-CRF model achieved 96.72 F1 score and BERT-LSTM-CRF received 96.70 F1 score. RoBERTa's performance on shallow parsing was slightly better. RoBERTa-CRF model's F1 score is 96.82 and RoBERTa-LSTM-CRF layer achieved 96.84. The performance of XLM on the chunking task outperformed our BERT-based models despite their similar bidirectional transformer-based architecture except XLM's translation language modeling objective during pre-training. XLM-CRF and XLM-LSTM-CRF models both achieved 96.83 F1 scores.

Table 3: One phrase example for every label existing in the CoNLL2000 dataset.

Pre-trained Models CRF LSTM-CRF	***CRF***	***LSTM-CRF***	***Dense Layer***
BERT	96.72	96.70	96.07
RoBERTa	96.82	96.84	96.34
XLM	96.83	96.83	96.50
XLM-RoBERTa	**96.92**	**96.92**	96.62
Funnel Transformer	96.83	96.53	96.50
BART	96.30	96.03	93.58
GPT2	85.00	84.46	82.57

XLM-RoBERTa achieved the highest scores between our transformer-based models. Both XLM-RoBERTa-CRF and XLM-RoBERTa-LSTM-CRF models received 96.92 F1 score. Funnel Transformer with its unique encoder-decoder transformer-based architecture achieved 96.83 F1 score in combination with CRF layer. The Funnel-LSTM-CRF model achieved 96.53 F1 score on IOBES mode.

The worst performance is reported based on GPT2. Due to the unidirectional architecture of GPT2, the model is adapted to learn left-to-right context of the text and its main goal is for text generation tasks; while in sequence labeling models, need to obtain left-to-right and right-to-left contextual information in order to achieve an acceptable result. The GPT2-CRF and the GPT2-LSTM-CRF models achieved 85 and 84.46 F1 scores, respectively.

Likewise, the unidirectional architecture of BART's decoder results in a relatively lower F1 score regarding bidirectional transformer-based models BERT, RoBERTa, XLM, XLM-RoBERTa, and Funnel Transformer. The results of BART, however, are significantly better than GPT2. The BART-CRF model achieved 96.30 and the BART-LSTM-CRF model result is 96.03.

We also tried a third possible architecture with a dense layer at the final stage. In this scenario, BERT achieved 96.07, RoBERTa's score is 96.34 and both Funnel Transformer and XLM received 96.50 and XLM-RoBERTa achieved 96.62, higher than others in this mode. BART achieved 93.58 and GPT2's score is 82.57.

As we expected, adding a LSTM layer to our CRF target layer not only did not improve our results, but caused them to drop in most cases. Since contextualized transformer-based models are stronger in extracting context information of text, adding a LSTM layer could not improve the ability of our model in capturing more information. In other words, the LSTM layer models similar information as transformers and it does not capture additional information compared to transformers. Moreover, considering the more advanced architecture of transformers in sequence modeling, using an LSTM network besides a transformer does not provide any benefit in the architecture. On the other hand, the CRF layer captures another type of information from the sequences of words by considering the relation between labels. As a result, models with just one CRF layer mostly outperformed their LSTM-CRF counterparts. Better results of the models with CRF in comparison with the models with just a dense layer in output was another expected observation as CRF has valuable information about the dependencies between tags which is not available in the other parts of the architecture.

In the next step, we compared our results with the state-of-the-art models in the literature as presented in Table 4. As can be seen, our model outperforms state-of-the-art models in the field, except the proposed model by Clark et al. [5] which is marginally better than ours. The main reason is that this model works based on multi-task learning and benefits from training data from other sequence modeling approaches in text chunking.

Overall, by comparing various transformer-based models and different architectures that are used after the embedding part, the following observations from the reported results are notable:

- Using novel transformer-based pre-trained models enhances the overall F1 score on text chunking due to the fact that they endow the precious information that they had learned by being trained on a huge amount of data to the model.
- Different pre-trained models which benefit from different architectures (except GPT2) have close performances on chunking and it is difficult to define an absolute winner.
- Despite other transformer-based models, GPT2's performance was relatively poor on text chunking, due to the fact that although the unidirectional architecture of GPT2 has made it a perfect model for text generation, it causes GPT2

to achieve lower results because predicting chunk labels require information from both sides of a particular token.

- A CRF layer in the final stage assists the contextualized pre-trained embedding models to consider the labels of other tokens when they are predicting the label of each particular word and enhances the overall performance.
- In the models in which contextualized pre-trained models were used, adding an LSTM layer does not improve the performance of the model, which could imply that removing LSTM could be a wiser decision since it does not help the model to capture more from the input tokens.

Overall the best results achieved by the XLM-RoBERTa model followed by a CRF layer, which achieved superior results compared to the state-of-the-art models in the field.

Table 4: One phrase example for every label existing in the CoNLL2000 dataset.

Model	***F1 score***
Collobert et al. [2011]	94.32
Huang et al. [2015]	94.46
S_gaard and Goldberg [2016]	95.28
Rei [2017]	93.88
Zhai et al. [2017]	94.72
Liu et al. [2017]	95.96
Peters et al. [2017]	96.37
Xin et al. [2018]	95.29
Akbik et al. [2018]	96.72
Clark et al. [2018]	97.00
Akhundov et al. [2018]	94.74
Lin et al. [2021]	93.70
Wei et al. [2021]	95.15
Ours: XLM-RoBERTa + CRF	96.92

5- Conclusions

We provided various architectures based on state-of-the-art transformer-based models for the chunking task. Well-known CoNLL2000 dataset was used for evaluating our models by F1 score. We compared our models' performance with other works done for sequence labeling tasks. Most previous models employed different types of RNNs such as LSTM and BiLSTM, CNN and character and word embeddings. We used novel transformer architectures like Funnel Transformer, XLM-RoBERTa, and BART as well as BERT, RoBERTa, XLM, and GPT2 to evaluate the effect of contextual embeddings and the combination of them with CRF and LSTM-CRF layers in shallow parsing.

In future, we will use the transformer-based models for other similar sequence labeling tasks like name entity recognition and parts of speech tagging. Due to the similarity of most sequence labeling tasks and the models that have been proposed for them, combining transformer-based models with previous state-of-the-art models could yield significant performance specially on rare languages.

References

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649, 2018.

[2] Adnan Akhundov, Dietrich Trautmann, and Georg Groh. Sequence labeling: A practical approach. arXiv preprint arXiv:1808.03926, 2018.

[3] Avi Chawla, Nidhi Mulay, Vikas Bishnoi, and Gaurav Dhama. Improving the performance of transformer context encoders for ner. In 2021 IEEE 24th International Conference on Information Fusion (FUSION), pages 1–8. IEEE, 2021.

[4] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.

[5] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. arXiv preprint arXiv:1809.08370, 2018.

[6] Ronan Collobert, Jason Weston, L´eon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. Journal of machine learning research, 12(ARTICLE):2493–2537, 2011.

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116, 2019.

[8] Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. arXiv preprint arXiv:2006.03236, 2020.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[10] Sean R Eddy. Hidden markov models. Current opinion in structural biology, 6(3):361–365, 1996.

[11] Alex Graves and Jurgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural networks, 18(5-6):602–610, 2005.

[12] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. arXiv preprint arXiv:1611.01587, 2016.

[13] Zhiyong He, Zanbo Wang, Wei Wei, Shanshan Feng, Xianling Mao, and Sheng Jiang. A survey on recent advances in sequence labeling from deep learning models. arXiv preprint arXiv:2011.06727, 2020.

[14] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[15] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991, 2015.

[16] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[17] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291, 2019.

[18] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, 2019.

[19] Bofang Li, Tao Liu, Zhe Zhao, and Xiaoyong Du. Attention-based recurrent neural network for sequence labeling. In Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, pages 340–348. Springer, 2018.

[20] Jerry Chun-Wei Lin, Yinan Shao, Youcef Djenouri, and Unil Yun. Asrnn: a recurrent neural network with an attention model for sequence labeling. Knowledge-Based Systems, 212:106548, 2021.

[21] Liyuan Liu, Jingbo Shang, Frank F Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han. Empower sequence labeling with task-aware neural language model. arXiv preprint arXiv:1709.04109, 2017.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[23] Chunping Ma, Huafei Zheng, Pengjun Xie, Chen Li, Linlin Li, and Luo Si. Dm nlp at semeval-2018 task 8: neural sequence labeling with linguistic features. In Proceedings of The 12th International Workshop on Semantic Evaluation, pages 707–711, 2018.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.

[25] Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. arXiv preprint arXiv:1705.00108, 2017.

[26] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1802.05365, 2018.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

[28] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), pages 147–155, 2009.

[29] Adwait Ratnaparkhi. A linear observed time statistical parser based on maximum entropy models. arXiv preprint cmp-lg/9706014, 1997.

[30] Marek Rei. Semi-supervised multitask learning for sequence labeling. arXiv preprint arXiv:1704.07156, 2017.

[31] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. nature, 323(6088):533–536, 1986.

[32] Erik F Sang and Sabine Buchholz. Introduction to the conll-2000 shared task: Chunking. arXiv preprint cs/0009008, 2000.

[33] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. IEEE transactions on Signal Processing , 45(11):2673–2681, 1997.

[34] Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 231–235, 2016.

[35] Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. arXiv preprint arXiv:1806.02847, 2018.

[36] Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. Small and practical bert models for sequence labeling. arXiv preprint arXiv:1909.00100, 2019.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[38] Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. Meta self-training for few-shot neural sequence labeling. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pages 1737–1747, 2021.

[39] Wei Wei, Zanbo Wang, Xianling Mao, Guangyou Zhou, Pan Zhou, and Sheng Jiang. Position-aware self-attention based neural sequence labeling. Pattern Recognition, 110:107636, 2021.

[40] Yingwei Xin, Ethan Hart, Vibhuti Mahajan, and Jean-David Ruvini. Learning better internal structure of words for sequence labeling. arXiv preprint arXiv:1810.12443, 2018.

[41] Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. Neural models for sequence chunking. arXiv preprint arXiv:1701.04027, 2017.

[42] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision, pages 19–27, 2015.

Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 185-195
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Deep Learning-based Educational User Profile and User Rating Recommendation System for E-Learning

Pradnya V. Kulkarni[1*], Dr. Rajneeshkaur Sachdeo[2], Dr. Sunil Rai[2], Dr. Rohini Kale[3]

[1].Department of Computer science and Engineering, MIT School of Engineering, MIT ADT University, India and School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune,India.
[2].Department of Computer science and Engineering, MIT School of Engineering, MIT ADT University, India.
[3].School of Polytechnique, Dr. Vishwanath Karad MIT World Peace University, Pune,India.

Received: 26 Sep 2021/ Revised: 04 May 2022/ Accepted: 25 Jun 2022

Abstract

In the current era of online learning, the recommendation system for the eLearning process is quite important. Since the COVID-19 pandemic, eLearning has undergone a complete transformation. Existing eLearning Recommendation Systems worked on collaborative filtering or content-based filtering based on historical data, students' previous grade, results, or user profiles. The eLearning system selected courses based on these parameters in a generalized manner rather than on a personalized basis. Personalized recommendations, information relevancy, choosing the proper course, and recommendation accuracy are some of the issues in eLearning recommendation systems. In this paper, existing conventional eLearning and course recommendation systems are studied in detail and compared with the proposed approach. We have used, the dataset of User Profile and User Rating for a recommendation of the course. K Nearest Neighbor, Support Vector Machine, Decision Tree, Random Forest, Nave Bayes, Linear Regression, Linear Discriminant Analysis, and Neural Network were among the Machine Learning techniques explored and deployed. The accuracy achieved for all these algorithms ranges from 0.81 to 0.97. The proposed algorithm uses a hybrid approach by combining collaborative filtering and deep learning. We have improved accuracy to 0.98 which indicate that the proposed model can provide personalized and accurate eLearning recommendation for the individual user.

Keywords: E-Learning; Recommendation System; Machine Learning; Deep Learning**.**

1- Introduction

To improve eLearning recommendation systems, it is critical to offer relevant data [12],[14],[17]. A customized eLearning recommendation system [11] is necessary to promote the acceptability of eLearning among students in all parts of India. Some of the drawbacks or issues in the recommendation framework are a cold start, reliable information would not be provided, searching from vast data, similar phrases are used for different reasons, and data relevance [25]. We are attempting to solve these concerns in our work.

Online education [1] is now widely utilized around the world, allowing young people to learn at their own pace while being safe at home. To satisfy the necessity, the proposed framework is introduced. The data science application for education is this type of recommendation system.

Information overload, adequate information, and reliable information delivered to learners are currently difficulties in eLearning [10]. A standard eLearning site should be available to students. eLearning has been around for a long time, and it provides a lot of benefits for students. It is founded on teaching values. The system presented in this study will aid in distance education.

Online degrees, distance learning courses, and online certifications are offered by many schools, universities, enterprises, and organizations throughout the world to promote eLearning. Many online courses and certifications are offered and announced by open courseware for MIT, Learndirect.com, NPTEL (National Enhanced Learning Technology Program), MOOC NPTEL (Massive Open Online Courses) [4].Companies, on the other hand, as they get to this higher level of personalization, increase the amount of information that buyers must process before

✉ **Pradnya V.Kulkarni ,**
pradnyav.kulkarni@mitwpu.edu.in

selecting which goods match their demands. The suggested approach is one of the answers to the problem of knowledge overload.

The majority of the decision-making in the recommendation framework is based on historical data or user's information.[15][25]. The recommendation framework has proposed a lot of work since the mid-1990s. Students who use eLearning are well aware that proposing systems is both a learning opportunity and a challenge [10]. The recommendations in the eLearning recommendation system are based on user feedback. There are two perspectives on this. The system is partially converted to manual by forcing students to rate courses; second, the system is automated by inferring learning from students' study patterns or click-stream activity. The suggested rating scores are expressed as a ratio of real study hours to total course hours and are converted to explicit rating values on a scale of 1 to 5.

The objectives of this paper are:

1.Proposal of a novel recommendation system model for online course recommendation with a hybrid approach consisting of the collaborative learning and deep learning approach.
2.To improve accuracy as compared with the current approaches.

The remainder of the paper is organized as follows. Section 2 defines the related work. Section 3 explains the proposed model and methodology section 4 explains experimentation and results. Section 5 concludes the paper and its future scope.

2- Related Work

In the paper [1], the author, Dr. Wahab Ali, suggested that the teachers who are using online learning platforms for teaching students should use technology and instruments to promote, during exceptional times mainly. He called for online and remote learning during the Pandemic as a necessity during lockdowns and social isolation.

The authors of the paper [6] suggested different approaches to analyzing trainers' and trainees' understanding of digital learning. Also, the authors addressed the technique of execution of digital learning. Their focus is on the active learning mode of the students during the digital learning phase, which includes the creative performance of the students and education system during the outbreak of COVID-19, to overcome the confusion of ongoing academic disruption and bring back the educational activities in place.

A hybrid real-time incremental stochastic gradient descent (RI-SGD) technique for implicit feedback Matrix Factorization recommendation system was developed in the study [16]. The study's numerical data revealed that the RI-SGD approach could reach almost the same recommendation accuracy. The proposed method can be utilized to improve the accuracy of various collaborative filtering algorithms in the future.

This paper [30] introduces a method that combines knowledge graph and collaborative filtering. They used the semantic information of the items and fused it with a collaborative filtering algorithm to recommend the course. The side information like social media, multimedia is used as parameters. User interests have not been considered herein our model we are considering the User Interests.

The Educational Data Mining technique is employed to handle academic management problems in this paper [31], and the Singular Value Decomposition (SVD) classifier is used to provide course information for students' optional course recommendations based on their grades. The accuracy of the advice is worse here because SVD has an issue with data density. The accuracy can be increased using the proposed method.

In this paper [31] Educational Data Mining algorithm is used to solve the academic management problems and a Singular Value Decomposition (SVD) classifier is used for giving course information for a recommendation of elective courses to the students based on their grades. Accuracy of recommendation is less here as SVD has a problem with the density of the data. Using our method, the accuracy is improved.

The authors of this paper [23] employed the Apriori and SPADE algorithms to select courses based on course history similarities. We increased the accuracy by combining the user's profile with the course ratings provided by users.

Similarly, the challenges in eLearning Recommendation Systems [13] are:

1. Existing work shows that User profile-based recommendation systems for eLearning are having an accuracy of less than 0.98. The comparison with the existing systems is shown in Table 1.
2. Current results have not utilized all the available information like mining users and item profiles, implicit feedback, contextual information, and review texts.
3. Scalability for the recommendation system is one of the problems.
4. Taking User's interests into consideration and Users' Ratings is also required for better accuracy in a recommendation.

In contrast to most current eLearning recommendation systems, such as [3],[6],[12], which only

used explicit feedback rating values in their recommendation method, the proposed approach considers multidimensional attributes of an education-based user profile and rating values in its recommendation method. The proposed system uses a user profile and ranking based on education for recommendation purposes, as well as deep learning [26][28] to improve recommendation accuracy.
To demonstrate the efficacy of the proposed method, it was tested against existing recommendation algorithms. Precision, recall, F1-score, and accuracy [9] are employed as performance indicators.

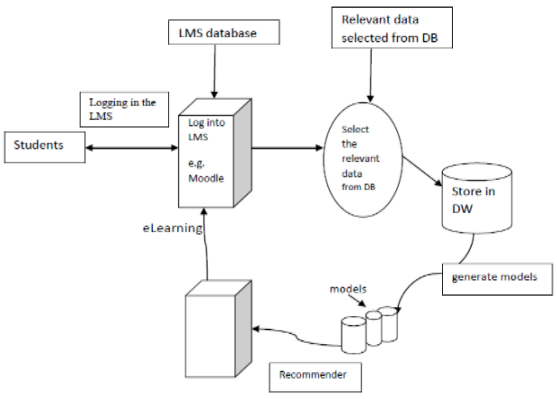
3- Proposed Methodology

The system architecture depicted in Fig.1 accepts User Profile data from Institutes or educational institutions as input data. The data is stored in the dataset through Learning Management Systems (LMS). The relevant data stored in the database is considered for the recommendation model. The information seeker's attention is drawn to the precise profiling technique [3]. User profiling [19] is used to develop recommendations for knowledge seekers using a hybrid approach that combines collaborative filtering with content-based filtering [16]. The proposed algorithm is as:

Algorithm:

Input: User Profile and User Rating Dataset
Output: Course Recommendation
1. Dataset is built from Engineering Students Profile.
2. Do stemming the field keyskills.
3. Tokenize the keyskills generated from step 2.
4. Remove the predefined frequent words,from tokenized keyskills.
5. Use TF-IDF to categorize and filter courses.
6. Extract Feature Vector, apply K-means algorithm and find out clusters.
7. Use the deep learning method to train the model.
8. Perform testing on new user profiles and predict the course recommendation list for the user.

Fig.1 System Architecture



3-1- Dataset

The data has been collected from the student profiles of Dr. Vishwanath Karad, MIT World Peace University (MITWPU), Pune. The data is collected from students through the Calyxpod tool. This data is for the year 2019-2020 for 5 different programs as mentioned in Table 2. Different User profile parameters considered are User Id, Specialization, Gender, Known Languages, Internship Experience, Projects, keyskills, Career Objectives, Certification.
As per the literature study, the parameters used for recommendations are students' grades, historical data, or User profiles [23]. We've used User Profile to determine the user's course requirements. Users Rating is also considered for recommendations. We were able to determine the course's quality as well as the user's grasp of the course by using ratings. Currently, this study only focuses on Engineering students. Engineering courses are fast and vast. For slow learners, the online courses will help to understand and speed up their studies. Further, this can be applied to another program as well.

Table 1: Summary of methods used for Course Recommendation System

Sr.No.	Authors and Year	Methodology	Parameters Used	Performance Measure
1	Zammer Gulzar, A.Anne Leema,Rerard Deepak(Dec 2017)[7]	Hybrid techniques are used N-Gram and Query Expansion approach and upgrade basic information retrieval technique along with the support of Ontology	User's personal information	Accuracy=0.95
2	Dussadee Praserttitipong and Wijak Srisujjalertwaja(Dec 2018)[31]	Education Data mining (EDM) and Singular Value Decomposition (SVD)	Student Grades	Accuracy=0.67
3	Raghad Obeidat, Rehab Duwairi, Ahmad Al-Aiad(Feb 2019)[23]	Apriori algorithm association rule mining and SPADE algorithm sequential pattern mining.	The similarity of student's course history	Rules Coverage using Apriori Algorithm Association Rule Mining =0.691
4	Gongwen Xu, Guangyu Jia, Lin Shi, Zhijun Zhang(Aug 2021)[30]	the algorithm based on knowledge graph and collaborative filtering	Side Information is used as social media, multimedia	Accuracy =0.96
5	Idowu Dauda Oladipo, Joseph Bamidele Awotunde, Muyideen Abdul Raheem Oluwasegun Osemudiame Ige Ghaniyyat Bolanle Balogun Adekola Rasheed Tomori Fatimoh Abidemi Taofeek-Ibrahim(Oct 2021)[8]	logistic regression model and deep recommender were used to classify students to recommend possible electives to them.	Students' previous grades	Accuracy=0.84
6	Our Method	K-means algorithm and Sequential deep learning algorithm	User Profile and User Rating	Accuracy=0.98

Table 2: Data Collection

Sr.No.	Specification of Data	Number of records
1	Computer Engineering	423
2	Electronics &Telecommunication	474
3	Mechanical	555
4	Civil	200
5	Chemical	233

Out of all the User Profile dataset parameters, the "keyskills" parameter is taken to decide the recommendation. The system is trained using all the records in the User Profiles dataset. Another dataset used for the recommendation is the User Rating dataset. This dataset is the matrix of User Id and Course Id. The User Id field is derived from the User Profile dataset. The Course Id comes from the list of courses. The course list is as shown in Table 3. There are a total of 20 courses we considered. Each course has a unique id, that is, course id. User Ratings from related clusters are taken into account when creating the User Rating dataset. The rating is given explicitly by the user in the range of 1 to 5 (1 is the lowest rating, and 5 is the highest rating).

3-2- Modelling Techniques

The proposed system's functioning is explained below.

Let the system be the set, S = {I, O, F}

$I: Set\ of\ Inputs$
$= \{U1, U2, \ldots . Un, K1, K2 \ldots . Kn, R1, R2, \ldots . . Rn\}$

Where

$U1, U2, \ldots . , Un = set\ of\ User\ Profiles$

$K1, K2, \ldots Kn$
$= set\ of\ keyskills\ taken\ from\ User\ Profiles$

$R1, R2, \ldots . . , Rn = Ser\ of\ User\ Ratings$

$O: Set\ of\ Outputs = \{RE1, RE2, \ldots . REn\}$

Where

$RE1, RE2, \ldots . REn = Recommendation\ list$

$F: Set\ of\ Functions = \{F1, F2, F3, F4\}$

where

$F1: Feature\ Extraction$
$F2: Clustering\ by\ Kmeans$
$F3: Training\ the\ model$
F4: Recommendation model

3-2-1-Data Pre-processing and Feature extraction (Function F_1)

To obtain the vector representation of each word, the TF-IDF (Term Frequency-Inverse Document Frequency) approaches [5] are used. TF-IDF is a traditional method used in the recommendation system.TF-IDF has the first item as Term Frequency (TF) can be represented as $tf_{t,d}$ where t is the word count, and d is the document where the counted word has appeared. $tf_{t,d}$ gives us a number that tells how many times the word t appears in document d. Their sub vectors compared each other, and similarity measure Euclidean distance is used. Tf-Idf weights are used to compare User Profiles. Term Frequency (TF) represents as $tf_{t,d}$.

The equation of TF weight is as

$$tf_{t,d} = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (1)$$

where $f_{t,d}$ is term count in a document
The equation of IDF weight is as

$$\text{idf} = \log \frac{N}{df_t} \quad (2)$$

Tf-Idf weight is

$$W_{t,d} = 1 + \log 10 tf_{t,d}\ X \log \frac{N}{df_t} \quad (3)$$

Where N is the total number of documents.

Table 3: Course List

✉ **Pradnya V.Kulkarni ,**
pradnyav.kulkarni@mitwpu.edu.in

Course ID	Course Name	Course ID	Course Name
1001	Core JAVA	1011	LINUX
1002	CPP	1012	PHOTOSHOP
1003	Data Structures	1013	CAD-CAM
1004	Python	1014	AuTOCAD
1005	C	1015	3D-modelling
1006	Mysql	1016	Project Managemnet
1007	Embedded C	1017	Hypermesh
1008	SQL	1018	D-A-U-E
1009	MATLAB	1019	Staad-pro
1010	Arduino	1020	Tekla

Each word in the dataset is converted into a vector. The input for the next function is the feature vector acquired from the F1. To extract the feature vector, the frequency matrix is computed first, then using frequency matrix term frequency (TF) is calculated, and inverse document frequency (IDF) is computed. The pseudo-code of the feature extraction is shown below.

```
ComputeFrequencyMatrix(userKeySkills):
    freqMatrix ← {}
    stopwords ← SET(stopwords.words("English"))
    ps = PorterStemmer()
    for each user in userKeySkills:
        freqTable ← []
        keyskills ← wordTokenise(userKeyskills)
        for each keyskill in keySkills:
            keyskill ← keyskill.lower()
            keyskill ← ps.stem(keyskill)
        if keyskill in stopwords:
            continue
        if keyskill in freqTable:
            freqTable[keyskill] ← freqTable[keyskill] + 1
        else:
            freqTable[keyskill] ← 1
        freqMatrix[user] ← freqTable
    return freqMatrix
```

```
ComputeTf(freqMatrix):
    tfMatrix ← {}
    for each user, freqTable in freqMatrix.items():
        tfTable ← []
        userKeyskillsCount ← length(freqTable)
        for each keyskill, count in freqTable.items():
            tfTable [keyskill] ← count / userKeyskillsCount
        tfMatrix[user] ← tfTable
    return tfMatrix
```

```
    idfTable ← []
    for each keyskill in freqTable.keys():
    idfTable[keyskill] ← math.log10(totalUsers / countUsersPerKeyskill[keyskill])
    idfMatrix[user] ← idfTable
return idfMatrix
```

```
ComputeTfIdf(tfMatrix, idfMatrix):
    tfidfMatrix ← {}
    for each (user, freqTable) in zip(tfMatrix.items(), idfMatrix.items()):
        tfidfTable ← []
for each (keyskill, value) in zip(tffreqTable.items(), idffreqTable.items():
            tfidfTable [keyskill1] ← (tfvalue . idfvalue)
        tfidfMatrix[user] ← tfidfTable
    return tfidfMatrix
```

3-2-2- Clustering using K-Means Algorithm (Function F2)

The feature vector is taken from the User Profile. This feature vector is fed into the K-means clustering algorithm [24] to determine which cluster the User Profile belongs to. To determine the set of K clusters, the K-means clustering technique is utilized. Every data point is assigned to the closest center, with the total number of such assignments reduced to a minimum. For low-dimensional data, the K-means clustering algorithm is utilized because it is simple and fast. It outperforms other hierarchical clustering methods in terms of computing speed. It also results in tighter clusters [24].

3-2-2-1-Elbow method to find optimal clusters

Both distortion and inertia elbow methods [27] are applied using the dataset and calculated the value of k. Fig.2 depicts the elbow to finalize the value of k using distortion. On the X-axis value of k and the Y-axis, the distortion value is displayed. Fig.3 depicts the elbow to finalize the value of k using inertia. On the X-axis value of k and Y-axis, the inertia value is displayed. The distortion is measured as the sum of the square distances of the respective clusters from the cluster centers. The Euclidean distance metric is used. Inertia is the number of square sample distances to the center of their nearest cluster to calculate distortion. Then using Euclidean distances between centroids and their items, the total intra-cluster variation is calculated.

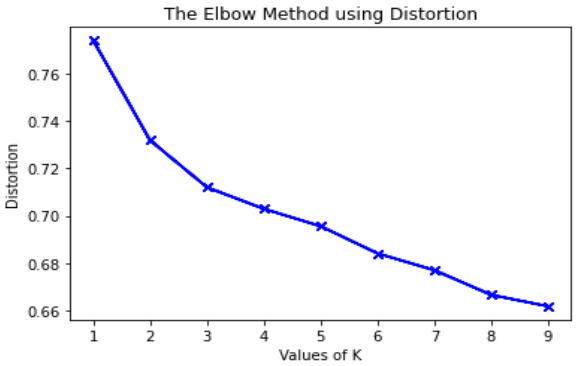


Fig. 2 The elbow method uses distortion

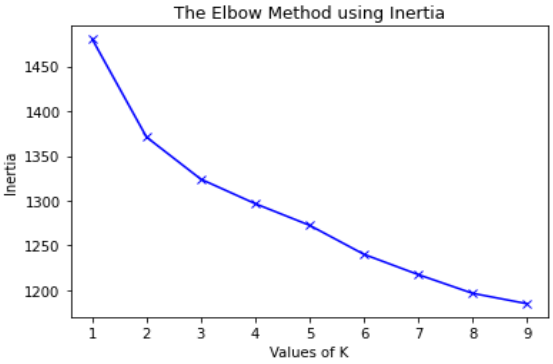


Fig.3 The elbow method uses inertia

The K value of the K-means algorithm can find out using the above elbow method. Once the K value is finalized, the clusters from the available feature vector are calculated using the standard K-Means algorithm. Euclidean Distance similarity measure [11] is used to find the similarity index in the K-Means algorithm.

The equation of Euclidean Distance similarity measure is:

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (4)$$

where p,q= two points in Euclidean n-space

q_i,p_i=Euclidean vectors, starting from the origin of the space (initial point)
n= n-space
The K-Means algorithm returns the clusters which the following model uses for training purposes.

3-2-3-Training the Model (Function F_3)

The model is trained using a deep learning-sequential model [2]. Fig.4 depicts the suggested system's sequential model. A sequential model is a series of layers that process input and output it through the output layer. It has a sequence of layers.

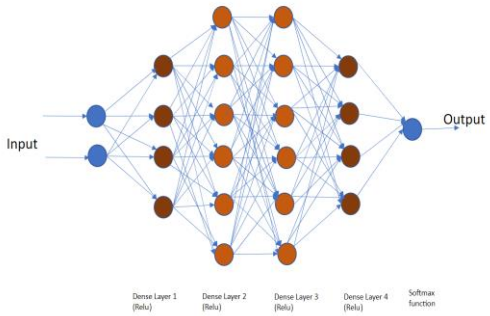


Fig.4: Deep Learning Sequential model

In the first dense layer,50 neurons and the relu() activation function is used. It is a rectilinear activation function. Mathematically, it is defined as y= max (0, x). It returns 0 if it gets negative input. For positive input, it gives the same value back. In the output layer the SoftMax() function[18] is used as activation function. It specifies the probability for each node in the output layer; when these probabilities are added together, the result is 1. That is, the result is between 0 and 1.

Softmax function is

$$\sigma(Z)_i = \frac{e^{Zi}}{\sum_{j=1}^{K} e^{Zi}} \; for\; i = K \; and \; z = (Z_1, \dots \dots Z_K)\epsilon\mathbb{R}^K \quad (5)$$

$$\sigma = softmax\; function$$

$$Z = input\; vector$$

e^{z_i} =standard exponential function for an input vector

K= number of classes in K-means

e^{z_j}=standard exponential function for output vector

In the second dense layer, 30 neurons, in the third layer 20, and the fourth layer10 neurons and relu() activation function is used in all these layers. After creating the model, an object is created that can perform training actions at each iteration. In compiling the model Stochastic Gradient Descent (SGD) optimizer [21], categorical cross-entropy [29] loss function is used because the multi-class classification and accuracy metric is used. The model is trained for 1000 iterations. In the end, the model performance analysis is done using loss and accuracy.

3-2-4-The Recommendation Model (Function F_4)

A recommendation list of courses is shown in Table 2. Then the next step is to read the User (individual user) Profile who is willing to use the proposed eLearning model. It reads a single user profile to which the course is to be recommended. The steps in section 3-2-1- will be carried out on the selected single user profile, and the feature vector is extracted from that profile. It needs to be found that the cluster would fit perfectly using this feature vector. A trained model will be used to find the predicted cluster that is the cluster to which that record belongs. The keyskills available to the user will be subtracted from the cluster's keyskills, and the user would be recommended for the remaining courses. There

may be more than the one suggested coue out of which the user can choose for one of the courses. From the recommended course list, a user is recommended a single course and dataset User Rating considered as mentioned in the dataset. The mean of User Rating will be calculated, and the course having maximum mean will be recommended to the user, using the User Rating dataset and recommendation list. Step 3.2.1 is applied to U1 (feature vector of a single user). The trained model from step 3.2.3 can be used on the feature vector U1 to get the predicted cluster. This is the cluster number from which cluster the user belongs. To get the recommendation list, subtract the keyskills of the user and the matching class. The result of the subtraction is the recommended list of courses which is depicted in the Venn diagram in Fig.5.

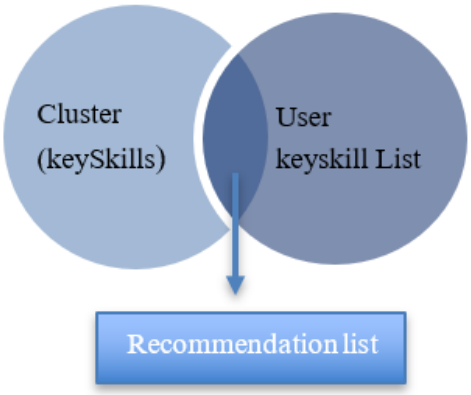


Fig.5 Venn diagram for a recommendation list

4- Results and Discussion

We investigated with the aid of a small educational illustration from eLearning. The execution of the algorithm mentioned above is performed using the datasets mentioned in the dataset section. The eLearning recommender model is implemented with the various Machine Learning Algorithms [20]. All the Machine Learning algorithms are executed, and results are determined for comparison with the proposed method. There are four classes for the mentioned dataset using K-means clustering algorithms, as shown in Table 4. The performance measures result is depicted in Table 5. Total

Class 0	Html, MongoDB, embedded, core, Matlab, SQL, MySQL, python, java, CPP
Class 1	Revit, skills, communication, design, cad, data, PCB, programming, CPP, Matlab
Class 2	Cad, office, creo, Matlab, Solidworks, Ansys, proe, ms, CATIA, AutoCAD
Class 3	Python, android, javascript, PHP, SQL, MySQL, Java, CPP, CSS, HTML

8 Machine Learning algorithms [22] are used, as shown in Table 5.

Table 4: Classes/Cluster list

Precision (6), recall (7), F1-score (8) are employed as performance measurements [22]. True positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the four parameters that are used to evaluate the model's overall predictive capability (9). The accuracy is defined as the ratio of appropriately recommended courses to the first n courses advised. The recall ratio is the number of correctly recommended courses divided by the total number of recommended courses (m).

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F = \frac{2*Precision*Recall}{Precision+Recall} \quad (8)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Table 5: Analysis of Machine Learning Algorithm

Sr.No.	Algorithms Used for eLearning Recommendation System	Average accuracy (%)	Performance Measures				
			Classes	Precision	Recall	F1-Score	Support
1	K Nearest Neighbour	91	0	0.83	0.85	0.84	85
			1	0.94	0.93	0.94	389
			2	0.83	0.86	0.84	72
			3	0.89	0.87	0.88	78
2	Support Vector Machine	97	0	0.98	0.95	0.96	85
			1	0.97	0.98	0.96	389
			2	0.94	0.97	0.97	72
			3	0.95	0.97	0.96	78
3	Decision Tree	95	0	0.97	0.88	0.94	85
			1	0.97	0.96	0.97	389
			2	0.9	0.97	0.93	72
			3	0.89	0.96	0.93	78
4	Random Forest	97	0	0.96	0.94	0.95	85
			1	0.98	0.97	0.98	389
			2	0.91	0.99	0.95	72
			3	0.94	0.96	0.95	78
5	Naïve Bayes	60	0	0.25	0.96	0.39	85
			1	0.88	0.35	0.5	389
			2	0.6	0.93	0.73	72
			3	0.68	0.24	0.36	78
6	Linear Regression	97	0	0.95	0.94	0.97	85
			1	0.97	1	0.98	389
			2	0.96	0.92	0.96	72
			3	0.96	0.95	0.95	78
7	LDA	92	0	0.82	0.92	0.87	85
			1	0.98	0.91	0.94	389
			2	0.87	0.93	0.9	72
			3	0.8	0.95	0.87	78
8	Neural Network	96	0	0.94	0.93	0.93	85
			1	0.98	0.97	0.98	389
			2	0.93	0.96	0.95	72
			3	0.94	0.96	0.95	78
9	Our Proposed algorithm using Deep Learning	98	0	0.94	0.95	0.95	85
			1	0.99	0.98	0.98	389
			2	0.97	0.97	0.97	72
			3	0.96	0.97	0.97	78

Fig.6 shows the graph of precision values for all four classes/clusters of all the algorithms. A high precision value indicates that there is a low false-positive rate. Fig.7 depicts the plot of recall values for all the four clusters of all algorithms. A high recall value indicates that there is a low false-negative rate. Fig.8 depicts the plot of the F1-score for all clusters of all algorithms and projects that large score value results in a correct positive outcome and preserving it.

Fig.9 shows the plot of the accuracy of all algorithms. The proposed algorithm accuracy achieved is 0.98 which shows that the system correctly recommends the courses to a user by using the User Profile and User Ratings dataset.

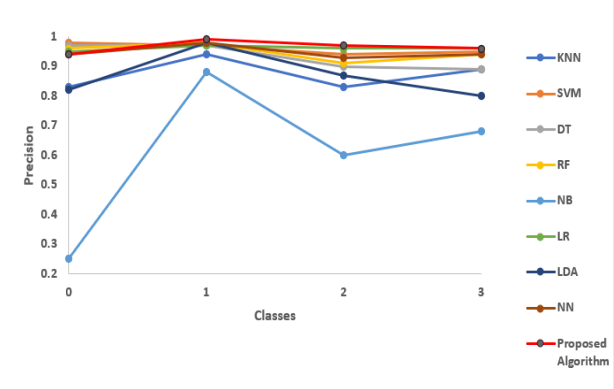


Fig.6 Plot of Precision

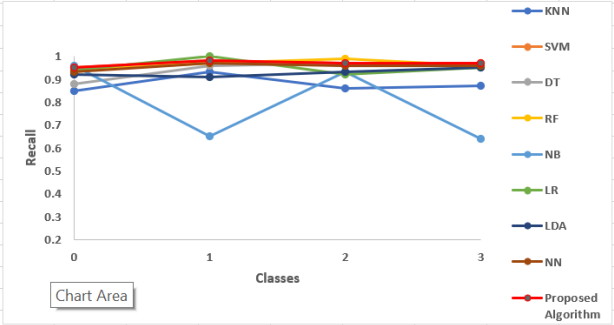


Fig.7 Plot of Recall

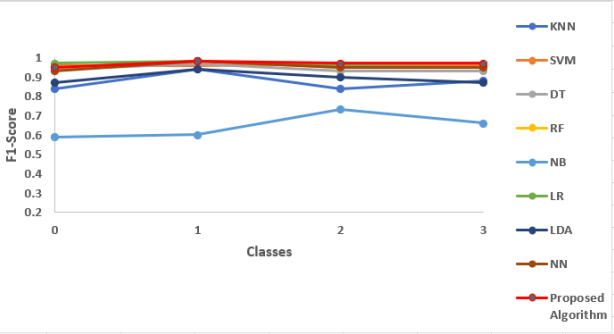


Fig.8 Plot of F1-score

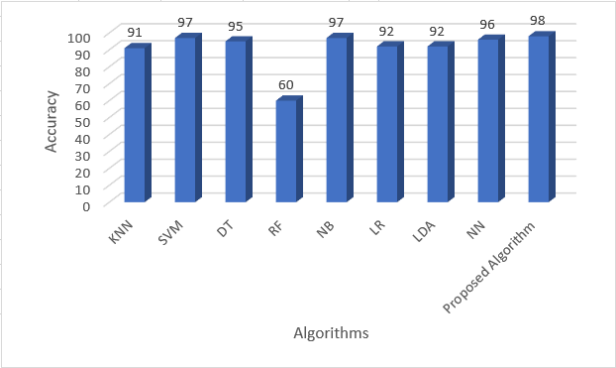


Fig.9 Comparison of accuracy

5- Conclusion and Future Work

To improve the accuracy and effectiveness for a proper recommendation of the course we proposed an eLearning Recommendation System based on collaborative learning and deep learning. The proposed system addresses the problems of collaborative filtering in the recommendation system algorithm which solely employs the users' history or grades data. To evaluate the performance of the proposed method, experiments on User Profile and User Rating between state-of-art methods and our method are conducted. Various machine learning algorithms are used to compare the results with the proposed method. The accuracy of machine learning algorithms is between 0.81 to 0.97 and we have improved the accuracy by using the proposed method to 0.98. The results suggested that the considered parameters in the dataset and deep learning approach provide more accurate results.

However proposed algorithm also has some limitations, which will not consider the weak topic of the user and do not consider implicit feedback of the user. It can be enhanced even more if implicit feedback-based recommendations are taken into account. Another limitation is the experimentation is done only on Engineering students. The model can further extend to any of the eLearning programs by modifying the course list.

References

[1] Muthuprasad, T., Aiswarya, S., Aditya, K. S., and Jha, G. K, "Students' perception and preference for online education in India during the COVID-19 pandemic", Social Sciences & Humanities Open, 3(1), 100101,2021.

[2] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., and Baskurt, "Sequential deep learning for human action recognition", in international workshop on human behavior understanding (pp. 29-39). Springer, Berlin, Heidelberg, A, 2011, November.

[3] Benhamdi, S., Babouri, A., and Chiky, R. "Personalized recommender system for an e-Learning environment", Education and Information Technologies, 22(4), 1455-1477, 2017.

[4] Bonk, C. J., Zhu, M., Kim, M., Xu, S., Sabir, N., and Sari, A. R," Pushing toward a more personalized MOOC: Exploring instructor selected activities, resources, and technologies for MOOC design and implementation", International Review of Research in Open and Distributed Learning, 19(4), 2018.

[5] Kim, S. W., and Gil, J. M," Research paper classification systems based on TF-IDF and LDA schemes", Human-centric Computing and Information Sciences, 9(1), 1-21,2019.

[6] Dahdouh, K., Oughdir, L., Dakkak, A., and Ibriz, A," Smart courses recommender system for an online learning platform", IEEE 5th International Congress on Information Science and Technology (CiSt) (pp. 328-333),2018.

[7] Gulzar, Z., Leema, A. A.,and Deepak, G," Pcrs: Personalized course recommender system based on hybrid approach", Procedia Computer Science, 125, 518-524, (2018).

[8] Oladipo, I. D., Awotunde, J. B., AbdulRaheem, M., Ige, O. O., Balogun, G. B., Tomori, A. R., and Taofeek-Ibrahim, F. A.)," An improved course recommendation system based on historical grade data using logistic regression", In International Conference on Applied Informatics (pp. 207-221), Springer, Cham,2021, October.

[9] Avazpour, I., Pitakrat, T., Grunske, L., and Grundy, J, "Dimensions and metrics for evaluating recommendation systems", In Recommendation systems in software engineering (pp. 245-273). Springer, Berlin, Heidelberg, 2014.

[10] Aboagye, E., Yawson, J. A., and Appiah, K," NCOVID-19 and E-learning: The challenges of students in tertiary institutions", Social Education Research, 1-8, 2021.

[11] Jain, G., Mahara, T., and Tripathi, K. N.," A survey of similarity measures for collaborative filtering-based recommender system",Soft computing: theories and applications, 343-352,2020.

[12] Jommanop, T., and Mekruksavanich, S," E-learning recommendation model based on multiple intelligence", in 2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP) (pp. 1-6). IEEE, (2019, October).

[13] Khusro, S., Ali, Z., and Ullah, I," Recommender systems: issues, challenges, and research opportunities", in Information Science and Applications (ICISA) 2016 (pp. 1179-1189). Springer, Singapore,2016.

[14] Kulkarni, P. V., Rai, S., and Kale, R, "Recommender system in eLearning: a survey", In Proceeding of International Conference on Computational Science and Applications (pp. 119-126). Springer, Singapore,2020.

[15] Lai, C. H., Liu, D. R., and Lin, S. R, " Document recommendation with implicit feedback based on matrix factorization and topic model", in 2018 IEEE International Conference on Applied System Invention (ICASI) (pp. 62-65). (2018, April).

[16] Lin, C. Y., Wang, L. C., and Tsai, K. H," Hybrid real-time matrix factorization for implicit feedback recommendation systems", IEEE Access, 6, 21369-21380,2018.

[17] Kulkarni, P. V., Phatak, R., Bhate, B., Deshpande, R., and Rai, S," Recommendation System for Enhancing eLearning using Deep Learning", in 2019 IEEE Pune Section International Conference (PuneCon) (pp. 1-4). IEEE, (2019, December).

[18] Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S," Activation functions: Comparison of trends in practice and research for deep learning", arXiv preprint arXiv:1811.03378,2018.

[19] Pannu, M., Anane, R., Odetayo, M., and James, A. "Explicit user profiles in web search personalization", in Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD) (pp. 416-421). IEEE, June 2011.

[20] Portugal, I., Alencar, P., and Cowan, D," The use of machine learning algorithms in recommender systems: A systematic review", Expert Systems with Applications, 97, 205-227,2018.

[21] Jentzen, A., Kuckuck, B., Neufeld, A., and von Wurstemberger, P," Strong error analysis for stochastic gradient descent optimization algorithms", IMA Journal of Numerical Analysis, 41(1), 455-492,2021.

[22] Ezz, M., Elshenawy, A," Adaptive recommendation system using machine learning algorithms for predicting student's best academic program", Education Information Technologies 25, 2733–2746.

[23] Obeidat, R., Duwairi, R., and Al-Aiad, A," A collaborative recommender system",2019.

[24] Vercellis, C," Business intelligence: data mining and optimization for decision making", John Wiley & Sons.2011.

[25] Wu, G., Swaminathan, V., Mitra, S., and Kumar, R. (2017, December)," Digital content recommendation system using implicit feedback data", in 2017 IEEE International Conference on Big Data (Big Data) (pp. 2766-2771). IEEE, December,2017.

[26] Xiao, Y., Li, X., Wang, H., Xu, M., and Liu, Y, "3-HBP: A three-level hidden Bayesian link prediction model in social networks", IEEE Transactions on Computational Social Systems, 5(2), 430-443, 2018.

[27] Yuan, C., and Yang, H," Research on K-value selection method of K-means clustering algorithm", J, 2(2), 226-235,2019.

[28] Zhang, S., Yao, L., Sun, A., and Tay, Y," Deep learning-based recommender system: A survey and new perspectives", ACM Computing Surveys (CSUR), 52(1), 1-38,2019.

[29] Zhang, Z., & Sabuncu, M. R, "Generalized cross entropy loss for training deep neural networks with noisy labels", in 32nd Conference on Neural Information Processing Systems (NeurIPS), January 2018.

[30] Xu, G., Jia, G., Shi, L., and Zhang, Z, "Personalized Course Recommendation System Fusing with Knowledge Graph and Collaborative Filtering", Computational Intelligence and Neuroscience, 2021.

[31] Praserttitipong, D., and Srisujjalertwaja, W, "Elective course recommendation model for higher education program", Songklanakarin Journal of Science & Technology, 40(6),2018.

Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 196-208
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Implementation of Machine Learning Algorithms for Customer Churn Prediction

Manal Loukili[1*], Fayçal Messaoudi[2], Mohammed El Ghazi[3], Raouya El Youbi[1]

[1].National School of Applied Sciences, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[2].National School of Business and Management, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[3].Superior School of Technology, Sidi Mohamed Ben Abdellah University, Fez, Morocco

Received: 20 Feb 2022/ Revised: 04 Jul 2022/ Accepted: 10 Aug 2022

Abstract
Churn prediction is one of the most critical issues in the telecommunications industry. The possibilities of predicting churn have increased considerably due to the remarkable progress made in the field of machine learning and artificial intelligence. In this context, we propose the following process which consists of six stages. The first phase consists of data pre-processing, followed by feature analysis. In the third phase, the selection of features. Then the data was divided into two parts: the training set and the test set. In the prediction process, the most popular predictive models were adopted, namely random forest, k-nearest neighbor, and support vector machine. In addition, we used cross-validation on the training set for hyperparameter tuning and to avoid model overfitting. Then, the results obtained on the test set were evaluated using the confusion matrix and the AUC curve. Finally, we found that the models used gave high accuracy values (over 79%). The highest AUC score, 84%, is achieved by the SVM and bagging classifiers as an ensemble method which surpasses them.

Keywords: Machine Learning; Churn Prediction; Bagging SVM; k-NN; Random Forest.

1- Introduction

The exponential growth in the number of operators in the market, due to globalization and advances in the telecommunications industry, is increasing competition. In this competitive era, it has become imperative to maximize profits regularly, for this reason, various approaches have been proposed, especially acquiring new customers, up-selling existing customers and increasing the retention period of current customers. Retaining existing customers is the simplest and least expensive strategy compared to the others. In order to adopt this strategy, companies need to reduce the eventual churn of customers. The main reason for this loss of customers is dissatisfaction with the services provided to consumers and the support mechanism. To solve this problem, the solution is to predict which customers are likely to churn [1]. Predicting churn is a crucial objective that helps in establishing customer retention and loyalty strategies. Along with the increasing competition in service delivery markets, the risk of customer churn is also increasing explosively. As a result, it has become imperative to implement strategies to keep track of loyal customers (non-churners). Churn models aim to identify early signals of churn and attempt to predict which customers leave voluntarily. Thus, many companies have realized that their existing database is one of their most valuable assets [2] and according to Abbasdimehr [3], churn prediction is a very useful tool to predict at-risk customers.

This article is organized as follows: the next section describes the problematic. Section 3 summarizes some related work. Section 4 represents a brief review of the selected classification techniques used for this study. The different steps of our methodology are discussed in Section 4. The results are presented in section 5. And section 6 to present the results and analyze the performance of each model. Finally, section 7 concludes the article.

2- Problem Description

To overcome the above problem, the company must correctly predict the customer's behavior. Churn

✉ **Manal Loukili**
manal.loukili@usmba.ac.ma

management can be done in two ways: reactive and proactive.

The reactive strategy is to wait for the cancellation request received from the customer and then to offer interesting plans to retain the customer. The proactive strategy, on the other hand, prevents the customer from unsubscribing. This is because the possibility of unsubscribing is anticipated, and plans are offered to customers accordingly. This results in a binary classification problem where churners are distinguished from non-churners.

To deal with this problem, machine learning algorithms have emerged as a very powerful technique for predicting information on the previously captured database [4], including linear regression, support vector machines, naive bayes, decision trees, random forests, etc. In machine learning models, the information is then used to predict the churn.

In machine learning models, after preprocessing, feature selection plays a major role in increasing the accuracy of classification. Researchers have developed a large number of approaches for feature selection that reduce dimensionality, overfitting, and computational complexity. For churn prediction, these features are taken from the given input vector and used for churn prediction. In this paper, to solve this problem, we used the following machine learning algorithms: Support vector machine, K-Nearest Neighbors, and Random Forest.

Support vector machines: This algorithm can be used in cases where there are two classes of objects (e.g., churners and non-churners). It can also be used when there are more than two classes of objects (e.g., churners, potential churners, and non-churners).

K-nearest neighbor: This algorithm is suitable for time series data, categorical data, and sparse datasets.

Random forest: This is a supervised machine learning algorithm which is based on classification and regression trees. It is designed to handle both categorical and continuous data types.

3- Related Work

This section briefly summarizes some related work proposed by leading researchers for the prediction of churn in the telecommunications sector.

The authors in [5] analyzed the variables that impact customer churn. They also conducted a comparative study of three machine learning models such as regression, neural network, and regression trees. The results showed that the decision tree performs better than the others because of its rule-based architecture. However, the accuracy obtained can be further improved by using one of the existing feature selection methods.

The authors in [6] adopted three machine learning approaches, namely support vector machine, neural network, and Bayesian networks for attrition rate prediction. A principal component analysis is taken into account in the feature selection process which allows for a reduction in data dimensions. However, the feature selection process can be improved to increase the accuracy of the classification by applying an optimization algorithm. The gain measure and the ROC curve were used to evaluate the performance.

In another study [7], the authors tried to solve the customer loss prediction problem using a support vector machine, a random forest and logistic regression. The performance of the SVM was approximately equal to that of the logistic regression and the random forest, but once optimal parameter selection was considered, the SVM outperformed the logistic regression and the random forest in terms of PCC and AUC.

Paper [8] presents an overview of all the machine learning models considered, as well as a detailed analysis of the feature selection techniques in use. The authors found that in the prediction models the decision tree had a higher efficiency than the others. In feature selection, optimization techniques also play an essential role that improves the prediction techniques. After a comparative analysis of existing methods.

In [9], the authors adopted two machine learning models, decision tree and logistic regression on a churn prediction dataset. They used the WEKA tool for experimentation. But the customer churn problem can be solved more effectively by using other machine learning methods.

4- A Brief Review of the Machine Learning Classification Algorithms Used

4-1- Bagging Support Vector Machine

Support Vector Machine or SVM is a supervised learning technique that aims to analyze data to detect patterns. There are two types of support vector machines: linear and nonlinear [10]. If the data domain can be divided linearly (e.g., straight line or hyperplane) to separate the classes in the original domain, it is referred to as a linear support vector machine. Nonlinear support vector machine is used when the data domain cannot be split linearly and can be translated to a space called the feature space where the data domain can be divided linearly to distinguish the classes [11]. On the basis of a set of training data, SVM attempts to determine the optimal separating hyperplanes between examples of distinct classes by representing observations as points in a high dimensional space. New instances are represented in the same space and assigned to a class depending on their closeness to the dividing gap [12].

Bagging, also called Bootstrap aggregating, is an ensemble learning approach that helps in the improvement of

performance and accuracy of a machine learning algorithm. It is mainly used to minimize a prediction model's variance and to deal with bias-variance tradeoffs. In [13] Various simulated results for IRIS data categorization and hand-written digit identification demonstrate that the proposed SVM ensembles with bagging significantly outperform a single SVM in terms of classification accuracy. When it comes to the customer churn issue, ensemble-learning techniques have been used as shown in [14], [15], [16].

4-2- K-Nearest Neighbors

When there is little or no prior knowledge about the distribution of the data, K-Nearest Neighbors (k-NN) classification is one of the most fundamental and straightforward classification procedures and should be one of the initial options for classification research [17]. The k-NN classification arose from the necessity to do discriminant analysis when valid parametric estimates of probability densities are unknown or impossible to calculate.

The k-NN method predicts the values of new data points based on "feature similarity", which implies that the new data point will be assigned a value depending on how closely it resembles the points in the training set. k-NN does not attempt to build an internal model, and no calculations are done until classification time. k-NN merely stores instances of the training data in the features space, and an instance's class is chosen by the majority vote of its neighbors. The class most prevalent among its neighbors is assigned to the instance. k-NN finds neighbors based on distance utilizing Euclidian, Manhattan, or Murkowski distance measures for continuous variables and hamming distance measures for categorical data [18].

4-3- Random Forest

Random forests, also known as random choice forests, are an ensemble learning approach for classification, regression, and other tasks that work by building a large number of decision trees during training. It contains multiple decision trees, each reflecting a unique instance of the random forest's classification of data input. The random forest approach examines each case independently, selecting the one with the most votes as the chosen prediction. The classification findings in [18] suggest that Random Forest outperforms Decision Tree (J48) for the same number of characteristics and big data sets, i.e., with a higher number of instances, but Decision Tree (J48) is useful for small data sets (less number of instances). In addition to that the study in [19] shows that, the best classification model out of naïve Bayes, decision tree, and random forest is random forest because of its high accuracy of 97.5 % when compared to the classification model of decision tree, which has an accuracy of 88.7 %.

5- Methodology

The steps and advantages of the proposed technique are as follows (Fig.1):

The gravitational search algorithm was used to select the features and reduce the dimensions of the data set, unlike the above existing approaches where the prediction accuracy is low due to inadequate feature selection.

After data preprocessing, some of the most important machine learning techniques used for predictions, including SVM, were applied. To avoid overfitting, cross-validation was performed, unlike other techniques where the overfitting prevention mechanism is not considered.

The power of ensemble learning was then utilized to optimize the algorithms and obtain better results, unlike the previously mentioned techniques where the performance of ensemble learning is not taken into account, which explains the low accuracies obtained.

The algorithms were then evaluated on the test set using the confusion matrix and the AUC curve to compare the best performing algorithm for the given data set.

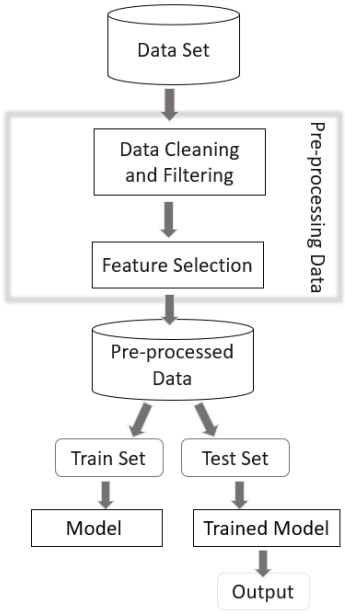


Fig. 1 Proposed system architecture

5-1- Presentation of the Data Set Used

The data set that we used in our experiments is "Telco Customer Churn" which is available on the Kaggle site, which contains a data set of 7043 customers. The data set includes information about:

• Customers who have left the company in the last month - the column is called Churn.

• The services to which each customer has subscribed: telephone, multiple lines, internet, online security, online backup, device protection, technical support and streaming TV and movies.

• Customer account information: how long they have been a customer, type of contract, method of payment, electronic billing, monthly charges, and total charges.

• Demographic information about customers: gender, age range, and whether they have partners and dependents.

The database consists of 21 attributes including a target value called Churn. The data set of the customer attributes alongside with their description is presented in Table. 1, Table. 2 presents the type of these attributes.

Table 1: Margin specifications

Attribute Name	*Type*	*Description*
CustomerID	Unique key	A code specific to each customer
Gender	Categorical	The gender of the customer
SeniorCitizen	Categorical	Whether the client is young or old
Partner	Categorical	Whether the client is married or not
Dependents	Categorical	If the client has someone dependent on him
Tenure	Integer	The number of months during which the customer is loyal
PhoneService	Categorical	Whether the customer has a telephone service or not
MultipleLines	Categorical	Whether the customer has a multitude of lines or not
InternetService	Categorical	If the customer has an internet service
OnlineSecurity	Categorical	If the customer has online security
OnlineBackup	Categorical	If the client has an online backup
DeviceProtection	Categorical	If the customer has device security
TechSupport	Categorical	If the customer has technical support
StreamingTV	Categorical	If the customer has on-demand television
StreamingMovie	Categorical	If the customer has the movies on demand
Contract	Categorical	The contract renewal period
PaperlessBilling	Categorical	Whether the customer has paperless or non-paperless billing
PaymentMethod	Categorical	The payment method of the customer
MonthlyCharges	Integer	The monthly charge of the client
TotalCharges	Integer	The total charge of the client
Churn	Categorical	If the customer cancels his contract or not

Table 2: Type of attributes

	Index
	Demographic Information
	Services
	Account information
	Target

5-2- Data Analysis

After importing data from the ".csv" file, the df.info() command was executed to display information about the database.

```
<class 'pandas.core.frame.DataFrame'>
Index: 7043 entries, 7590-VHVEG to 3186-AJIEK
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   gender            7043 non-null   object
 1   SeniorCitizen     7043 non-null   int64
 2   Partner           7043 non-null   object
 3   Dependents        7043 non-null   object
 4   tenure            7043 non-null   int64
 5   PhoneService      7043 non-null   object
 6   MultipleLines     7043 non-null   object
 7   InternetService   7043 non-null   object
 8   OnlineSecurity    7043 non-null   object
 9   OnlineBackup      7043 non-null   object
 10  DeviceProtection  7043 non-null   object
 11  TechSupport       7043 non-null   object
 12  StreamingTV       7043 non-null   object
 13  StreamingMovies   7043 non-null   object
 14  Contract          7043 non-null   object
 15  PaperlessBilling  7043 non-null   object
 16  PaymentMethod     7043 non-null   object
 17  MonthlyCharges    7043 non-null   float64
 18  TotalCharges      7043 non-null   object
 19  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(17)
memory usage: 1.1+ MB
```

It has been noticed that there is a problem with the types of the attributes "SeniorCitizen" and "TotalCharges" in which "SeniorCitizen" needs to be converted to a string type, and "TotalCharges" needs to be converted to an integer type. The conversion of these attributes to their appropriate types will be carried out as the first step.

```
df["TotalCharges"] =
pd.to_numeric(df["TotalCharges"],
errors='coerce')
df['SeniorCitizen'].replace(to_replace=1,
value='Yes', inplace=True)
df['SeniorCitizen'].replace(to_replace=0,
value='No', inplace=True)
```

After the conversion, it was observed that the "TotalCharges" column has 11 missing values. It is known that the "TotalCharges" variable can be calculated by multiplying the two variables "Tenure" and "MonthlyCharges". However, for all the entries in the "TotalCharges" column, the corresponding "Tenure" value is 0, indicating that these customers are in their first month. Therefore, the value of "MonthlyCharges" will be directly assigned to them as their "TotalCharges" value.

```
df_na =
df[df['TotalCharges'].isna()].copy()
df_na['TotalCharges'] =
df_na['MonthlyCharges']
df[df['TotalCharges'].isna()] = df_na
```

The database has been cleaned and is now prepared for visualization.

5-3- Data Visualization

a) Qualitative Variables

The qualitative variables were visualized using Python, as depicted in the figures below Fig. 2-Fig. 18:

Fig. 2 Male/female distribution

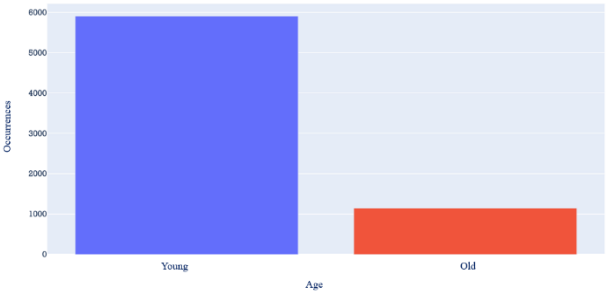


Fig. 3 Young/old people distribution

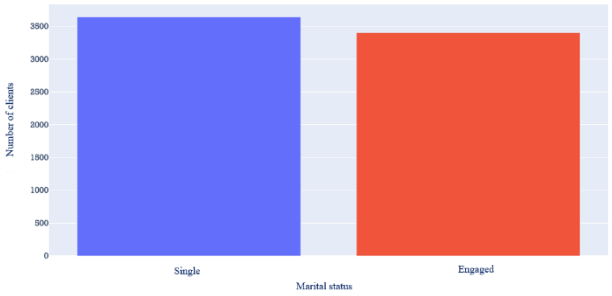


Fig. 4 Single/engaged people distribution

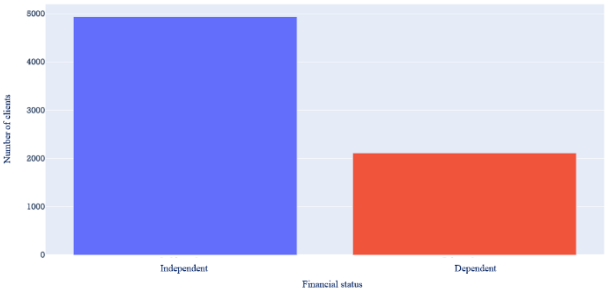


Fig. 5 Independent/ dependent people distribution

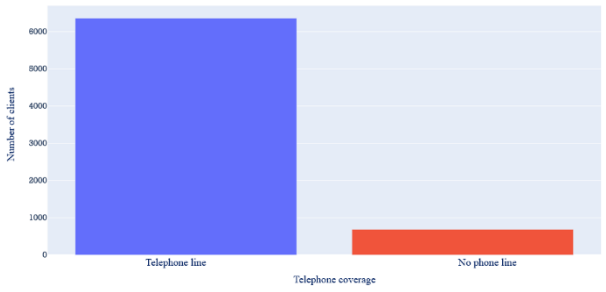


Fig. 6 Distribution of the customers having a telephone line at disposition

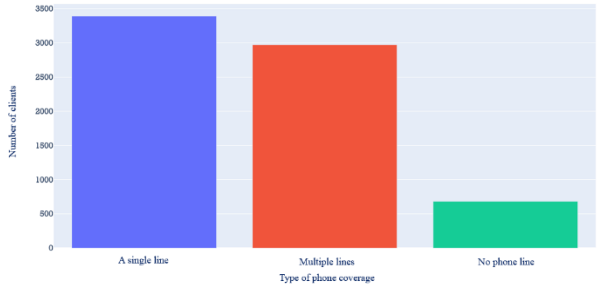


Fig. 7 Distribution of customers with several telephone lines available

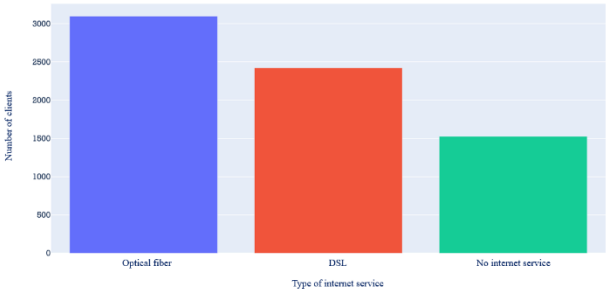


Fig. 8 Type of the customer's internet service provider

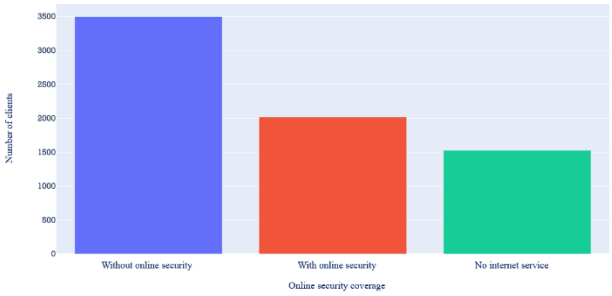


Fig. 9 Distribution of customers with online security

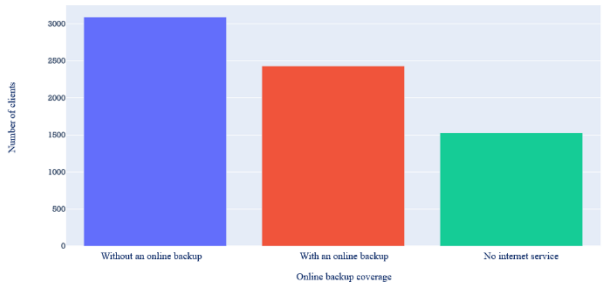


Fig. 10 Distribution of customers with an online backup available

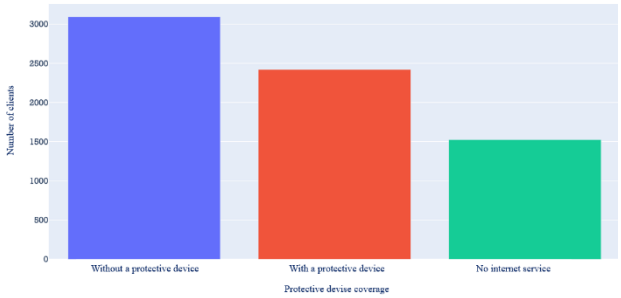


Fig. 11 Distribution of customers with a protective device

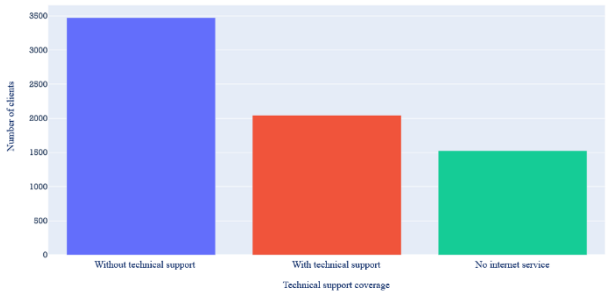


Fig. 12 Distribution of customers with technical support

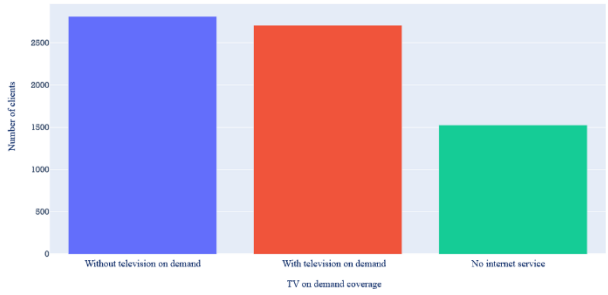


Fig. 13 Distribution of customers with on-demand television available

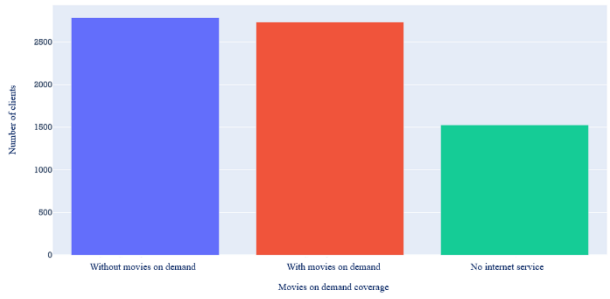


Fig. 14 Distribution of customers with on-demand movies available

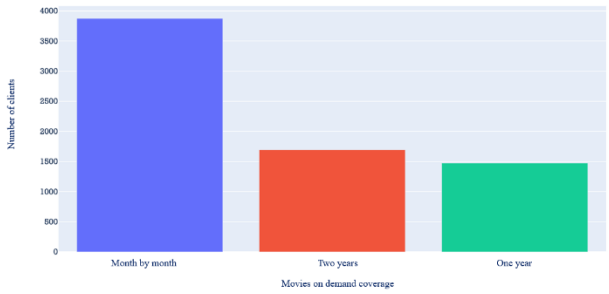


Fig. 15 Distribution of customers according to the duration of their contract

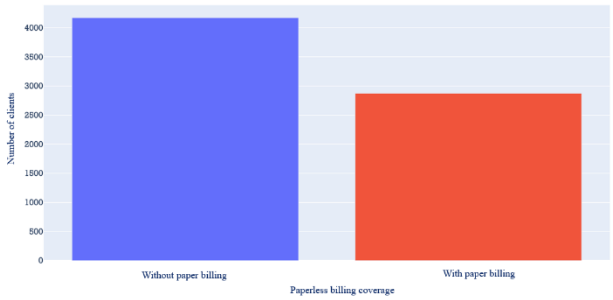


Fig. 16 Distribution of customers according to their billing

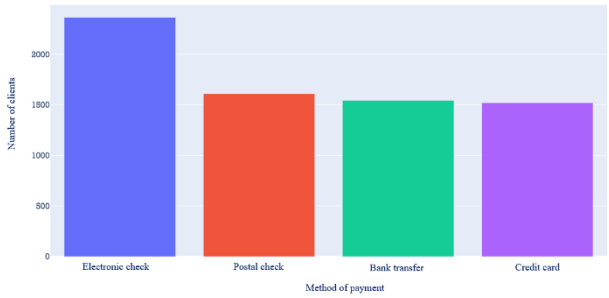


Fig. 17 Distribution of customers according to their type of payment

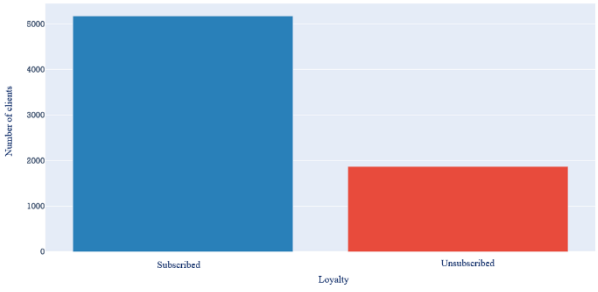


Fig. 18 Distribution of customers according to their loyalty

b) Quantitative Variables

The quantitative variables were visualized as shown in the figures below Fig. 19-Fig. 21:

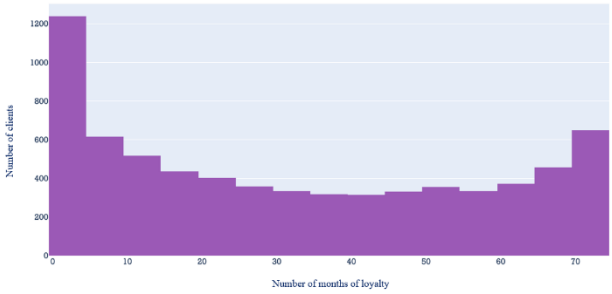


Fig. 19 Distribution number of months of loyalty

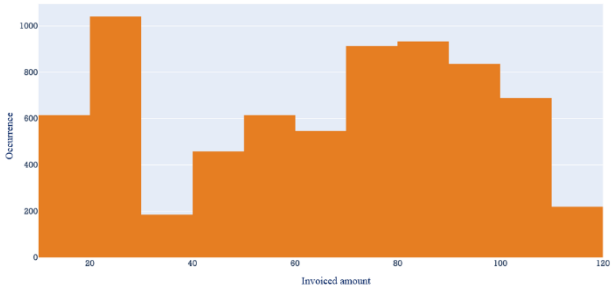


Fig. 20 Distribution of the amount invoiced to the customer each month

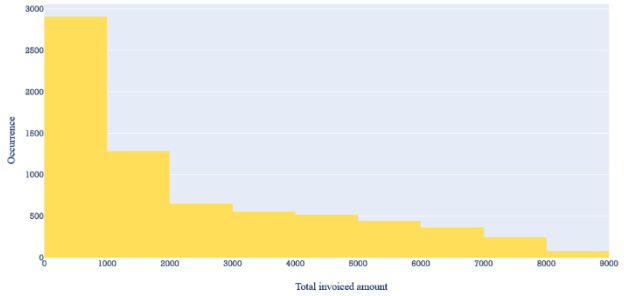


Fig. 21 Distribution of the amount invoiced to the customer in total

5-4- Study of Correlation

The figures that follow represent the correlation between each of the variables and the target variable Fig. 22-Fig. 36:

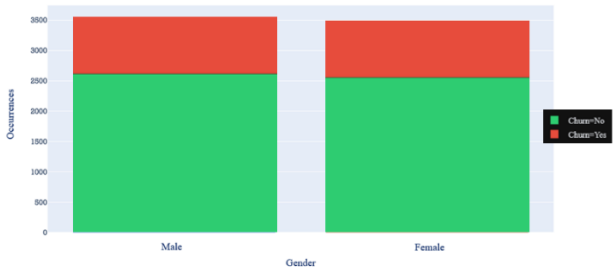


Fig. 22 Gender and churn correlation

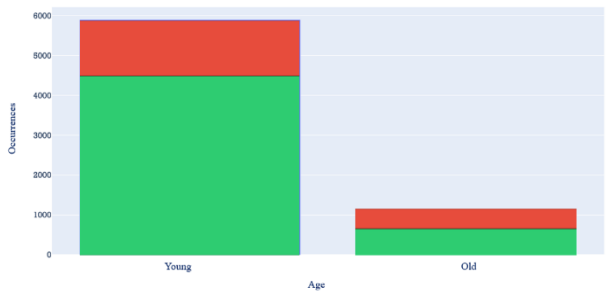


Fig. 23 Age and churn correlation

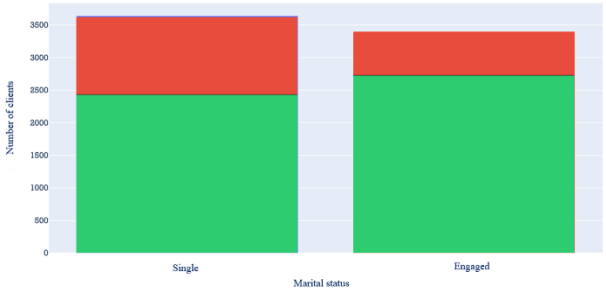


Fig. 24 Marital status and churn correlation

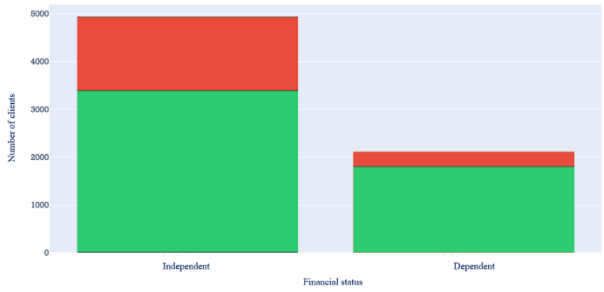


Fig. 25 Financial status and churn correlation

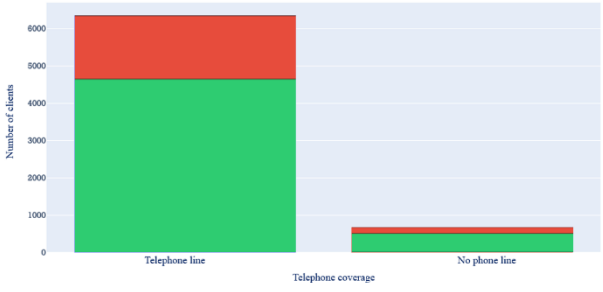


Fig. 26 Telephone coverage and churn correlation

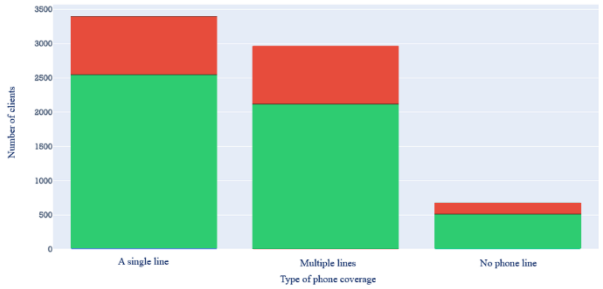


Fig. 27 Type of phone coverage and churn correlation

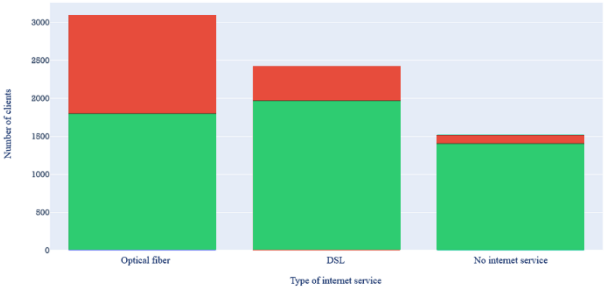


Fig. 28 Type of internet service and churn correlation

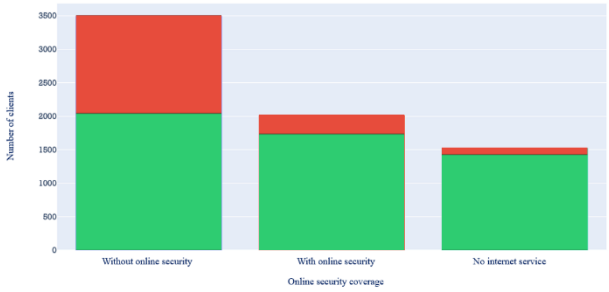


Fig. 29 Online security coverage and churn correlation

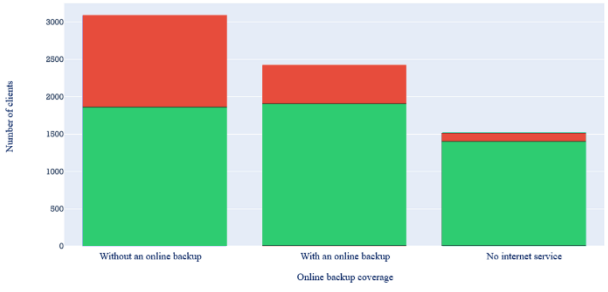


Fig. 30 Online backup coverage and churn correlation

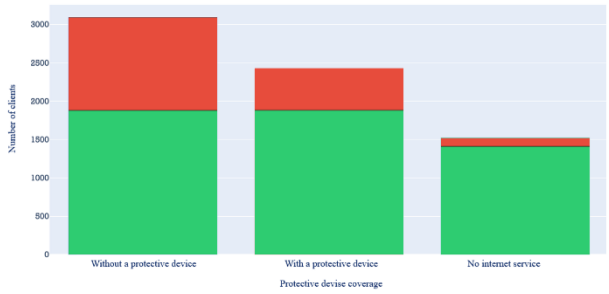


Fig. 31 Protective devise coverage and churn correlation

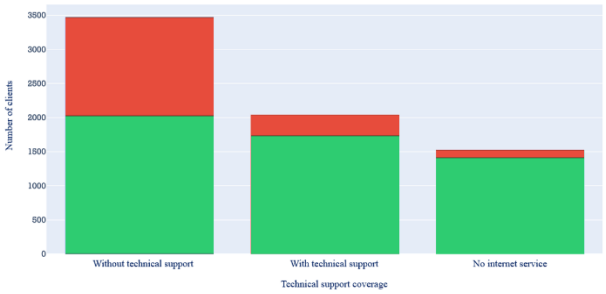


Fig. 32 Technical support coverage and churn correlation

Fig. 33 TV on demand coverage and churn correlation

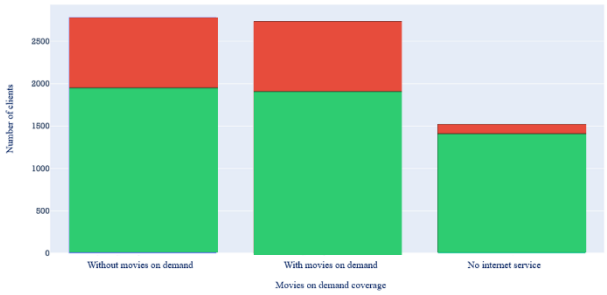


Fig. 34 Movies on demand coverage and churn correlation

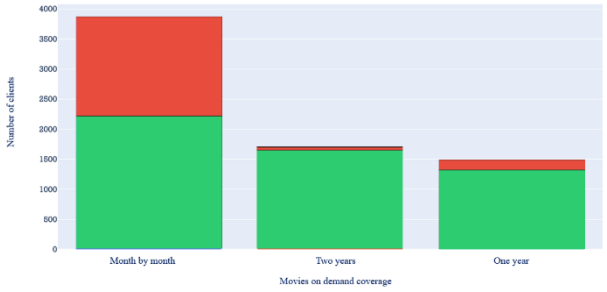


Fig. 35 Movies on demand's duration of contract and churn correlation

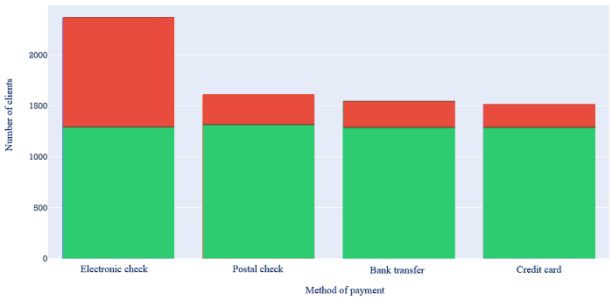


Fig. 36 Method of payment and churn correlation

The Fig. 37 shows the correlation between the churn variable and the other variables:

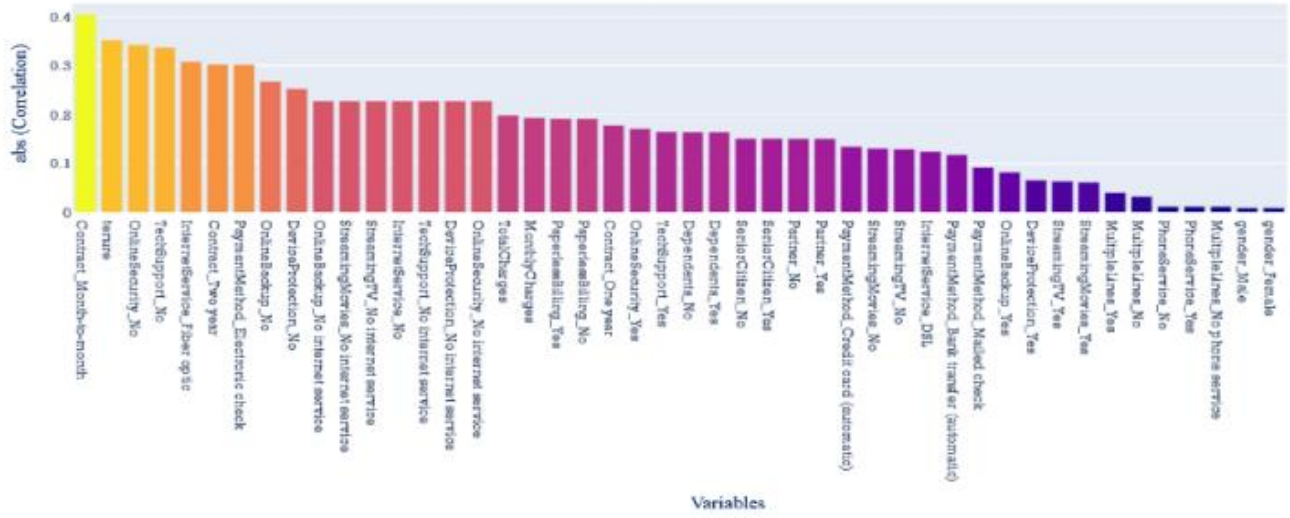


Fig. 37 The correlation between churn and other variables

Based on the correlation diagram and the results obtained from the visualization of the independence of each variable with the target variable, it has been observed that there are variables with weak correlation. However, these variables will be retained as the objective is not to generalize the model, but rather to specifically fit it to this particular company. This approach may result in the model being overfitted to match the company's specific characteristics.

5-5- The Learning Phase

The learning in this study is based on three machine learning algorithms: Bagging SVM, Random Forest and the k-NN algorithm.
The first step in the learning is to convert all qualitative variables into quantitative variables.

```
df['Churn'].replace(to_replace='Yes',
 value=1, inplace=True)
df['Churn'].replace(to_replace='No',
value=0, inplace=True)
df_dum = pd.get_dummies(df)
```

.
After the execution, the "df_dum" DataFrame now consisted solely of quantitative variables.
Subsequently, the various methods and algorithms available in the "Scikit-Learn" library were implemented using Python language.

```
from sklearn.preprocessing import
StandardScaler
from sklearn.pipeline import
make_pipeline
from sklearn.svm import SVC
from sklearn.ensemble import
BaggingClassifier,RandomForestClassifie
r
from sklearn.neighbors import
KNeighborsClassifier
from sklearn.model_selection import
train_test_split
```

5-6- Algorithms Implementation

Bagging SVM

```
model = None
score = 0
for i in range(10):
  X_train, X_test, y_train, y_test = train_te
st_split(X, y, test_size=0.30)
  mdl = make_pipeline(StandardScaler(),
BaggingClassifier(base_estimator=SVC(gamma="a
uto"), n_estimators=10))
  mdl.fit(X_train, y_train)
  a = mdl.score(X_test, y_test)
  if a > score:     model = mdl
                    score = a
```

K-Nearest Neighbors

```
model = None
score = 0
for i in range(10):
  X_train, X_test, y_train, y_test = train_tes
t_split(X, y, test_size=0.30)
  mdl = KNeighborsClassifier(n_neighbors=10)
  mdl.fit(X_train, y_train)
  a = mdl.score(X_test, y_test)
  if a > score : model = mdl
                 score = a
```

Random Forest

```
model = None
score = 0
for i in range(10):
  X_train, X_test, y_train, y_test = train_t
est_split(X, y, test_size=0.30)
  mdl = RandomForestClassifier(max_depth=5)
  mdl.fit(X_train, y_train)
  a = mdl.score(X_test, y_test)
  if a > score :    model = mdl
                    score = a
```

6- Results and Performance Analysis

6-1- Confusion Matrix

In order to evaluate the performance of the applied models and the prediction rate of customer churn on the test set, various metrics were utilized, including precision, recall, accuracy, and F-score..They measure the ability of the predictive models to correctly predict the customer churn. The four indicators previously mentioned are calculated from the information captured using the confusion matrix and are presented in Table 3. The representation of the confusion matrix is shown in Fig. 38. True positives and false positives are denoted Tp and Fp, while false negatives and true negatives are denoted Fn and Tn [20].

- True positive (TP): The number of clients who are in the churn class and whom the predictive model correctly predicted.
- True negative (TN): The number of clients who are in the non-churners class and that the predictive model correctly predicted.
- False positive (FP): The number of customers who are not churners but that the predictive algorithm identified as churners.
- False Negative (FN): The number of customers who are churners but that the predictive model identified as non-churners.

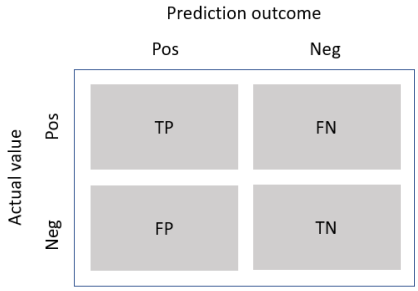


Fig. 38 Confusion matrix

6-2- Performance Indicators

- Recall:

The recall is the ratio of true churners or true positives (TP), and is calculated as follows:

Recall = TP/(TP+FN)

- Precision:

The precision is the ratio of predicted correct churners, its formula is as follows:

Precision = TP/(TP+FP)

- Accuracy:

The accuracy is the ration of the number of all correct predictions and is written as:

Accuracy = (TP+TN)/(TP+FP+TN+FN)

- F-score:

The F-score is the harmonic mean of precision and recall and is written as follows:

F-score = (2* Precision*Recall)/(Precision+Recall)

A value closer to 1 implies a better combination of precision and recall achieved by the classifier.

6-3- AUC Curve

To evaluate the performance of the models on the positive and negative classes of the test set, we employed the AUC curve. A high value of the AUC score indicates that the model performs better on the positive and negative classes. The AUC scores achieved for the three predictive models used to predict the target variable Churn are shown in Table 3 and Fig. 39, graphically represents the AUC scores obtained for Bagging SVM, k-NN and Random Forest. According to the AUC scores, all the selected models perform well on the test set. However, the most adequate classifier is Bagging SVM with an AUC score of 84%.

Table 3: Comparison of the model used

Model	***Accuracy %***	***Recall %***	***Precision %***	***F-score %***	***AUC Score %***
Bagging SVM	80.26	80.63	79.71	80.16	84
KNN	79.03	76.30	76.30	76.92	80
Random Forest	79.47	79.44	78.91	79.14	82

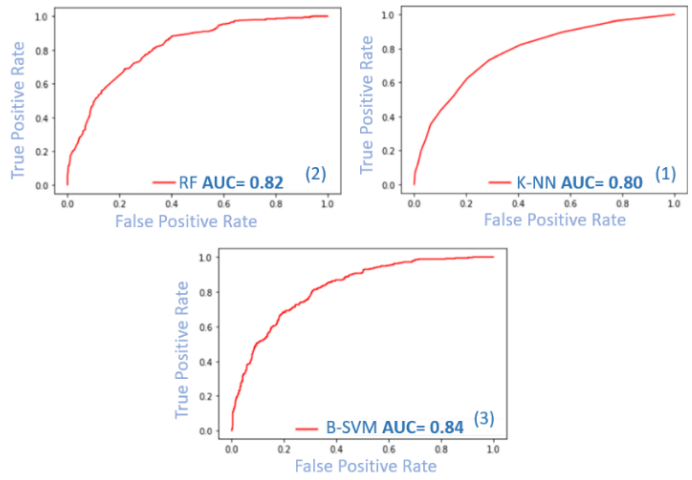


Fig. 39 Models AUC curve. (1): K-Nearest Neighbors. (2): Random Forest. (3): Bagging SVM.

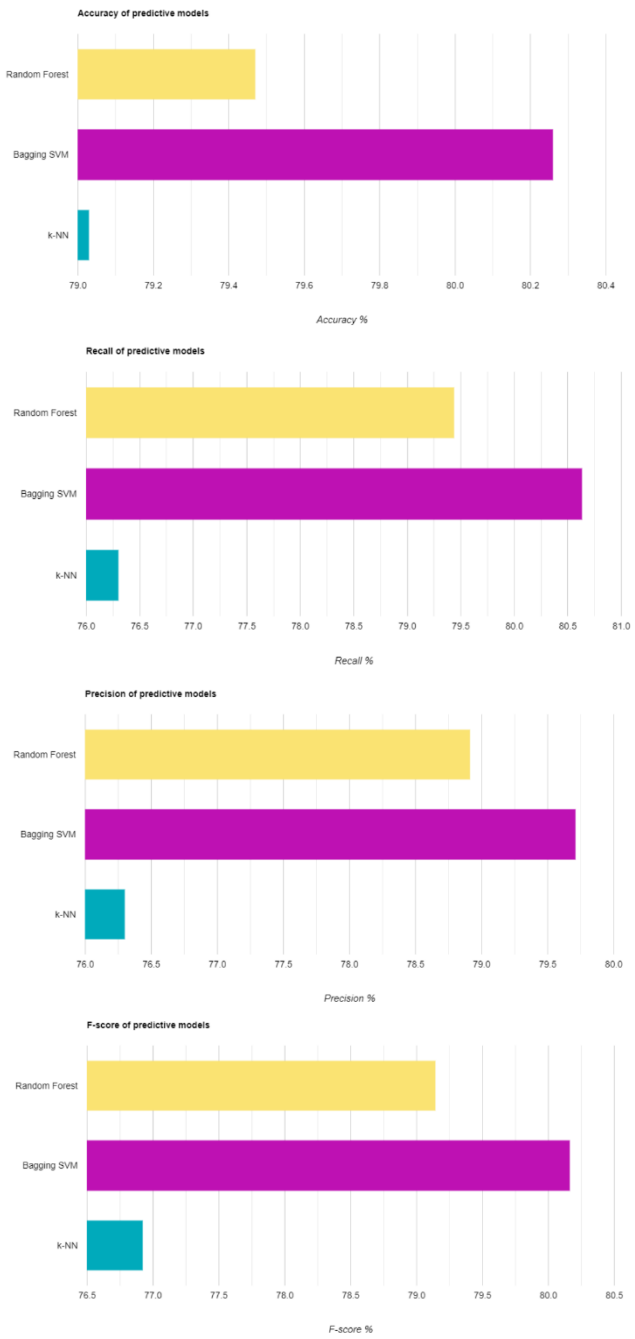


Fig. 40 Predictive models performance indicators: Accuracy, Recall, Precision, F-score

7- Conclusion

In the telecommunications sector, churn prediction is an issue that has attracted the interest of various researchers in recent years. It is becoming one of the sources of revenue for companies and helps to prevent customers from terminating their contracts, it opens the possibility of renegotiation in order to retain the customer by implementing retention strategies.

In this paper, we provide a comparative study of churn prediction in the telecom sector using well-known machine learning techniques such as random forest, k-nearest neighbor, and support vector machine. The experimental results show that all three machine learning techniques give high accuracy for churn prediction. The SVM and bagging classifiers as an ensemble method outperformed the other algorithms in terms of all performance measures such as accuracy, precision, F-measure, recall and AUC score. Predicting churn for a business is proving to be a very tedious task so there is stiff competition in the market to retain customers by providing services that are beneficial to both parties. In futurity, with the upcoming concepts and frameworks in the field of reinforcement learning and deep learning, machine learning proves to be one of the most used and efficient ways to overcome the problems like churn prediction with better accuracy and precision.

References

[1] M. Loukili, F. Messaoudi, and M. El Ghazi, "Supervised Learning Algorithms for Predicting Customer Churn with Hyperparameter Optimization", International Journal of Advances in Soft Computing & Its Applications, Vol. 14, No. 3, 2022, pp. 49-63. doi: 10.15849/IJASCA.221128.04

[2] K. Matuszelański, and K. Kopczewska, "Customer Churn in Retail E-Commerce Business: Spatial and Machine Learning Approach". J. Theor. Appl. Electron. Commer. Res. 2022, 17, pp. 165-198. https://doi.org/10.3390/jtaer17010009.

[3] H. Abbasimehr, M Setak, and M Tarokh, "A neuro-fuzzy classifier for customer churn prediction", International Journal of Computer Applications, Vol. 19, No. 8, 2011, pp. 35-41.

[4] A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform". Journal of Big Data, Vol. 6, No. 1, 2019, pp. 28

[5] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, "Churn prediction : Does technology matter", International Journal of Intelligent Technology, Vol. 1, No. 2, 2006, pp. 104-110

[6] I. Brându¸soiu, G. Toderean, and H. Beleiu, "Methods for churn prediction in the pre-paid mobile telecommunications industry", in 2016 International conference on communications (COMM), IEEE, 2016, pp. 97-100.

[7] K. Coussement, and D. Van den Poel, "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques", Expert systems with applications, Vol. 34, No. 1, pp. 313-327

[8] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, "Computer assisted customer churn management: State-of-the-art and future trends", Computers & Operations Research Vol. 34, No. 10, 2007, pp. 02-29.

[9] K. Dahiya, and S. Bhatia, "Customer churn analysis in telecom industry", in 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO), Trends and Future Directions, 2015, pp. 1-6.

[10] L. Bottou, "Large-scale machine learning with stochastic gradient descent", in Proceedings of COMPSTAT'2010, 2010, Physica-Verlag HD, pp. 177-186.

[11] S. Suthaharan, "Support Vector Machine in Machine learning Models and Algorithms for Big Data Classification", Integrated Series in Information Systems, Springer, New York, Vol. 36, 2016, pp. 207-235.

[12] S. F. Sabbeh, "Machine-learning techniques for customer retention: A comparative study", International Journal of Advanced Computer Science and Applications, Vol. 9, No. 2, 2018.

[13] H. C. Kim, S. Pang, H. M. Je, D. Kim, and S. Y. Bang, "Support vector machine ensemble with bagging", Berlin, Heidelberg, Springer, 2002, pp. 397-408.

[14] H. Abbasimehr, M. Setak, and M. J. Tarokh, "A Comparative Assessment of the Performance of Ensemble Learning in Customer Churn Prediction", Int. Arab J. Inf. Technol, Vol. 11, No. 6, 2014, pp. 599-606.

[15] S. Tavassoli, and H. Koosha, "Hybrid Ensemble Learning Approaches to Customer Churn Prediction", Kybernetes, 2021.

[16] A. Mishra, and U. S. Reddy, "A comparative study of customer churn prediction in telecom industry using ensemble-based classifiers", in 2017 International Conference on Inventive Computing and Informatics (ICICI), 2017, IEEE, pp. 721-725.

[17] N. Ali, D. Neagu, and P. Trundle, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets", SN Applied Sciences, Vol. 1, 2019, pp. 1-15.

[18] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random Forests and Decision Trees, International", Journal of Computer Science Issues, Vol. 9, No. 5, 2012, pp. 272-275.

[19] A. Alamsyah, and N. Salma, "A Comparative Study of Employee Churn Prediction Model", in 2018 4th International Conference on Science and Technology, IEEE, 2018, pp. 1-4.

[20] M. Loukili, F. Messaoudi, and M. El Ghazi, "Sentiment Analysis of Product Reviews for E-Commerce Recommendation based on Machine Learning", International Journal of Advances in Soft Computing & Its Applications, Vol. 15, No. 1, 2023, pp. 1-13.

Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 209-221
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Recognition of Facial and Vocal Emotional Expressions by SOAR Model

Matin Ramzani Shahrestani[1], Sara Motamed[2*], Mohammadreza Yamaghani[3]

[1].Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran
[2].Department of Computer, Fouman & Shaft Branch, Islamic Azad University, Fouman, Iran
[3].Department of Computer, Lahijan Branch, Islamic Azad University, Lahijan, Iran

Received: 23 Oct 2022/ Revised: 04 Dec 2022/ Accepted: 10 Jan 2023

Abstract

Todays, facial and vocal emotional expression recognition is considered one of the most important ways of human communication and responding to the ambient and the attractive fields of machine vision. This application can be used in different cases, including emotion analysis. This article uses six basic emotional expressions (anger, disgust, fear, happiness, sadness, and surprise), and its main goal is to present a new method in cognitive science, based on the functioning of the human brain system. The stages of the proposed model include four main parts: pre-processing, feature extraction, feature selection, and classification. In the pre-processing stage, facial images and verbal signals are extracted from videos taken from the enterface'05 dataset, noise removal and resizing is performed on them. In the feature extraction stage, PCA is applied to the images, and the 3D-CNN network is used to find the best features of the images. Moreover, MFCC is applied to emotional verbal signals, and the CNN Network will also be applied to find the best features. Then, fusion is performed on the resulted features and finally Soar classification will be applied to the fused features, to calculate the recognition rate of emotional expression based on face and speech. This model will be compared with competing models in order to examine the performance of the proposed model. The highest rate of recognition based on audio-image was related to the emotional expression of disgust with a rate of 88.1%, and the lowest rate of recognition was related to fear with a rate of 73.8%.

Keywords: Emotion Recognition; Facial and Vocal Emotional Expressions; Cognitive Model; Soar Model.

1- Introduction

Emotion recognition is the task of automatically discovering the precise emotional ideas of a person, which is one of the most important and challenging problems in the field of human-machine communication. Various computational models of emotional learning have been studied in the literature to understand the emotional expression of people. Since the brain is the main organ of the human central nervous system, computational models inspired by the human brain have a high ability to recognize and classify the pattern [1]. Emotion analysis thought mining, and analyzing subjectivity are related fields of research that use different techniques derived from Natural Language Processing (NLP), Information Retrieval (IR), and Structured and unstructured Data Mining (DM). Most parts of available data throughout the world are unstructured (like text, speech, verbal and visual), which leads to significant research challenges [2]. Ekman et al. have examined facial changes in the form of these muscles' activity, and have collected some of them in the form of the Facial Action Coding System (FACS) [3]. In this system, action units refer to a change in the face that, firstly, can be done alone, and secondly, is indivisible. The limitation of this system is that the expression of action units is only based on local specifications [4]. A new type of facial feature that is commonly used is classified into two main categories of appearance and geometric features. Geometric features refer to the shape and location components of the face, such as eyes, eyebrows, lips, etc., and display the appearance feature, and facial texture, such as wrinkles, ridges, and dimples. Geometric features are obtained based on the alignment results of face components through the Active Appearance

✉ **Sara Motamed**
motamed.sarah@gmail.com

Model (AAM) method [8]. Meanwhile, the representation of local binary patterns and local phase quantization are two representations of appearance-based feature extraction methods [9-15].

Existing research, in general, uses two types of dynamic and static classification. Dynamic classifiers (Hidden Markov Model (HMM)) use several video frames and perform classification by analyzing time patterns from the analyzed areas, or extracted features. Static classifiers classify each frame in a video into a category of facial expressions, based on specific video frame results. In general, these methods are such that first some features of the images are extracted, then they are classified into a classification system, and as a result, one of the emotional categories will be selected. Automatic emotion recognition from face video images encountered many challenges, including finding the face in the image, localizing the dimensions of the eyes, nose, and mouth, revealing the changes in the face and its components, during a certain period of time, and also establishing the relationship between these changes with the person's emotional expression. Each of these issues has its own variation, depending on environmental and personal conditions. For example, at the time detecting a face and finding the exact location of its components, the composition, and makeup of the facial appearance, such as wearing glasses and the angle of the head, results in several problems, each of which has been the subject of extensive independent research. In order to identify emotion from video images, Yacoob and Black model locally the facial action in different areas [16].

In [41], a novel approach is proposed to utilize two face images. In the proposed method, the face component displacements are highlighted by subtracting neutral image from emotional image; then, LBP features are extracted from the difference image. The proposed method is evaluated on standard databases and the results show a significant accuracy improvement compared to DLBPHS.

In [42], we propose an Affine Graph Regularized Sparse Coding approach for resolving this problem. We apply the sparse coding and graph regularized sparse coding approaches by adding the affinity constraint to the objective function to improve the recognition rate. Several experiments has been done on well-known face datasets such as ORL and YALE. The first experiment has been done on ORL dataset for face recognition and the second one has been done on YALE dataset for face expression detection. Both experiments have been compared with the basic approaches for evaluating the proposed method. The simulation results show that the proposed method can significantly outperform previous methods in face classification. In addition, the proposed method is applied to KTH action dataset and the results show that the proposed sparse coding approach could be applied for action recognition applications too.

In [43], focuses on facial expression to identify seven universal human emotions i.e. anger, disgust, fear, happiness, sadness, surprise, and neu7tral. Unlike the majority of other approaches which use the whole face or interested regions of face, we restrict our facial emotion recognition (FER) method to analyze human emotional states based on eye region changes. The reason of using this region is that eye region is one of the most informative regions to represent facial expression. Furthermore, it leads to lower feature dimension as well as lower computational complexity. The facial expressions are described by appearance features obtained from texture encoded with Gabor filter and geometric features. The Support Vector Machine with RBF and poly-kernel functions is used for proper classification of different types of emotions. The Facial Expressions and Emotion Database (FG-Net), which contains spontaneous emotions and Cohn-Kanade(CK) Database with posed emotions have been used in experiments. The proposed Method was trained on two databases separately and achieved the accuracy rate of 96.63% for spontaneous emotions recognition and 96.6% for posed expression recognition, respectively.

In [44], introduces a new classification method using multi-constraints partitioning approach on emotional speech signals. To classify the rate of speech emotion signals, the features vectors are extracted using Mel frequency Cepstrum coefficient (MFCC) and auto correlation function coefficient (ACFC) and a combination of these two models. This study found the way that features' number and fusion method can impress in the rate of emotional speech recognition. The proposed model has been compared with MLP model of recognition. Results revealed that the proposed algorithm has a powerful capability to identify and explore human emotion.

Kumari et al., (2015), stated that the facial expression recognition system has many applications, and is not just limited to understanding human behavior, revealing mental disorders, and expressing human structure. Two popular methods for automated FER systems are geometry-based and appearance-based [37]. The emotional expression of the face is divided into six categories: anger (combination of lowering of the eyebrows, raising of the upper eyelid, narrowing of the eyelids, tightening of the lips), disgust (combination of the wrinkle of the nose, lowering of the corners of the lips, lowering of the upper lip), fear (combination of raising the eyebrow, raising the upper lip, narrowing of the eyelids, widening of the lips, dropping the jaw), happiness (combination of raising the chin, closing the corners of the lips), sadness (combination of raising the eyebrow, closing the corner of the lip), surprise (combination of raising the eyebrow, raising the upper eyelid, dropping the jaw) [7,17]. The main advantage of the deep learning neural network approach in facial emotion recognition is its capacity to learn a large

amount of data. Secondly, recent neural network architectures allow end-to-end training from the representation stage to the classification stage, and finally, they have shown high efficiency compared to non-neural network systems and can learn the limits of complex decision-making in classification. The only disadvantage of these networks is their lack of interpretability. Identifying features play an important and influential role in forecasting results. Moreover, to train deep learning neural networks, many data sets are needed for better performance. This article will use the combination model of deep learning neural networks and the Soar model to solve these problems.

Soar is a symbolic cognitive architecture, which implements problem-solving as behavior in line with the goal, and includes research through the problem and learning from its results [5]. This architecture is used for a wide range of applications, such as routine tasks and solving open problems, which are common for artificial intelligence, and also for interacting with the outside world, either in simulation or in reality [6]. In Soar architecture, the problem is solved by decomposing the goal into hierarchical sub-problems. Procedural memory stores previous states of problem-solving, while semantic memory deals with known evidence.

The proposed model of this article consists of the main parts of pre-processing, feature extraction, feature selection, and classification. First, pre-processing stage, facial images, and verbal signals are extracted from videos, and noise removal and resizing are performed on them. The output of this stage will move to the next stage, and in the feature extraction stage, PCA and MFCC will be applied for face image and verbal signal, respectively. In the best features selection stage, 3D-CNN face images and CNN verbal signals will be applied to the obtained features. Finally, and in the fusion stage, fusion is performed on the best features, and the final vector will be sent to the Soar classifier, to calculate the recognition rate of emotional expression based on facial and verbal emotional expressions.

In the feature extraction stage, the reason for choosing PCA on face images is to use different components to achieve more details and simplify them, and choosing MFCC on verbal signals leads to the necessary compression and understanding of the verbal source, and contributes to the necessary identification and diagnosis. Moreover, in the feature selection stage, the reason for choosing 3D-CNN on face images is that no dimension is removed from the images, and considering the temporal information in dynamic images, it leads to more efficient and better feature selection. CNN networks are used to select the best verbal features because they are very resistant to changing the image's scale and size, and even in noisy images, they provide a suitable response to the output. On the other hand, fusion leads to obtaining a uniform vector for single and optimal use in decision making. This is why we use the fusion method in this paper before using the Soar classifier. This article is organized as follows: Section 2 cognitive science models are introduced. In Section 3, the proposed method is introduced. Experimental results and conclusions are presented in Sections 4 and 5, respectively.

2- Recognition of Facial and Vocal Emotional Expression

The verbal signal is the fastest and most natural way of communication. Accordingly, speech is used as a fast and efficient method for human-computer interaction. So far, many efforts have been made to verbal recognition. Despite many advances in this field, there is a long gap between natural human-computer interactions. The main reason is the computer's inability to understand the user's feelings. Therefore, in the last few years, emotion recognition by speech is one of the challenging issues in the field of verbal processing. In addition, emotion recognition from speech can be used to extract useful meanings from speech, as well as, is used to recognize the user's personality, which is used in many organizations, including political, military, etc. Emotion recognition from speech has various applications, which have been addressed from different aspects. Among its applications are the development of automatic verbal recognition systems, text-to-speech conversion, driver's mental states report, computer games [12], diagnosis tools in medical sciences [13], for disabled people or autistic children, in order to communicate with others [14], and also as an application in mobile phones [15]. Considering the importance of recognizing facial and vocal emotional expressions, this article presents a practical method inspired by the human brain system.

3- Proposed Model

In order to recognize the emotional verbal and facial expressions, this article has used the fusion of appropriate features resulting from signals and emotional images, and then applied Soar classification. Considering Fig. 1, the procedure sequence of the proposed model includes four main parts: preprocessing, feature extraction, feature selection, and classification. In the first steps, in order to extract suitable features, first pre-processing, and then PCA and MFCC will be applied to face images and verbal signals, respectively. The output of the extracted features is entered into CNN and 3D-CNN so that more suitable features are selected. Before applying the classifier, fusion is applied to all the selected features, and finally, the Soar

classifier will be applied in order to examine the emotional verbal-visual recognition rate.

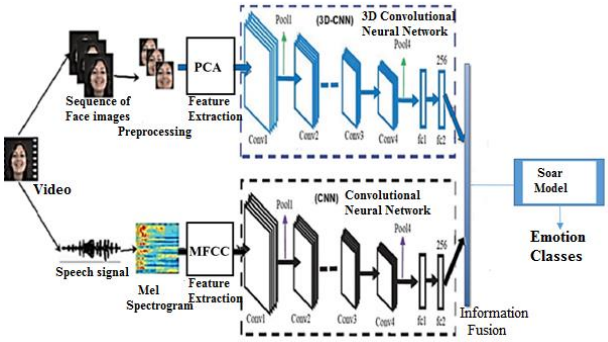


Fig. 1. Outline of the proposed method

According to Fig. 1, the first step is to read the videos from the enterface'05 dataset [30]. To separate the verbal signal and extract the image frames from the video, the separation algorithm with a certain frame rate, and also the separation of the speech from the video will be applied to all the samples. Face images and verbal signals are stored in specific folders, and then pre-processing operations are applied to the frames of images and resulted in speech from the video. In the following, PCA and MFCC are used to extract features from face images, and for the verbal signal, respectively. PCA is an accurate quantitative method to achieve this simplification. This method produces a new set of variables, called principal components. MFCC also compresses the information related to the verbal system, based on understanding, to a small number of coefficients, which this simplification greatly contributes to overall recognition. Moreover, video samples are divided into three categories: training, validation, and testing. In the end, the image frames of each video sample are sent to the 3D-CNN network in the form of a four-dimensional matrix, including the number of sequences of frames of each video, width, height, and the number of images channels [19]. The resulted speech from the video is also sent to CNN for feature selection. The reason for selecting 3D-CNN is that this method is can store data in the entire network, and also can deal with incomplete knowledge, and also tolerate high error. Therefore, it is expected to select appropriate features. The reason for selecting CNN, in addition to the above reasons and speech will be two-dimensional. Finally, in order to classify the obtained features, the Soar method has been used. In Soar, all goal-oriented symbolic tasks are formulated in problem spaces. A problem space consists of a set of states and a set of operators. States represent situations, and operators refer to actions that, when applied to states, lead to other states. Each functional context contains a goal, roles for a problem state, a state, and an operator. Problem-solving is driven by decisions that lead to the selection of problem spaces, states, and operators for the respective context roles. According to a goal, a problem space should be selected, in which the achievement of the goal can be pursued. Then, an initial state must be selected, which represents the initial state. Then, to apply in the initial state, an operator must be selected [24]. This process continues until a sequence of operators is discovered, which transforms the initial state into a state in which the goal is achieved [31].

Emotion recognition is a problem with many parameters and close to each other, and since the Soar architecture has not been used in this field and this model has many free parameters, it can be a suitable model for prediction. Also, this model is inspired by the human brain system, and we believe that any model inspired by the human biological system can work properly.

3-1- Pre-Processing

According to Fig. 1, the first step is to read the video from the enterface'05 dataset and pre-process the images. The separation of image and voice will be performed using call and separation functions in PYTHON. To extract the frames of face images from the video, the separation algorithm with a certain frame rate will be applied to all samples. In these tests, the frame rate of the images is 10 (that is, it extracts 10 frames per second). The images are stored in specific folders, and then, pre-processing operations are applied to the resulted frames from the video. This operation includes face detection, face alignment, and image resizing. The image size of each video frame is changed to 3x100x96 resolution. Moreover, the Mel spectrogram function has been used for verbal conversion, and the general algorithm has been used to calculate the Mel spectrogram.

3-2- Feature Extraction from Images and Speech by PCA and MCCF

PCA is an accurate quantitative method to achieve this simplification. This method results in a new set of variables called principal components. The PCA function is used to find the principal components. PCA is a quantitatively accurate method to achieve this simplification. This method produces a new set of variables called principal components. The reason for choosing PCA is to use different components to achieve more details and simplify them. Fig. 2 shows an example of feature extraction from an image using PCA.

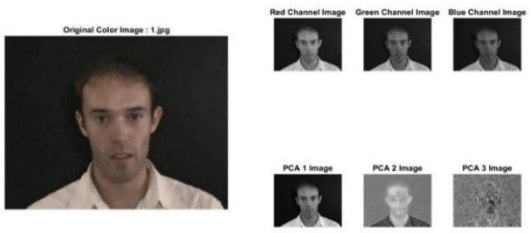


Fig. 2. An example of feature extraction from an image by PCA

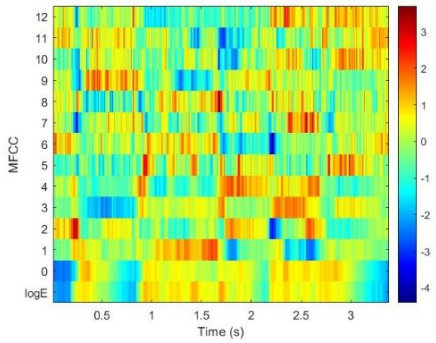


Fig. 3. Example of feature extraction from speech file images by MFCC

Mel-frequency spectrum coefficients are popular features extracted from verbal signals for use in recognition tasks. In the source-filter model of speech, brain coefficients are understood as a filter representative (verbal system). The frequency response of the speech system is relatively smooth, while the spoken verbal source can be modeled as a shock train. As a result, the verbal system can be estimated by the spectral coverage of a verbal segment. The motivational idea of the Mel frequency envelope coefficients is to compress the information about the verbal system (smooth spectrum) into a small number of coefficients based on cochlear perception. Although there is no exact standard to calculate coefficients, the basic steps are outlined in the diagram. Fig. 3 shows an example of feature extraction from speech using MFCC.
According to Fig. 3, the horizontal axis shows the time, and the vertical axis is the calculated value for 12 filters, in which the signal input energy in the time and frequency domain results in a distance between these filters, and different colors.

3-3- Selection of Image and Speech Frame Features using 3D-CNN and CNN

3D-CNN (convolutional neural network) model is used to represent the features of the facial emotional expression in the video [21]. Fig. 4 shows the architecture of the facial expression recognition model using 3D-CNN.
of the facial expression recognition model using 3D-CNN.

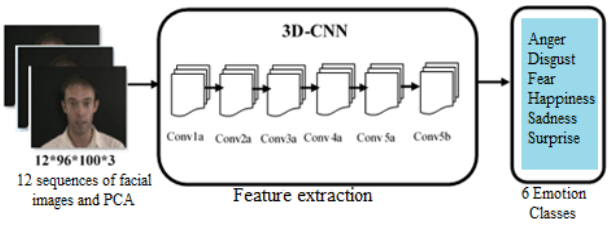


Fig. 4. The architecture of facial expressions recognition model, using 3D-CNN [20]

According to Fig. 4, 12 3D-sequences, with a size of 96*100*3, are entered into the 3DCNN as a feature selection step, and Conv and Pool layers are applied to them. This function and Leaky ReLU have been used due to the popularity of the RelU function in transfer functions, and its operation is shown according to Eqn. 1 and Eqn. 2.

$$RelU \Rightarrow \max(0, x) \quad (1)$$

$$Leaky\ RelU \Rightarrow \max(0.1x, x) \quad (2)$$

3-4- Information Fusion

Information fusion is a set of activities that by using information from several resources obtain more and more accurate information about a subject [32]. The need for such activities is evident in several aspects. In classification problems, considering the structure of the problem and the selected solution, the information fusion is performed in four levels: data fusion, features, classification, and decision making. If different sensors measure the same quantities of the same phenomenon (such as seeing an object from different angles, or from a combination of basic colors), the fusion process is a process at the data level. Moreover, the fusion process can be delayed to the feature level. Many psychological studies have shown theoretically and experimentally, the importance of integrating information from several modalities (for example, emotional verbal, and facial expressions in this design) to obtain a coherent representation and inference of emotion [32-33]. As a result, in recent years, many studies have been conducted on the recognition of human verbal-visual emotion [34-35]. In addition, there are studies in the field of other resources, such as information fusion of facial expressions and head movement [31] and also the fusion of facial expressions and body movement or behavior [36], most of which are independent of the verbal text. In 1971, Friesen and Ekman identified six distinct emotional categories; each characterized by a specific facial expression. These six categories, which include happiness, sadness, anger, surprise, fear, and disgust, are common to all human beings of all nationalities and are called basic emotions [38]. Cohen et al. (2000) have dealt with the automatic face recognition of live video, and to this end, they used dynamic programming methods of pattern pairing and the

Hidden Markov Model (HMM). This has presented existing methods and a new HMM structure for automatic human face recognition from video sequences. The advantage of this structure is that both segmentation and face recognition have been performed automatically using HMM, increasing the differentiation between different face classes. In this work, the dependent and independent expression of the face was investigated, and the average total ratios of detecting the dependent emotional expressions of the person's face, using the special sense of one-dimensional HMM and multi-level HMM, were 78.49 to 82.46%, respectively [6]. Fusion has different levels: information fusion at the feature level, information fusion at the classification level, and information fusion at the decision-making level. According to the performed investigations, we decided to focus on fusion at the feature level due to the combination of the product of the independent features of verbal and facial expression.

Using Fusion for data classification provides results with high accuracy, sensitivity, specificity, accuracy and consistency. This leads to improved modeling metrics such as MCC (Matthews Correlation Coefficient), which is reliable, robust and effective. Also, the amount of rating error is reduced. It extracts better, more important, useful, richer, valuable, relevant, and meaningful information and improves the signal-to-noise ratio (compared to irrelevant information) and provides promising, better, and reasonable results, resulting in a decision. It gets better [39].

3-5- Classification by Soar learning Model

To recognize emotion from facial expressions in the video, the Soar learning model is used [5, 6]. Soar is used as a method based on brain architecture and pattern-based learning. In this method, first, patterns are learned on the basis of deep learning and memory mechanisms, and then, are recognized on the basis of the learned model.

Soar is a symbolic cognitive architecture, which implements problem-solving as behavior in line with the goal, and includes research through the problem and learning from its results. This architecture is used for a wide range of applications, such as routine tasks and solving open problems, which are common for artificial intelligence, and also for interacting with the outside world, either in simulation or in reality. In Soar architecture, the problem is solved by decomposing the goal into hierarchical sub-problems. Procedural memory stores previous states of problem-solving, while semantic memory deals with known evidence [5].

The working memory of an agent is used to evaluate its current position. To this end, this memory uses the perceptual information received through its sensors, as well as the information stored in the long-term memory. Working memory is also responsible for making motor commands or selective actions by the decision-making process module, selecting operators, and tracking potential bottlenecks. In Soar design, decision-making is controlled by the pattern matching part of production rules. An important issue in Soar decision-making is the selection of operators used to change the problem state. Multiple operators can be selected for a problem, and in the case of conflict, it can be resolved through a preference structure. The preference structure can select operators according to different criteria. By going through these options, the seriousness and difficulty of the criteria are reduced, and they show a trend of more adaptability and more innovation in generating ideas; therefore, the preference structure can reflect the problem-solving style as a kind of superiority, which an operator may choose, and move along different problem-solving paths [6].

Algorithm: Generalized SOAR

Input: A, q_0, p_0 and an integer order n

Output: the orthonormal matrix Q_n

1. $\beta = \|q_0\|$
2. $\begin{bmatrix} q_1 \\ p_1 \end{bmatrix} = \frac{1}{\beta}\begin{bmatrix} q_0 \\ p_0 \end{bmatrix}$
3. For $j = 1, 2, \cdots, n-1$
4. $\quad \begin{bmatrix} q_{j+1} \\ p_{j+1} \end{bmatrix} = A\begin{bmatrix} q_j \\ p_j \end{bmatrix}$
5. $\quad$ For $i = 1, 2, \cdots, j$
6. $\quad\quad h_{ij} = q_i^T q_{j+1}$
7. $\quad\quad \begin{bmatrix} q_{j+1} \\ p_{j+1} \end{bmatrix} = \begin{bmatrix} q_{j+1} \\ p_{j+1} \end{bmatrix} - h_{ij}\begin{bmatrix} q_j \\ p_j \end{bmatrix}$
8. $\quad$ End
9. $\quad h_{j+1\ j} = \|q_{j+1}\|$
10. $\quad$ If $h_{j+1\ j} \cong 0$, stop (breakdown)
11. $\quad \begin{bmatrix} q_{j+1} \\ p_{j+1} \end{bmatrix} = \frac{1}{h_{j+1\ j}}\begin{bmatrix} q_{j+1} \\ p_{j+1} \end{bmatrix}$
12. End
13. $Q_n = [q_1 \quad \cdots \quad q_n]$

Fig. 5. Soar architecture algorithm [5]

Q is the canonical response matrix consisting of <mode, state> pairs, which are performed according to the execution of different steps. And the inputs are the states that are determined based on the input matrix A with the matrix P. The details of the method were described in [40]. The decision-making process in Soar is the same as in other systems: it involves matching and firing rules, which is, in fact, a context-dependent representation of knowledge. These conditions describe the current situation of the agent, and the body of rules describes the actions that lead to the creation of structures related to the current conditions in the working memory [22]. Fig. 5 shows the architectures of the classic and current versions of Soar. The classical Soar architecture includes two types of memory: (1) a long-term symbolic memory, which is

encoded using production rules, and (2) a short-term working memory, which is encoded as a graph structure, to make possible the representation of objects in detail and relationships between them [6]. Soar has a deep learning component, which is related both to procedural knowledge and working memory (the latter through the evaluation component). Reinforcement learning in Soar is relatively simple. This adjusted action selection is based on environmental numerical rewards. This evaluation has been applied to all goals and sub-goals in the system, which enables quick identification of good and bad operators for specific conditions [23]. Evaluations lead to evaluations of how goals are achieved, which in turn affects the feelings of the agent. These emotions express the intensity of emotion and thus act as an intrinsic reward for reinforcement learning. Currently, emotions in Soar only connect reinforcement learning with working memory [24]. Fig. 5 shows the Soar algorithm.

4- Results

Python software is used to simulate the proposed model. A validity test was used in all experiments using the K-Fold method and K=10 [25].

4-1- Enterface'05 Dataset Verbal-Visual Emotion

The verbal-visual database enterface'05 was developed in 2005 and included a number of video files [26]. This database consisted of 44 people, each of whom expresses 6 basic emotions (anger, disgust, fear, happiness, sadness, surprise) with five different scenarios and sentences. In total, the database contains 1320 samples of verbal-visual clips, whose verbal signal was recorded with a sampling rate of 48000 Hz, with a resolution of 16 bits and one channel. The average length of each video is 3-4 seconds, and the database language is English. Fig. 6 shows an example of arranged images of different facial expressions in the enterface'05 database [27].

Fig. 6. Some examples of arranged images of different expressions in the enterface'05 database

4-2- Results' Analysis

4.2.1 Analyzing the Results of Facial Emotional Expression

In order to emotion recognition from the facial emotional expressions in the video, in the first step, the database is prepared, and then, the pre-processing operation is performed. In the database preparation phase, first, the database is divided into 2 training and testing sections. So that 70% of the database is considered for training, and 30% for testing. Separation of the image formation is done using call and separation functions in PYTHON. In order to feature extraction, PCA has been used. The complete set of principal components is the same as the principal set of variables. But it is common that the sum of the variances of several principal components exceeds 80% of the total variance of the original data. By examining plots of these new multivariate, researchers often gain a deeper understanding of the driving forces that produced the original data. To use PCA, actual measured data to be analyzed are required [28]. As shown in Fig. 4, a sequence of 10 frames of color face images, each with a size of 96x100, is fed to the 3D convolutional neural network, and the output is equal to the number of basic emotion classes. Table 1 shows the number of layers, the filters of each layer, and the size of each filter in the structure of the 3D-CNN model.

Table 1. The structure details of the used layers in the 3D-CNN model

32 Filters 3*3*3	conv1
2 steps	pool1
64 Filters 3*3*3	conv2
2 steps	pool2
128 Filters 3*3*3	conv3
2 steps	pool3
1024	Fully Connected
256	Fully Connected
Classes Count=6	Softmax

The evaluation indices in this article are accuracy rate, precision rate, and recall rate [29]. The proposed model applied 3D-CNN to select the best features from the emotional expressions of facial images, and the results are shown in Fig. 7.

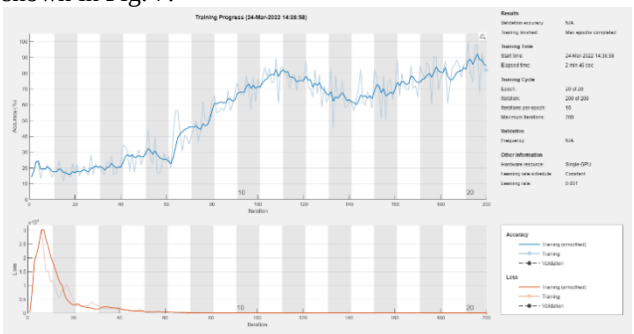


Fig. 7. The Soar output on the 3D-CNN features selected from images

According to Fig. 7, the training in the network started with very low values close to an accuracy of 20%, and

these values increased with the considered number of iterations, which finally reached the final value of 82% in the amount of 200 iterations. In the bottom part of the figure, the values of missing information or failed goals can be seen, which were very high in the initial iterations, and in the following, the value was close to zero. According to the proposed method, the number of input frames to this stage can be different, and Table 2 shows the recognition value in different numbers of frames.

Table 2. Comparison of recognition accuracy according to different sequences of facial expressions images

Count sequences of facial expressions images	accuracy
6	57%
8	62%
10	62.5%
12	63%

As shown in Table 2, by using 10 sequences of images, the recognition accuracy has reached 62.5, which is the highest value compared to 6 and 8 sequences. Of course, this accuracy is only 0.5% different compared to 8 sequences, and if the speed and quantity of identification are considered, 8 sequences can be considered, and in the case of increasing the quality index, 10 sequences can be considered. 12 sequences were considered in the proposed method. After performing the pre-processing operation, the number of sample frames can be less or more than the frames required for processing. If it is less, we copy several copies of the last frame as much as the difference with the required number of frames, and if it is more, we select the coefficients of the frames.

For the proposed facial expression recognition model using 3D-CNN, the learning values are determined as follows. The size of each batch is 32, the number of iteration steps is 500, the learning rate is 0.001, and momentum rate is 0.0001. In this method, the early stopping technique is used during training. To examine the used layers in this model, we have displayed the image of the middle layers for each frame. The facial expression recognition test was performed on the enterface'05 database for 1005 training video samples and 250 test video samples in the form of cross-validation. The results of this test are shown in Tables 3 and 4.

According to Table 3, by training the model and memorizing the detection method in the proposed Soar model, the test data set centered on each of the data set emotions has been presented to the model, and the results have been recorded. Considering the way of selecting the test set, and completely different examples from the training set, "disgust" and "surprise" have shown the highest and lowest recognition rates, respectively. For example, out of 100% of test samples with "disgust " emotion, 86% were correctly recognized, and 14% of the rest were recognized in other emotions. According to Table 4, "disgust " and "happiness" have the highest recognition accuracy in recognizing facial expressions, and "anger" and "fear" have the lowest recognition accuracy.

Table 3. Confusion matrix of facial expression recognition on the enterface'05 database

	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	47%	7%	15%	6%	10%	15%
Disgust	2%	86%	5%	0%	5%	2%
Fear	7%	7%	55%	5%	17%	10%
Happiness	2%	12%	5%	67%	5%	10%
Sadness	5%	2%	14%	2%	67%	10%
Surprise	10%	5%	17%	10%	15%	44%

Table 4. Confusion matrix of facial expression recognition on the enterface'05 database

	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	77%	3%	7%	3%	4%	6%
Disgust	1%	91%	3%	1%	2%	2%
Fear	1%	4%	81%	5%	7%	2%
Happiness	2%	3%	2%	90%	2%	1%
Sadness	2%	4%	2%	4%	87%	1%
Surprise	2%	4%	6%	2%	2%	84%

According to Tables 3 and 4, on average, the recognition percentage of facial expressions for 6 basic emotions, for the test data set from the enterface'05 database, was 73%. The recognition rate of the above two tables, which are specified for each of the emotions, is added together and divided by the number of emotions in the two tables, which, based on this calculation, and the overall recognition rate presented in the simulation, the value of 73% has been recorded by the proposed model as the result of the overall recognition of emotions.

Fig. 8 and Fig. 9 show the recognition accuracy and loss error function for different iteration stages and the validation data set in the training stage of the 3D-CNN network for recognizing facial expressions on the enterface'05 database. The horizontal axis in this figure is the number of iteration steps, the vertical axis in Fig. 8 is the recognition accuracy and in Fig. 9, the loss error. According to the figure, the program stopped after about 130 apks.

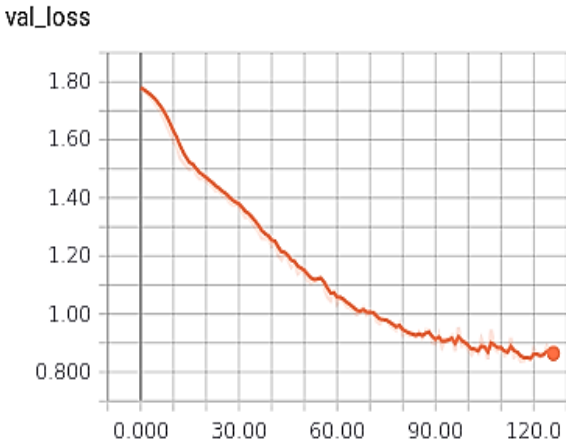


Fig. 8. The loss error of the validation data set, for the number of different iterations in the training phase

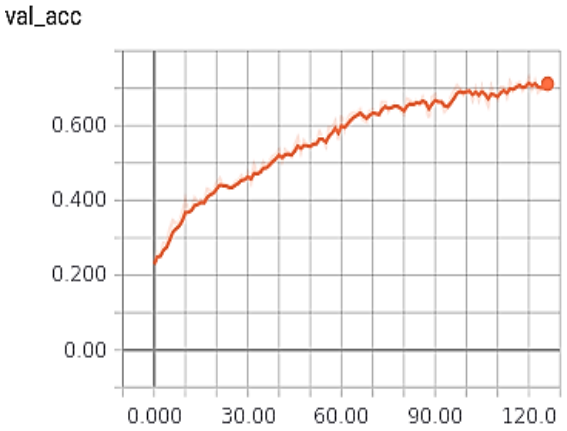


Fig. 9. Recognition accuracy of the validation data set, for the number of different iterations in the training phase

4.2.2 Analyzing the Results of Verbal Emotional Expression

In the part of verbal expression recognition, the Mel spectrogram function follows the general algorithm to calculate the Mel spectrogram, which is explained below. In this algorithm, the audio input is first buffered in frames of the number of samples (windows). Frames are overlapped by the number of overlap length samples. The specified window is applied to each frame, and then the frame is converted to a frequency domain representation with the number of FFTLength points. The frequency domain representation can be specified in magnitude or power by Spectrum Type. If Window Normalization is adjusted to true, the spectrum is normalized by the window. Each frame of the frequency domain display passes through a Mel filter bank. The output spectral values of the Mel filter bank are summed, and then the channels are concatenated so that each frame becomes a column vector of NumBands-Element numbers. MFCC has been used to extract features in verbal expression.

The set of extracted verbal features is given to the CNN network to select features, and the output of this network finally indicates the basic emotion class of each verbal file sample. In this experiment, the size of each batch is 32 and the number of iteration steps is 500, the learning rate is 0.001 and the momentum value is 0.0001. In Table 5, the confusion matrix of the presented model for verbal emotion recognition, for 6 basic emotion classes, is shown on the enterface'05 database.

Table 5. Confusion matrix of verbal emotion recognition, using CNN, for 6 basic emotion classes on the enterface'05 database

	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	82%	7%	0%	3%	3%	5%
Disgust	12%	55%	10%	7%	7%	10%
Fear	2%	5%	60%	14%	14%	5%
Happiness	12%	2%	5%	69%	2%	10%
Sadness	5%	5%	10%	0%	71%	10%
Surprise	5%	10%	2%	15%	5%	63%

The horizontal axis in this figure is the number of iteration steps, the vertical axis in Fig. 10 is the recognition accuracy and in Fig. 10, the loss error. According to the figure, the program stopped after about 170 apps (Because from this apps onwards, the recognition accuracy of the validation data will no longer increase)

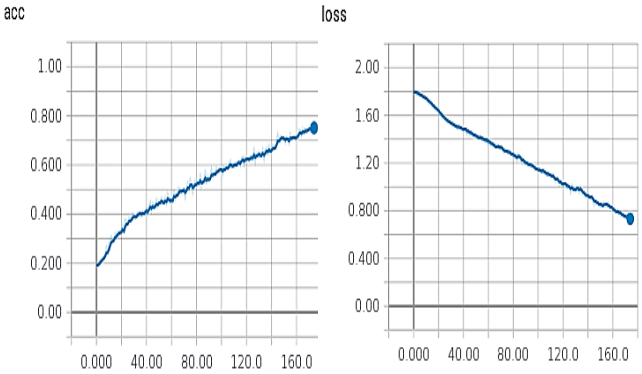


Fig. 10. Recognition accuracy and loss error, for different numbers of iterations in the training phase

4.2.3 Analyzing the Results of Facial and Verbal Emotional Expressions

Obtained features from facial expressions and verbal emotion have been used for fusion. Accordingly, 12 sequences of face images are given to the 3D-CNN network, and after applying different layers, a 256-character feature vector is obtained in the last fully

connected layer. Moreover, the verbal Mel-spectrogram images, in the size of 96x300x1 are given to the CNN neural network, and after applying different layers, a 256 feature vector is obtained from speech. Then, these features are combined and fed into the Soar classification model. This network, after feature fusion, results in the output of the final emotion class.

Feature fusion is the process of combining two feature vectors to obtain a single feature vector, which is more distinct than any of the input feature vectors. The architecture of the proposed emotion recognition model based on the fusion of two verbal-visual processes is shown in Fig. 1. According to this structure, information is obtained from two different paths. One is visual processing, in which after preprocessing, the facial images frames enter the visual cortex of the brain in the thalamus, and through PCA and then the 3D convolutional neural network, the best features of the face images are obtained and the other is audial processing, which after pre-processing operations such as framing and windowing, is converted into Mel-Spectrogram coefficients, and entered into CNN, to select the best features of the signals.

In order to address the advantage of the proposed facial expression recognition model, the efficiency of this model was compared with other methods that used the enterface'05 database. According to Table 6, the results of the proposed model are better than KNN and SVM. Even in order to examine the performance of the proposed model in the experiments, the feature selection part with 3D-CNN is also omitted in the calculations. Due to examining the recognition rate in Table 6, it can be seen that the proposed model provides higher recognition accuracy than other methods.

Table 6. Comparison of facial expression recognition accuracy, between different methods and the proposed model, for 6 classes on the enterface'05 database

Method	Accuracy
PCA+SVM	%64
My Method Without 3D-CNN	%65.5
PCA+KNN	%66.3
My Method	%73

The accuracy of emotion recognition in the proposed model, compared to other models in this field, was investigated and evaluated by researchers, and the results are shown in Table 7.

Table 7. Comparison of the effectiveness of the proposed model in bimodal emotion recognition with other methods performed on the enterface'05 database

Reference	Accuracy
Savestani et al. [28]	70.1%
Mansoorizadeh et al. [10]	71%
Bejani et al. [15]	77.7%
Zhalehpour et al. [24]	77.02%
My Method	81.7%

In order to show the improvement of the algorithms used in this thesis, Table 8 shows the recognition accuracy of all algorithms for different emotional expressions.

Table 8. Comparison of the recognition accuracy of the proposed models for different emotional expressions on the enterface'05 database

Suggested models	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Total Accuracy
Verbal emotional expressions	82%	55%	60%	69%	71%	63%	66.7%
Facial emotional expressions	47%	86%	55%	67%	67%	44%	62%
Fusion of features	85%	88.1%	73.8%	78.6%	81%	78%	81.7%

According to Table 8, the highest recognition rate is related to disgust and anger with a rate of 88.1 and 85%, and the lowest recognition rate is related to fear and surprise with a rate of 73.8 and 78%.

4.2.4 Results Analyze by Selecting the Feature

According to the proposed method, after the stages of feature extraction from facial expressions, using PCA, and from verbal expressions, using MFCC, a feature selection step using CNN and 3DCNN, respectively for verbal and facial expressions has been used, which then, fusion and final classification will take place on it; Table 9 shows the confusion matrix of emotion recognition based on this model, with feature selection.

Table 9. The confusion matrix of emotion recognition, on the basis of the fusion of verbal and visual features, using the Soar model by selecting the feature.

	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	85%	7.1%	2.3%	2.3%	0%	2.3%
Disgust	2.3%	88.1%	2.3%	2.3%	4.7%	0%
Fear	2.3%	0%	73.8%	7.1%	14%	2.3%
Happiness	7.1%	2.3%	4.7%	78.6%	2.3%	4.7%
Sadness	2.3%	0%	9.5%	0%	81%	9.5%
Surprise	2.3%	0%	4.7%	7.1%	7.1%	78%

According to Table 9, the recognition rate related to disgust and anger is 88.1%, and 85%, respectively, which is one of the highest emotion recognition rates, and on the other hand, the recognition rate of fear and surprise is recorded at 73.8% and 78%, respectively, which was the lowest recognition rate.

The emotion recognition accuracy for all classes, using this proposed model, was 81.7% on average. The interesting point is that according to Table 9, in the corresponding enterface'05 database, the emotions "disgust" and "anger" have the highest recognition accuracy, and surprise and fear have the lowest recognition accuracy, with less than 80%; while, the accuracy of other emotions is more than 80%. Moreover, the overlap of fear and sadness is more so that 14% of fear samples are in sadness. In addition, according to the table recognizing "disgust" with 90% accuracy is easier than other emotions.

4.2.5 Results Analyze without Selecting the Feature

According to the proposed method, after the stages of feature extraction from facial expressions, using PCA, and from verbal expressions, using MFCC, a feature selection step using CNN and 3DCNN, respectively for verbal and facial expressions has been used, which then, fusion and final classification will take place on it; Table 10 shows the confusion matrix of emotion recognition based on this model, without feature selection, and accordingly, after feature extraction, there are fusion and classification.

Table 10. The confusion matrix of emotion recognition, on the basis of the fusion of verbal and visual features, using the Soar model without selecting the feature.

	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	82%	8.1%	2.3%	3.3%	0%	3.3%
Disgust	3.3%	86.1%	2.3%	3.3%	4.7%	0%
Fear	3.3%	0%	70.8%	8.1%	14%	3.3%
Happiness	7.1%	3.3%	4.7%	76.6%	3.3%	4.7%
Sadness	3.3%	0%	10.5%	0%	78%	10.5%
Surprise	3.3%	0%	5.7%	7.1%	7.1%	76%

According to Table 10, the recognition rate related to disgust and anger is 86.1%, and 82%, respectively, which is one of the highest emotion recognition rates, and on the other hand, the recognition rate of fear and surprise is recorded at 70.8% and 76%, respectively, which was the lowest recognition rate.

The emotion recognition accuracy for all classes, using this proposed model, was 79.2% on average. The interesting point is that according to Table 7, in the corresponding enterface'05 database, the emotions "disgust" and "anger" have the highest recognition accuracy, and surprise and fear have the lowest recognition accuracy, with less than 80%; while, the accuracy of other emotions is more than 80%.

5- Conclusions

Emotion plays an effective role in humans' non-verbal communication with each other. Automatic emotion recognition can lead to the construction of computer systems that have the ability to understand human emotion and appropriate responses to it. In these computers, natural interaction is with humans. That is, the computer receives and interprets the emotion expressed in facial expressions, and gives an appropriate response based on its judgment. This article presented a method to recognize emotion from people's faces in video images, extract image features due to receiving all discriminating features from PCA, extract verbal features due to compression of the verbal resource and its wide range, and extract MFCC is used appropriately. Then, in order to select the best features in verbal signals and for facial images, CNN and 3DCNN were used, respectively. Next, feature level fusion was performed on the output of the previous stage to combine and achieve the appropriate shape of speech feature vectors and facial expressions. Finally, by using Soar classification, the recognition rate of audio-visual emotional expression was calculated. The highest rate of recognition based on audio-image was related to the emotional expression of disgust with a rate of 88.1%, and the lowest rate of recognition was related to fear with a rate of 73.8%. Moreover, in order to examine the performance of the proposed model, the tests were performed again without selecting the features, the highest recognition rate was related to disgust with a rate of 86.1%, and the lowest was related to fear with a rate of 70.8%. The proposed model has recorded an average improvement of 4 percent compared to the closest model. In addition, compared to the well-known SVM and KNN machine learning models, the results showed that the proposed method had a 6.7% improvement in recognition.

Among the future works, I can mention face recognition using head and body movements, which are involved in the expression of emotions in a special way. Also, in the continuation of this research, other modalities such as text, head and body movements, thermal and infrared images and other physiological features such as EEG can be added to it.

References

[1] Senthilkumar, N., S. Karpakam, M. Gayathri Devi, R. Balakumaresan, and P. Dhilipkumar. "Speech emotion recognition based on Bi-directional LSTM architecture and deep belief networks." Materials Today: Proceedings 57 (2022): 2180-2184.

[2] Crisp, Nicholas H., Peter CE Roberts, Sabrina Livadiotti, A. Macario Rojas, Vitor Toshiyuki Abrao Oiko, Steve Edmondson, S. J. Haigh et al. "In-orbit aerodynamic coefficient measurements using SOAR (Satellite for Orbital Aerodynamics Research)." Acta Astronautica 180 (2021): 85-99.

[3] Vallejo, Carlos, Jun Ho Jang, Carlo Finelli, Efreen Montaño Figueroa, Lalita Norasetthada, Rodrigo T. Calado, Mehmet Turgut et al. "Efficacy and Safety of Eltrombopag Combined with Cyclosporine As First-Line Therapy in Adults with Severe Acquired Aplastic Anemia: Results of the Interventional Phase 2 Single-Arm Soar Study." Blood 138 (2021): 2174.

[4] Whittaker, Jackie L., Linda K. Truong, Trish Silvester-Lee, Justin M. Losciale, Maxi Miciak, Andrea Pajkic, Christina Y. Le et al. "Feasibility of the SOAR (stop OsteoARthritis) program." Osteoarthritis and Cartilage Open 4, no. 1 (2022): 100239.

[5] Laird, John Edwin, Keegan R. Kinkade, Shiwali Mohan, and Joseph Z. Xu. "Cognitive robotics using the soar cognitive architecture." In Workshops at the twenty-sixth AAAI conference on artificial intelligence. 2012.

[6] Stavros, J., and G. Saint. "SOAR: Chapter 18: Linking strategy and OD to sustainable performance." WJ Rothwell, JM Stavros, R. Sullivan, and A. Sullivan, Practicing organization development: A guide for leading change. San Francisco, CA: Jossey-Bass (2010).

[7] Lucey, Simon, Ahmed Bilal Ashraf, and Jeffrey F. Cohn. Investigating spontaneous facial action recognition through aam representations of the face. Vol. 2. INTECH Open Access Publisher, 2007.

[8] Chang, Ya, Changbo Hu, Rogerio Feris, and Matthew Turk. "Manifold based analysis of facial expression." Image and Vision Computing 24, no. 6 (2006): 605-614.

[9] Pantic, Maja, and Ioannis Patras. "Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 36, no. 2 (2006): 433-449.

[10] Guo, Guodong, and Charles R. Dyer. "Learning from examples in the small sample case: face expression recognition." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 35, no. 3 (2005): 477-488.

[11] Anderson, Keith, and Peter W. McOwan. "A real-time automated system for the recognition of human facial expressions." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 36, no. 1 (2006): 96-105.

[12] Whitehill, Jacob, and Christian W. Omlin. "Haar features for FACS AU recognition." In 7th international conference on automatic face and gesture recognition (FGR06), pp. 5-pp. IEEE, 2006.

[13] Pantic, Maja, and Ioannis Patras. "Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 36, no. 2 (2006): 433-449.

[14] Zhao, Guoying, and Matti Pietikainen. "Dynamic texture recognition using local binary patterns with an application to facial expressions." IEEE transactions on pattern analysis and machine intelligence 29, no. 6 (2007): 915-928.

[15] Dhall, Abhinav, Akshay Asthana, Roland Goecke, and Tom Gedeon. "Emotion recognition using PHOG and LPQ features." In 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG), pp. 878-883. IEEE, 2011.

[16] Black, Michael J., and Yaser Yacoob. "Recognizing facial expressions in image sequences using local parameterized models of image motion." International Journal of Computer Vision 25, no. 1 (1997): 23-48.

[17] Soleymani, Mohammad, David Garcia, Brendan Jou, Björn Schuller, Shih-Fu Chang, and Maja Pantic. "A survey of multimodal sentiment analysis." Image and Vision Computing 65 (2017): 3-14.

[18] Rosenbloom, Paul S., John E. Laird, Allen Newell, and Robert McCarl. "A preliminary analysis of the Soar architecture as a basis for general intelligence." Artificial Intelligence 47, no. 1-3 (1991): 289-325.

[19] Livanos, Nicole. "Mobility for Healthcare Professional Workforce Continues to Soar." Journal of Nursing Regulation 10, no. 4 (2020): 54-56.

[20] Ngai, Wang Kay, Haoran Xie, Di Zou, and Kee-Lee Chou. "Emotion recognition based on convolutional neural networks and heterogeneous bio-signal data sources." Information Fusion 77 (2022): 107-117.

[21] Lu, Cheng, Yuan Zong, Wenming Zheng, Yang Li, Chuangao Tang, and Björn W. Schuller. "Domain invariant feature learning for speaker-independent speech emotion recognition." IEEE/ACM Transactions on Audio, Speech, and Language Processing 30 (2022): 2217-2230.

[22] Cecchi, Ariel S. "Cognitive penetration of early vision in face perception." Consciousness and Cognition 63 (2018): 254-266.

[23] Torfi, Amirsina, Seyed Mehdi Iranmanesh, Nasser Nasrabadi, and Jeremy Dawson. "3d convolutional neural networks for cross audio-visual matching recognition." IEEE Access 5 (2017): 22081-22091.

[24] Wu, Xun, Wei-Long Zheng, Ziyi Li, and Bao-Liang Lu. "Investigating EEG-based functional connectivity patterns for multimodal emotion recognition." Journal of neural engineering 19, no. 1 (2022): 016012.

[25] Jin, Qin, and Junwei Liang. "Video description generation using audio and visual cues." In Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, pp. 239-242. 2016.

[26] Zhang, Shiqing, Shiliang Zhang, Tiejun Huang, Wen Gao, and Qi Tian. "Learning affective features with a hybrid deep model for audio–visual emotion recognition." IEEE Transactions on Circuits and Systems for Video Technology 28, no. 10 (2017): 3030-3043.

[27] Neverova, Natalia, Christian Wolf, Graham Taylor, and Florian Nebout. "Moddrop: adaptive multi-modal gesture recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 38, no. 8 (2015): 1692-1706.

[28] Koromilas, Panagiotis, and Theodoros Giannakopoulos. "Deep multimodal emotion recognition on human speech: A review." Applied Sciences 11, no. 17 (2021): 7962.

[29] Pantic, Maja, and Leon JM Rothkrantz. "Facial action recognition for facial expression analysis from static face images." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34, no. 3 (2004): 1449-1461.

[30] Martin, Olivier, Irene Kotsia, Benoit Macq, and Ioannis Pitas. "The eNTERFACE'05 audio-visual emotion database." In 22nd International Conference on Data Engineering Workshops (ICDEW'06), pp. 8-8. IEEE, 2006.

[31] Farhoudi, Zeinab, and Saeed Setayeshi. "Fusion of deep learning features with mixture of brain emotional learning for audio-visual emotion recognition." Speech Communication 127 (2021): 92-103.

[32] Bloch, Isabelle. "Information combination operators for data fusion: A comparative review with classification." IEEE Transactions on systems, man, and cybernetics-Part A: systems and humans 26, no. 1 (1996): 52-67.

[33] Zhang, Yongmian, and Qiang Ji. "Active and dynamic information fusion for facial expression understanding from image sequences." IEEE Transactions on pattern analysis and machine intelligence 27, no. 5 (2005): 699-714.

[34] James, Alex Pappachen, and Belur V. Dasarathy. "Medical image fusion: A survey of the state of the art." Information fusion 19 (2014): 4-19.

[35] Chen, JunKai, Zenghai Chen, Zheru Chi, and Hong Fu. "Emotion recognition in the wild with feature fusion and multiple kernel learning." In Proceedings of the 16th International Conference on Multimodal Interaction, pp. 508-513. 2014.

[36] Teissier, Pascal, Jordi Robert-Ribes, J-L. Schwartz, and Anne Guérin-Dugué. "Comparing models for audiovisual fusion in a noisy-vowel recognition task." IEEE Transactions on Speech and Audio Processing 7, no. 6 (1999): 629-642.

[37] Kumari, Jyoti, Reghunadhan Rajesh, and K. M. Pooja. "Facial expression recognition: A survey." Procedia computer science 58 (2015): 486-491.

[38] Ekman, Paul, and Wallace V. Friesen. "Facial action coding system." Environmental Psychology & Nonverbal Behavior (1978).

[39] Nazari, Elham, Rizwana Biviji, Amir Hossein Farzin, Parnian Asgari, and Hamed Tabesh. "Advantages and challenges of information fusion technique for big data analysis: proposed framework." Journal of Biostatistics and Epidemiology 7, no. 2 (2021): 189-216.

[40] Su, Yangfeng, Jian Wang, Xuan Zeng, Zhaojun Bai, Charles Chiang, and Dian Zhou. "SAPOR: Second-order Arnoldi method for passive order reduction of RCS circuits." In IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004., pp. 74-79. IEEE, 2004.

[41] Sadeghi, Hamid, Abolghasem-Asadollah Raie, and Mohammad-Reza Mohammadi. "Facial expression recognition using texture description of displacement image." Journal of Information Systems and Telecommunication (JIST) 2, no. 4 (2014): 205-212.

[42] Nikpoor, Mohsen, Mohammad Reza Karami-Mollaei, and Reza Ghaderi. "A new Sparse Coding Approach for Human Face and Action Recognition." Journal of Information Systems and Telecommunication (JIST) 1, no. 17 (2017): 1.

[43] Navraan, Mina, Nasrollah Moghadam Charkari, and Muharram Mansoorizadeh. "Automatic Facial Emotion Recognition Method Based on Eye Region Changes." Journal of Information Systems and Telecommunication (JIST) 4, no. 4 (2016): 221-231.

[44] Motamed, Sara, Saeed Setayeshi, Azam Rabiee, and Arash Sharifi. "Speech emotion recognition based on fusion method." Journal of Information Systems and Telecommunication 5 (2017): 50-56.

Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 222-231
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Long-Term Software Fault Prediction Model with Linear Regression and Data Transformation

Momotaz Begum[1], Jahid Hasan Rony[1], Md. Rashedul Islam[1,2], Jia Uddin[3*]

[1].Department of Computer Science and Engineering, Dhaka University of Engineering & Technology, Gazipur-1707, Dhaka, Bangladesh;
[2].Department of Computer Science and Engineering, International University of Business Agriculture and Technology
[3].AI and Big Data Department, Endicott College, Woosong University, Daejeon, South Korea

Received: 12 Apr 2022/ Revised: 04 Sep 2022/ Accepted: 27 Oct 2022

Abstract

The validation performance is obligatory to ensure the software reliability by determining the characteristics of an implemented software system. To ensure the reliability of software, not only detecting and solving occurred faults but also predicting the future fault is required. It is performed before any actual testing phase initiates. As a result, various works on software fault prediction have been done. In this paper presents, we present a software fault prediction model where different data transformation methods are applied with Poisson fault count data. For data pre-processing from Poisson data to Gaussian data, Box-Cox power transformation (Box-Cox_T), Yeo-Johnson power transformation (Yeo-Johnson_T), and Anscombe transformation (Anscombe_T) are used here. And then, to predict long-term software fault prediction, linear regression is applied. Linear regression shows the linear relationship between the dependent and independent variable correspondingly relative error and testing days. For synthesis analysis, three real software fault count datasets are used, where we compare the proposed approach with Naïve gauss, exponential smoothing time series forecasting model, and conventional method software reliability growth models (SRGMs) in terms of data transformation (With_T) and non-data transformation (Non_T). Our datasets contain days and cumulative software faults represented in (62, 133), (181, 225), and (114, 189) formats, respectively. Box-Cox power transformation with linear regression (L_Box-Cox_T) method, has outperformed all other methods with regard to average relative error from the short to long term.

Keywords: Software Reliability; Software Faults; Forecasting; Long Term Prediction; Relative Error.

1- Introduction

In the modern era, the software acts as an essential part of our life. The software ensures the performance of our digital devices and helps to maintain our lifestyle, manage businesses, and so on. It has become impossible to pass even a single day without software usage in our daily life. When software is responsible for a massive operation, making a minor software fault, the entire system can collapse. For example, in 2018, a fully autonomous uber test car hit a pedestrian and accidentally killed her [1]. Because of the object detection software fault, the system failed to detect the human who was crossing the road with her bike. In addition, the Hawaii missile false alarm is another example of major suffering due to software failure [2]. Such incidents could have been avoided by reliable software with a software failure prediction system which is a popular approach in software engineering.

The engineering approach of systematic application development can be defined as the term, software engineering. The test effort, optimal cost analysis, correction prediction, security, effort, reusability, and quality-related prediction are a few vital parts issues of software engineering. To find a versatile method of prediction analysis further research is still going on in this area. On the other hand, to ensure software reliability, software fault prediction performance has to make sure. Software reliability is the probability of failure-free software. Long-term software failure can detect the possibility of a software failure so that impact of the failure can be minimized by taking necessary steps and precautions [3].

The development of software is expected to be perfect. However, it is impracticable to design and develop

✉ **Jia Uddin**
jia.uddin@wsu.ac.kr

software with 100% accuracy and dependability. From a previous study, already it has been established that proficiency of fault prediction is caused due to the lack of proper evaluation criteria of performance and different fault distribution in a fault dataset [4]. But day by day the importance of software fault prediction has gained lot of attention because of the capability of providing faults number as well as the occurrence pattern of a certain system. Subsequently, it is also helpful for the quality assurance team as it can reduce testing time and cost.

The purpose of the software fault prediction is to identify the fault before sending it to the testing phase in the basis of software structural characteristics. In addition, to ensure software quality, professional stakeholders use prediction systems for optimal cost and effort during the operational phases. In this regard, we focused on the long-term software fault prediction using a linear regression method. Besides that, we have compared the model with Naïve gauss, exponential smoothing time series forecasting model, and two existing software reliability growth models name log extreme minimum (SRGM_LEM) [5] and pareto (SRGM_Pareto) [6]. Besides that, we have used software fault count data instead of fault detection time data because of the availability and the usefulness. Furthermore, three real data sets have been used for this study and the three most popular data transformations Box-Cox_T, Yeo-Johnson_T as well as Anscombe_T methods have been applied for the Poisson data into Gaussian data.

The organization of the rest of the paper is as follows: the related study is explained with conventional NHPP-based SRGMs in Section 2. Then in Section 3, the system architecture is described with suitable figures associated with data pre-processing techniques and forecasting models. The fault prediction with the proposed methodology is presented in Section 4. After that, Section 5 represents the experimental illustration assist with system setup, performance measurement, and result analysis. Finally, the paper concludes with future direction in Section 6.

2- Related Works

Various works have been done and are still going on in this field to predict faults of software to ensure reliability. Software reliability growth models (SRGMs) are one of the oldest with some limitations, such as the maximum likelihood estimation requires high computation power, and from a large number of SRGMs, researchers get confused to select the suitable model for every software data [3]. Nowadays, another popular classification method is an artificial neural network (ANN) with backpropagation (BP) learning algorithms used in software fault prediction [5], [67], [8].

Recently, Begum et al. proposed a robust prediction interval method using a refined artificial intelligence approach, where 5 data transformation methods are used for pre-processing and compared with the traditional method SRGMs [9]. They have constructed prediction intervals using their proposed method, and performance analysis is conducted by coverage rate and mean prediction interval width as well as compared with the existing delta method. However, the architecture of neural networks is complex; as a result, computation time is very high. The same author has related works [10–13] based on multilayer perceptron to address optimal software release problems.

Furthermore, the paper [14] presented a neuro base software fault prediction method using the Box-Cox transformation scheme. They have also investigated the optimal value of transformation parameter λ in case of average relative errors. Subsequently, they compared their result with traditional SRGMs and showed that their method outperformed in the early testing phase. On the other hand, multiplayer neural network architecture is used for identifying optimal software testing time [15]. For underlying software fault count, they have also pre-processed the data using a well-known data transformation technique. Where experimental result was conducted from four (4) actual software fault count data.

In [16] a study showed a study about software fault prediction and the different components and parameters of software fault prediction. Then the paper focused on the accomplishment in this area as well as recent research trends. In addition, they have discussed major future challenges of software fault prediction. But the advantage and disadvantage of recent studies has not prevailed. Different software fault prediction technique [17]–[20] have been proposed previously but none of those fully fills the long-term software fault prediction criteria as well as could not provide enough quality assurance resources and logistics.

Recently semantic long short-term memory (LSTM) network is used to train a model that can be self-directed to identify fault prediction and performed on real projects where the proposed model outperformed state-of-the-art approaches of fault prediction [21], [22]. After that, for finding fault in real life, a deep learning-based fault prediction model is presented in [23]. Which matches the abstract of source code syntax tree representation based on tree structure LSTM. They have evaluated their model using Samsung and a public PROMISE repo dataset. Then in a study [34], author estimate the reliability improvement that the recently suggested SDAFlex&Rel software development methodology, which aims to create reliable but flexible software, promises. By laying the groundwork for formal modelling, refinement, and verification which in turn avoid and eliminate possible faults, that method increases the dependability of software.

On the other hand, Deepak and Pravin introduced object-oriented metrics that are used for main factor findings [24]. In, Diego J. Pedregal [25] presents a paper where a few time series forecasting methods such as regression, Naïve methods, ARIMA, Transfer Functions, VAR(X), exponential smoothing (ETS), unobserved components (UC) models are used to develop a user-friendly graphical interface tool based on MATLAB for automatic outlier identification and detection. Also, a regression technique is applied for prediction analysis which provides a different approach for the software prediction process. For example, the linear regression method is used in optical network fault tracing to reduce high cost and fault detection time cited in [26].

In this paper, we presented different transformation methods as pre-processing for the real dataset and applied linear regression, Naïve Gauss, and exponential smoothing time series forecasting methods to predict software faults. Finally compared the outcome from various perspectives with existing popular SRGMs.

2-1- Non-Homogeneous Poisson Process-based SRGM

Let the group number of software fault represented by $G(t)$ at time t. Suppose (i) $G(0) = 0$, (ii) $G(t)$ has independent increments, (iii) $Pro[G(t+q)- G(t) \geq 2] = O(q)$, (iv) $Pro[G(t+q)- G(t) =1] = \lambda(t; \Theta)q+O(q)$, Here, non-homogeneous Poisson process(NHPP) intensity function is $\lambda(t; \Theta)$, Θ is the parameter of the model and infinitesimal time q higher term is $O(q)$. So that, probability of $G(t) = y$ is calculated by,

$$Pro[G(t) = y] = \frac{\Lambda(t;\Theta)^y}{y!} * \exp[-\Lambda(t;\Theta)] \quad (1)$$

where mean value function is,

$$\Lambda(t;\Theta) = \int_0^t \lambda(y;\Theta)dy \quad (2)$$

By estimating model parameter Θ, mean value function λ(t;Θ) or Λ(t;Θ) can be specified the NHPP, regarded as parametric SRGMs. In [32], using SRATS model parameter can be estimated. After finding model parameter, the mean value function of random time t_(n+1) presented below:

$$\Lambda(t_{n+1};\hat{\Theta}) = \int_0^{t_{n+1}} \lambda(y;\hat{\Theta})dy \quad (3)$$

$$\Lambda(t_{n+1} \mid G(t_n) = y_n;\hat{\Theta}) = y_n + \int_{t_n}^{t_{n+1}} \lambda(y;\hat{\Theta})dy = y_n + \Lambda(t_{n+l};\hat{\Theta}) - \Lambda(t_n;\hat{\Theta}) \quad (4)$$

here, l represents the length of prediction and y_n denotes cumulative fault for n-th testing days. For log extreme minimum (SRGM_LEM) [5] and pareto (SRGM_Pareto) [6], the equation are:

$$\Lambda(t;\Theta) = a\big(1 - F(-\log t)\big), \theta \in (a,b,c)$$
$$F(t) = exp(-\exp[\left(-\frac{t-c}{b}\right)]) \quad (5)$$

$$\Lambda(t;\Theta) = aF(t), \theta \in (a,b,c),$$
$$F(t) = 1 - (\frac{c}{t+c})^b \quad (6)$$

3- System Architecture of Proposed Model

The system architecture of the proposed model is shown in Figure 1, where the best method is highlighted in bold. As we know the fault count data is Poisson count data and used as an integer value in the software fault prediction so the underlying data requires to be transformed from Poisson to the Gaussian data in advance. The real data noted by Data Set are passed into the pre-processing state where categorized; With_T and Non_T. With_T represents pre-processed data by the well-known transformation methods; Box-Cox_T, Anscombe_T and Yeo-Johnson_T. Non_T define the non-transformed data. The methodology has two categories; linear regression and time series forecasting. Where Naïve Gauss, and exponential smoothing method of time series forecasting is used. Furthermore, two popular SRGMs; SRGM_LEM and SRGM_Pareto is applied to justify the accuracy. After that, the result is sent to Inverse_Trans state for inverse transformation. For determining better accuracy of long-term software fault prediction, we have compared linear regression with Naïve gauss, exponential smoothing methods, and SRGMs in terms of average and relative error.

3-1- Pre-Processing

In the pre-processing stage, we have tested software fault data for both cases, with and without data transformation. In the case of transformation, Box-Cox [27], Anscombe [28], and Yeo-Johnson [29] methods are used in both linear regression and time series forecasting. In case of linear regression, L_Box-Cox_T, L_Anscombe_T, and L_Yeo-Johnson_T represent Box-Cox, Anscomb and Yeo-Johnson respectively. Besides that, two time series forecasting Naïve gauss and exponential smoothing are represented as NG_Box-Cox_T, NG_Anscomb_T, NG_Yeo-Johnson_T, ES_Box-Cox_T, ES_Anscomb_T and ES_Yeo-Johnson_T sequentially.

Researcher Box and Cox develop Box-Cox power transformation (Box-Cox_T) to transform data into normal shapes for determining the appropriate value of a variable λ. It is used for transforming from arbitrary random data to Gaussian data. But the value of variable λ is difficult to find using log transformation. Additionally, when the data is reverted from transformed to the original state, it always provides a median prediction.

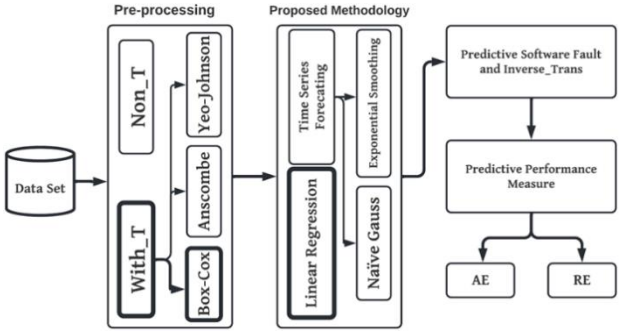


Fig. 1 System architecture of the proposed method.

Another power transformation operator, which is an extension of Box-Cox_T, Yeo-Johnson transformation (Yeo-Johnson_T), is applied to transform the original data into normally distributed numbers. and useful method. Where the distribution is far from Gaussian, distribution is skewed more often. Yeo-Johnson_T is another useful method to handle those data and make them normal (Gaussian).

Table 1: Data Transformation Formulae.

SI No	Name of method		Transformation Formula	Inverse Transformation Formula
1	Box-Cox_T	$\lambda=0$	$y_i = log(x_i)$	$exp(y_{n+l})$
		$\lambda \neq 0$	$y_i = \frac{x_i^{\lambda}-1}{\lambda}$	$(\lambda y_{n+l}+1)^{1/\lambda}$
2	Anscombe_T		$y_i = 2\sqrt{x_i+\frac{3}{8}}$	$\left(\frac{y_{n+l}}{2}\right)^2-\frac{3}{8}$
3	Yeo Johnson	If $\lambda\neq0, y\geq0$	$((x_i+1)^{\lambda}-1)/\lambda$	$(y_{n+l}\lambda+1)^{\frac{1}{\lambda}}-1$
		If $\lambda=0, y\geq0$	$log(x_i+1)$	$exp(y_{n+l})$-1
		If $\lambda\neq2, y<0$	$-((-x_i+1)^{(2-\lambda)}-1)/(2-\lambda)$	$1-(-y_{n+l}(2-\lambda)+1)^{\frac{1}{2-\lambda}}$
		If $\lambda=2, y<0$	$-log(-x_i+1)$	$1-exp(-y_{n+l})$

The Anscombe transform (Anscombe_T) is a variance stabilizing transformation in statistics that converts a random variable with a Poisson distribution into one with a Gaussian distribution that is approximately standard. In photon-limited imaging (astronomy, X-ray), where images naturally obey the Poisson law, it is also commonly employed. In Table 1, different transformation and inverse transformation equations for Box-Cox_T, Anscombe_T, and Yeo-Johnson_T methods are shown. Where input *i*, denotes the cumulative number of software faults detected at *i* (1, 2, 3, … n)-th testing day, output [*l* =1,2,3,…], *l* is the length of prediction. After that, the data inverse transformation formula is used for the final predictive output. When λ=2, Yeo Johnson_T showed ambiguous results so we have to exclude that.

3-2- Forecasting Models

In machine learning, we are required to form predictive outcomes based on different rules, attributes, and patterns from the provided data set. In addition, regression is a well-known method for generating predictive outcomes as a numeric value from given datasets between multiple independent and one targeted single dependent variable. In this paper, we have tested both the transformed and non-transformed dataset in our system using linear regression and time series forecasting methods, such as Naïve gauss and exponential smoothing [30].

Naïve gauss is a sophisticated forecasting model where the last outcome is used as the current predictive value and is widely used for economic and financial time series-related analysis [30] Additionally, another commonly used predictor model is Exponential Smoothing, where prediction is made via an exponentially weighted average of previous observations.

Among different regression models, linear regression is one of the easiest [31], most widely used methods for predictive analysis and checks whether the set of variables generates a better outcome with dependent variables or not. As a supervised method, here, an independent variable (assume *x*) is given, and the linear regression predicts a suitable dependent variable (assume *y*) and finds out linear relations between those variables. Let the specific set of input values x_i (i =1, 2, 3,…n) where n denotes the number of input and the predicted output for the set of input variables by y_i.

In our software fault data, t_i, x_i denotes the number of testing days and a cumulative number of software faults, respectively. So, without loss of generality, the linear regression equation combines a specific set (t_i, x_i) represented by (y_i, x_i). Hence, the input and predicted output are in numeric value. Suppose that the input variable is (x_i= x_1, x_2, x_3 ,...x_n) and the parameters of the model are β_i = [β_0, β_1, β_2,... β_n]. After that, the prediction model would be:

$$y_i \approx \beta_0 + \sum_{i=1}^{n} \beta_i \times x_i \qquad (7)$$

If x_i increases to (x_i= *1, x_1, x_2, x_3 ,...x_n)* after that, y_i would be a parameter of dot product and an independent variable that is

$$y_i \approx \beta_0 + \sum_{i=1}^{n} \beta_i \times x_i \ = \vec{\beta}.\vec{x}_i \qquad (8)$$

Apart from the algebra equation of the straight line, let the slope be "*m*" then the equation become

$$y = mx + b \qquad (9)$$

Then if the slope is "*a*" for the linear regression, the equation become

$$y = ax + b \qquad (10)$$

Where *b* is the *y*-axis intercept and *a* is the slope, then we can easily find the value of the coefficient of *a*, *b*,

$$a = \frac{(\sum_{i=1}^{n} y)(\sum_{i=1}^{n} {y_i}^2) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} x_i y_i)}{n(\sum_{i=1}^{n} {x_i}^2) - (\sum_{i=1}^{n} x_i)^2} \qquad (11)$$

$$b = \frac{n(\sum_{i=1}^{n} x_i y_i) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n(\sum_{i=1}^{n} {x_i}^2) - (\sum_{i=1}^{n} x_i)^2} \qquad (12)$$

In the cases, where the coefficient is zero, the effect of input variable x_i eliminates the model's effectiveness; therefore, the predicted output will be (0 * x = 0). It has a consequence with the regularization method so that learning algorithms of regression models decrease the complexity of the algorithm by putting pressure on the size of the coefficient.

4- Fault Prediction through Proposed Methodology

For software fault prediction, we have to use the previous fault data to predict next faults. In our proposed system, we have applied linear regression, naïve gauss and exponential smoothing time series forecasting model. For theorical better understanding we have illustrated the configuration of our training and prediction scheme in Figure 2 where the x-axis shows the time and y-axis shows the number of faults In the training phase (0, *n*), after pre-processing, processed input data x_i, [i=1,2,3,…, n] and output $y_{n+l,}$ [l =1,2,3…] where prediction length l=5, 10, 15.To predict the cumulative number of software faults for l testing days from the training point t_n, the prediction has to be made for the $n+l$ days. In this scheme, with given λ the Box-Cox, Yeo-Johnson and Anscombe transformations transformed data (x_1, x_2, …, x_n) are used for the input, and the predicted fault (y_{n+1}, y_{n+2}, … , y_{n+l}) are used for the evaluation in the prediction phase.

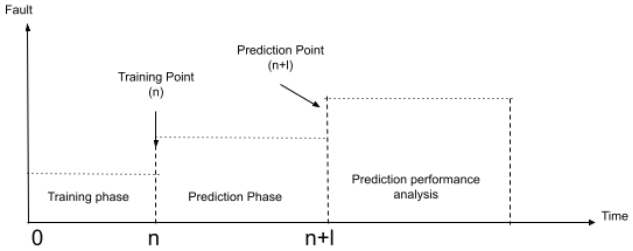


Fig. 2 Training and prediction scheme.

$$y_{n+l} = ax_i + b \qquad (13)$$

In the equation of linear regression (13), the coefficient of *a* and *b* are generated from equation (11) and (12). However, it is not necessary to train all the unknown patterns in principle by linear regression.

$$y_{n+l} = x_n \qquad (14)$$

where x_n is processed input at n-th testing days. Equation (13) represents Naïve gauss [31].

$$y_{n+l} = \alpha x_n + (1 - \alpha) y_n \qquad (15)$$

Where, α is the smoothing constant (0.25, 0.5, 0.75, 1) and y_n is the predictive software fault at n-th testing days. Exponential smoothing is presented in equation (15).

5- Experimental Setup and Results Analysis

5-1- Experimental Setup

We have used three real project datasets cited in the reference [33]; DS_1, DS_2, and DS_3, which contain days and cumulative software fault (grouped) shown in Table 2.

DS_1 contains 62 days, cumulative software fault 133 and the project type is "command and control subsystem". Then the "command and data subsystem" type project data is used in DS_2 and DS_3 where the cumulative number of faults is 225 and 189 for 181 days and 114 days respectively. In the datasets, the length of total training and testing of software faults are given by(10X5), (20X10), (30X10), (40X15). Here, (10X5) means ten inputs, five defined as short and in (40X15) architecture presents forty inputs, fifteen predicted outputs which represented as long time prediction, respectively.

Table 2: Structure of the Data Set in Study.

Dataset	Number of Days	Number of Faults	Type of the project
DS_1	62	133	Command and Control subsystem
DS_2	181	225	Command and Data subsystem
DS_3	114	189	Command and Data subsystem

To find out the desired output via the linear regression, naïve model, and exponential smoothing model, we have implemented our proposed system in python and used NumPy's for pre-processing the data. In Figure 3, the X-axis represents testing days, and the Y-axis shows a cumulative number of software faults. Where the data set DS_1, DS_2 and DS_3 is represented using different colours (red, purple and green).

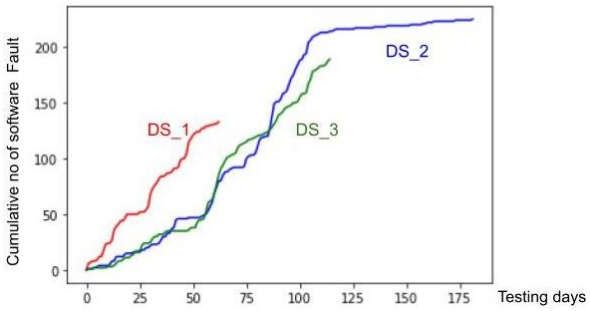


Fig. 3 Dataset Structure

5-2- Performance Measurement Criterion

To measure the performance of the system, we have followed two criteria: average error and relative error. Average error represents a degree to which a group of incorrect values with respect to absolute value. In addition, the relative error is a ratio of absolute error and actual error. For *n*-th testing day, the observation point is given tn. Here (n - l) software fault counts data are used for training in linear regression models. The prediction model is measured by the average error.

$$AE_l = \frac{\sum_{s=1}^{l} RE_s}{l} \qquad (16)$$

where REs is called the relative error for the future time $t = n + s$ and is given by

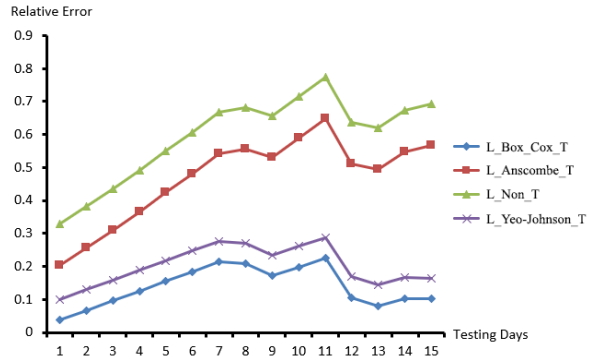
$$RE_s = \left| \frac{\tilde{x}_{n+s} - y_{n+s}}{\tilde{x}_{n+s}} \right| \qquad (17)$$

Here, $\tilde{x}_{n+s}$ is actual software fault and y_{n+s} is predicted software fault at (n+s)-th testing days. So, we regard the prediction model with smaller AE as a better prediction model.

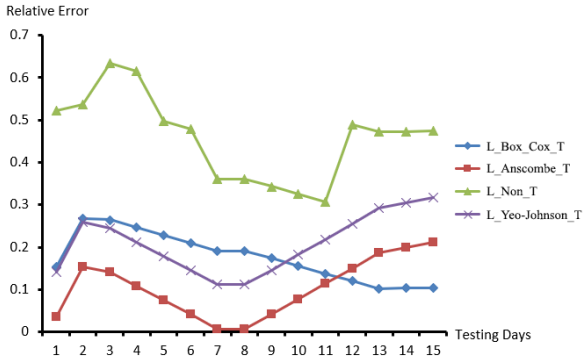
5-3- Result Analysis

For better understanding, we have analysed the proposed result and then compared it with other time series forecasting models (Naïve, Exponential Smoothing) and SRGMs. Figure 4 shows the comparative graphical results where the relative error is shown on the Y-axis with respect to testing days on the X-axis, and the length of the prediction is 15 days based on linear regression. In the datasets DS_1 and DS_3, L_Box-Cox_T (when λ=1) shows less error, whereas in DS_2, L_Anscombe_T is better, and L_Non_T gives the worst condition in all the cases. In the figure, all the behavior of the data transformation in the case of relative error is shown in different colors such as, red with a square representing shows L_Anscombe_T, light green with a triangle representing L_Non_T, L_Box-Cox_T represented in sky blue color with a diamond sign and the L_Yeo -Johnson_T is showed in violet with a cross symbol. In the DS_1, L_Non_T and L_Yeo -Johnson_T continuously changed the states while the L_Box-Cox_T showed stable and comparatively less error. In addition, furthermore, in DS_2, the worst result was provided by the L_Non_T on the other hand other methods showed the average result. Furthermore, L_Box_Cox_T clearly outperformed all other methods in DS_3.

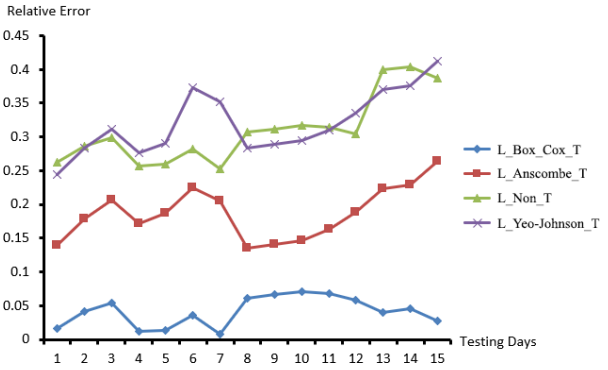
In Table 3, prediction performance results of software fault based on average error (AE) for three datasets denoted as DS_1, DS_2, and DS_3, respectively. Here, the architecture represents the number of input and predicted output. From the data table, we can say that for different values of λ, L_Box-Cox_T showed better results. In the 10X5 architecture, for both DS_2 and DS_3, L_Box-Cox_T was preferable, but in DS_1, L_Yeo-Johnson_T performed better. Besides that, L_Box-Cox_T outperformed other methods in 20X10 architecture for all datasets and in DS_1 and DS_3 for 40X15 architecture. Exceptionally in 30X10 architecture, the L_Box-Cox_T, L_Yeo-Johnson_T as well Non_T give less error in DS_2, but in DS_1 L_Yeo-Johnson_T and in DS_3 L_Anscombe_T was better. The best performance of every architecture is highlighted in bold text. Comparison of AEs for the naïve gauss and exponential smoothing time series forecasting model considering the three datasets is shown Table 4 and Table 5.



(a)



(b)



(c)

Fig. 4 Behavior of relative error with linear regression when l=15 for (a) DS_1, (b) DS_2, (c) DS_3.

By analyzing the data, we got that naïve gauss provides better result than exponential smoothing in every data set. To be mentioned that, when $\alpha = 1$, Box-Cox and the Yeo-Johnson transformed data provides same error. In addition, usually, whenever we have increased the input days, then the result showed an improved outcome (shown in 40X15 architecture). But the linear regression model for all architecture has shown superior predictive results than the time series forecasting models.

Table 3: Comparison of AEs for linear Regression Model with Different Transformation Methods.

Architecture	Transformation Method	Value of λ	DS_1	DS_2	DS_3
10X5	L_Box-Cox_T	0.5	0.6330	0.4296	0.3160
		1	0.7236	0.6142	0.1526
		2	1.6462	0.7844	0.3904
	L_Anscombe_T	-	0.6682	0.4592	0.3256
	L_Yeo-Johnson_T	0	0.3440	0.9122	2.1502
		1	0.9236	0.7646	0.1975
	L_Non_T	-	1.1236	0.6142	0.3526
20X10	L_Box-Cox_T	0.5	2.8586	0.7093	0.3983
		1	1.2853	0.1631	0.6625
		2	0.3592	0.1543	0.8293
	L_Anscombe_T	-	2.8256	0.6618	0.4263
	L_Yeo-Johnson_T	0	5.4305	1.2803	1.6912
		1	1.0853	0.1584	0.6846
	L_Non_T	-	1.2853	0.1631	0.6625
30X10	L_Box-Cox_T	0.5	0.9270	0.3523	0.2743
		1	0.6003	0.3034	0.4853
		2	1.6852	0.6728	0.8468
	L_Anscombe_T	-	0.8992	0.3089	0.2525
	L_Yeo-Johnson_T	0	8.3375	2.3368	2.1974
		1	0.6000	0.3034	0.4853
	L_Non_T	-	0.6867	0.3034	0.4853
40X15	L_Box-Cox_T	0.5	1.4036	0.3351	1.2571
		1	0.0411	0.1764	0.1380
		2	1.1772	0.9081	0.1415
	L_Anscombe_T	-	0.1866	0.1031	1.2107
	L_Yeo-Johnson_T	0	7.4240	2.4961	0.9557
		1	0.3202	0.2077	0.3477
	L_Non_T	-	0.3096	0.4596	0.5941

Table 4: Comparison of AEs for Naïve Gauss Model with Different Transformation Methods.

Architecture	Transformation Method	Value of λ	DS_1	DS_2	DS_3
10X5	NG_Box-Cox_T	0.5	5.1018	5.9957	11.2750
		1	9.7363	11.3213	20.3810
		2	17.8097	20.2702	42.9644
	NG_Anscombe_T	-	4.9690	5.8260	10.3004
	NG_Yeo-Johnson_T	0	2.9077	4.3616	11.4400
		1	9.7363	11.3213	20.3810
20X10	NG_Box-Cox_T	0.5	1.8409	1.6517	3.8768
		1	3.5839	3.2436	7.4225

		2	6.8030	6.2574	26.9249
	NG_Anscombe_T	-	1.4830	1.6263	3.7993
	NG_Yeo-Johnson_T	0	0.8928	0.9966	2.5400
		1	3.5839	3.2436	7.4225
30X10	NG_Box-Cox_T	0.5	1.0929	2.1003	1.1349
		1	2.1679	4.1139	2.2350
		2	4.2653	7.8961	13.0742
	NG_Anscombe_T	-	1.2691	2.0781	1.1212
	NG_Yeo-Johnson_T	0	0.4916	1.1539	0.6425
		1	2.1679	4.1139	2.2350
40X15	NG_Box-Cox_T	0.5	1.1487	0.8004	1.3202
		1	2.2689	1.5627	2.5800
		2	4.4279	2.9826	7.0538
	NG_Anscombe_T	-	1.2033	0.7945	1.3086
	NG_Yeo-Johnson_T	0	0.4886	0.4067	0.6954
		1	2.2689	1.5627	2.5800

Comparison of two SRGMs, SRGM_Pareto and SRGM_LEM for the same three dataset is presented in Table 6. After getting the mean value function using Equation (5) and (6), we have calculated long term prediction by Equation (4). SRGM_Pareto performed well in the 10X5 architecture however the SRGM_Pareto provided superior result for DS_2 and DS_3 in all other architectures.

In Table 7, the best method according to AEs from linear regression (Table 3), time series forecasting (Table 4 and Table 5) and SRGMs (Table 6) is presented. Linear regression with transformation performed comparatively better result than the other models. Then to be more specific, Box-Cox transformation with linear regression (L_Box-Cox_T) outrun all others. Notwithstanding, SRGMs showed good performance in 30X10 architecture. Subsequently, among the three datasets, only in one case, architecture 40X15 L_Anscombe_T offers superior results than other data models. From the result, we can also observe that the traditional method L_Non_T has not shown better results in any cases.

Table 5: Comparison of AEs for Exponential Smoothing model with different transformation methods.

Architecture	**Transformation Method**	**Value of λ**	**DS_1**	**DS_2**	**DS_3**
10X5	ES_Box-Cox_T	0.5	6.8332	7.7045	15.0115
		1	12.8170	14.4343	26.2358
		2	22.7536	25.4941	26.2358
	ES_Anscombe_T	-	7.1665	7.4876	13.5335
	ES_Yeo-Johnson_T	0	3.9619	5.6239	15.4431
		1	12.8171	14.4345	20.3810
20X10	ES_Box-Cox_T	0.5	2.2531	2.1116	5.1992
		1	4.3536	4.1366	9.8457
		2	8.1509	7.9421	7.4225
	ES_Anscombe_T	-	1.7020	2.0790	5.0920
	ES_Yeo-Johnson_T	0	1.0996	1.2775	2.5412
		1	4.3536	4.1366	2.3970
30X10	ES_Box-Cox_T	0.5	1.5773	2.7959	1.7096
		1	3.0995	5.4466	3.3406
		2	5.9938	10.3474	7.4225
	ES_Anscombe_T	-	1.9551	2.7662	1.6887
	ES_Yeo-Johnson_T	0	0.7176	1.5450	1.6425
		1	3.0995	5.4466	
40X15	ES_Box-Cox_T	0.5	1.5464	1.0887	1.6128
		1	3.0418	2.1135	3.1425
		2	5.8889	3.9919	2.2350
	ES_Anscombe_T	-	1.6053	1.0805	1.5985
	ES_Yeo-Johnson_T	0	0.6603	0.5561	1.6954
		1	3.0418	2.1135	2.5800

Table 6: Comparison of AEs for SRGMs with Different Transformation Methods.

Architecture	SRGM Model	DS_1	DS_2	DS_3
10X5	SRGM_LEM	1.4259	4.7747	14.4099
	SRGM_Pareto	1.2472	0.9279	1.5445
20X10	SRGM_LEM	0.4354	1.00171	1.2371
	SRGM_Pareto	0.6770	0.3921	0.2609
30X10	SRGM_LEM	0.2005	0.4511	0.4904
	SRGM_Pareto	0.4054	0.2425	0.1474
40X15	SRGM_LEM	0.1128	0.1948	0.3073
	SRGM_Pareto	0.2814	0.1423	0.1242

Table 7: Best Method based on AEs (Among linear Regression and Time Series Forecasting and SRGMs.)

Architecture	**DS_1**	**DS_2**	**DS_3**
10X5	L_Yeo-Johnson_T	L_Box-Cox_T	L_Box-Cox_T
20X10	L_Box-Cox_T	L_Box-Cox_T	SRGM _Pareto
30X10	SRGM_LEM	SRGM_Pareto	SRGM_Pareto
40X15	L_Box-Cox_T	L_Anscombe_T	SRGM_Pareto

6- Conclusion

In this paper, we have presented a long-term software fault prediction model based on linear regression with data transformation. In our model, we have pre-processed three actual software development project datasets with three

different transformation methods. Through a comprehensive analysis with non-transformation, time series forecasting method and conventional SGRMs, it has been shown that linear regression with Box-Cox (L_Box-Cox_T) could work well to predict the software fault in short time prediction. On the other hand, SRGM_Pareto showed better result for 30X10 architecture. Additionally, when we have increased input testing days for long-term prediction the relative errors have decreased. As a result, linear regression-based model was much attractive, though SGRMs are used often for long term software fault prediction. For further development, we will study to find an optimal value of λ for Box-Cox and Yeo-Johnson power transformation. Moreover, we will construct the prediction interval for better accuracy by well-known methods and apply our method in experimental or simulation data using Monte Carlo simulation methods for better prediction.

Acknowledgement.

This research is funded by Woosong University Academic Research in 2023.

References

[1] J. Stilgoe, "Who Killed Elaine Herzberg?," in Who's Driving Innovation? New Technologies and the Collaborative State, J. Stilgoe, Ed. Cham: Springer International Publishing, 2020, pp. 1–6. doi: 10.1007/978-3-030-32320-2_1.

[2] B. P. Murthy, N. Krishna, T. Jones, A. Wolkin, R. N. Avchen, and S. J. Vagi, "Public Health Emergency Risk Communication and Social Media Reactions to an Errant Warning of a Ballistic Missile Threat — Hawaii, January 2018," *Morb. Mortal. Wkly. Rep.*, vol. 68, no. 7, pp. 174–176, Feb. 2019, doi: 10.15585/mmwr.mm6807a2.

[3] H. Pham, *System Software Reliability*. Springer Science & Business Media, 2007.

[4] T. Menzies, Z. Milton, B. Turhan, B. Cukic, Y. Jiang, and A. Bener, "Defect prediction from static code features: current results, limitations, new approaches," *Autom. Softw. Eng.*, vol. 17, no. 4, pp. 375–407, Dec. 2010, doi: 10.1007/s10515-010-0069-5.

[5] A. L. Goel, "Software Reliability Models: Assumptions, Limitations, and Applicability," *IEEE Trans. Softw. Eng.*, vol. SE-11, no. 12, pp. 1411–1423, Dec. 1985, doi: 10.1109/TSE.1985.232177.

[6] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood, "Evaluation of competing software reliability predictions," *IEEE Trans. Softw. Eng.*, vol. SE-12, no. 9, pp. 950–967, Sep. 1986, doi: 10.1109/TSE.1986.6313050.

[7] S. Santosa, R. A. Pramunendar, D. P. Prabowo, and Y. P. Santosa, "Wood Types Classification using Back-Propagation Neural Network based on Genetic Algorithm with Gray Level Co-occurrence Matrix for Features Extraction," 2019.

[8] Y. Wang, D. Niu, and L. Ji, "Short-term power load forecasting based on IVL-BP neural network technology," *Syst. Eng. Procedia*, vol. 4, pp. 168–174, Jan. 2012, doi: 10.1016/j.sepro.2011.11.062.

[9] "Long-term Software Fault Prediction with Robust Prediction Interval Analysi...: EBSCOhost."

[10] M. Begum and T. Dohi, "Optimal Release Time Estimation of Software System using Box-Cox Transformation and Neural Network," *Int. J. Math. Eng. Manag. Sci.*, vol. 3, pp. 177–194, Jun. 2018, doi: 10.33889/IJMEMS.2018.3.2-014.

[11] M. Begum and T. Dohi, "Estimating prediction interval of cumulative number of software faults using back propagation algorithm," May 2016.

[12] M. Begum and T. Dohi, *optimal software release decision via artificial neural network approach with bug count data*. 2016.

[13] M. Begum and T. Dohi, "Prediction Interval of Cumulative Number of Software Faults Using Multilayer Perceptron," vol. 619, pp. 43–58, Jan. 2016, doi: 10.1007/978-3-319-26396-0_4.

[14] M. Begum and T. Dohi, "A Neuro-Based Software Fault Prediction with Box-Cox Power Transformation," *J. Softw. Eng. Appl.*, vol. 10, no. 3, Art. no. 3, Mar. 2017, doi: 10.4236/jsea.2017.103017.

[15] M. Begum and T. Dohi, "Optimal stopping time of software system test via artificial neural network with fault count data," *J. Qual. Maint. Eng.*, vol. 24, pp. 00–00, Jan. 2018, doi: 10.1108/JQME-12-2016-0082.

[16] Y. Kamei and E. Shihab, "Defect Prediction: Accomplishments and Future Challenges," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Mar. 2016, vol. 5, pp. 33–45. doi: 10.1109/SANER.2016.56.

[17] V. R. Basili, "The experimental paradigm in software engineering," in *Experimental Software Engineering Issues: Critical Assessment and Future Directions*, Berlin, Heidelberg, 1993, pp. 1–12. doi: 10.1007/3-540-57092-6_91.

[18] T. M. Khoshgoftaar *et al.*, "Predicting fault-prone modules with case-based reasoning," in *Proceedings The Eighth International Symposium on Software Reliability Engineering*, Nov. 1997, pp. 27–35. doi: 10.1109/ISSRE.1997.630845.

[19] C. Catal, "Software fault prediction: A literature review and current trends," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 4626–4636, Apr. 2011, doi: 10.1016/j.eswa.2010.10.024.

[20] K. Thantirige, A. K. Rathore, S. K. Panda, S. Mukherjee, M. A. Zagrodnik, and A. K. Gupta, "An open-switch fault detection method for cascaded H-bridge multilevel inverter fed industrial drives," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2016, pp. 2159–2165. doi: 10.1109/IECON.2016.7794032.

[21] M. Islam, M. Akhtar, and M. Begum, *Long short-term memory (LSTM) networks based software fault prediction using data transformation methods*. 2022, p. 6. doi: 10.1109/ICAEEE54957.2022.9836388.

[22] M. Islam, M. Begum and M. Akhtar, *Recursive Approach for Multiple Step-Ahead Software Fault Prediction through Long Short-Term Memory (LSTM)*. p. 10.

[23] H. K. Dam *et al.*, "Lessons Learned from Using a Deep Tree-Based Model for Software Defect Prediction in Practice," in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, May 2019, pp. 46–57. doi: 10.1109/MSR.2019.00017.

[24] D. Sharma and P. Chandra, "Linear regression with factor analysis in fault prediction of software," *J. Interdiscip. Math.*, vol. 23, pp. 11–19, Jan. 2020, doi: 10.1080/09720502.2020.1721641.

[25] D. J. Pedregal, "Time series analysis and forecasting with ECOTOOL," *PLOS ONE*, vol. 14, no. 10, p. e0221238, Oct. 2019, doi: 10.1371/journal.pone.0221238.

[26] O. Nyarko-Boateng, A. F. Adekoya, and B. A. Weyori, "Predicting the actual location of faults in underground optical networks using linear regression," *Eng. Rep.*, vol. 3, no. 3, p. eng212304, 2021, doi: 10.1002/eng2.12304.

[27] G. E. P. Box and D. R. Cox, "An Analysis of Transformations," *J. R. Stat. Soc. Ser. B Methodol.*, vol. 26, no. 2, pp. 211–252, 1964.

[28] F. J. Anscombe, "The Transformation of Poisson, Binomial and Negative-Binomial Data," *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948, doi: 10.2307/2332343.

[29] S. Weisberg, "Yeo-Johnson Power Transformations." 2001.

[30] E. S. Gardner, "Exponential smoothing: The state of the art—Part II," *Int. J. Forecast.*, vol. 22, no. 4, pp. 637–666, Oct. 2006, doi: 10.1016/j.ijforecast.2006.03.005.

[31] X. Su, X. Yan, and C.-L. Tsai, "Linear regression," *WIREs Comput. Stat.*, vol. 4, no. 3, pp. 275–294, 2012, doi: 10.1002/wics.1198.

[32] H. Okamura and T. Dohi, "SRATS: Software reliability assessment tool on spreadsheet (Experience report)," in *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)*, Nov. 2013, pp. 100–107. doi: 10.1109/ISSRE.2013.6698909.

[33] M. R. Lyu, Ed., *Handbook of Software Reliability Engineering*. Los Alamitos, Calif.: New York: McGraw-Hill, 1996.

[34] A. Rasoolzadegan, "A new approach to the quantitative measurement of software reliability," 2015.

Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 232-240
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

A survey on NFC Payment: Applications, Research Challenges, and Future Directions

Mehdi Sattarivand[1], Shahram Babaie[2*], Amir Masoud Rahmani[1]

[1].Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2].Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

Received: 19 Apr 2021/ Revised: 04 Dec 2022/ Accepted: 10 Jan 2023

Abstract

Near Field Communication (NFC), as a short-range wireless connectivity technology, makes it easier for electronic devices to stay in touch. This technology, due to its advantages such as secure access, compatibility, and ease of use, can be utilized in multiple applications in various domains such as banking, file transferring reservations, booking tickets, redeeming, entry/exit passes, and payment. In this survey paper, various aspects of this technology, including operating modes, their protocol stacks, and standard message format are investigated. Moreover, future direction of NFC in terms of design, improvement, and user-friendliness is presented for further research. In addition, due to the disadvantages of banknote-based payment methods such as the high temptation to steal and the need for a safe, mobile payments, which include mobile wallets and mobile money transfers, are explored as a new alternative to these methods. In addition, the traditional payment methods and their limitations are surveyed along with NFC payment as a prominent application of this technology. Furthermore, security threats of NFC payment along with future research directions for NFC payment and its challenges, including protocols and standards, and NFC payment security requirements are addressed in this paper. It is hoped that effective policies for NFC payment development will be provided by addressing the important challenges and formulating appropriate standards.

Keywords: Near Field Communication (NFC); NFC Payment; Card Emulation; NFC Security.

1- Introduction

In recent years, the electronic industry is improving significantly in various aspects and profoundly affected people's lives by dispelling some traditional problems. In NFC technology, two NFC-enabled devices can create a point-to-point connection via a wireless channel. The NFC technology has been firstly introduced by two leading manufacturers in the electronics industry, Sony and Philips in 2002. Some advantages of this technology such as flexibility, versatility, and ease to use caused it to extend its application in many fields such as file transferring, health monitoring systems, indoor navigation, ticketing, and financial [1].

Although many electronic devices can be equipped with NFC technology, this technology is mostly installed on smartphones. Also, other mobile network technologies such as 3G, 4G, and 5G have been added to mobile smartphones to provide broadband high-speed internet connection for users [2]. Nowadays, NFC is the most leading technology that can be embedded in smartphones and tablets. The theoretical maximum working distance of this technology is 20 centimeters, moreover, in practice, the range of reliable communication is much smaller, usually about 5 centimeters. In general, a combination of the contactless identification and interconnection technologies can compose an NFC system, which requires two NFC-compatible devices close to each other for a proper response [3].

In general, mobile payment due to its specific features such as safe, secure, convenient, and fast is an increasingly attractive method to pay. It should be noted that smartphone technology plays an essential role in the growth of these transactions [4]. It is reported that over 3.4 billion smartphones will be equipped with Android Pay, Apple Pay, and Samsung Pay by the end of 2017; likewise,

✉ **Shahram Babaie**
Sh.babaie@iaut.ac.ir

this may reach 5.3 billion by 2021. Moreover, Apple Pay transactions grew 450 percent in the first quarter of 2017 in comparison to the same quarter of 2016. In stores, mobile payments can carry out through NFC technology, which stands for Near Field Communication. Also, NFC can also be combined with RFID technology and read its tags also applied as a file transfer system. Therefore, NFC-enabled payments due to the three key reasons, i.e., secure, fast, and convenient can convince vendors to provide their products with this technology. Also, digital payment instead of paying for products with cash, checks, or physical credit cards causes people to carry less cash [5].

In general, safety and security are vital requirements in financial transactions. NFC technology due to the short support range can be considered as a reliable method for these applications [6]. The NFC connection is initiated when two NFC-enabled devices are close to each other almost 5 centimeters. Therefore, eavesdropping on the transaction's data is hard for a third party. Supporting a shorter range leads to improved security as a key NFC advantage over other communication systems. In general, the NFC technology is a short-range half-duplex communication protocol, working at 13.56 MHz frequency band, which is designed for particular applications [7]. NFC can enable a variety of innovative services. As a result, it is right to say that NFC provides smooth content delivery, brings simplicity to transactions, and enables secure information sharing. Moreover, NFC can build various opportunities for vendors, banks, mobile operators, and transport operators [8].

There are two operating modes in NFC, active and passive modes [9]. In the active mode, both devices generate the RF field to establish the connection and transfer data. Whereas, in the passive mode, one active device generates the RF field to establish the connection and transfer the data. Table 1 shows the NFC communication modes; also, Table 2 compares the WPAN technologies.

Table 1. Communication Modes in NFC [10].

Device A	Device B	RF Field Generation	RF Mode
Active	Active	Generated by Both Devices	Active Mode
Active	Passive	Generated by Device A Only	Passive Mode
Passive	Active	Generated by Device B Only	Passive Mode

NFC supports three data transfer rates i.e. 106, 212, and 424 KB/s [11], also, a variety of data coding techniques such as Manchester, modified miller, and NRZ-L are applied in this technology [12]. NFC also supports both 10% ASK and 100% ASK modulation. In fact, data rate, operation mode, RF signaling, and standards are major parameters to determine the modulation type. Figure 1 shows modulation and coding of an NFC connection.

Table 2. Comparison of WPAN Technologies [11].

Parameter	Bluetooth	ZigBee	NFC
Range	10-100 m	10-100 m	4-10 cm
Data Rate	0.8-2.1 Mbps	0.02-0.2 Mbps	0.02-0.4 Mbps
Cost	Low	Low	Low
Power Consumption	High	Medium	Low
Spectrum	2.4 GHz	2.4 GHz	13.54 MHz
Security	Low	Low	High
Network Topology	Piconets, Scatternets	Star, Tree, Mesh	One to One
Devices per Network	8	2-65,000	2
Usability	Moderate, Data Centric	Easy, Data Centric	Easy, Human centric
Personalization	Medium	Low	High
Flexibility	High	High	High
Setup Time	Approx. 6 s	Approx. 0.5 s	Less Than 0.1 s

According to [13], secure elements play a principal role when a secure transaction is needed in an NFC connection for either transmitting and storing data in NFC-enabled devices. Secure elements can create a secure environment for the sensitive data that is transferred and stored in an NFC connection such as users' credit card information and access key information when using card emulator mode. Also, HCE as the latest SE of NFC is another security approach to stores and manages the user's private data in the smartphone.

In this paper, the main goal is to survey the current NFC payment methods including their related architecture, standards, and challenges. Moreover, the existing solutions to overcome these challenges are presented, and future research directions of NFC payment are investigated in terms of design, improvement, and user-friendliness to dispel the challenges. In addition, along with presenting the drawbacks of banknote-based payment methods such as high temptation to steal, mobile payments and NFC payment, which include mobile wallets and mobile money transfers, are explored as a new effective alternative. Furthermore, security challenges of NFC payment are investigated; also various approaches that are applied for securing the NFC payment are introduced and compared in terms of security requirements. Finally, it has been endeavored that new future directions are addressed to improve protocols and standards for and NFC payment security requirements

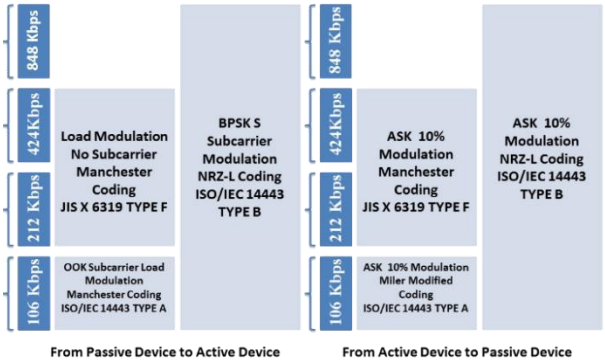


Fig. 1. Modulation and Coding of an NFC Connection [11].

The rest of this paper is organized as follows: NFC technology and its operating modes are introduced in section 2. Current payment methods and their limitations are discussed in section 3. Afterward, in section 4 the concept of NFC payment is represented. Section 5 presents the NFC payment security and security threats. The future direction of NFC payment is subject to be investigated in section 6. Finally, section 7 concludes the paper.

2- NFC Architecture, Standards, and Protocols

In this section NFC architecture, standards, and protocol are introduced in detail. In some studies, some modules have been proposed to improve the performance of NFC technology. Likewise, secure elements play an essential role in NFC applications, especially in payment cases. A mobile device equipped with NFC technology depending on its application may also have more than one SE. There are three operating modes in NFC, which are investigated in the following subsections.

2-1- Operating Modes

NFC technology similar to other communication systems needs a set of standards and protocols to control how the network elements operate, communicate, and guarantee security. Therefore, increasing the number and improving these standards result in ease of access and guaranteed security; meanwhile, it may make the NFC slightly more complicated than its status. The most significant standardization association for NFC technology is NFC Forum, which develops and improves all of NFC essentials, as well as NFC standards. NFC Forum is a non-profit association, which aims to turn NFC into a better network environment also extend it worldwide [14].

2-1-1- Reader/Writer Mode

Reader/writer mode is an operation mode of the NFC in which an active mobile NFC-enabled device initiates a wireless communication; it can also read and modify the stored data in the NFC tag accordingly. [15]. In this mode, the NFC-enabled device can read and write the NFC Forum's standard tags, which can be in four types. Consequently, the user can read, write the stored data on the NFC transponder, and carry out the proper action later. Following NFC Forum, messages of the NFC communication have a specific format, which is called NDEF. In general, several NDEF records are chained to each other to create an NDEF message, so that a more substantial payload can be transferred. The protocol stack of the reader/writer mode is also depicted in Fig. 2(a).

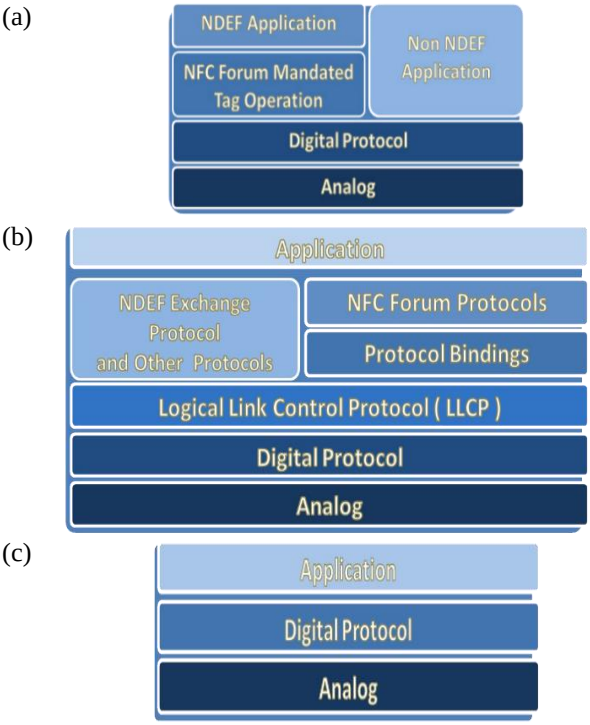


Fig. 2. Protocol Stack of the (a) Reader/Writer Mode, (b) Peer-to-Peer Mode, (c) Card Emulation Mode [17].

2-1-2- Peer-to-Peer Mode

In this mode, a bidirectional connection is made between two NFC-enabled devices, in which two devices can actively transmit any data such as business information, payment transaction information, and file sharing [16]. The RF communication interface in the peer-to-peer connection of NFC is standardized by ISO/IEC 18092 protocol, which is called NFCIP-1 that enables the "request and response" feature for both devices to make sharing data capability. In the mode, both of the NFC

devices, i.e., initiator and target are active. Also, the RF uses 13.56 MHz, frequency band. The protocol stack of the peer-to-peer mode of NFC is illustrated in Fig. 2(b).

2-1-3- Card Emulation Mode

In card emulation mode, both NFC devices use the same analog and digital techniques based on the ISO/IEC 14443 type A, type B, and SONY FeliCa. In this operation mode, when a mobile user's NFC device touches an NFC reader, its NFC device acts as a smart card also the NFC reader deals with the emulate smart card security element.

In this mode, the NFC reader is active, and the smartphone is passive. Applications of the card emulation mode are access control and ticketing. In general, some standards such as ISO/IEC 14443 type A, type B, and SONY FeliCa communication interfaces are applied in these applications. Figure 2(c) indicates the protocol stack of the card emulation mode. A summary of NFC operating modes including reader/writer mode, peer-to-peer mode, and card emulation mode is shown in Fig. 3.

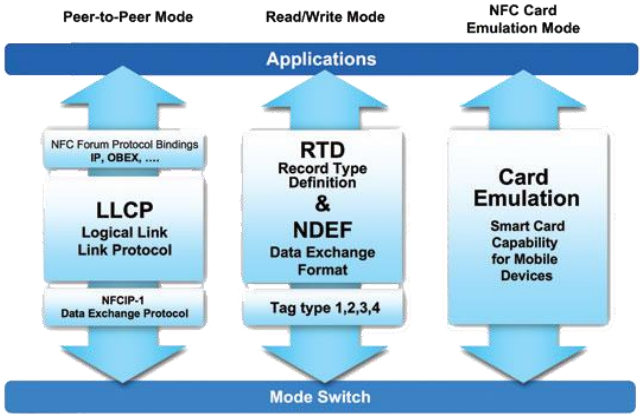


Fig. 3. Three Operation Modes of NFC [18].

2-2- Pros and Cons of NFC

In general, each system and technology has some advantages and disadvantages. Consequently, NFC as new technology has some pros and cons as follows.

2-2-1- Pros of NFC

- NFC uses electromagnetic waves instead of wired connections. Therefore, the wired connection is not necessary for users, so that the transactions, data transferring, media sharing are carried out faster and more reliable.
- The proximity of the two devices is necessary for NFC technology that causes eavesdropping is hard. As a result, this technology is almost secured.
- The NFC technology by reducing the time of sweeping the card, choosing a menu, and entering the password decreases the total time of a transaction; also, it can lead to a faster and more flexible transaction process when it applied in the transactions and card emulation.
- Lower energy consumption should be considered in the NFC design phase [19]. As a result, more energy power is saved in comparison to other payment technologies such as RFID in the overtime and considering the high number of users [20].
- The time needed for an NFC connection to be made less than 0.1 seconds. Therefore, the NFC is an appropriate choice, where there are many users, and they need to perform their payments tandem one after another, such as subway stations.
- NFC is a flexible technology, which can work under a variety of environmental conditions. Therefore, a user is not restricted to environmental conditions and the user can easily use this technology.
- NFC is versatile equipment where many tasks can be carried out. For example, a user can be checking out at a store, read information from a smart poster, and purchase and load a concert ticket through a single NFC-enabled device.

2-2-2- Cons of NFC

Besides the mentioned advantages, this technology has a few disadvantages as follows.

- Most NFC disadvantages are related to their costs and prices. So that increasing the costs in NFC applications is inevitable due to requiring both tags and readers. Generally, NFC device's price is approximately high; also, the maintenance and development of NFC devices are not cheap.
- Although NFC is almost secure due to it needs the devices to be close to each other, remains some challenges of NFC security.
- Another point to be mentioned as an NFC disadvantage is that this technology is not yet fully spread all around the world so that NFC technology is not applied in many countries. Likewise, for various reasons, this technology is not applicable in every country.

2-3- Future Research Directions on NFC

There are some future directions and open issues in NFC tags, NFC protocols, and NFC communication modes as follows. Although some of the proposed research topics are novel, recently some innovative efforts have been made on some matters, which makes these topics as future directions of NFC technology. The suggested future

research directions for topics of this section including SE and HCE except for security parts of NFC is listed:

- Analysis of antenna design, antenna coupling, RF efficiency, and simulation of an NFC communication on other proper frequency bands and proposing alternative designs for antennas of NFC.
- Proposing an alternative or developed protocol for SNEP for peer-to-peer transactions.
- Developing a new modulation scheme and a coding technique considering the power management and security issues of this technology.
- Developing alternative NFC protocols for other possible applications and a revision of LLCP and NFCIP-1.
- Proposing a more user-friendly software for Android, IOS, and Windows smartphones with a professionally designed GUI to increase the willingness of users to use this technology.
- Exploring possible ways of setting dynamic NFC tags as alternative hardware for static NFC tags.

3- Current Payment Methods

There are various methods, which can be applied to payment applications as follows. The NFC payment is not considered in this section.

3-1- EMV® Card

EMV® Card is the abbreviation form of "Euro pay, MasterCard®, and Visa®", which are three major companies that handle payments and transactions from little to large amounts of money. These companies have produced payment cards based on the chip technologies that are disclosing the account information is nearly impossible for anyone who might steal the card. To this end, the microprocessor chip inside the card generates a unique code for each transaction. As a result, due to the uniqueness of transaction codes, the gained codes will be useless when a criminal can get the code from a store.

3-2- Magnetic Stripe Card

A magnetic stripe card can store transaction data by changing the magnetism of a tiny iron-based magnetic layer of the card, which is called magstripe or sweep card. The magstripe cards work by sweeping the magnetic reading head to POS and POP devices. This type of card is most suitable for identity cards, credit cards, and transportation tickets [21]. Limitations of these payment methods are:

- **Bulky Wallets**

It is inevitable to have more than one payment card if someone has multiple bank accounts. As a result, managing these cards and their information will not be accessible. Also, keeping multiple cards cause a thicker wallet.

- **Loss of the Card**

Card-based payment methods have a high risk of being stolen and loss of cards. Furthermore, the extra time and more effort are required for the cardholder to recover the card when its card is lost or stolen.

- **Fake Identity**

When a criminal can steal a credit/debit card (EMV® card or magnetic stripe card), fake identity issues can occur. Therefore, a criminal can claim that it is the cardholder and cause severe and irrecoverable problems for the real cardholder.

4- The Concept of NFC Payment

Generally, mobile payments, which include mobile wallets and mobile money transfers, are actual transactions that take place on mobile devices. Mobile payment technology allows that payment to carry out digitally instead of paying for stuff with cash, checks, or physical credit cards. This type of payment can be applied in both "peer to peer" context and for paying at a brick-and-mortar business. A mobile payment application is applied to pay in a peer-to-peer mobile payment; also, instead of cash or a card, a mobile application is applied to pay for particular commodity and services at the checkout counter in the mobile payment of a brick-and-mortar business. In this instance, a specific type of point-of-sale device is required to process the transactions.

In general, the success of a payment method depends on some factors. To this end, Staykova and Damsgaard [22] investigated the effective parameters of a payment method that are based on entry and expansion. They have built a framework to analyze the entry and expansion strategies of digital payment solutions. They have concluded that the importance of the timing of expansion and importance of the timing of entry are identical. The important factors affecting mobile payment in public transportation have been investigated in [23]. In this study, the role of mobile payment as a payment system has been investigated in the public transport of Oporto, Portugal and Beijing, China. Their research has indicated that users' age is an essential factor in determining the usage of mobile payment systems so that younger people have tended to use mobile and digital payment systems more than older citizens. According to their statistical results, innovation in representing the service is an important factor that affects users to choose a digital mobile payment system [24].

User acceptance is a major success factor for each new technology. In [25], the user acceptance influence of success a new technology has been studied by Ramos-de-

Luna and Montoro-Rı´os. To meet this objective, they have prepared a questionnaire that 191 audiences have filled it. They have also used PLS v3.0 software to analyze the results statistically. Their results show that attitude, subjective norms, and innovation are vital parameters that can determinant the future intention of using mobile payments.

Likewise, Cocosila and Trabelsi studied the user acceptance of credit/debit cards and other contactless payment methods [26]. Their results show that NFC payment has a great advantage over other methods, but it also has some user doubts. They also surveyed value and risk perception, which had about 290 participants. The authors have claimed that the integrated value-risk perception is a significant factor in the adoption of NFC payments with smartphones so that the user stimulation to be used due to having utilitarian and enjoyment values; also, psychological and privacy risks are the most significant deterrents [27].

5- NFC Payment Security Issues and its Challenges

When a new payment system replaces traditional payment systems, the first issue to be taken care of is its security. In fact, when a user decides to switch from conventional payment methods to NFC payment, the most critical factor is how well the security will be achieved in this transaction. In general, the NFC technology has a proper security property due to its low support range. Although this feature gives users a sense of reliability, some security threats have remained as well. Also, some reports indicate the manipulation of thirds parties has occurred in the NFC transaction [15]. In order to better overcome this challenge, SE and HCE have been added to new NFC generation devices. The SE of NFC provides the security properties for an NFC-enabled device. SE has the following types:

- Embedded hardware: this SE type is the static element in an NFC-enabled device, which is embedded in the smartphone or tablet; also, the user enables to personalize the security settings after buying the smartphone. In fact, this type of SE is embedded in the smartphone at the manufacturing company.
- SMC: a smart memory card element and a smart card controller can make this type of SE. This type of smart card can supply a high level of security. Although many NFC versions are equipped with security elements, SMC is compatible with nearly all of the interfaces and standards such as ISO/IEC 7816, Global Platform, and Java Card [28]. As regards, the SMC is removable and provides a flexible scheme; also it has a large memory capacity and can easily handle the transactions in payments.
- UICC: this type of security element, which is called the Universal Integrated Circuit Card is based on the smart SIM card technique. Therefore, these cards are added to SE of NFC to protect the personal information of the users [13].
- TMB: Mobey Forum has introduced this technology as future technology and a new NFC secure element. This type of SE is located in the CPU of smartphones, which can handle a wide range of NFC's security applications. This facility causes it as a proper candidate for becoming an independent future SE.

There is another classification for SE of NFC technology so that the SE schemes can be classified into non-removable SEs, removable SEs, flexible SE solutions. As mentioned before, HCE is the latest SE for NFC technology and can manage the security applications of NFC with high reliability. In fact, HCE separates the card emulation mode, which is mostly used in payment transactions from SE. In fact, in using the HCE as a SE, a smartphone can perform the card emulation mode, while its security data is stored at the different location such as in another device or even in cloud storage. Figure 4 shows a comparison between SE and HCE.

Although HCE is a perfect approach to ensure an NFC transaction's security issues, there are some remaining challenges. The major challenge for HCE is when its data is stored in cloud memory. In fact, when a user intends to carry out an NFC transaction, maybe he/she does not access the internet, which fails the transaction [29].

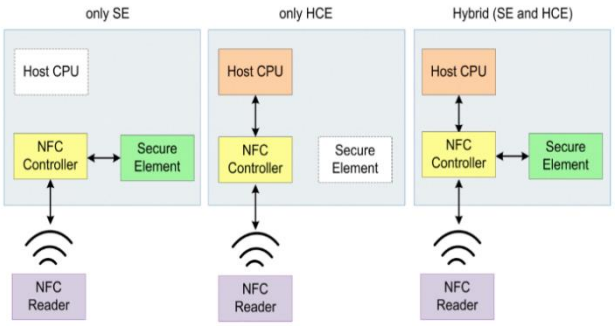


Fig. 4. Comparison of SE and HCE Architecture [30].

Although a decent research background exists for some issues of NFC applications, the future research directions for SE of NFC and HCE are suggested as follows.

- Proposing access control models to secure NFC elements compatible with smartphone OSs.
- Developing an alternative model for HCE as a security element.
- Proposing a secure communication protocol for cloud-based HCE in NFC.

- Developing an alternative model for removable SEs also and improving security SMC and UICC to achieve a more secure SE.
- Exploring possible methods for proposing a more reliable HCE architecture.
- Proposing a novel non-removable SE to apply in the current embedded hardware SEs.
- Evaluation and analysis of the cloud-based HCE for a hybrid system.
- Extensive research concerning market needs and device compatibility with the satisfaction of users who are interested in SE and security issues.
- Analysis of the connectivity between HCE and various smartphone OSs.
- Analysis of the power management and power-saving strategies for HCE and cloud-based HCE [31].

5-1- Security Threats of NFC Payment

There are various security threats for NFC payments that are introduced as follows.

5-1-1- Eavesdropping

Eavesdropping means that an attacker can hear the RF signal of an NFC transaction. Generally, eavesdropping is comfortable in an active NFC-enabled device. According to [32], an active NFC device, which generates its RF signals, can be heard at 10 meters, whereas in a passive device this distance is reduced to 1 meter. In the eavesdropping attack, an attacker uses a powerful active antenna. It seems that physically separating the channel between the two NFC devices can solve this problem.

5-1-2- Data Modification or Corruption

In this attack, an attacker uses a powerful antenna to manipulate the transferred data between two NFC devices at the valid frequency and at the right time. The manipulate data can be rubbish data to interrupt the connection and purposeful data to modify the transaction properties and information. The victim of this type of security threat is passive devices. The threats of this type of attack increase when a weak coding scheme is applied; whereas, a secure coding scheme can prevent the attacker's success. However, there is a trade-off between the security protection level in eavesdropping and data modification or corruption attacks so that the bidirectional active mode with baud data rate equal to 106 is the most unsafe condition to eavesdropping [33].

5-1-3- Man-in-the-Middle Attack

In this attack, a third party cuts off the NFC communication between the two NFC-enabled devices. In fact, the attacker acts as a relay between the NFC devices also transmits the changed information to the actual receiver device without the two devices getting to know that there is a third party. As regards, the support range of this technology is deficient; the occurrence probability of this type of attack in the NFC connection is low. In general, using an active-passive communication link between the NFC-enabled devices can identify the third party and reduce the risk of this type of security threat. In fact, applying both active devices prevent detection of the attacker.

Considering these types of NFC payment attacks, possible future research topics on this subject are suggested as follows.

- Proposing a physical channel protection scheme to avoid eavesdropping.
- Exploring the power-saving mechanisms to develop energy-efficient and functionally secure communications.
- Designing an extra passive antenna to be placed near the NFC payment devices or even inside the NFC device to detect the possible attacker to avoid the man-in-the-middle attack, data corruption, and modification accordingly.
- Modeling a convex problem to solve the trade-off situation in security protection level of eavesdropping attacks, data corruption, and modification attacks. In general, the distance, signal power, and coding schemes can be suitable parameters to model and design the convex problem.

6-Future Directions of NFC Payment

In general, NFC payment is a popular application of the NFC technology, which allows users to perform their transactions. Today, many shops and stores are equipped with this technology, but new efforts are still needed to extend this technology to more stores and smartphones. As a result, the first aim of the NFC Forum is that spreading this technology to more countries, shops, and smartphones. Also, user trust is another important factor in expanding NFC payments in the future, so that the users insist that their money be must securely transfer. The mentioned aims can be achieved by some research that is carried out by statistical experts that survey the completed questionnaires related to users' experience about the NFC payment.

Apart from the market-related issue and matters of the development of NFC technology, protocols, standards, and security are three significant problems related to the NFC payment, which directly affects the NFC payments future

directions. In the previous sections, the future directions related to each subject have been suggested. The presented future directions can give useful insights about how NFC payments can be a faster and more secure contactless payment technology to carry out the transactions with real ease. However, a point that should not be forgotten is that much research has to be done to refine NFC's new protocols and standards, also fill out the security gaps of NFC payments for making a bright future for this technology. Although extraordinary research efforts have been carried out to improve NFC technology and NFC payment, there still are some primary future directions as follows.

- Designing the more powerful NFC antennas to dominate the possible attacks and threats.
- Proposing a new communication standard that provides a higher definition of QoS in NFC.
- Proposal for efficient modulation and coding techniques that make eavesdropping is impossible for an attacker.
- Proposing an efficient protocol for NFC-enabled devices for payment applications based on power management and power saving issues.
- Evaluating the integration of this technology with new technologies such as the IoT to accelerate the usage of this technology.

7- Conclusions

In this paper, we have been investigated the current payment methods, also the pros and cons of NFC payment. Our evaluations show that NFC payment can outperform in comparison to the traditional payment methods in terms of speed, reliability, security, and ease of use. Moreover, we conclude NFC technology is growing faster in countries with a younger population. Likewise, security issues and challenges of the NFC payment have been surveyed in this paper. We have been tried to focus on the technical problems since the market-related issues have better solved by users' questionnaires also have analyzed by statistical and marketing experts. The technical challenges discussed in this article mainly related to security, protocols, and standard issues. Moreover, future directions have been proposed for NFC technology and NFC payments in terms of designing, architecture, and protocols.

References

[1] P. Chandrasekar and A. Dutta, "Recent Developments in Near Field Communication: A Study," *Wirel. Pers. Commun.*, pp. 1–20, Sep. 2020.

[2] S. Chabbi, R. Boudour, F. Semchedine, and D. Chefrour, "Dynamic array PIN:A novel approach to secure NFC electronic payment between ATM and smartphone," *Inf. Secur. J. A Glob. Perspect.*, vol. 29, no. 6, pp. 327–340, Nov. 2020.

[3] P. Escobedo, M. Bhattacharjee, F. Nikbakhtnasrabadi, and R. Dahiya, "Flexible Strain and Temperature Sensing NFC Tag for Smart Food Packaging Applications," *IEEE Sens. J.*, vol. 21, no. 23, pp. 26406–26414, Dec. 2021.

[4] N. Song, Q. Wang, D. Jiao, H. Pan, L. Shi, and P. Ding, "Highly thermally conductive SiO2-coated NFC/BNNS hybrid films with water resistance," *Compos. Part A Appl. Sci. Manuf.*, vol. 143, p. 106261, Apr. 2021.

[5] R. Tso, "Untraceable and Anonymous Mobile Payment Scheme Based on Near Field Communication," *Symmetry (Basel).*, vol. 10, no. 12, p. 685, Dec. 2018.

[6] H. Seddiqi and S. Babaie, "A New Protection-based Approach for Link Failure Management of Software-Defined Networks," *IEEE Trans. Netw. Sci. Eng.*, pp. 1–10, 2021.

[7] M. Chung, "Short distance data transmission method using inaudible high-frequencies between smart devices," *Telecommun. Syst.*, vol. 70, no. 4, pp. 583–594, Apr. 2019.

[8] F. Liébana-Cabanillas, S. Molinillo, and M. Ruiz-Montañez, "To use or not to use, that is the question: Analysis of the determining factors for using NFC mobile payment systems in public transportation," *Technol. Forecast. Soc. Change*, vol. 139, pp. 266–276, Feb. 2019.

[9] E. L. Wadii, J. Boutahar, and S. E. L. Ghazi, "NFC Technology for Contactless Payment Echosystems," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 5, pp. 391–397, 2017.

[10] R. Lyu, W. Cheng, and W. Zhang, "Modeling and Performance Analysis of OAM-NFC Systems," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 7986–8001, Dec. 2021.

[11] V. Coskun, K. Ok, and B. Ozdenizci, *Near field communication (NFC): from theory to practice*. John Wiley & Sons, 2011.

[12] J. Besnoff, M. Abbasi, and D. S. Ricketts, "High data-rate communication in near-field RFID and wireless power using higher order modulation," *IEEE Trans. Microw. Theory Tech.*, vol. 64, no. 2, pp. 401–413, 2016.

[13] M. de Reuver and J. Ondrus, "When Technological Superiority is not Enough: The Struggle to Impose the SIM Card as the NFC Secure Element for mobile payment platforms," *Telecomm. Policy*, vol. 41, no. 4, pp. 253–262, 2017.

[14] M. D. Steinberg, C. Slottved Kimbriel, and L. S. d'Hont, "Autonomous near-field communication (NFC) sensors for long-term preventive care of fine art objects," *Sensors Actuators A Phys.*, vol. 285, pp. 456–467, Jan. 2019.

[15] N. Druml *et al.*, "Secured miniaturized system-in-package contactless and passive authentication devices featuring NFC," *Microprocess. Microsyst.*, vol. 53, pp. 120–129, 2017.

[16] S. Ghosh, A. Majumder, J. Goswami, A. Kumar, S. P. Mohanty, and B. K. Bhattacharyya, "Swing-Pay: One Card Meets All User Payment and Identity Needs: A Digital Card

Module using NFC and Biometric Authentication for Peer-to-Peer Payment," *IEEE Consum. Electron. Mag.*, vol. 6, no. 1, pp. 82–93, 2017.

[17] P. Teengam *et al.*, "NFC-enabling smartphone-based portable amperometric immunosensor for hepatitis B virus detection," *Sensors Actuators B Chem.*, vol. 326, p. 128825, Jan. 2021.

[18] Y. W. Juen and D. Balachandran, "Predicting the diffusion of NFC-enabled smartphone payment in Malaysia," *Int. J. Model. Oper. Manag.*, vol. 8, no. 3, p. 266, 2021.

[19] A. E. Varjovi and S. Babaie, "Green Internet of Things (GIoT): Vision, applications and research challenges," *Sustain. Comput. Informatics Syst.*, p. 100448, Sep. 2020.

[20] S. Naraparaju, P. Jalapati, and K. Nara, "Smart Poster for Tourism Promotion Through NFC Technology," Springer, Singapore, 2019, pp. 507–519.

[21] U. Demir Alan and D. Birant, "Server-Based Intelligent Public Transportation System with NFC," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 1, pp. 30–46, 2018.

[22] K. S. Staykova and J. Damsgaard, "The race to dominate the mobile payments platform: Entry and expansion strategies," *Electron. Commer. Res. Appl.*, vol. 14, no. 5, pp. 319–330, 2015.

[23] Y. J. Ng, "Near field communication (NFC) mobile payment in Malaysia: a partial least square-structural equation modelling (PLS-SEM) approach," *Int. J. Model. Oper. Manag.*, vol. 7, no. 2, p. 134, 2019.

[24] D. Veloz-Cherrez and J. Suárez, "NFC-Based Payment System Using Smartphones for Public Transport Service," Springer, Cham, 2019, pp. 34–44.

[25] I. Ramos-de-Luna, F. Montoro-Ríose, and F. Liébana-Cabanillas, "Determinants of the intention to use NFC technology as a payment system: an acceptance model approach," *Inf. Syst. E-bus. Manag.*, vol. 14, no. 2, pp. 293–314, 2016.

[26] M. Cocosila and H. Trabelsi, "An integrated value-risk investigation of contactless mobile payments adoption," *Electron. Commer. Res. Appl.*, vol. 20, pp. 159–170, 2016.

[27] X. Pu, F. T. S. Chan, A. Y. L. Chong, and B. Niu, "The adoption of NFC-based mobile payment services: an empirical analysis of Apple Pay in China," *Int. J. Mob. Commun.*, vol. 18, no. 3, p. 343, 2020.

[28] D. A. Ortiz-Yepes, "A review of technical approaches to realizing near-field communication mobile payments," *IEEE Secur. Priv.*, vol. 14, no. 4, pp. 54–62, 2016.

[29] M. M. Gharamaleki and S. Babaie, "A New Distributed Fault Detection Method for Wireless Sensor Networks," *IEEE Syst. J.*, vol. 14, no. 4, pp. 4883–4890, 2020.

[30] F. S. M. Tafti, S. Mohammadi, and M. Babagoli, "A new NFC mobile payment protocol using improved GSM based authentication," *J. Inf. Secur. Appl.*, vol. 62, p. 102997, Nov. 2021.

[31] C. Peres, M. Emam, H. Jafarzadeh, M. Belcastro, and B. O'Flynn, "Development of a Low-Power Underwater NFC-Enabled Sensor Device for Seaweed Monitoring," *Sensors*, vol. 21, no. 14, p. 4649, Jul. 2021.

[32] A. B. M. Alim Al Islam, T. Chakraborty, T. A. Khan, M. Zoraf, and C. S. Hyder, "Towards defending eavesdropping on NFC," *J. Netw. Comput. Appl.*, vol. 100, pp. 11–23, Dec. 2017.

[33] C. Thammarat and W. Kurutach, "A lightweight and secure NFC-base mobile payment protocol ensuring fair exchange based on a hybrid encryption algorithm with formal verification," *Int. J. Commun. Syst.*, vol. 32, no. 12, p. e3991, Aug. 2019.

Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 241-250
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Content-based Retrieval of Tiles and Ceramics Images based on Grouping of Images and Minimal Feature Extraction

Simin RajaeeNejad[1], Farahnaz Mohanna[1*]

[1]. Department of Communication Engineering, University of Sistan and Baluchestan, Zahedan, Iran

Received: 11 Feb 2022/ Revised: 04 Oct 2022/ Accepted: 27 Nov 2022

Abstract

One of the most important databases in the e-commerce is tile and ceramic database, for which no specific retrieval method has been provided so far. In this paper, a method is proposed for the content-based retrieval of digital images of tiles and ceramics databases. First, a database is created by photographing different tiles and ceramics on the market from different angles and directions, including 520 images. Then a query image and the database images are divided into nine equal sub-images and all are grouped based on their sub-images. Next, the selected color and texture features are extracted from the sub-images of the database images and query image, so, each image has a feature vector. The selected features are the minimum features that are required to reduce the amount of computations and information stored, as well as speed up the retrieval. Average precision is calculated for the similarity measure. Finally, comparing the query feature vector with the feature vectors of all database images leads to retrieval. According to the retrieving results by the proposed method, its accuracy and speed are improved by 16.55% and 23.88%, respectively, compared to the most similar methods.

Keywords: Content-based Retrieval; Feature Vector; Tile and Ceramic; Accuracy and Speed of Retrieval.

1- Introduction

Nowadays with the development of computer technology, easy access to images has been achieved. On the other hand, with the expansion of digital images, a search for the desired images has become a more complex process. Also, with increasing the number of images, large databases have been created that are essential for many systems to organize, store, and retrieve these databases. Meanwhile, image retrieval methods have also grown [1]. The image retrieval process allows the database users to search for their favorite images. Initially, an image was retrieved with keywords that captured the semantic information of an image. This method was very useful for small databases because the total number of images of databases could be described with just a few hundred keywords. But with an expansion of the database, the size of images has also changed. Larger images have different components that a keyword is not enough to describe [2]. Therefore, content-based image-based retrieval (CBIR) was proposed. The CBIR system has three main components including, extracting low-level image features such as color, texture, and shape, storing these features, and measuring similarity. The main differences between different CBIR systems are in the method of extracting features and how similarity is measured [3].

One of the most important databases in e-commerce is the tile and ceramic database, for which no specific retrieval method has been provided so far. In this paper, the aim is to design a retrieval method based on the content of tiles and ceramics images that saves the user time and money. For this purpose, to prepare this image database, 520 different tiles and ceramics available in the market were photographed at different angles and directions. Then other information such as thickness, degree, etc. were attached to each image. In such a CBIR system, when the user inquires about any type of tile or ceramic image of interest, in addition to viewing all the similar images in the database, also receives additional information that cannot be detected just by viewing the image. On the other hand, due to the similarity of the images, it was not possible to use retrieval methods based on the keywords. As a result, a CBIR system is used to achieve this goal. Therefore, suitable features have to be selected for different images of tiles and ceramics.

So, after reviewing and implementing several previous research [4-24], the Average Color Dominance (ACD) and HSV (Hue-Saturation-Value) color space features and the texture features including energy and contrast are selected.

✉ **Corresponding Author**
f_mohanna@ece.usb.ac.ir

Then, according to the existing designs of tiles and ceramics in the market, a database including 520 different tiles and ceramics images is created. Then, each image of the database is divided into nine sub-images. Next, all the images in our tiles and ceramics database are grouped into three clusters based on their sub-images. In continuing, selected features are extracted from each image in the database and query image. These selected features are the minimum features that are required to reduce the amount of computations and information stored, as well as speed up the retrieval. Finally, Feature vectors are formed for each image, and the feature vector of the query image is retrieved among the feature vectors of images in our database. The retrieval results of the proposed method showed that its accuracy and speed are improved by 16.55% and 23.88%, respectively, compared to the most similar methods. In the following, related works are presented in section 2. The proposed method is introduced in section 3. In section 4, retrieval results of the proposed method are shown on our database of tile and ceramic, and the proposed method is also compared with a number of the most similar methods. The paper concludes with a conclusion in section 5.

2- Related Work

The use of color and texture visual features in the CBIR was enhanced in [1] by adding a new color feature called average color dominance (ACD) which was tried to enhance color description using the dominant colors of an image. Images were also retrieved in [2] by their contents such as color, texture, shape, or objects. Thus, the degree of similarity between the query image and database images was measured by color, texture features extraction, shape feature extraction similarity, or object presence between the two images. It was shown that a better way to retrieve images was using multi-visual features. Furthermore, it was provided in [3] both scale and rotational invariance in images for the CBIR. A CBIR method was proposed in [4], which was a modified form of the color averaging technique. In the previous technique, row mean, column mean, forward diagonal mean, backward diagonal mean, sum of row, columns, and diagonal intensity values respectively were calculated for the similarity measure. However, it was possible that the sum was the same, but the intensity values were not the same for the same corresponding positions in both the query image and database image, as a result, irrelevant images were retrieved. To overcome this problem, the position of intensity values presented in both the query image and database images were considered for the similarity measure as well. A CBIR method was also introduced in [5] using both the color and texture features. To extract the color feature, the color moment was calculated where an image was in the HSV color space. To extract the texture feature, the Ranklet transform was performed on the grey-scale image, and then from the Ranklet images, the texture feature was extracted by calculating the texture moments. Furthermore, it was shown in [6] that the image retrieval using a single feature did not provide a good solution for the accuracy and efficiency. It was also shown the most important visual features were color and texture. Therefore, a technique for retrieving the images was presented based on their content namely color, texture, and combination of both. In addition, an algorithm was introduced in [7] that incorporated all three features of color, shape and texture to give the advantages of various other algorithms to improve the accuracy and performance of image retrieval. The accuracy of HSV color space-based color histogram-based matching gave better retrieval results. The speed of shape-based retrieval was enhanced by considering approximate shape rather than the exact shape. The Gray-Level Co-occurrence Matrix (GLCM) was used to extract the texture features. The feature matching procedure was based on the Canberra distance. A CBIR method was proposed in [8] by extracting both color and texture feature vectors using the discrete Wavelet transform and the Self Organizing Map (SOM) artificial neural networks. At query time, texture vectors were compared using a similarity measure of the Euclidean distance, and the most similar image was retrieved. In addition, other relevant images were retrieved using the neighborhood of the most similar image from the clustered data set via SOM. Two algorithms of the color histogram and Wavelet-based color histogram (WBCH) were implemented in [9] which were used respectively for the color and combined color and texture features extraction from both the query image and database images. Then the retrieved images by both algorithms were compared on the basic values of parameters that showed the WBCH algorithm was better than the color histogram algorithm in terms of the retrieval time as it took less time for the image retrieval. A method was introduced in [10] which presented the query image as an input and got related images as the output that matched the content of the query image. The findings were not only based on the color, texture, and shape but also traced the underlying points of an image. At first, the images were retrieved based on the color, then taken after by texture, and finally traced by the underlying graphical structure. Two algorithms of color and texture extraction were introduced in [11]. The Color histogram that was rotation invariant about the view axis, was used to represent color features but it could not entirely characterize the image. The texture feature extraction was presented based on the Gabor filter. A CBIR system was also presented in [12] that used a combination of the completed local binary pattern (CLBP) and color autocorrelogram. The CLBP features were extracted on a multi-resolution multi-direction filtered domain of value component. The color

autocorrelogram features were extracted in two dimensions of hue and saturation components. Furthermore, an image retrieval method was presented in [13] based on the texture structure histogram (TSH) and Gabor texture feature extraction. In the TSH technique to describe the texture feature, the edge orientation and HSV color information methods were used. To make the image content more reasonable non-equal interval quantization scheme was used. The image texture was retrieved using the mean and variance of the Gabor filtered image. To get the same dominant direction, rotation normalization was used. The comparison of both texture techniques was discussed. A hybrid-feature extraction approach was described in [14] to solve the problem of designing a CBIR system manually. Two features were used to retrieve images such as color and texture. The color feature was extracted using different color spaces. The texture feature was extracted using the GLCM. An image was retrieved by combining the color and texture features. A method was introduced in [15] that combined both color and texture features in a hierarchical manner to retrieve an image and showed its advantage. They also introduced a method of image segmentation for feature extraction. The proposed hierarchical approach was applied to the standard INRIA dataset. A two phases approach was proposed in [16] to retrieve images from the data set based on color and texture. In the first phase, the HSV global color histogram was used and an automatic cropping technique was introduced to accelerate the features extraction process and enhanced the retrieval accuracy. In the second phase, the joint histogram and GLCM were deployed and the color and texture features were combined to enhance the retrieval accuracy. Finally, the images were classified and retrieved using the K-means algorithm. Two experiments were conducted using the WANG database consisting of 10 different classes each with 100 images. A three-level hierarchical CBIR system was presented in [17] where, each level of the hierarchy used either the texture, shape, or color image features to reduce the size of the image database by discarding the irrelevant images and at the final level of the hierarchy, it extracted the most analogous images from the reduced image database. The adaptive Tetrolet transform was used to extract the texture features from the regions of interest in images. To extract the shape features, the edge joint histogram was proposed which used the orientation of edge pixels and their distance from the origin together to create the joint histogram. For the color feature extraction, another color channel correlation histogram was introduced. The order of the three different feature extraction processes on each level of the hierarchy was not rigid because it was difficult to predict the proper order for the highest retrieval.

A CBIR approach was proposed in [18] that combined the visual and textual features to retrieve images. Firstly, the method classified the query image as textual and non-textual. The textual query image formed a bag of textual words. The visual salient features were extracted from the non-textual query image and formed a bag of visual words. Next, the method fused the visual and textual features, and the top similar images were retrieved based on the fused feature vector. Three modes of retrieval, image query, keywords, and a combination of both were used. A technique was proposed in [19] to fuse the spatial color information with the shaped extracted features and object recognition. For the RGB (Red-Blue-Green) channels L2 spatial color arrangements were applied and features were extracted, thereby fused with the intensity-ranged shapes formed by connecting the discovered edges and corners for the grey level image. They used the perifoveal receptive field estimation with 128-bit cascade matching with symmetric sampling on the detected interest points to discover information for the complex, overlay, foreground, and background objects. Firstly, the process was accomplished by reducing the massive feature vectors, selecting a high variance coefficient, and secondly obtaining the indexing and retrieval by employing a Bag-of-Words approach. An image retrieval method was presented in [20] based on a combination of the local texture information derived from two different texture descriptors. First, the color channels of the input image were separated. The texture information was extracted using two descriptors as the evaluated local binary patterns and predefined pattern units. After extracting the features, the similarity matching was done based on the distance criteria. A deep cross-modality Hashing network was proposed in [21] that first, the optical images with three channels were transformed into four different types of single channel images to increase the diversity of training modalities. This helped the network to focus mainly on extracting the contour and texture shared features and made it less sensitive to the color information for images across modalities. Second, it combined any type of randomly selected transformed images and their corresponding optical images to form image pairs that were fed into the networks. The training strategy, with paired image data, eliminated the large cross-modality variations caused by different modalities. Finally, the triplet loss, in combination with the Hash function, helped the modal to extract the discriminative features of images and upgraded the retrieval efficiency. An image retrieval algorithm was introduced in [22] by integrating the image color information and surface geometry principal curvatures information. First, the color histogram of the quantized color image was obtained. Simultaneously, the Hessian matrix was used to extract the image texture information, and the joint histogram of the oriented gradient with the mixed sampling and multi-scale was constructed. Then, the color histogram and histogram of the oriented gradient were fused to obtain the final joint histogram. A CBIR method was presented in [23] that first modified the

traditional microstructure descriptor (MSD) to capture the direct relationship between the shape and texture features and that between the color and texture features. Then, the image uniform local binary patterns (LBP) histogram was extracted to capture the color difference information. At the image comparison stage, first, the image descriptors were compared to compute their similarities. The similarity between each pair of images was then updated by considering the similarities to comparable images within the dataset. Accordingly, the final similarities of the images were obtained. A CBIR technique was proposed in [24] that focused on extraction and reduction in the multiple features. To obtain the multi-level decomposition of the image, the discrete Wavelet transformation was applied to the RGB channels initially, and a local binary pattern texture descriptor was applied to the transformed images. The additional information was also extracted using the magnitude information. Then, the GLCM description was used to extract the statistical characteristics for texture image classification. The proposed technique was applied to the CORAL dataset with the help of the particle swarm optimization-based feature selector to minimize the number of features that could be used during the classification process. In addition, three classifiers including, the support vector machine, K-nearest neighbor, and decision tree, were trained and tested. A CBIR system was presented in [25] in a hierarchical mode based on three visual features like color, texture, and shape for retrieving the most relevant images from a large-scale image database where in each stage the retrieval process discarded the irrelevant images by filtering process and as a result, the search space was reduced in subsequent stages. The system implementation results were evaluated on the GHIM-10k and Corel-1k databases according to the average precision criterion. The value of average precision was reported at 74.87% and 80.08% on the GHIM-10k and Corel-1k databases respectively. A feature descriptor named "correlated microstructure descriptor (CMSD) was proposed in [26] to incorporate the high-level semantic concepts for image retrieval. The CMSD relied on the various aspects of an input image by correlating color, texture orientation, and intensity information. The correlated information was mapped with the microstructure to extract all the fine detail within the subject area. A richer multi-directional edge orientation map was also constructed by quantizing the edge obtained by the 4D Sobel operator into 6 levels. Moreover, the local information was obtained by quantizing the V component of HSV into 10 levels. Experiments were performed on standard databases of Corel-1k, Corel-5k, and Corel-10k. The method implementation results evaluation according to the average precision criterion showed the best value of 78.54%.

A relevance feedback retrieval (RFRM) method was introduced in [27] for the content-based image medical retrieval (CBMIR). The feedback was based on voting values performed by each class in the image repository. A group of color and texture features was extracted based on the color moments and GLCM texture features. For similarity measure, eight common similarity coefficients were used. After briefly researching and applying a single random image query, the top images retrieved from each class were used as voters to select the most effective similarity coefficient that was used for the final searching process. The method was implemented on the Kvasir database, which has 4000 images divided into 8 classes. The method implementation results showed an average precision of 85% on the Kvasir database. A method was proposed in [28] that mainly concentrated on extracting the dominant color information of the image using the clustering process. The clustering process was initiated by the proposed seed point selection approach. This approach derived the number of seed points using the first-order statistical measure and the maximum range of the distributed pixel values. Moreover, the method gave equal priority to dominant color and its occurrence information in calculating the similarity between query and database images. The performance of the method was investigated on SIMPLIcity, Corel-10k, OT-scene, Oxford flower, and GHIM databases. A feature detector was presented in [29] by performing non-max suppression after detecting edges and corners based on corner score and pixel derivation-based shapes on intensity-based interest points. Thereafter, interest point description is applied to interest point features set by using symmetric sampling to cascade matching produced by validating dense distributed receptive fields after estimating perifoveal receptive fields. Spatial color-based features vector was fused with retinal and color-based feature vector extracted after applying L2 normalization on a spatially arranged color image. Dimensions were reduced by using PCA on massive feature vectors produced after symmetric sampling and transmitted to bag-of-word in fused form for indexing and retrieval of images. Experiments were performed on Corel-1k, Corel-10k, Caltech-101, image net, alot, coil, ftvl, 102-flowers, and 17-flowers databases. The method was also compared with seven other descriptors. A comprehensive survey of deep learning based developments in the past decade was presented in [30] for the CBIR. The categorization of existing methods from different perspectives was performed for the greater understanding of progress. The taxonomy used in this survey covered different supervision, different networks, different descriptor type and different retrieval type. A performance analysis was also performed using the state-of-the-art methods.

3- Proposed Method

The steps of the proposed retrieval method are shown in Fig. 1. These steps are described as follows.

3-1- Image Division

In this step, tiles and ceramics images in the database that were photographed by a Sony Cyber-shot camera, 1/12 Megapixels, are resized to 240*150 pixels. Then, each image is divided into nine equal sub-images of size 50*80 pixels. Since many tiles and ceramics designs are concentrated in the middle, this division method causes the original design of tile or ceramic to be located in the middle and around the perimeter. The image division process increases the retrieval accuracy, but it also increases the retrieval time. On the other hand, based on the existing designs of tiles and ceramics in the market, the images of our database can be grouped into three clusters; images that have a design in the middle, a design in the margins, and both. Therefore, we group all the tiles and ceramics in our database into three clusters in order to retrieve them faster, with less computation, and compensate for the time increasing according to the image division process.

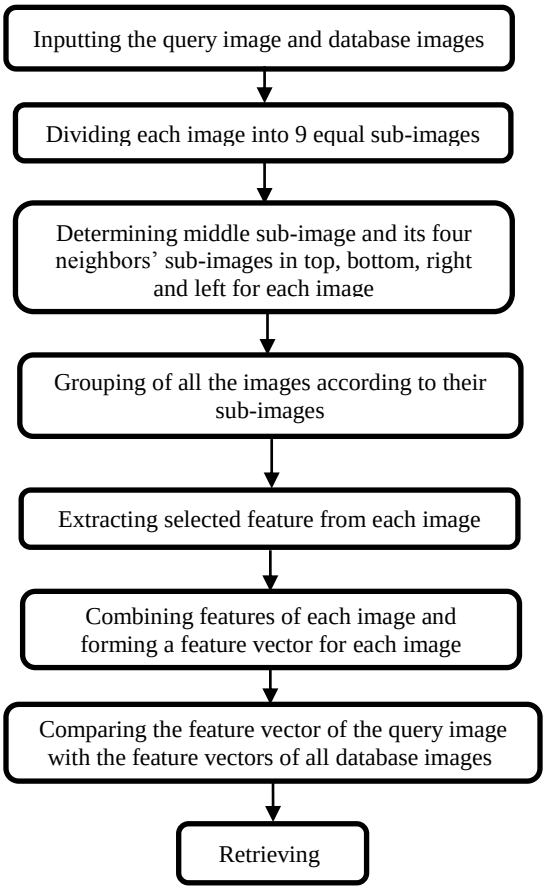


Fig. 1 Steps of the proposed retrieval method for tile and ceramic images database

3-2- Grouping of the Database Images

The grouping process is performed according to the flowchart that is shown in Fig. 2. After determining the five sub-images of each database image, if in the middle sub-image, a dominant color occupies more than half of the pixels, the design is not in the middle but in the margin (group 1). Therefore, there is no need to examine the dominant colors of the other four sub-images. However, the feature vector extraction has to be performed from the four sub-images. Because in the images of group 1, the color design may be at the top, left, right, and bottom of the image (Fig. 3). Finally, the feature vectors of four sub-images are combined and formed a feature vector for the original image.

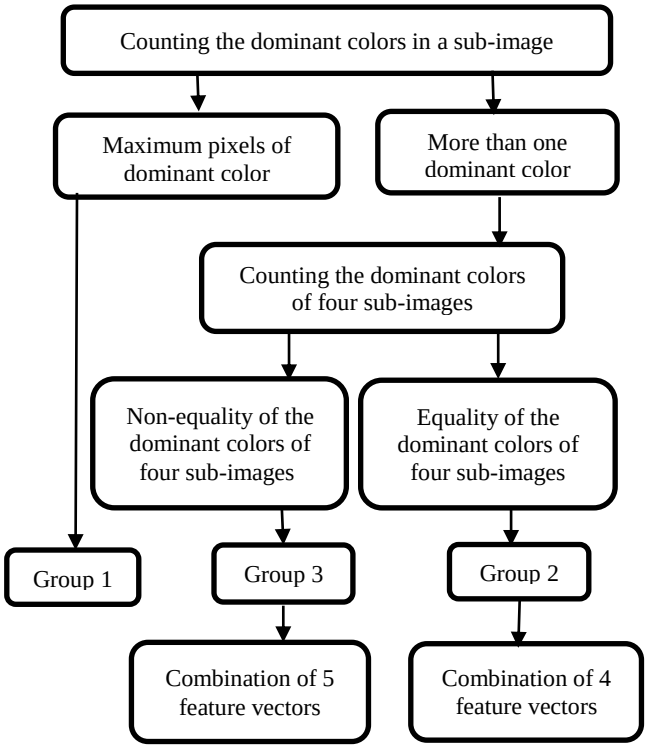


Fig. 2 Grouping process of the database of tile and ceramic images

In the second group, the design is in the middle of the images. Therefore, it is existed more than one color in counting the number of pixels of the dominant colors in the middle sub-image. So the number of pixels of the image is divided between two or three colors. In this case, the number of pixels of the dominant colors in the four sub-images also has to be counted. If the number of dominant colors of all four sub-images is equal, then for certain, the design is in the middle, and the feature vector has to be extracted only from the middle sub-image (Fig. 4). The third group includes images that the design is in the middle or in the margin. Therefore, the number of dominant colors in the middle sub-image and the four

other sub-images, is more than one dominant color and the number of dominant colors in the four sub-images is not equal. So, the feature vector has to be extracted from all the five sub-images of the original image and finally, the five feature vectors are combined and formed a feature vector for the original image (Fig. 5).

3-3- Feature Extraction

In the CBIR retrieval, features have to be selected somehow to increase accuracy and decrease retrieval time. Therefore, the smaller the number of features and amount of computation, the faster is the retrieval. Of course, a small number of features should be sufficient to maintain retrieval

Fig. 3 The sample images of tiles and ceramics in group 1

Fig. 4 The sample images of tiles and ceramics in group 2

Fig. 5 The sample images of tiles and ceramics in group 3

Accuracy. In the proposed method, in order to select the appropriate features for tiles and ceramics images, we reviewed previous research available in various references. Then we tested several features and selected the ACD, HSV, energy, and contrast features according to the same reasons mentioned above. The results of these experiments are shown in section 4. Regarding the feature extraction method, by specifying the group, the feature extraction is done only in a few sub-images instead of in the whole of the original image, which leads to a reduction in the computational volume and thus speeds up the retrieval.

3-3-1- HSV Color Feature

The HSV color space is closer to the human eye vision due to its uniformity and is more suitable for retrieval. Each of the HSV components (S, V, and H) has a wide range of values. In the proposed method, the H component is quantized into 8 bins and each of the S and V components into 3 bins. Therefore, 72 color features are obtained according to Equations (1). Then, these three feature vectors are combined by Equation (2), and the final HSV feature vector is formed [4].

$$S=\begin{cases}0 & if\ s\epsilon[0,0.2)\\ 1 & if\ s\epsilon[0.2,0.7)\\ 2 & if\ s\epsilon[0.7,1)\end{cases}$$

$$V=\begin{cases}0 & if\ v\epsilon[0,0.2)\\ 1 & if\ v\epsilon[0.2,0.7)\\ 2 & if\ v\epsilon[0.7,1)\end{cases}$$

$$H=\begin{cases}0 & if\ h\epsilon[316,20)\\ 1 & if\ h\epsilon[21,40)\\ 2 & if\ h\epsilon[41,75)\\ 3 & if\ h\epsilon[76,155)\\ 4 & if\ h\epsilon[156,190)\\ 5 & if\ h\epsilon[191,270)\\ 6 & if\ h\epsilon[271,295)\\ 7 & if\ h\epsilon[296,315)\end{cases} \quad (1)$$

$$G=9H+3S+V \quad (2)$$

3-3-2- ACD Color Feature

To extract the ACD color feature [1], first, the original image is quantized into 38 colors, then pixels that the pixel itself and its 4 neighbors have the same color, are determined. Next, from these pixels, the top 3 most counted colors that have a similar color to their four neighbors, are separated and for each of these three colors, the average of the B, R, and G components are calculated. Finally, these three average values are again re-averaged and this value is considered as the ADC color feature of the original image.

3-3-3- Contrast and Energy

To determine the contrast and energy as the texture features, first, the gray-level co-occurrence matrix, $C_d(i,j)$ of each image is formed then, energy and contrast are calculated according to Equations (3) and (4) [11].

$$Contrast = \sum_i \sum_j (i-j)^2 C_d(i,j) \quad (3)$$

$$Energy = \sum_i \sum_j C_d^2(i,j) \quad (4)$$

Where $C_d(i,j)$ is defined by first specifying a displacement vector d and counting all pairs of pixels separated by d having gray levels i and j. Energy is a measure of the textual uniformity of an image. Contrast is the difference moment of the C matrix and it measures the amount of local variation in an image.

3-3-4- Feature Vector of Each Group

Finally, in our database, for 150 images of tiles and ceramics in group 1, the first four feature vectors are extracted from 4 sub-images of each image and placed side by side in a feature vector with a size of 4*4. Then 150 feature vectors in size of 4*4 are combined and placed side by side in a feature vector with a size of 4*4*150 for group 1. Similarly, for 120 images of tiles and ceramics in group 2 of our database, because we only consider the middle sub-image, the final feature vector becomes 1*4*120. In addition, the final feature vector of group 3 with 250 images is in size of 5*4*250, because we consider 5 sub-images of the original image.

3-3-5- Retrieving

When a query image is given to the proposed CBIR system, the query image is first divided into the nine equal sub-images, and then its query group is determined according to the flowchart in Fig. 2. Next, the feature vector of the query image is extracted. Finally, only the feature vector of the specified group is compared to the feature vector of the query image based on the least Euclidean distance. Note that all the previous steps of the proposed method are performed offline, and only retrieving step is online.

4- Simulation Results

4-1- Evaluation criterion

To compare the retrieval performance of the proposed method with the most similar methods, the precision criterion is calculated according to Equation (5) [4].

$$Precision = \frac{Number\ of\ retrieval\ relevant\ images}{Total\ number\ of\ retrieval\ images} \quad (5)$$

4-2- Simulation Results

Methods in [1] and [4] are very similar to the proposed method. Therefore, first, we compare the retrieval accuracy and retrieval time of the proposed method with these two methods. The comparison results are shown in Table 1 according to the average precision criterion and the online running time of the methods. As it is seen, the retrieval accuracy of the proposed method is increased and its retrieval time is decreased on our tiles and ceramics database by selecting fewer features and extracting them from only a few numbers of the sub-images of the query image. In addition, the retrieval accuracy of the proposed method is also compared with a number of other methods that are mentioned in the references section. The results are also shown in Table 2 by using the average precision criterion. Furthermore, the proposed method is tested on other famous image databases to show its retrieval performance on different image databases as well. The results are shown in Table 3.

4-3- Results of the Best Features Selection

Retrieval of all images of tiles and ceramics in our database was performed by applying the proposed method based on the various features extraction such as the ACD, HSV, color moment, energy, contrast, and Ranklet. The selection of these features was based on the research that was carried out in 10 years from 2012 to 2022. Evaluations of these retrieval results are reported in Fig. 6 and Fig. 7, using both time and precision criteria. As it is seen, the ACD, HSV, energy, and contrast are the best choice between different features. Therefore, in the feature extraction stage of the proposed method, only these four features are extracted, which are the minimum number of features that leads to a reduction in the computational volume and thus speeds up the retrieval.

Table 1: Retrieval accuracy and time of the proposed method, in comparison with the two most similar methods on our database

Method	*Average Precision*	*Average Time (Sec.)*
in [1]	61.42%	1.53
in [4]	72.56%	0.67
Proposed	**84.57%**	**0.51**

Table 2: Retrieval accuracy of the proposed method, in comparison with a number of other methods on our database

Method	*Average Precision*	*Method*	*Average Precision*
in [9]	83.15%	in [20]	81.23%
in [12]	82.47%	in [22]	83.30%
in [15]	83.08%	in [23]	58.52%
in [16]	83.22%	in [24]	78.52%
in [19]	83.12%	in [25]	78.92%
in [26]	76.57%	in [28]	83.29%
in [29]	82.79%	**Proposed**	**84.57%**

Table 3: Retrieval accuracy of the proposed method on other databases

Proposed Method	*Average Precision*	*Database*	*Total number of images*
	89.73%	WANG	1000
	95.11%	INRIA	1500
	94.57%	CORAL	1000
	84.57%	Our	520

4-4- Examples of Retrieval Results

A few examples of the query images and their retrieval results by using the proposed method are shown in Figs. 8, 9, and 10.

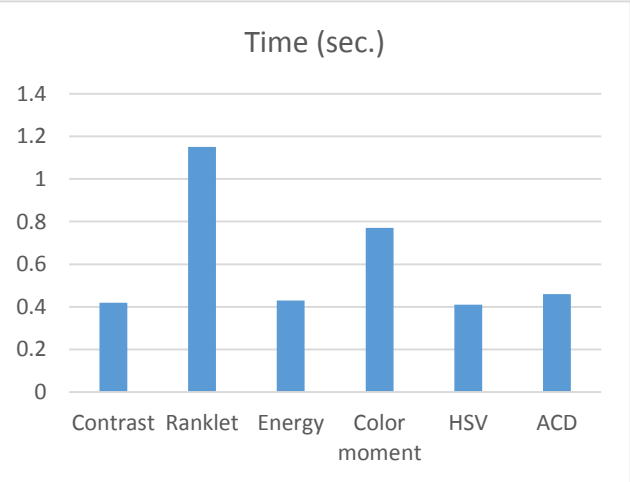


Fig. 6 Retrieval time of the proposed method on our database based on extracting different features

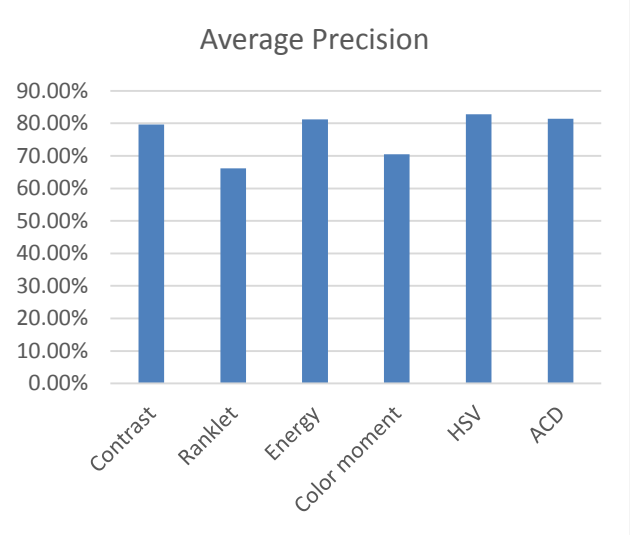


Fig. 7 Retrieval evaluation of the proposed method on our database based on various features extraction using the average precision criterion

5- Conclusion

To have a specific retrieval approach for the tiles and ceramics database in e-commerce, a CBIR method was proposed. In the proposed method, according to the division of our database images into nine sub-images, the images were grouped. Then, four selected features were extracted from the main sub-images in each group and the feature vectors of the database images were formed. The same process was performed for a query image until its feature vector was formed. Finally, by comparing the query feature vector with the feature vector of each database image based on the Euclidean distance, retrieval results were presented. After implementing the proposed method on our database, the results were compared first to the most similar methods, which showed the retrieval accuracy and speed were improved by 16.55% and 23.88% respectively. In addition, these results were also compared to a number of the recent methods with less similarity, and still, the proposed method retrieval accuracy even with fewer extracted features was 1.5% higher. Furthermore, the proposed method was run on other famous image databases as well. The results showed that in the standard databases with high-quality imaging and more images, the accuracy of the proposed method was even higher than in our own database.

The implementation results on the best features selection for the retrieving showed that the contrast, energy, HSV, and ACD were the best choice between different features. The retrieval time of the proposed method was 0.51 sec. which is enough fast as a real-time method. As a result, minimum feature selection but sufficient was very effective in improving the speed and accuracy of retrieval. Also, using only the main sub-images as defined in each group instead of the whole input image reduced the computational volume and therefore reduced storage space and increased speed without compromising accuracy. For future work, an investigation of the number of sub-images of the original image in the division process is recommended.

Fig. 8 The query image and its retrieval images by applying the proposed method on our database. All the similar images, even with different angles and directions, are retrieved for the query image.

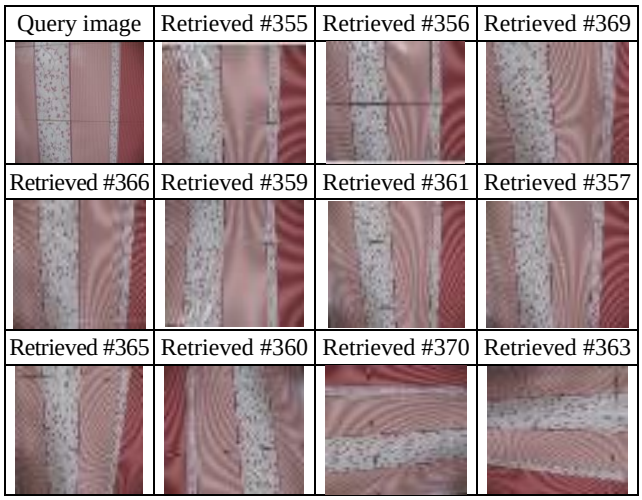


Fig. 9 The query image and its retrieval images by applying the proposed method on our database. All the similar images are retrieved.

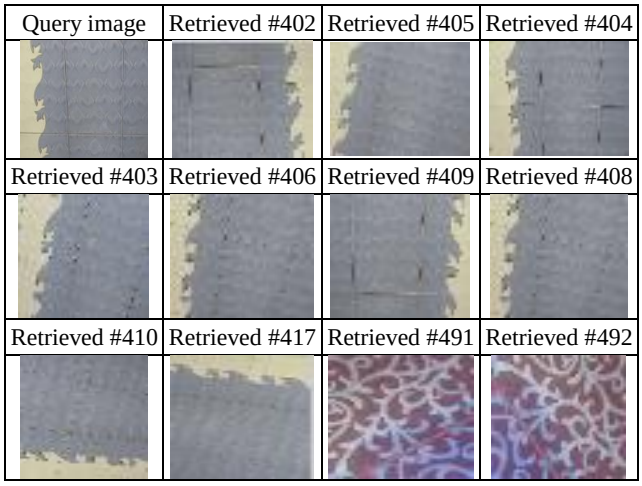


Fig. 10 The query image and its retrieval images by applying the proposed method on our database. Images #491 and #492 were retrieved incorrectly.

References

[1] T. Mehyar, and J. O. Atoum, "An enhancement on content-based image retrieval using color and texture features", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No. 4, 2012, pp. 488-496.

[2] C. M. Ibraheem and G. U. Reddy, "Content based image retrieval using HSV color, shape, and GLCM texture", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, No. 10, 2015, pp. 1-6.

[3] P. Sharma, and D. Dubey, "Color averaging technique using dominant color for content based image retrieval", International Journal of Computer Science, Vol. 10, No. 3, 2013, pp. 603-607.

[4] Ahmed J. Afifi, and Wesam M. Ashour, "Content-based image retrieval using invariant color and texture features", in International Conference on Digital Image Computing Techniques and Applications, 2012, pp. 1-6. DOI:10.1109/DICTA.2012.6411665

[5] T. Prathiba, N. M. Mary Sindhuja, and S. Nisharani, "Content based image retrieval based on spatial constraints using Lab view", International Journal of Engineering Research and Technology, Vol. 2, No. 1, 2013, pp. 1-6.

[6] V. Khandave, and N. Mishra, "CBIR by integration of color and texture features", International Journal of Recent Development in Engineering and Technology, Vol. 2, No. 1, 2014, PP. 1-6.

[7] R. K. Lingadalli and N. Ramesh, "Content based image retrieval using color, shape and texture", International Advanced Research Journal in Science, Engineering and Technology, Vol. 2, No. 6, 2015, pp. 40-48.

[8] A. Huneiti, and M. Daoud, "Content-based image retrieval using SOM and DWT", Journal of Soft Engineering and Applications, Vol. 8, 2015, pp. 51-61.

[9] S. Kaur, and N. Kaur, "Content based image retrieval using color histogram and Wavelet based color histogram algorithms", International Journal of Engineering Research and General Science, Vol. 4, No. 3, 2016, pp. 530-535.

[10] R. Bulli Babu, V. Vanitha, and K. Sai Anish, "Content based image retrieval using color, texture, shape and active re-ranking method", Indian Journal of Science and Technology, Vol. 9, No. 17, 2016, pp. 1-5.

[11] N. Jain, and S.S. Salankar, "Content based image retrieval using combined color and texture features", IOSR Journal of Electrical and Electronics Engineering, Vol. 11, No. 6, 2016, pp. 53-58.

[12] H. H. Bu, N. C. Kim, C. J. Moon, and J. H. Kim, "Content-based image retrieval using multi-resolution multi-direction filtering-based CLBP texture features and color autocorrelogram features", Journal of Information Processing Systems, Vol.16, No. 4, 2020, pp. 991-1000.

[13] M. B. Suresh, and B. Mohankumar Naik, "Content based image retrieval using texture structure histogram and texture features", International Journal of Computational Intelligence Research, Vol. 13, No. 9, 2017, pp. 2237-2245.

[14] M. B. Suresh, and B. Mohankumar Naik, "Content based image retrieval using color and texture content", International Journal of Computer Trends and Technology, Vol. 48, No. 2, 2017, pp. 78-84.

[15] D. Sarala, T. Kanikdaley, S. Jogi, and R. K. Chaurasiya, "Content-based image retrieval using hierarchical color and texture similarity calculation", International Journal of Advanced Trends in Computer Science and Engineering, Vol. 7, No. 2, 2018, pp. 11-16.

[16] J. Q. Alnihoud, "Image retrieval system based on color global and local features combined with GLCM for texture features", International Journal of Advanced Computer Science and Applications, Vol. 9, No. 9, 2018, pp. 164-171.

[17] J. Pradhan, S. Kumar, A. Kumarpal, and H. Banka, "A hierarchical CBIR framework using adaptive tetrolet transform and novel histograms from color and shape features", Digital Signal Processing, Vol. 82, 2018, pp. 258-281.

[18] S. Unar, X. Wang, C. Wang, and Y. Wang, "A decisive content based image retrieval approach for feature fusion in visual and textual images", Journal of Knowledge-Based Systems, Vol. 179, 2019, pp. 8-20.

[19] K. T. Ahmed, S. Ummesafi, and M. Iqbal, "Content based image retrieval using image features information fusion", Journal of Information Fusion, Vol. 51, 2019, pp.76-99.

[20] N. Hor and S. Fekri-Ershad, "Image retrieval approach based on local texture information derived from predefined patters and spatial domain information", International Journal of Computer Science Engineering, Vol. 8, No. 6, 2019, pp.246-254.

[21] W. Xiong, Z. Xiong, Y. Zhang, Y. Cui, and X. Gu, "A deep cross-modality Hashing network for SAR and optical remote sensing images retrieval", IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing, Vol. 13, 2020, pp. 5284-5296.

[22] H. Zhang, M. Jiang, and Q. Kou, "Color image retrieval algorithm fusing color and principal curvatures information", IEEE Access, Vol. 8, 2020, pp. 184945-184954.

[23] D. Niu, X. Zhao, X. Lin, and C. Zhang, "A novel image retrieval method based on multi-features fusion", Signal Processing: Image Communication, Vol. 87, 2020.

[24] M. Garg and G. Dhiman, "A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variations", Neural Computing and Applications, Vol. 33, 2021, pp. 1311-1328.

[25] N. Varish, A. Kumar Pal, R. Hassan, M. K. Hasan, A. Khan, N. Parveen, D. Banerjee, V. Pellakuri, N. Ul Haq, and I. Memon, "Image retrieval scheme using quantized bins of color image components and adaptive Tetrolet transform", IEEE Access, Vol. 8, 2020, pp. 117639-117665.

[26] Hussain Dawood, M. H. Alkinani, A. Raza, Hassan Dawood, R. Mehboob, and S. Shabbir, "Correlated microstructure descriptor for image retrieval", IEEE Access, Vol. 7, 2019, pp. 55206-55228.

[27] Ali Ahmed, "Implementing relevance feedback for content-based medical image retrieval", IEEE Access, Vol. 8, 202, pp. 79969-79976.

[28] L. K. Pavithra, and T. Sree Sharmila, "An improved seed point selection based unsupervised color clustering for content-based image retrieval application", Computer Journal, 2020. DOI:10.1093/Comjnl/bxz017

[29] K. T. Ahmed, S. Aslam, H. Afzal, S. Iqbal, A. Mehmood, and G. S. Choi, "Symmetric image contents analysis and retrieval using decimation, pattern analysis, orientation, and features fusion", IEEE Access, Vol.9, 2021, pp. 57215-57242.

[30] S. Ram Dubey, "A decade survey of content based image retrieval using deep learning", IEEE Transactions on Circuits and Systems for Video Technology, Vol.32, No. 5, 2021, pp. 2687-2704.

Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 251-259
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Spectrum Sensing of OFDM Signals Utilizing Higher Order Statistics under Noise Uncertainty Environments in Cognitive Radio Systems

Mousumi Haque[1*,2], Tetsuya Shimamura[2]

[1].Department of Information and Communication Engineering, University of Rajshahi, Bangladesh.
[2].Graduate School of Science and Engineering, Saitama University, Japan.

Received: 03 Jun 2022/ Revised: 12 Nov 2022/ Accepted: 03 Dec 2022

Abstract

Cognitive radio (CR) is an important issue to solve the spectrum scarcity problem for modern and forthcoming wireless communication systems. Spectrum sensing is the ability of the CR systems to sense the primary user signal to detect an ideal portion of the radio spectrum. Spectrum sensing is mandatory to solve the spectrum scarcity problem and the interference problem of the primary user. Noise uncertainty consideration for orthogonal frequency division multiplexing (OFDM) transmitted signals in severe noise environments is a challenging issue for measuring the performance of spectrum sensing. This paper proposed a method using higher order statistics (HOS) functions including skewness and kurtosis for improving the sensing performance of a cyclic prefix (CP) based OFDM transmitted signal for noise uncertainty. The detection performance of OFDM systems is measured for various CP sizes using a higher order digital modulation technique over a multipath Rayleigh fading channel for low signal-to-noise ratio (SNR) cases. In the proposed method, the CP-based OFDM transmitted signal sensing performance is measured and compared with the conventional methods under noise uncertainty environments. Through comprehensive evaluation of simulation, it is demonstrated that the sensing performance of this method significantly outperforms conventional schemes in the case of noise uncertainty in severe noise environments.

Keywords: Spectrum Sensing; Orthogonal Frequency Division Multiplexing; Skewness; Kurtosis; Cognitive Radio.

1- Introduction

Spectrum sensing in cognitive radio (CR) systems is an important issue in the modern era. The Federal Communications Commission (FCC) reported that some radio frequency bands are heavily used by licensed systems, but there are also many radio frequency bands that are only partly occupied [1]. CR is an approach for solving the scarcity problems of the frequency spectrum [2], [3]. In CR, the radio spectrum status is identified by the spectrum sensing. In recent years, numerous spectrum sensing methods have been proposed to solve spectrum sensing problems [4–6].

The energy detection based spectrum sensing method utilizes the energy of the received primary signal [7–9]. The performance of the energy detection method is not very poor for low signal-to-noise ratio (SNR) cases. Cyclostationary features have been used for detecting the signal for detecting the primary user [10]. When the primary user signal is used for sensing, matched filter detection provides the best sensing performance [11], [12]. However, spectrum sensing in the presence of noise uncertainty is not considered, and the computational complexity is very high.

The correlation based spectrum-sensing methods are very popular due to their low computational complexity and provide good performance over the fading channel [13]. The autocorrelation based spectrum sensing is classified into this category. The time domain autocorrelation property of a cyclic prefix (CP) based orthogonal frequency division multiplexing (OFDM) primary user signal was used for spectrum sensing [14–16]. The spectrum sensing of an OFDM signal under noise

✉ **Mousumi Haque**
mouice@ru.ac.bd

uncertainty conditions for low SNR cases is challenging. Conventional autocorrelation based methods utilize the knowledge of the CP for spectrum sensing [14], [15]. However, in practice, this is very difficult in the real cases. In addition, in [16] CP unknown case was considered. The noise uncertainty is not considered and the detection performance of OFDM transmitted signals is unsatisfactory in severe noise environments.

Higher order statistics are useful in digital signal processing, communication systems, signal detection, and a variety of other applications [17–20]. The higher order statistics, including third order statistics and fourth order statistics, are utilized for sensing OFDM transmitted signals, where the noise uncertainty cases are not considered for OFDM sensing [21]. Although some recent works considered spectrum sensing in noise uncertainty environments, the sensing performance is not very good in severe noise environments [22], [23]. However, the sensing of OFDM signals in noise uncertainty environments for low SNR cases is very important [23]. For these reasons, the major limitations of the existing spectrum-sensing methods result in their poor spectrum sensing performance for low SNR cases in noise uncertainty environments. Therefore, we were motivated for sensing OFDM transmitted signals in the case of noise uncertainty in severe noise environments.

The proposed method utilizes higher order statistics for sensing OFDM signals in noise uncertainty environments. The skewness calculation is utilized for sensing OFDM systems in noise uncertainty environments. Moreover, the kurtosis function is used for further sensing performance improvement of OFDM signals under noise uncertainty in severe noise environments. In the proposed method, the detection performance is evaluated for various CP sizes of OFDM systems under 64-QAM over multipath fading channels with additive white Gaussian noise (AWGN) in noise uncertainty environments. The proposed spectrum sensing method is compared with conventional methods [22], [23] over multipath fading channels under the effect of noise uncertainty. When the noise uncertainty effect is taken into account, the performance of OFDM detection improves dramatically when using skewness based spectrum sensing in severe noise environments. Furthermore, the sensing capability of OFDM transmitted signals increases markedly by utilizing kurtosis based spectrum sensing for low SNR cases under noise uncertainty.

The major contributions of this paper are as follows:

• Firstly, the skewness calculation is used for spectrum sensing under the effect of noise uncertainty. Furthermore, the proposed method is compared for skewness based sensing with and without noise uncertainty cases.
• Secondly, the proposed method is investigated for kurtosis function based spectrum sensing under two scenarios (with and without noise uncertainties).
• Thirdly, we have compared our proposed method with conventional methods [22], [23] in noise uncertainty environments. Simulation results demonstrated that our proposed skewness and kurtosis based spectrum sensing methods significantly improve the sensing performance.

The rest of the paper is arranged as follows. Section 2 presents a brief overview of the related work in spectrum sensing. The problem formulation of spectrum sensing is provided in Section 3. Section 4 represents the methodology of spectrum sensing. The transmitted OFDM signal is used as a primary user in the proposed method which is discussed in Section 5. The detailed explanation of the proposed method is presented in Section 6. Section 7 describes the performance evaluation of the proposed method by simulation results. Finally, the conclusion of this paper is drawn in Section 8. The notation employed in this paper is summarized in Table 1.

2- Related Work

Spectrum sensing detects the primary user's transmitted signals in the CR systems. In the modern era, spectrum sensing for OFDM signals is very important issues in CR systems [24], [25]. The spectrum sensing method for OFDM signals under consideration of noise uncertainty is a severe challenging issue. Several spectrum sensing methods considering noise uncertainty have been proposed and investigated in recent years. Energy detection is a very popular technique that is used for spectrum sensing considering the noise uncertainty [26]. The spectrum sensing performance was very good for low SNR cases. However, OFDM systems were not considered, and only the AWGN channel was considered in this proposed method. In addition, recently, single and double threshold energy detection algorithms were implemented in [27], and a Frequency domain Goodness of Fit Test (FGoF) based spectrum sensing method was proposed to detect primary user signals [28]. The primary user detection performance was not very good in severe noise environments considering noise uncertainty and OFDM detection was not considered in those methods.

An Improved Energy Detection (IED) algorithm was proposed for spectrum sensing with an experimental hardware setup [29]. However, in low SNR environments, spectrum sensing performance degrades significantly due to noise uncertainty. In the presence of noise uncertainty, an adaptive double threshold based spectrum sensing

method was recently proposed [22]. The sensing performance for low SNR cases was not very satisfactory using this method. In that case, the OFDM primary user transmitted signal was not considered. Effective energy detection was proposed in [23] to overcome the problem of OFDM signal detection considering noise uncertainty cases. However, the OFDM transmitted signal performance is not very good for large values of noise uncertainty in severe noise environments. In this paper, the proposed method improves the detection performance of the OFDM transmitted signal under both noise uncertainty and low SNR cases.

3- Problem Formulation of Spectrum Sensing

The principle of spectrum sensing is where the primary user transmitter sends data to the primary user receiver in their allotted licensed radio frequency spectrum band, while a pair of secondary users also intend to access the spectrum at the same instant. Spectrum sensing should be performed to detect the presence of the primary user receiver present within the coverage of the secondary user transmitter to protect the primary user transmission.

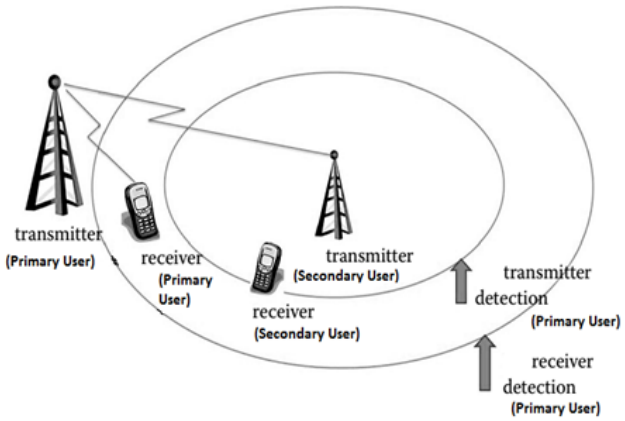


Fig. 1: Spectrum sensing principle [30].

Fig. 1 shows the principle of spectrum sensing [30]. Noise uncertainty is a very important challenge for the spectrum sensing methods. In a practical scenario, determining the noise power is very difficult. It needs to be estimated that it may contain calibration errors due to changes in thermal noise. Therefore, it is necessary to have more sensitive spectrum sensing under noise uncertainty environments in the practical situation.

In this paper, the primary user transmitted signal is an OFDM signal received at the spectrum sensing receiver. The spectrum sensing of OFDM signals is very important for modern broadcasting applications. Furthermore, spectrum sensing for OFDM signals in severe noise environments under consideration of noise uncertainty cases is challenging for modern and forthcoming wireless communication systems.

4- Methodology of Spectrum Sensing

In the proposed spectrum sensing method, there are two hypotheses that are defined by two states such as idle state, H_0, and active state, H_1, of the primary user in the spectrum sensing model. The secondary user's receiver evaluates a test statistic, T_f, based on its observed signal and compares it with a specific threshold, λ, to decide the situation between the two hypotheses.

The two hypotheses are given by

$$H_0 : T_f < \lambda \quad (1)$$

$$H_1 : T_f \geq \lambda \quad (2)$$

The spectrum sensing performance can be characterized using some important parameters. The probability of detection, P_d, indicates the primary user correctly detects its active mode [12] as

$$P_d = P(H_1; H_1) = Pr\{T_f > \lambda \mid H_1\} \quad (3)$$

Table 1: Notation used in this paper.

Parameter	***Definition***
P_d	Probability of detection
P_{fa}	Probability of false alarm
P_m	Probability of miss detection
H_0	Idle state
H_1	Active state
T_f	Test statistics
T_{f1}	Test statistics using skewness
T_{f2}	Test statistics using kurtosis
λ	Threshold
σ_w^2	Noise variance
Ω	Noise Uncertainty
N_c	CP Size
s	Number of OFDM Symbols
$S_l(y)$	Skewness function
$K_l(y)$	Kurtosis function

There are two types of errors: the probability of false alarm, P_{fa}, and the probability of miss detection, P_m.

These can be given [12] as

$$P_{fa} = P(H_1; H_0) \quad (4)$$

$$P_m = P(H_0; H_1) \quad (5)$$

In this paper, it is assumed that the primary user signal is a CP-OFDM signal received at the sensing station. The spectrum sensing of OFDM signals considering noise uncertainty cases in severe noise environments is challenging for wireless communication systems.

5- OFDM Signal

The OFDM is a multi-carrier modulation technique where the input data streams are convolutionally encoded by a 1/2-rated convolutional encoder and an interleaver is used to reduce errors that occur in bursts. Fig. 2 represents that the coded bits are digitally modulated, resulting in a complex symbol stream $X(0), X(1), \ldots\ldots, X(N-1)$. This symbol stream is converted to parallel subchannels by serial-to-parallel converter which are the frequency components. Then the frequency components are converted into time samples utilizing the inverse fast Fourier transform (IFFT).

The IFFT gives the OFDM symbol consisting of the sequence $x(0), x(1), \ldots\ldots, x(N-1)$ of length N as

$$IFFT\{X(k)\} = x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k)\, e^{j2\pi kn/N} \quad (6)$$

We consider the l^{th} OFDM symbol, $x_l(n)$, as

$$x_l(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_l(k)\, e^{j2\pi kn/N} \quad (7)$$

where $1 \le l \le s$. s is the number of OFDM symbols and $X_l(k)$ corresponds to the l_{th} OFDM symbol at the k_{th} subcarrier. The CP is added between consecutive OFDM symbols to combat the intersymbol interference (ISI) problem of OFDM signals. Fig. 3 shows the CP insertion of the OFDM systems. The CP for $x_l(n)$ can be defined by $x_l(N-C), \ldots\ldots, x_l(N-1)$, where it consists of the last C samples of the $x_l(n)$ sequence. For each input sequence of length N, these last C samples are appended to the beginning of the sequence. This gives a new sequence, $\tilde{x}_l(n)$, of length $M = N + C$, which is the each OFDM symbol length. The resulting transmitted CP-OFDM signal $\tilde{x}_l(n)$ is converted by a parallel-to-serial converter.

The transmitted signal is filtered by the channel impulse response, $h_l(n)$, and corrupted by the additive noise, $w_l(n)$. Finally, the received signal is given by

$$h_l(n) = \tilde{x}_l(n) * h_l(n) + w_l(n) = v_l(n) + w_l(n). \quad (8)$$

where the asterisk denotes convolution and $v_l(n) = \tilde{x}_l(n)$ is used for simplicity.

6- Proposed Method

In the proposed method, the skewness calculation or kurtosis calculation is used for sensing OFDM primary user transmitted signals. Fig. 2 shows a block diagram of the proposed spectrum sensing. In this method, a noise uncertainty environment is considered which is very important for spectrum sensing method.

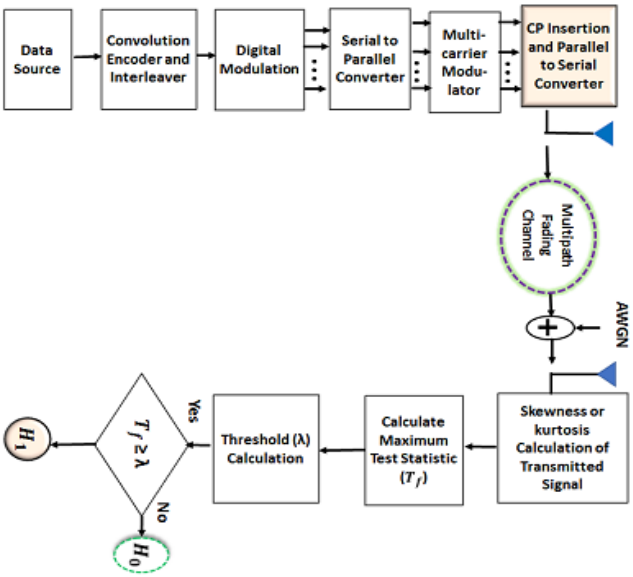


Fig. 2: Block diagram of the proposed spectrum sensing.

The OFDM signal is absent or present which is represented by H_0 and H_1 as

$$H_0 : y_l(n) = w_l(n) \quad (9)$$

$$H_1 : y_l(n) = v_l(n) + w_l(n) \quad (10)$$

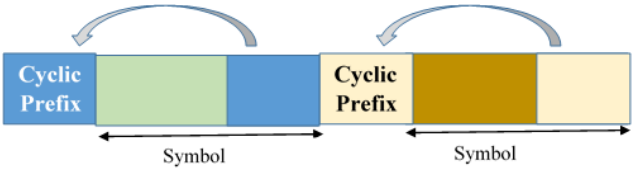


Fig. 3: CP Insertion of OFDM Signals

6-1- Skewness Function for Spectrum Sensing

Skewness, one of the third order statistics, is a statistical measure of the asymmetry of the probability distribution

around the sample mean. Here, the skewness function method can be effectively utilized for OFDM signal sensing in noise uncertainty environments. The skewness function, $S_l(y)$, of a random variable, $y_l(n)$, is calculated [31], [32] as

$$S_l(y) = \frac{E[(y_l(n)-\mu)^3]}{\sigma^3} \quad (11)$$

where E{.} represents the expected value operator. In the proposed method, $y_l(n)$ is the OFDM-transmitted signal, μ is the mean of $y_l(n)$, and σ is the standard deviation of $y_l(n)$.

For a sample of variables $y_{l,1}(n)$, $y_{l,2}(n)$, ..., $y_{1,M}(n)$ the skewness can be estimated from Equation (11)

$$\hat{S}_l(y) = \frac{\frac{1}{M}\sum_{t=1}^{M}(y_{l,t}(n)-\hat{\mu})^3}{\left(\sqrt{\frac{1}{M}\sum_{t=1}^{M}(y_{l,t}(n)-\hat{\mu})^2}\right)^3} \quad (12)$$

$$\hat{\mu} = \frac{1}{M}\sum_{t=1}^{M} y_{l,t}(n) \quad (13)$$

where $\hat{.}$ means an estimate and M is the number of variables. In the proposed method, M corresponds to the number of OFDM samples.

The sample skewness Equation (12) is effectively used for the test statistic calculation for spectrum sensing of OFDM transmitted signals. The test statistic using skewness function, T_{f1}, selection for the detection can be calculated by averaging the absolute value of the sample skewness function of the l_{th} OFDM symbol $\hat{S}_l(y)$

$$T_{f1} = \frac{1}{s}\sum_{l=1}^{s} |\hat{S}_l(y)| \quad (14)$$

The threshold value λ is obtained from the noise variance, σ_w^2, and the P_{fa}. The noise variance σ_w^2 is obtained as a priori information from the channel, where the technique in [33] is available for example. In the proposed spectrum sensing method, calculate the threshold value λ in the same way as that in [34] by

$$\lambda = \sqrt{-\ln P_{fa}.\sigma_w^2}. \quad (15)$$

In the proposed higher order statistics based spectrum sensing method, noise variance σ_w^2 is used for threshold calculation for OFDM transmitted signal detection. Here, we have considered that the accurate noise variance σ_w^2 is calculated. However, it is very difficult to estimate noise variance σ_w^2 perfectly for some practical applications. For this reason, it is very important to consider the influence of noise uncertainty for proposed spectrum sensing based on higher order statistics including skewness and kurtosis functions.

The noise uncertainty, Ω, is measured in dB as

$$\Omega = 10\, log10\, \Psi \quad (16)$$

where Ψ is a parameter which estimates the noise uncertainty [35], [36].

When the influence of noise uncertainty is estimated in the spectrum sensing method using skewness function, the threshold value λ is calculated by Equation (15) using $\Psi\sigma_w^2$ instead of σ_w^2.

Finally, the spectrum sensing receiver gives the final output H_0 and H_1 by

$$H_0: T_{f1} < \lambda \quad (17)$$

$$H_1: T_{f1} \geq \lambda. \quad (18)$$

6-2- Kurtosis Function for Spectrum Sensing

The kurtosis function is considered in the proposed method for OFDM transmitted signal detection. The kurtosis is a statistical measure of the peakness of the probability distribution. In the proposed method, the fourth order function can also be utilized effectively to detect the OFDM signal.

The kurtosis function, $\hat{K}_l(y)$, for a sample of variables $y_{l,1}(n)$, $y_{l,2}(n)$,…….,$y_{l,M}(n)$ is estimated [31], [32] as

$$\hat{K}_l(y) = \frac{1/M\sum_{t=1}^{M}(y_{l,t}(n)-\hat{\mu})^4}{\left(1/M\sum_{t=1}^{M}(y_{l,t}(n)-\hat{\mu})^2\right)^2} \quad (19)$$

The test statistic using kurtosis function, T_{f2}, can be obtained from the Equation (18) as

$$T_{f2} = \frac{1}{s}\sum_{l=1}^{s}\left|\hat{K}_l(y)\right| \quad (20)$$

In the kurtosis based spectrum sensing method, the threshold value λ is calculated by Equation (15).

Furthermore, the threshold value λ is calculated by Equation (15) using $\Psi\sigma_w^2$ instead of σ_w^2 when the influence of noise uncertainty environments is considered in the kurtosis function based spectrum sensing method.

For the proposed method, the spectrum sensing gives the final output by

$$H_0: T_{f2} < \lambda \quad (21)$$

$$H_1: T_{f2} \geq \lambda. \quad (22)$$

The test statistics and threshold values for the skewness and kurtosis based spectrum sensing methods do not depend upon the OFDM primary user transmitted information, hence the proposed scheme is semi blind.

7- Simulation Result

The detection performance of OFDM primary user signals is investigated to evaluate the performance of the proposed spectrum-sensing scheme. The transmitted signals consist of a WLAN radio interface (IEEE 802.11a), which is an OFDM-based system. The simulation parameters for the proposed method are digital 64-QAM, an FFT size N of 64, a CP ratio N_c of 1/4, 1/8, 1/16, or 1/32, 150 OFDM symbols, and an SNR range of -35 dB to 0 dB. Monte Carlo simulations are conducted by averaging results over 5000 iterations. All the simulations are carried out on a multipath fading channel. The multipath fading channel consists of a five-tap impulse response with a maximum delay of 8. The simulation parameters used throughout the paper are tabulated in Table 2 (where the CP ratio is defined as $N_c = C/N$, unless otherwise stated).
The noise uncertainty is a very important parameter for spectrum sensing performance evaluation. The effect of noise uncertainty on the detection of OFDM signals is analyzed for spectrum sensing performance evaluation.

Table 2: Simulation Parameters

Parameters	***Types***
OFDM System	WLAN
Digital Modulation	64-QAM
FFT Size	64
Number of OFDM Symbols	150
Noise Uncertainty	0.5 dB and 1 dB
CP Size N_c	1/4, 1/8, 1/16 and 1/32
Channel	Multipath Rayleigh Fading with AWGN Channel
SNR Range	-35 dB to 0 dB
Higher order statistics	Skewness and Kurtosis
Number of Iterations	1500

In the proposed method, the detection performance of OFDM signals for N_c 1/4 at fixed P_{fa} 0.01 under 64-QAM is presented in Fig. 4. Here, for OFDM signal with noise uncertainty and without noise uncertainty cases are considered for sensing the primary user. In terms of noise uncertainty, which is 1 dB, the skewness function achieves the maximum P_d ($\geq$ 0.9) at -31 dB SNR and the kurtosis function achieves the maximum P_d ($\geq$ 0.9) at -13 dB SNR. For the skewness and kurtosis functions, the P_d for noise uncertainty 0.5 dB is slightly better than that of the noise uncertainty 1dB.

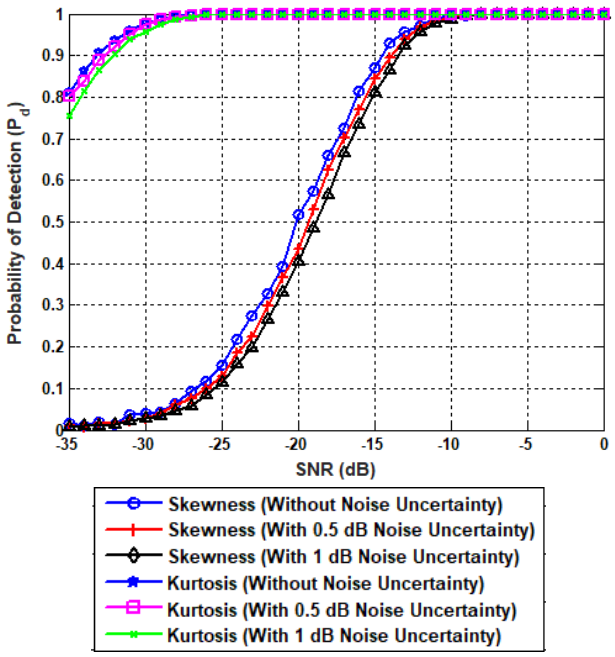


Fig. 4: Performance of the proposed spectrum-sensing method for OFDM signals under 64-QAM for N_c of 1/4 with noise uncertainty.

For without noise uncertainty case, the maximum P_d is obtained at -32 dB SNR for the skewness function and at -34 dB SNR for the kurtosis function. As a result, when the kurtosis function is used, the performance of the proposed spectrum sensing increases markedly compared with that of the skewness function.

For skewness and kurtosis functions of the proposed method, the P_d of CP-OFDM signals for N_c 1/8 in cases of 0.5 dB and 1 dB noise uncertainties are shown in Fig. 5. In the proposed spectrum sensing, FFT size (N) of 64, 64-QAM, 150 OFDM symbols, and fixed P_{fa} 0.01 are considered to measure the effect of noise uncertainty. It is observed from the plot that when the noise uncertainty is increased, the OFDM transmitted signal detection probability P_d is decreased slightly with SNR for both functions. On the other hand, the P_d of our proposed spectrum sensing method show the satisfactory results with 0.5 dB and 1 dB noise uncertainties for skewness and kurtosis functions.

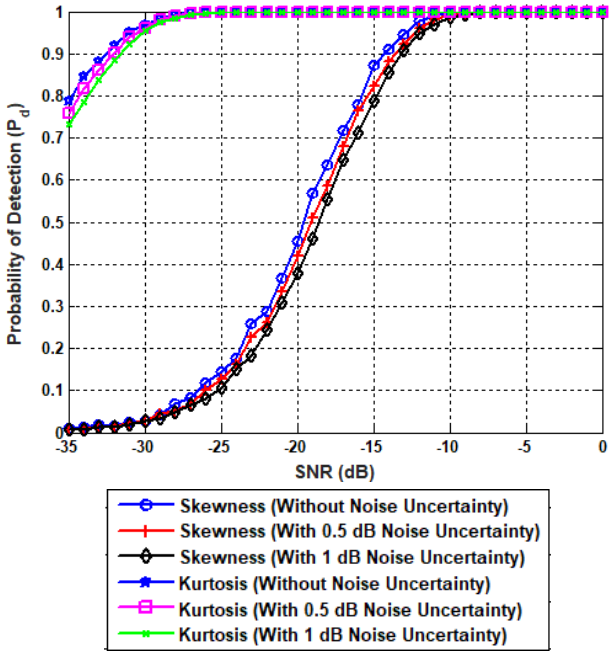


Fig. 5: Performance of the proposed spectrum-sensing method for OFDM signals under 64-QAM for N_c of 1/8 with noise uncertainty.

Fig. 6 shows the detection performance of OFDM primary user signals for N_c 1/16 under 0.5 dB and 1 dB noise uncertainty. It is clear from Fig. 6 that when the noise uncertainty is increased, the OFDM signal detection probability P_d is decreased slightly with SNR. However, the P_d of our proposed spectrum sensing method show the satisfactory results with 0.5 dB and 1 dB noise uncertainties, which is very important for spectrum sensing in CR systems.

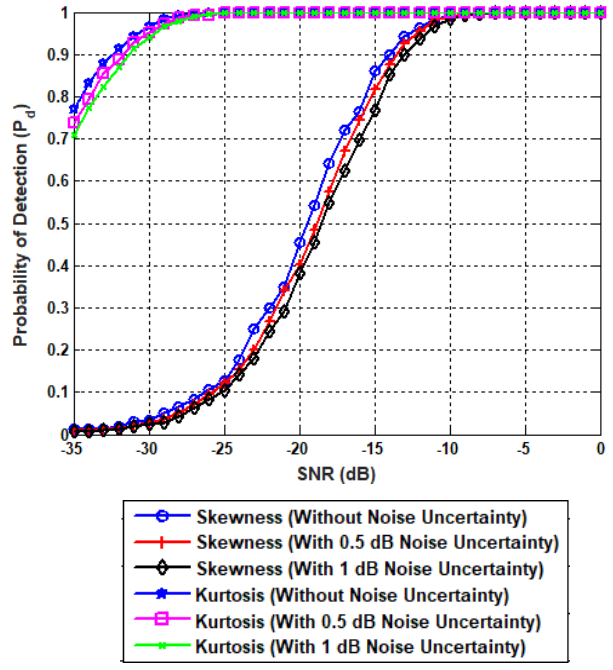


Fig. 6: Performance of the proposed spectrum-sensing method for OFDM signals under 64-QAM for N_c of 1/16 with noise uncertainty.

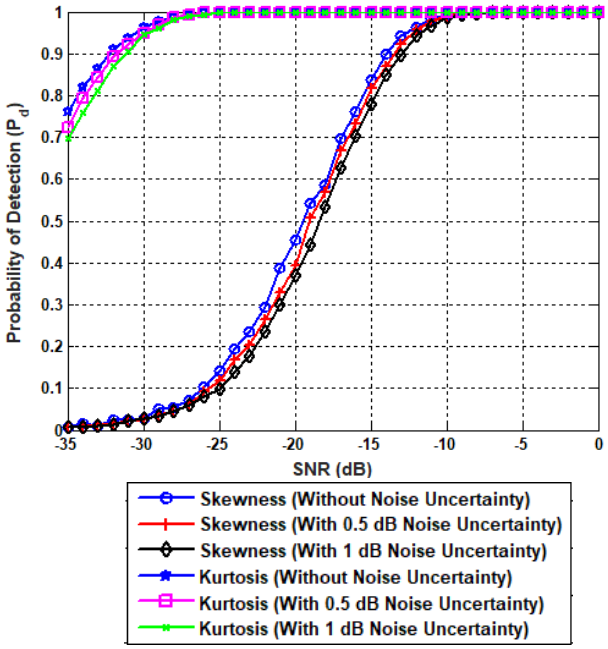


Fig. 7: Performance of the proposed spectrum-sensing method for OFDM signals under 64-QAM for N_c of 1/32 with noise uncertainty.

In Fig. 7, the noise uncertainty of 0.5 dB and 1 dB is considered in cases of the skewness and kurtosis functions to detect the OFDM signal for CP ratio N_c 1/32. It is clear from Fig. 7 that when the noise uncertainty increases, the P_d of the proposed spectrum sensing slightly decreases with SNR for skewness and kurtosis functions. However, the satisfactory P_d of OFDM transmitted signal is sustained even at 1 dB noise uncertainty in highly noisy environments.

It can be concluded from Fig. 4 to Fig. 7 that the proposed method shows the excellent P_d of OFDM primary user signal for skewness and kurtosis functions under both moderate and high noise uncertainty in highly noisy environments, which is essential for spectrum sensing in CR systems.

7-1- Performance Comparison

The proposed method is compared with an adaptive double threshold based spectrum sensing method [22] and an effective energy detection spectrum sensing method [23] in Fig. 8.

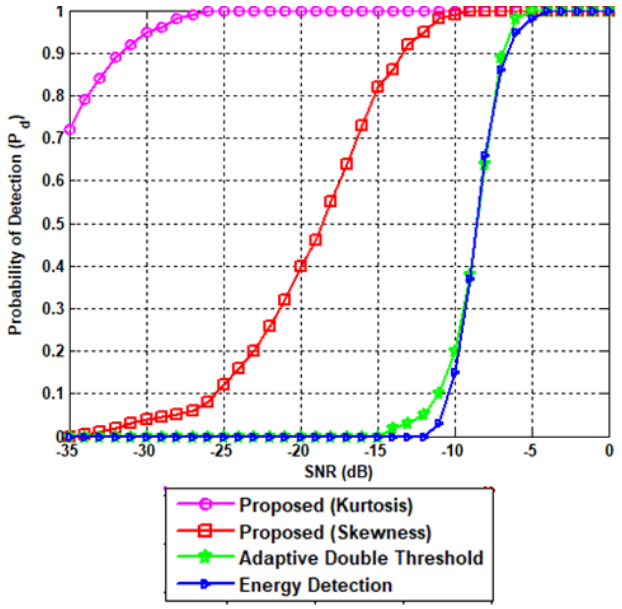


Fig. 8: Performance comparison under noise uncertainty

Here, a fixed probability of false alarm P_{fa} 0.01 and noise uncertainty 0.5 dB are considered for detection performance simulation. The results show that the sensing performance of our proposed method increases dramatically more than the conventional methods.

Table 3: Probability of detection ($\boldsymbol{P_d}$) at fixed SNR

Spectrum Sensing Method	***SNR (dB)***	$\boldsymbol{P_d}$
Proposed (Skewness)	-10	1
Proposed (Kurtosis)	-26	1
Adaptive Double Threshold [22]	-5	1
Energy Detection [23]	-4	1

Table 3 shows the detection performance of the proposed and conventional spectrum sensing methods under the noise uncertainty. It is clear from Table 3 that the proposed spectrum sensing scheme using skewness function offers a 5 dB SNR improvement compared with that of adaptive double threshold based spectrum sensing and a 21 dB SNR gain compared with that of energy detection. Furthermore, in the proposed method, when the kurtosis function is used for spectrum sensing, it provides a 6 dB SNR improvement relative to that of the adaptive double threshold based spectrum sensing and a 22 dB SNR gain relative to that of the energy detection of the conventional autocorrelation based spectrum sensing method. As a result, the proposed method significantly outperforms the conventional spectrum sensing methods.

8- Conclusions

The proposed method is an effective spectrum sensing scheme considering the poor sensing performance in severe noise and noise uncertainty environments. The higher order statistics, such as skewness function and kurtosis function, are exploited for sensing OFDM primary user signals. The proposed method is applicable under higher order digital modulation for various CP radios with noise uncertainty cases, which is very important for various OFDM based systems. The skewness calculation significantly improves the detection performance of the OFDM transmitted signal compared to the conventional methods, considering the noise uncertainty effect in severe noise environments. In the proposed method, the kurtosis calculation has a tendency to provide a further SNR gain relative to the corresponding skewness calculation considering the effect of noise uncertainty.

Acknowledgments

The authors would like to thanks Saitama University for supporting this research academically.

References

[1] FCC, "Spectrum Policy Task Force Report", Federal Communications Commission, ET Docket No. 02–135, 2002.
[2] J. Mitola, "Cognitive radio: an integrated agent architecture for software defined radio", Ph.D. Dissertation, Royal Institute of Technology, Stockholm, Sweden, 2000.
[3] S. Haykin, "Cognitive radio: brain-empowered wireless communications", IEEE Journal on Selected Areas in Communications, Vol. 23, 2005, pp. 201–220.
[4] G. Ding, Y. Jiao, J. Wang, Y. Zou, Q. Wu, Y. Yao, and L. Hanzo, "Spectrum inference in cognitive radio networks: algorithms and applications", IEEE Communications Surveys Tutorials, Vol. 20, 2018, pp. 150–182.
[5] M. Amjad, M. H. Rehmani, and S. Mao, "Wireless multimedia cognitive radio networks: a comprehensive survey", IEEE Communications Surveys Tutorials, Vol. 20, 2018, pp. 1056–1103.
[6] F. Hu, B. Chen, and K. Zhu, "Full spectrum sharing in cognitive radio networks toward 5g: a survey", IEEE Access, Vol. 6, 2018, pp. 15754–15776.
[7] P. C. Sofotasios, E. Rebeiz, L. Zhang, T. A. Tsiftsis, D. Cabric, and S. Freear, "Energy detection based spectrum sensing over kappa-mu and kappa-mu extreme fading channels", IEEE Transection on Vehicle Technology, Vol. 62, 2013, pp. 1031–1040.
[8] F. F. Digham, M. S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels", IEEE Transactions on Communications, Vol. 55, 2007, pp. 21–24.
[9] R. Umar, A. U. H. Sheikh, and M. Deriche, "Unveiling the hidden assumptions of energy detector based spectrum

sensing for cognitive radios", IEEE Transections on Vehicle Technology, Vol. 59, July 2010, pp. 2940–2950.

[10] A. Tani, and R. Fantacci, "A low-complexity cyclostationary-based spectrum sensing for UWB and WiMAX coexistence with noise uncertainty", IEEE Transactions on Signal Processing, Vol. 61, 2013, pp. 3931–3943.

[11] S. Kapoor, S. Rao, and G. Singh, "Opportunistic spectrum sensing by employing matched filter in cognitive radio network", in Proceedings International Conference on Communication Systems and Network Technologies, 2011, pp. 580–583.

[12] S. M. Kay, Fundamentals of Statistical Signal Processing: Detection Theory, Prentice Hall, 1993.

[13] W. Han, C. Huang, J. Li, Z. Li, and S. Cui, "Correlation-based spectrum sensing with oversampling in cognitive radio", IEEE Journal on Selected Areas in Communications, Vol. 33, 2015, pp. 788–802.

[14] S. Chaudhari, V. Koivunen, and H. V. Poor, "Collaborative autocorrelation-based spectrum sensing of OFDM signals in cognitive radios", in Proceedings Annual Conf. on Information Sciences and Systems (CISS), 2008, pp. 191–196.

[15] S. Chaudhari, V. Koivunen, and H. V. Poor, "Distributed autocorrelation based sequential detection of OFDM signals in cognitive radios", in Proceedings IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Commun. (CROWNCOM), 2008, pp. 1–6.

[16] S. Chaudhari, V. Koivunen, and H. V. Poor, "Autocorrelation-based decentralized sequential detection of OFDM signals in cognitive radios", IEEE Transactions on Signal Processing, Vol. 57, 2009, pp. 2690–2700.

[17] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications", in Proceedings of the IEEE, 1991, Vol. 79, pp. 278–305.

[18] C. L. Nikias, and J. M. Mendel, "Signal processing with higher order spectra", IEEE Signal Processing Magazine, Vol. 10, 1993, pp. 10–37.

[19] P. A. Delaney, and D. O. Walsh, "Performance analysis of the incoherent and skewness matched filter detectors in multipath environments", IEEE Journal Oceanic Engineering, Vol. 20, 1995, pp. 80–84.

[20] M. Sanaullah, "A review of higher order statistics and spectra in communication systems", Global Journal of Science Frontier Research Physics and Space Science, Vol. 13, 2013, pp. 31–50.

[21] M. Haque, Y. Sugiura, and T. Shimamura, "Spectrum Sensing Based on Higher Order Statistics for OFDM Systems over Multipath Fading Channels in Cognitive Radio", Journal of Signal Processing, Vol. 23, 2019, pp. 257–266.

[22] G. Mahendru, A. K. Shukla, P. Banerjee, and L. M. Patnaik, "Adaptive double threshold based spectrum sensing to overcome sensing failure in presence of noise uncertainty", in Proceedings of IEEE International Conference on Signal Processing and Integrated Networks (SPIN), 2019, pp. 466–471.

[23] J. Yao, M. Jin, Q. Guo, Y. Li, and J. Xi, "Effective energy detection for IoT systems against noise uncertainty at low SNR", IEEE Internet of Things Journal, Vol. 6, 2019, pp. 2327–4662.

[24] H. Sadeghi, and P. Azmi, "Cyclic Correlation-Based Cooperative Detection for OFDM-Based Primary Users", Journal of Information Systems and Telecommunication, Vol. 1, 2013, pp. 155-164.

[25] S. S. Kashef, P. Azmi, and H. Sadeghi, "GoF-Based Spectrum Sensing of OFDM Signals over Fading Channels", Journal of Information Systems and Telecommunication, Vol. 2, 2014, pp. 103-112.

[26] E. E. A. Medina, and S. E. Barbin, "Performance of Spectrum Sensing Based on Energy Detection for Cognitive Radios", in Proceedings of IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC), 2018, pp. 948–951.

[27] M. Alijani, and A. Osman, "Performance improvement of energy detection in cognitive radio under noise uncertainty", in Proceedings of IEEE International Conference on Innovations in Information Technology (IIT), 2020, pp. 148–153.

[28] R. Gao, P. Qi, and Z. Zhang, "Frequency domain goodness of fit test based spectrum sensing method with dynamically varying noise", China Communication, Vol. 17, 2020, pp. 172–179.

[29] B. Gajera, D. K. Patel, B. Soni, and M. Lopez-Benitez, "Performance evaluation of improved energy detection under signal and noise uncertainties in cognitive radio networks", in Proceedings of IEEE International Conference on Signals and Systems (ICSigSys), 2019, pp. 131–137.

[30] G. Tomar, A. Bagwari, and J. Kanti, "Introduction to Cognitive Radio Networks and Applications", Taylor & Francis Group, 2017.

[31] S. Brown, "Measures of shape: skewness and kurtosis", Available: http://brownmath.com/stat/shape.htm.

[32] D. N. Joanes, and C. A. Gill, "Comparing measures of sample skewness and kurtosis", Journal of the Royal Statistical Society, Vol. 47, 1998, pp. 183–189.

[33] F. X. Socheleau, A. A. E. Bey, and S. Houcke, "Non data-aided SNR estimation of OFDM signals", IEEE Communications Letters, Vol. 12, 2008, pp. 813–815.

[34] E. Hong, K. Kim, and D. Har, "Spectrum sensing by parallel pairs of cross-correlations and comb filters for OFDM systems with pilot tones", IEEE Sensors Journal, Vol. 12, 2012, pp. 2380–2383.

[35] S. Dikmese, P. C. Sofotasios, M. Renfors, M. Valkama, and M. Ghogho, "Analysis of noise uncertainty and frequency selectivity effects in wideband multimode spectrum sensing", in Proceedings of IEEE Global Communications Conference (GLOBECOM), 2015, pp. 36–40.

[36] R. Tandra, and A. Sahai, "SNR walls for signal detection", IEEE Journal of Selected Topics in Signal Processing, Vol. 2, 2008, pp. 4–17.

Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 260-268
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Trip Timing Algorithm for GTFS Data with Redis Model to Improve the Performance

Mustafa Alzaidi[1*], Aniko Vagner[2]

[1].Department of Information Technology Faculty of Informatics University of Debrecen, Hungary
[2].Department of Information Technology Faculty of Informatics University of Debrecen, Hungary

Received: 05 May 2022/ Revised: 04 Dec 2022/ Accepted: 09 Jan 2023

Abstract

Accessing public transport plays an essential role in the daily life productivity of people in urban regions. Therefore, it is necessary to represent the spatiotemporal diversity of transit services to evaluate public transit accessibility appropriately. That can be accomplished by determining the shortest path or shortest travel time trip plan. Many applications like ArcGIS provide tools to estimate the trip time using GTFS data. They can perform well in finding travel time. Still, they can be computationally inefficient and impractical with increasing the data dimensions like searching all day time or in case of huge data. Some research proposed recently provides more computationally efficient algorithms to solve the problem. This paper presents a new algorithm to find the timing information for a trip plan between two start and destination points. Also, we introduce RMH (Range Mapping Hash) as a new approach using Redis NoSQL to find and calculate the accessibility of a trip plan with fixed time complexity of O(2) regardless of the city size (GTFS size). We experimented with the performance of this approach and compared it with the traditional run-time algorithm using GTFS data of Debrecen and Budapest. This Redis model can be applied to similar problems where input can be divided into ranges with the same output.

Keywords: Author Guide; Article; Camera-Ready Format; Paper Specifications; Paper Submission.

1- Introduction

For two key reasons, public transportation accessibility has become a hot topic for scholars and transit organizations. First, improved transit accessibility promotes active transportation (such as walking and bicycling) while decreasing private vehicle use. As a result, it will enhance public health and reduce GHG emissions [1]–[4]. Second, transit-dependent people primarily rely on public transportation to reach key services (e.g., workplace, university, and shopping center). Thus, transit accessibility is crucial to attaining socioeconomic fairness[5], [6]. Also, transit accessibility research may help guide decisions about transportation investment and land use development[7].

GTFS stands for General Transit Feed Specification. It was developed by TriMet and Google in Portland [8]. Google announced transit feed specs in 2007, enabling transit agencies to develop and publish transit data online as open sources using the GTFS format. The feed rapidly became the most extensively used standard for exchanging static transit data in Canada and the United States [9] [10]. Additionally, the transit sector has embraced the GTFS format as a standard for communicating schedule data due to its expanding popularity. Subsequently, software such as OpenTripPlanner, GoogleMaps, and Bing Maps was developed and updated to use GTFS and provide services like, stop locations, timetables, and route planning. The trip (or route) planner is essential to these applications since it examines GTFS data for possible routes between two places, which is the most demanded service.

Handling a user trip planning request requires two actions or steps. First, identifying all feasible pathways or routes between two places as candidate solutions; second, filtering and validating these solutions according to the user's schedule and start time. Route planning is more complex than identifying a path or route in a graph since it considers journeys, directions, and intermediate transfers between bus stops or stations. The shortest path of a graph is a frequent issue; various algorithms have been developed to solve it [11]–[14]. Depending on the factors used to calculate the weight of graph edges, the algorithm may be bi- or multi-criteria. For instance, if the graph edges depict roads, the road weight may be a bi-criterion, taking distance and cost into account, or a multi-criterion,

✉ **Mustafa Alzaidi**
mustafa.alzaidi@inf.unideb.hu

taking other parameters into account. Numerous techniques attempt to solve the shortest path issue by decreasing the considered factors to a single value; these algorithms fall into types like two-phases algorithms [15], kth shortest-path [16], label correction, and setting algorithms[17]–[23], and others [23]–[28]. A subset of these techniques is used in certain studies [29][30] to locate pathways in local transportation networks. We have already introduced a trip planning algorithm variation[31] this algorithm ignores the weight criteria while checking for all possible next transitions from the current and uses a limit for the number of made transitions to find the best route (trip plan). This trip planning algorithm can find the possible trip plans without considering the timing factor. However, some plans may be rejected due to time conflict between the plan trips and the GTFS trips timetable or trip unavailability at the time the user determines to start the journey. Therefore, we must calculate and find the trips time and transit accessibility as a next step.

To calculate transit accessibility in spatiotemporal dimensions, trip time for station pairs must be calculated at any particular time of day, which is practically impossible with a standard computer as it is time-consuming and needs high computation power[32]Although previous studies [33] introduce algorithms that try to calculate the trip time and transit accessibility while reducing the time complexity and computational power, there is still a need to find an approach to simplify the complexity of such problems solution and reduce the required time and resources, what is the aim of this paper. Algorithms enhancement is a common research topic[34]–[36]. The contribution of this paper has two parts. First, we introduce a new time validation algorithm that can find the timing information for a trip plan or reject the plan if there is a time conflict according to the trips timetable in the GTFS. Second, we go beyond the algorithm enhancement and propose RMH (Range Mapping Hash), which is a new method that can find and extract the timing information for any trip using GTFS data with O(2) time complexity. Our new approach (RMH) eliminates the need for an algorithm to search the GTFS timing records. We use Redis NoSQL Hash to create RMH. Thus we provide a solution by turning the problem of simplifying the existing algorithms into a simple database interaction that can run even on a stander computer. The idea is that for a route going through a station, at any minute between the last going bus and the next bus, the answer for the question "when is the next bus time" will be the time of the next bus. The RMH is applicable not only for the GTFS timing data but also for improving the performance of similar problems, as we describe later. We experiment with the performance of RMH and compare it to run-time search algorithm performance using arbitrary search input for 30 pairs of origin and destination stops using the GTFS data of Debrecen and Budapest. We implement the algorithm and RMH as an open-source project using C# and Redis available on https://github.com/mustafamajid/GTFS-csharp. The project also includes our published trip planning algorithm[31].

Next, we review our route planning algorithm and its output data structure[31]. Then we introduce the time validation algorithm, which will use this output to provide the trip's timing information. Later, we introduce the RMH approach and the Redis implementation. Finally, we list our experiment's results and performance evaluation.

2- GTFS and Trip Planning

The GTFS data is a set of tables, usually in CSV file format. There are three main objects in the GTFS: route, trips, and stop [37]. The routes represent the pathway used by the vehicle, a bus, tram, train, etc., and are usually denoted by the vehicle name. The route visits a set of stops in a specific sequence where the stop can be a bus or tram stop or a train or subway station.

Planning a trip between two locations (the start and destination) requires finding all possible single or combinations of trips that can take the passenger from start to destination points. Finding trip plans can be divided into two steps. First, find all possible routes that can connect the start to the destination point, and these will be the candidate solutions list. Then find the trip's timing information and check for any time conflict in the candidate's plans. To understand the problem, we use Figure 1 as an example of the GTFS data. The figure shows three routes, A, B, and C going through a set of stops denoted by circles with numbers. We consider that the user wants to start the trip from stop 7 at 11:10:00 going to stop 6. Therefore, the candidate solutions will be as follows: first, the user takes route C to stop 9 and then takes route A from stop 9 to stop 6. The second solution is that the user walks from stop 7 to stop 3 and then takes route B to stop 6 if the distance is walkable [38]. The next step is to validate the solution according to the timetable. For the first solution, as in the figure, if the user starts at 11:07:00, 11:17:00, or 11:27:00, he will arrive to stop nine at 11:35:00, 11:35:00, or 11:45:00, respectively. Thus, the user will take the trip at 11:17:00 because it is the earlier trip and then arrive at stop 9 at 11:35:00, where the next trip using route A will be at 11:45:00 and reach the destination at 12:05:00. In the same way, we can find the time information for the second solution. As we mentioned earlier, a solution can be rejected if there is a time conflict; for example, the first solution may be rejected if there is no outgoing trip using route A from stop 9 any time after 11:35:00.

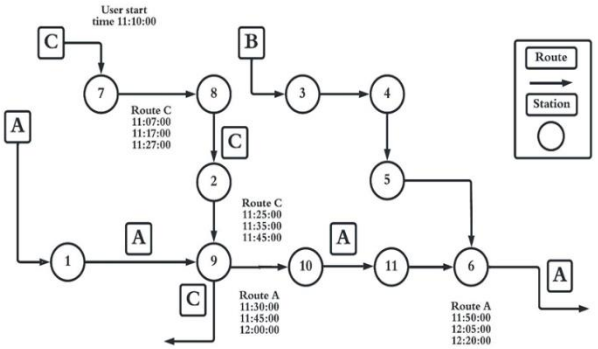


Fig 1. GTFS routes and stops example.

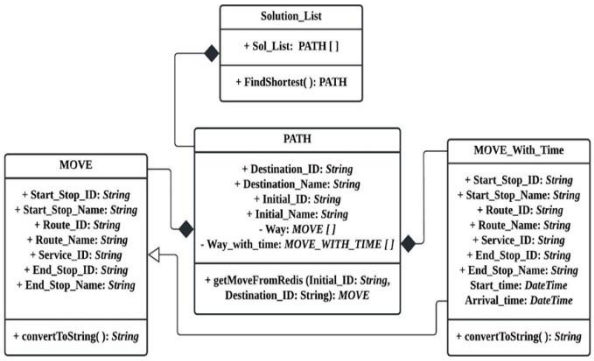


Fig 2. Algorithm data structure.

3- Find Trip Time Information

3-1- Data Structure

Finding trip plans according to the time is a complex problem and needs computation power that is not provided by standard computers [33]. For this work, we use the output structure (the candidate solutions) provided by our previously proposed trip routes planning algorithm [31] as an input to introduce our new trip timing algorithm. Figure 2 shows the UML design of the trip routes planning algorithm output with additional fields to store the time data. The structure contains three main objects Solution_LIST, PATH, and MOVE. Each MOVE represents a single transition from a start-stop to an end-stop using a route (e.g., a bus), and PATH denotes a trip plan or solution containing at least one or more transitions (MOVE) stored in a list called Way. The algorithm's final output is a list of PATH called the Solution_List. The MOVE_WITH_TIME class was inherited from the MOVE class and contained the arrival and departure time fields. Finally, a list of MOVE_WITH_TIME is added to the PATH class called Way_With_Time. The task of the next trip timing algorithm is to validate the PATH by checking every MOVE object in its Way list. If time conflict is found in any MOVE, the whole PATH will be rejected; Otherwise, a new MOVE_WITH_TIME object will be created from the current MOVE by adding the timing fields. The newly created list of MOVE_WITH_TIME objects will form the Way_With_Time list.

3-2- Algorithm

The stoptimes.txt file list a set of records for each trip; each record contains stop ID, trip ID, trip arrival, and departure time at that stops. The set of trips records is present in the file ordered by trip ID and the arrival time. Thus, if the file starts to list a trip that visits ten stops at row number N, then the row N contains timing data about the first stop, row N + 9 shows the data about the last stop that the trip visits, and row N + 10 will list data for the first stop of another new trip if any. Every PATH must be checked by examining the MOVEs in its WAY list using T's time. The stoptimes.txt file record is checked sequentially to find the trip with the closest time to T. Initially, T is set to the time determined by the user (USER_TIME) to start the trip, and during the next MOVEs check, T is set to the arrival time at the last checked MOVE end stop. The check starts from the first record in the stoptimes.txt until finding the first record (i) with stop_id equal to the MOVE start_stop_id and with the same route used by the MOVE and the departure time is greater than T and one of the next record (i + j) in the same trip with stop_id equal to the MOVE end_stop_id. Where j is the number of intermediate stops, if such records are found, a MOVE_WITH_TIME object is created using the examined MOVE and record (i) departure time as Start_time and record (i + j) arrival_time as arrive time for the new MOVE_WITH_TIME object and as the new T value for the next MOVE check. If no such record is found in the stoptimes.txt file, then the whole PATH is rejected and mentioned as an unacceptable solution. The new resulting MOVE_WITH_TIME objects are used to form a WAY_WITH_TIME list. Figure 3 shows the Trim timing algorithm that validates the MOVE according to the trip's timing information. The algorithm input is the start-stop from which the MOVE starts, the end-stop where the MOVE ends, the route used to make that MOVE, and the user's time. The algorithm output must be the trip on that route with the nearest time to T.

Input: WAY a List of MOVE, USER_TIME.

OutPut: WAY_WITH_TIME as List of MOVE_WITH_TIME objects

Step1: T=USER_TIME , WAY_WITH_TIME = *empty*

Step2: ForEach MOVE M in WAY

Do **Step4** To **Step5**

Step3: Set M_WITH_TIME =*NULL,*

Step4: For (i=0 ; i< Stoptimes.Length-2 ; i++**)**

IF(Stoptimes[i].stop_id == M.start_stop_id **&&** Stoptimes[i].Route == M.route_id **&&** Stoptimes[i].Departure_time >T **) Then:**

For (j=i+1 ; i< Stoptimes.Length-1 ; j++**)**

IF (Stoptimes[j].stop_id == M. end_stop_id)

M_WITH_TIME = MOVE_WITH_TIME (M,Stoptimes[i].Derparture_time Stoptimes[j].Arraival_time).

ADD M_WITH_TIME **to** WAY_WITH_TIME,

T= Stoptimes[j].Arrival_time

Goto **Step2** check the next MOVE

Step5: IF M_WITH_TIME ==NULL **Then:**

Return FALS and Exit **Else**

Step6: Return WAY_WITH_TIME

Fig 3. Trip timing algorithm.

3-3- Time Complexity

Searching the stoptimes.txt file record is a time-consuming process as any linear search, the algorithm performs a linear search for timing information. Let N is the number of records present in the stoptimes file. Then, the best case is if the stop with time greater than T is at the stoptimes file's first record. The worst case is O (N-1), and the average case is O((N-1)/2). Thus, the algorithm time is increased by increasing the number of records. We ignore the number of records between the start and the end stops, as this number is minimal compared to N.

4- Redis

Redis is a high-performance in-memory NoSQL database written with C and worked on most POSIX platforms [39]. Redis is a message broker and session manager that stores data in key-value pairs. An HTML page including its resources may be serialized to a string and saved in a Redis to enable a high-speed page load. Thus, software organizations prefer Redis for its fast performance and scalability. Strings, Lists, Sets, Hashes, and Sorted Sets are the five data structures available in Redis. In this research, we will depend only on the Hash structure. All our object data will be converted to a string and concatenated before being stored in the Redis Hashes. A wide range of programming languages supports Redis. Each language has its libraries and packages for communicating with and manipulating the Redis server. In this project, we utilized StackExchange.Redis, which can be installed using NuGet Package Manager.

5- Range Mapping Hash (RMH)

We propose the RMH as a Redis model to avoid the time-consuming liner data scanning by mapping the input parameters to the output directly without any liner search or scan using the power of hash structure in Redis. For each route between any two stops, we need to map a route and two stops ID and time T to the ID of the trip with the nearest time to T on that route, the trip departure time at the start-stop, and trip arrival time at the end stop. The RMH consists of two structures. Both structures querying results are combined to form the timing answer. We use a Redis Hash structure for the implementation. The Hash structure syntax has three-part, the KEY, which refers to the Hash name, the FIELD that uniquely identifies a row in the hash; and the VALUE. The HGET and HSET commands are used to retrieve and insert data into Redis Hash [39].

5-1- RMH Trip Departure Time Structure

The first structure is used to map the start-stop ID, route ID, and Time (T) into the next trip's ID and time at this stop using that route. For this structure, we create a Redis hash for each stop route pair to store all trip visiting time at the stop. The Hash KEY part will mention the stop ID and the route ID separated by the "__" string. The FIELD will contain the time to be examined and denote it as T. The hash VALUE part holds the time of the next coming trip according to T, and the trip ID, separated by the "__" string. Figure 4 shows the trip time structure.

All this information is available in the stoptimes file except the time (T). Any possible T value belongs to the set of sharp minutes in the day. Thus a maximum of 1440 entries is needed to cover all the possibilities. For each stop ID, route ID pair from the stoptimes data, and a time T entry, the value field contains the time of the next coming trip and the trip ID separated by "___". For example, in Figure 4, we take route 10 and stop 1100905. Three trips, 208,209, and 210 are listed in the stoptimes file with departure times 11:44:00, 11:54:00, and 12:04:00. At the first hash entry, where the T value is 11:43:00, the answer (the Hash value field) shows the time 11:44:00 and trip ID 208. For any value of T equal to or greater than 11:44:00, the answer will be trip 209 as its time is 11:54:00. When T is greater or equal to 11:54:00, the answer will be trip 210 and its time 12:04:00.

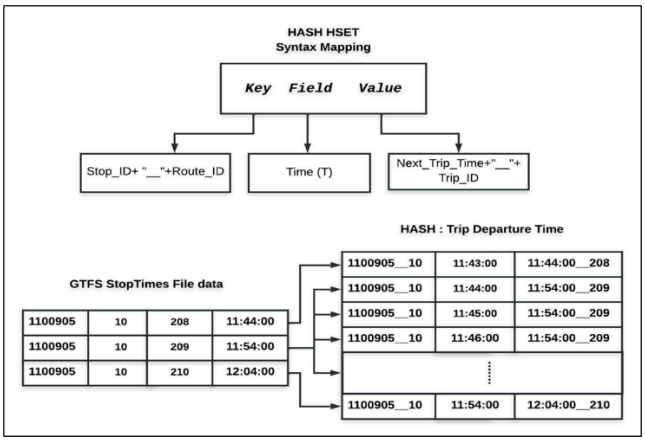


Fig 4. RMH trip departure time structure.

5-2- RHM Arrival Time Structure

After using the trip departure time structure, we have the departure time from the start-stop and the trip ID. Finally, we can find the end stop's arrival time using the end-stop ID and the trip ID using the arrival time structure. Figure 5 shows the arrival time structure, a single Redis Hash for each stop, to list all the trip arrival time combinations. The KEY part of the Redis Hash is used to refer to stop using the stop ID; the FIELD is used to refer to the trip ID, where the VALUE stores the arrival time.

For example, in Figure 5, the trip planning algorithm results in a MOVE with start-stop 1100905, end stop 1002315, and route 10. If the T value is 11:45:00, the timing validation will work as follow: HGET statement using the trip departure time structure is used to retrieve the trip ID and the departure time from the start-stop. The HGET KEY will be "1100905__10" (start-stop ID and the route ID); the FIELD part is "11:45:00" (T). the returned value from this HGET statement will be "11:54:00__209" (the departure time and the trip ID), as shown in figure 8. Now, the trip is known, and the end stop's arrival time is the only missing part. Another HGET statement with KEY is "1002315", and FIELD "209" is used with the arrival time structure; the return value from this statement will be "12:02:00 (the arrival time at the end stop) as shown in Figure 9. If any of the two structures did not return a match for the HGET command, the MOVE is rejected and the whole PATH (path). Thus the total time complexity of RMH formed by two Hash table read operations only. As the time complexity for reading from a Hash structure is O(1)[40], then RMH total complexity is O(2).

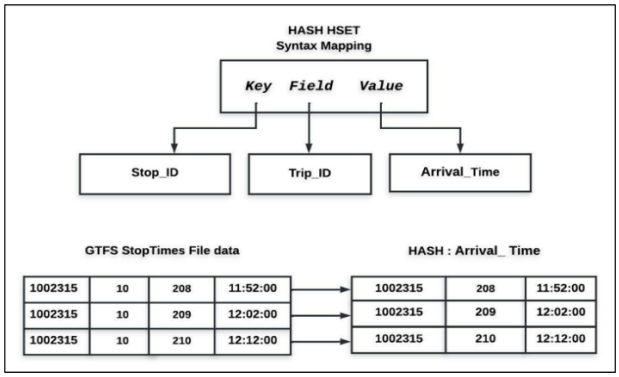


Fig 5. RHM arrival time structure.

6- Experiment and Results

6-1- Experiment Tool

We implement the trip timing algorithm as a C# project. The project is a WinForm application (GUI) containing a set of classes: GTFSData for loading and preprocessing the GTFS data, Algorithm class contains the implementation of our previously published trip planning algorithm, TimeCalculator class contains the implementation of the trip timing algorithm (trip timing), Redis action class include the code for connecting to Redis database load the data to Redis and retrieve the solution and other classes. Figure 6 shows the UML design of the main classes in the project.

We used the GetMovesWithTime( ) function from the TimeCalculator class for this experiment, which takes a MOVE list and start-time as a parameter and returns a MOVE_WITH_TIME list. We calculate the execution time for this function and compare it with the execution time for reviving the timing information using the RMH using the GetSolFromRedis( ) function from the RedisAction class.
The GetMovesWithTime() illustrates the implementation of the trip timing algorithm given earlier, whereas the GetSolFromRedis() represents the interaction with the Redis RMH model, which is implemented as mentioned before. This function will retrive the same result returned by the GetMovesWithTime() function (for the same parameters). The Redis RMH serves in a similar way to a data warehouse model where redundant data is stored and utilized to serve the application purpose.

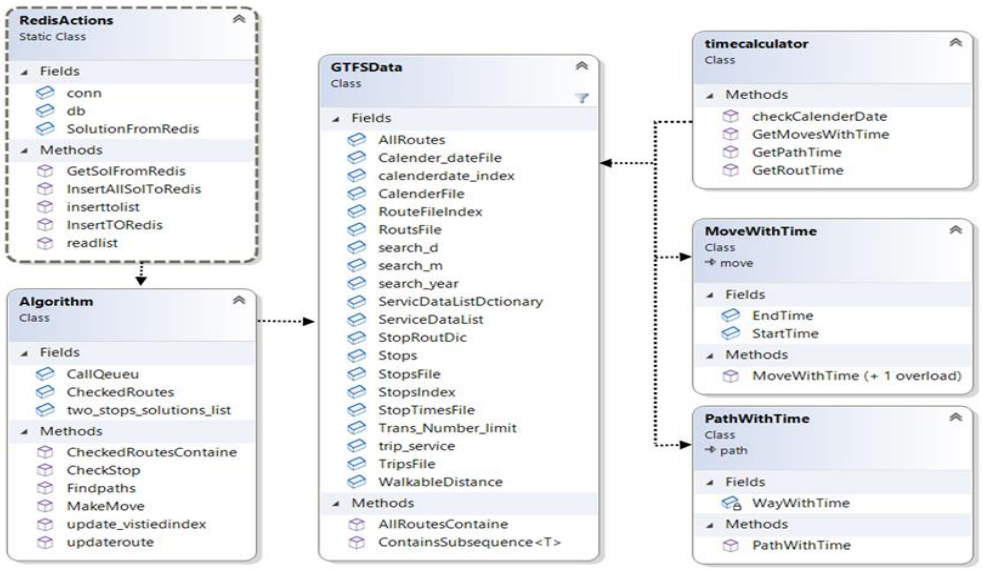


Fig 6.Test tool UML design.

6-2- Result

We experiment using RMH (Redis model) with Budapest and Debrecen cities GTFS data for 30 random start and end stop combinations. Each experiment finds the timing data with and without using Redis (the RMH) and records the time (in milliseconds) that the computer takes to retrieve the result for each combination. Table 1 shows the recorded results.

Table 1. Experiments results.

NO	Budapest Without Redis	Budapest Using Redis	Debrecen Without Redis	Debrecen Using Redis
1	1202.4	8.368	1204.6	8.559
2	1305.4	8.238	1106.7	8.278
3	1105.2	8.38	908.2	8.48
4	1409.3	8.749	907.3	7.237
5	1408.4	8.198	904.1	8.107
6	1206.3	8.558	1008.1	7.848
7	1201.9	7.387	906.7	8.298
8	1106.3	7.938	909.4	6.897
9	1002.6	8.638	907.5	8.318
10	1402.3	8.727	906.3	8.278
11	1404.2	8.648	1009.8	8.67
12	1103.2	7.018	908.7	6.917
13	1107.1	7.449	1105.7	7.53
14	1303.4	7.418	1009.2	7.227
15	1324.1	8.027	1205.4	7.887
16	909.7	8.199	1204.2	8.509
17	1213	8.73	1004.5	7.488
18	908.5	8.319	1003.3	6.809
20	1308.7	7.679	1202.9	8.037
22	1408.4	6.829	1103.2	7.748
23	1203.8	6.839	1208.9	7.05
24	1304.8	8.307	1009.1	7.807
25	1303.5	8.678	907.7	6.909
26	1204.3	8.098	908.5	7.677
27	1305.1	7.099	1101.8	8.3
28	1001.1	7.697	901.8	8.437
29	1003.9	7.158	1001.2	8.047
30	1421.4	7.758	1202.4	6.849
Average	1219.89	7.985	1025.94	7.808

Figure 7 visualize the execution time difference between the time taken for finding the timing information for a trip using the run-time algorithm without Redis and the execution time for retrieving the exact data for the same pair of start and end stops using Redis. We can notice that the run-time performance varies during the experiments, around an average of 1219.89 milliseconds. Conversely, Redis's execution time is more stable. It has ignorable variation during the experiments, with an average of 7.985

milliseconds forming a straight line in the chart close to zero compared to the run-time performance.

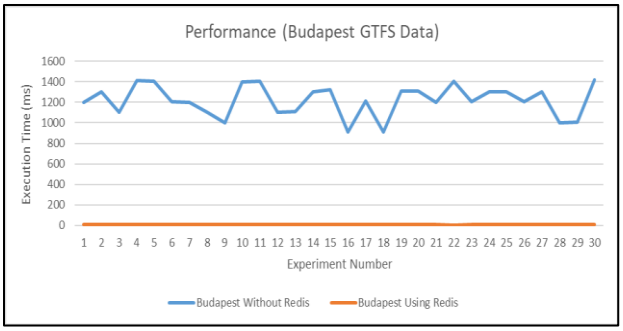


Fig 7.Budapest data experiments result.

With Debrecen data, the experiment shows a similar performance compared to Budapest experiments. The average execution time is 7.808 milliseconds which is very close to the execution time for Redis with Budapest data. However, again, the experiments show notifiable variation in the performance using the run-time algorithm. Figure 8 shows Debrecen data experiments' performance using Redis and the run-time algorithm (without Redis).

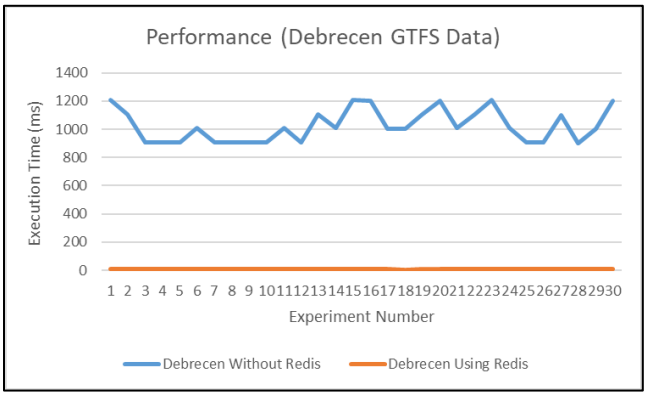


Fig 8.Debrecen experiments result.

The experiments also show that the difference between GTFS data size for the cities (Debrecen 1483 KB and Budapest 42128 KB) affects the performance in the case of run-time algorithm use. This effect can be clear if we compare the average execution time with both cities' data, as shown in Figure 9. We can notice that the average run-time execution time increases with larger cities (in this case, Budapest), while the data size has no effect in the case of using Redis.

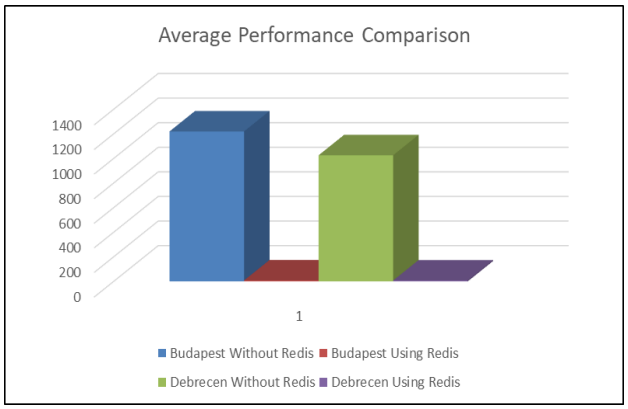


Fig 9.Experiments results average comparison.

7- Conclusions

In contemporary and Smart Cities, sharing transportation data is crucial for a successful transportation system. As a result, the necessity for a uniform format for communicating transportation data has grown. Transportation authority's extensively use GTFS (General Transit Feed Specification) across the globe as a standard format for sharing and publishing data. In addition, trip planning and computing transit accessibility are common topics between researchers and transit organizations as they can affect society's life and productivity. However, computing transit accessibility and finding a trip plan with timing information is complex and requires more computation than a standard computer can provide.

Find a trip plan and transit accessibility consist of two steps. First, find all possible routes that can lead from the start to the destination, mark them as candidate solutions, and then validate them according to the user time to start the trip(start time) and the trip timetable in the GTFS. The first part is done using the trip route planning algorithm, and the second part is accomplished using the trip timing algorithm. Also, they can be combined in one algorithm. This work uses our already published trip planning algorithm output as input to introduce a new trip timing algorithm. The trip planning algorithm output is a set of trip plans; each has one or more transitions. The trip timing algorithm in this paper validates these transitions by searching the trips to find the trip with the closest departure time to the time T (where T initially is specified by the user to start the trip) on the specified route.

Some researchers try to improve the time complexity of trip planning and trip timing algorithms. We introduced the Range Mapping Hash RMH as a Redis model that provides fast access to the timing data and eliminates the need to run the trip timing algorithm as it does the same task with better performance.

The model contains two structures. The first structure can map any route ID, start-stop ID, and the time T; to the next going trip's ID, and the departure time at the start-stop. The idea behind this method is that T can be any time during the day with sharp minutes part. Thus, we have 1440 possible values for T during the day. We use a Redis hash for the implementation. For each stop route combination, we create a hash with 1440 entries such that the key part will mention the stop ID and the route ID separated by the string "__", the hash fields will hold the T possible values, and the value field will hold the next trip_ID and its departure time. If T falls between two trips' departure times, then the answer (the value field) should be the trip with a later departure time. If T is earlier than the departure time of the first trip, then the answer will be the first trip and its departure time. If T is later than the departure time of the last trip in the GTFS data, then no entry will be stored in the hash, and a null value will be returned for such search, leading to rejecting the transition and the plan, and in this case, we will have less than 1440 entries in the hash list. The second structure is a Redis hash with the key part holding the destination stop ID, the field part containing the trip ID goes through that stop, and the value part containing the arrival time. Thus, both structures can form answers for any trip timing request. Both structures are Redis hash, and each can provide the response within O(1) complexity. Thus, the RMH can solve the timing problem with (2) complexity.

Using GTFS data from Budapest and Debrecen, we tested the performance of RMH and the normal trip planning algorithm using the same computation hardware and software specifications. Experiments show that RHM can provide better complexity than the time validation algorithm. The experiments also show that RHM provides consistent time independent of data size (city size) in comparison to the run-time algorithm, where the performance is decreased when the data size is increased. In future work, the RMH can be applied to any similar problem where the input can be divided into sets or ranges with identical output. The RMH sacrifices the space to provide better performance.

Acknowledgment

The work is supported by the EFOP-3.6.1-16-2016-00022 project. The project is co-nanced by the European Union and the European Social Fund.

References

[1] T. Litman, "Integrating Public Health Objectives in Transportation Decision-Making," *American Journal of Health Promotion*, vol. 18, no. 1, pp. 103–108, 2003, doi: 10.4278/0890-1171-18.1.103.

[2] T. Litman, "Exploring the Paradigm Shifts Needed To Reconcile Transportation and Sustainability Objectives," *Transp Res Rec*, vol. 1670, no. 1, pp. 8–12, Jan. 1999, doi: 10.3141/1670-02.

[3] J. F. Sallis, L. D. Frank, B. E. Saelens, and M. K. Kraft, "Active transportation and physical activity: opportunities for collaboration on transportation and public health research," *Transp Res Part A Policy Pract*, vol. 38, no. 4, pp. 249–268, 2004, doi: https://doi.org/10.1016/j.tra.2003.11.003.

[4] T. Shannon, B. Giles-Corti, T. Pikora, M. Bulsara, T. Shilton, and F. Bull, "Active commuting in a university setting: Assessing commuting habits and potential for modal change," *Transp Policy (Oxf)*, vol. 13, no. 3, pp. 240–253, 2006, doi: https://doi.org/10.1016/j.tranpol.2005.11.002.

[5] A. Golub and K. Martens, "Using principles of justice to assess the modal equity of regional transportation plans," *J Transp Geogr*, vol. 41, pp. 10–20, 2014, doi: https://doi.org/10.1016/j.jtrangeo.2014.07.014.

[6] K. Martens, A. Golub, and G. Robinson, "A justice-theoretic approach to the distribution of transportation benefits: Implications for transportation planning practice in the United States," *Transp Res Part A Policy Pract*, vol. 46, no. 4, pp. 684–695, 2012, doi: https://doi.org/10.1016/j.tra.2012.01.004.

[7] K. Coffel *et al.*, "Guidelines for Providing Access to Public Transportation Stations," 2012.

[8] M. Catala, S. Dowling, and D. M. Hayward, "Expanding the Google Transit Feed Specification to Support Operations and Planning," 2011.

[9] J. Wong, "Leveraging the General Transit Feed Specification for Efficient Transit Analysis," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2338, pp. 11–19, Dec. 2013, doi: 10.3141/2338-02.

[10] J. Wong, L. Reed, K. Watkins, and R. Hammond, "Open Transit Data: State of the Practice and Experiences from Participating Agencies in the United States," 2013.

[11] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer Math (Heidelb)*, vol. 1, no. 1, pp. 269–271, 1959, doi: 10.1007/BF01386390.

[12] R. Bellman, "On a routing problem," *Q Appl Math*, vol. 16, no. 1, pp. 87–90, 1958.

[13] R. W. Floyd, "Algorithm 97: Shortest path," *Commun ACM*, vol. 5, no. 6, p. 345, 1962, doi: http://doi.acm.org/10.1145/367766.368168.

[14] D. B. Johnson, "Efficient Algorithms for Shortest Paths in Sparse Networks," *J. ACM*, vol. 24, no. 1, pp. 1–13, Jan. 1977, doi: 10.1145/321992.321993.

[15] J. Mote, I. Murthy, and D. L. Olson, "A parametric approach to solving bicriterion shortest path problems," *Eur J Oper Res*, vol. 53, no. 1, pp. 81–92, 1991, doi: https://doi.org/10.1016/0377-2217(91)90094-C.

[16] J. C. Namorado Climaco and E. Queirós Vieira Martins, "A bicriterion shortest path algorithm," *Eur J Oper Res*, vol. 11, no. 4, pp. 399–404, 1982, doi: https://doi.org/10.1016/0377-2217(82)90205-3.

[17] E. Q. V. Martins, "On a multicriteria shortest path problem," *Eur J Oper Res*, vol. 16, no. 2, pp. 236–245, 1984, doi: https://doi.org/10.1016/0377-2217(84)90077-8.

[18] C. Tung Tung and K. Lin Chew, "A multicriteria Pareto-optimal path algorithm," *Eur J Oper Res*, vol. 62, no. 2, pp. 203–209, 1992, doi: https://doi.org/10.1016/0377-2217(92)90248-8.

[19] J. Brumbaugh-Smith and D. Shier, "An empirical investigation of some bicriterion shortest path algorithms," *Eur J Oper Res*, vol. 43, no. 2, pp. 216–224, 1989, doi: https://doi.org/10.1016/0377-2217(89)90215-4.

[20] H. W. Corley and I. D. Moon, "Shortest Paths in Networks with Vector Weights," *J. Optim. Theory Appl.*, vol. 46, no. 1, pp. 79–86, May 1985, doi: 10.1007/BF00938761.

[21] H. G. Daellenbach and C. A. De Kluyver, "Note on Multiple Objective Dynamic Programming," *Journal of the Operational Research Society*, vol. 31, no. 7, pp. 591–594, Jul. 1980, doi: 10.1057/jors.1980.114.

[22] A. J. V Skriver and K. Andersen, "A label correcting approach for solving bicriterion shortest-path problems," *Comput Oper Res*, vol. 27, pp. 507–524, May 2000, doi: 10.1016/S0305-0548(99)00037-4.

[23] P. Dell'Olmo, M. Gentili, and A. Scozzari, "On Finding Dissimilar Pareto-Optimal Paths," *Eur J Oper Res*, vol. 162, pp. 70–82, Apr. 2005, doi: 10.1016/j.ejor.2003.10.033.

[24] E. Machuca, L. Mandow, and J. Cruz, "An evaluation of heuristic functions for bicriterion shortest path problems," *New Trends in Artificial Intelligence. Proceedings of EPIA'09*, Jan. 2009.

[25] L. Mandow and J. L. de la Cruz, "Frontier Search for Bicriterion Shortest Path Problems," in *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, NLD: IOS Press, 2008, pp. 480–484.

[26] L. Mandow and J. L. Pérez de la Cruz, "Path recovery in frontier search for multiobjective shortest path problems," *J Intell Manuf*, vol. 21, no. 1, pp. 89–99, 2010, doi: 10.1007/s10845-008-0169-2.

[27] R. Mart\'\i, J. Luis González Velarde, and A. Duarte, "Heuristics for the Bi-Objective Path Dissimilarity Problem," *Comput. Oper. Res.*, vol. 36, no. 11, pp. 2905–2912, Nov. 2009, doi: 10.1016/j.cor.2009.01.003.

[28] A. Raith and M. Ehrgott, "A comparison of solution strategies for biobjective shortest path problems," *Comput Oper Res*, vol. 36, pp. 1299–1331, Apr. 2009, doi: 10.1016/j.cor.2008.02.002.

[29] J. Widuch, "A Label Correcting Algorithm for the Bus Routing Problem," *Fundam Inform*, vol. 118, pp. 305–326, Aug. 2012, doi: 10.3233/FI-2012-716.

[30] C.-L. Liu, T.-W. Pai, C.-T. Chang, and C.-M. Hsieh, "Path-planning algorithms for public transportation systems," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, 2001, pp. 1061–1066. doi: 10.1109/ITSC.2001.948809.

[31] A. V. MUSTAFA ALZAIDI, "Trip Planning Algorithm For Gtfs Data With Nosql Structure To Improve The Performance," *J Theor Appl Inf Technol*, vol. Vol.99. No, no. 10 31st May 2021, pp. 2290–2300, May 2021.

[32] S. Farber, B. Ritter, and L. Fu, "Space–time mismatch between transit service and observed travel patterns in the Wasatch Front, Utah: A social equity perspective," *Travel Behav Soc*, vol. 4, pp. 40–48, 2016, doi: https://doi.org/10.1016/j.tbs.2016.01.001.

[33] S. K. Fayyaz S., X. C. Liu, and G. Zhang, "An efficient General Transit Feed Specification (GTFS) enabled algorithm for dynamic transit accessibility analysis," *PLoS One*, vol. 12, no. 10, pp. e0185333-, Oct. 2017, [Online]. Available: https://doi.org/10.1371/journal.pone.0185333

[34] S. Motamed, A. Broumandnia, and A. Nourbakhsh, "Multimodal biometric recognition using particle swarm optimization-based selected features," *Journal of Information Systems and Telecommunication*, vol. 1, pp. 79–87, Mar. 2013, doi: 10.7508/jist.2013.02.002.

[35] P. Goli and M. M. R. KARAMI, "Speech Intelligibility Improvement in Noisy Environments for Near-End Listening Enhancement," 2016.

[36] G. M. Saeed, H. B. N. Babak, and L. Mojtaba, "Achieving Better Performance of S-MMA Algorithm in the OFDM Modulation," 2013.

[37] Q. Zervaas, *The Definitive Guide to GTFS (Consuming open public transportation data with the General Transit Feed Specifcation)*, First Edit. 2014. [Online]. Available: http://gtfsbook.com/gtfs-book-sample.pdf

[38] N. Amirah, D. Mohamad, and A. Hilmy, *Acceptable walking distance accessible to the nearest bus stop considering the service coverage*. 2021. doi: 10.1109/ICOTEN52080.2021.9493435.

[39] "Introduction to Redis – Redis." https://redis.io/topics/introduction (accessed Jan. 18, 2021).

[40] S. Tapia-Fernández, D. García-García, and P. García-Hernandez, "Key Concepts, Weakness and Benchmark on Hash Table Data Structures," *Algorithms*, vol. 15, no. 3, 2022, doi: 10.3390/a15030100.