A New Data Reduction Technique for Efficient Arabic Data Sentiment Analysis

Remah Younisse
Department of Computer Science
Princess Sumaya University, Jordan
r.baniyounisse@psut.edu.jo

Arafat Awajan
Department of Computer Science
Princess Sumaya University, Jordan
awajan@psut.edu.jo

Maram Bani Younes
Department of Cybersecurity
American University of Madaba, Jordan
Mbani047@uottawa.ca

Abstract: *Sentiment Analysis (SA) has become popular for determining opinions and feelings from textual data. The huge amount of text fed to sentiment analysis models can be considered an obstacle that slows the models' execution. Besides, it requires a large memory to run these models. Thus, data reduction and feature extraction processes can enhance these models' performance in terms of time complexity and memory usage. However, the reduction process should not affect the classification models' performance with the sentiment analysis process to split textual data according to its polarity. In this work, we present an analytical study of the role of data reduction techniques in improving analysis time and accuracy conducted on Arabic datasets. A structured performance assessment of features is produced. The Bidirectional Encoder Representations from Transformers (BERT) models are used as a data reduction tool, and then the performance results of these models are compared to the performance of the Frequency-Inverse Document Frequency (TF-IDF) model. The results show that the quality of the features extracted via BERT models is more valuable for sentiment analysis tasks and can enhance the required time by eight different classifiers. For example, the performance of the Random Forest classifier was improved by 3% when BERT models were used for feature extraction rather than the TF-IDF method, and the time taken by the Random Forest Classifier (RFC) was reduced to one-tenth compared to its performance when the TF-IDF was used as a feature extraction tool.*

Keywords: *Arabic language, sentiment analysis, BERT, data reduction, TF-IDF.*

Received February 13, 2025; accepted June 11, 2025
https://doi.org/10.34028/iajit/22/5/7

1. Introduction

Sentiment Analysis (SA) is a popular Natural Language Processing (NLP) application that aims to extract the opinions or emotions of textual content [34]. The textual data are usually sourced from users' reviews, comments, or social media posts. SA is typically performed to classify texts into positive and negative content. In other cases, the contents are classified into positive, negative, or neutral content [27]. SA is essential for evaluating users' attitudes towards certain cases (i.e., politics, sports, culture) or sometimes products or services [25]. The purpose of these SA models is to extract the polarity of the text. Hence, in many cases, certain parts (words) of the text can be sufficient to conclude the text's polarity. Datasets used in SA processes tend to be huge and rich in text. The enormous textual content of these models is considered an essential hindrance confronting the development and evaluation of SA models [19]. The size of the dataset critically consumes memory and increases classifiers' training and testing time.

Data reduction methods such as Principal Component Analysis (PCA) and auto-encoders were proven to be efficient tools in this scope. They are widely used to reduce the data size while keeping essential data content. Thus, data reduction has been used on large datasets in different works as a step before training and testing SA models. It is a fact that the value and the amount of sentiment content of texts differ according to the language [9]. For the Arabic language, many SA works have been presented and proposed. According to our knowledge, merging data reduction models with SA NLP models has yet to be mentioned comparatively in the literature for Arabic content. Hence, this work aims to present an SA model for Arabic texts where the datasets used throughout this work are prepared through pretrained transformer encoders Bidirectional Encoder Representations from Transformers (BERT) models. Pretrained transformer encoders aim to reduce the size of the dataset while keeping the fundamental text content. The SA model receives the outcome of the encoder part of the BERT model, presented by the pooler layer output; then, it classifies the texts into positive or negative texts.

The main contribution of this work is to investigate the efficiency of Arabic text SA models when the Arabic text is processed using famous reduction techniques such as BERT models. The presented work compares the accuracy and the required time to train and test SA models when the data reduction process is applied to the data versus when non-reduced data is used. Then, a structured performance assessment of features produced by BERT models is presented. BERT models are mainly considered for data reduction, the performance of these models is compared to the performance of the traditional

TF-IDF model.

The rest of the paper is organized as follows: Section 2 presents the related work and previous studies in this field of research focusing on the existing gaps. Section 3 contains the proposed reduction technique used throughout this work. Section 4 presents the results and results analysis, followed by the conclusion in section 5.

2. Related Work

Different sentiment analysis works have been presented in the literature for languages such as English and Arabic [2, 3, 6, 20, 21, 34]. In these studies, several data collections, feature selection, and extraction methods have been used. Data collection sources can be obtained from social media, forums, weblogs, and e-commerce websites [10, 22, 31] The feature selection methods help to build classification models. Many techniques have been developed to help understand word combinations for SA, such as unigrams, bigrams, and trigrams [5, 7, 11, 22, 31]. Meanwhile, pragmatic feature extraction enhances the process for SA. Extracting features such as emoticons, punctuation, slang, and negations is highlighted as a useful process to empower the SA [14, 16, 21, 22, 36]. Feature extraction methods include term frequency, parts of speech tagging, negations, and sentiment boundaries, which play crucial roles, as well as the bag of words approach. Word embedding techniques such as Word2vec, GloVe, FastText, and Embeddings from Language Models (ELMo) are also found useful in many SA works as well [22]. Feature selection approaches focus on eliminating irrelevant data content to enhance classification processes. The main discussed feature selection approaches are filter wrapper and embedded approaches. Feature extraction with SA was recently highly discussed and investigated [2, 10, 12, 14, 21, 31]. Kaur and Sharma [21] emphasized the role data reduction techniques play in SA models. The accuracy and F1 measures of different SA models were enhanced when a proposed feature extraction method was used.

Most studies in this field have considered the English language. However, some studies have been developed specifically to investigate and develop sentiment analysis in the Arabic language. Some studies have reported and declared the distinguished limitations and challenges in the recent Arabic-based sentiment analysis [1, 17]. Alhumoud and Al Wazrah [5] have focused on selecting the best mechanism or technique to apply the sentiment analysis for Arabic. Touahri [33] presented a road map for preparing Arabic textual data to be used in classification models. Positive and negative Bag-of-Words (BoWs) are extracted from datasets. These BoWs are then used through a classification process to detect the polarity of different texts. Then, three deep learning models were tested by Elhassan *et al*. [15] in the scope of SA with different word embedding techniques. They proposed an architecture for a Convolutional Neural Network (CNN) model and used it throughout their work. The work compared the CNN model to Long Short-Term Memory (LSTM) and the hybrid CNN-LSTM model. The results showed that using the fastText word embedding technique with the CNN model achieved the best accuracy measures.

Furthermore, Alyaba and Palade [4] proposed an approach that combines CNNs with LSTM networks to improve sentiment classification. The approach is based on removing the max-pooling layer from CNNs since the max-pooling layer reduces the length of generated feature vectors after convolving feature filters on the input data. The authors investigated different methods for preparing and representing the text features in order to increase the accuracy of Arabic sentiment classification. In these aforementioned studies, data reduction methods were not implemented in the used approach, yet the authors propose an efficient SA approach for Arabic datasets.

Few studies have considered data reduction mechanisms to enhance the efficiency of the SA in terms of reducing the processing time and the allocated memory such as Mhatre *et al*. [24]. Oussous *et al*. [26] built two Arabic SA deep learning models; the first model uses convolutional neural networks, and the second model uses a LSTM model. Different preprocessing steps, such as stemming, normalization, tokenization, and stop word removal, have been used to enhance the analysis. Finally, feature extraction methods to reduce text size, such as Boolean weighting, Term Frequency (TF) weighting, Inverse Document Frequency (IDF) weighting, and TF-IDF have been. The work proved the superiority of deep learning models with SA tasks over traditional classification methods such as support vector machines, naive bayes classifiers, and maximum entropy. However, the efficiency of using the data reduction technique has not been investigated in Oussous *et al*. [26]. Two different data reduction methods were tested in the scope of SA for the Arabic language [28]. The tested reduction methods are the unsupervised transformation method Principal Component Analysis (PCA) and the supervised transformation method Latent Dirichlet Allocation (LDA). The study showed better accuracy results when datasets are reduced using LDA than when PCA is used. The study covered only the enhancement of accuracy when data reduction methods are used. The reduction volume and the time measurements are not illustrated.

In general, some studies and research have considered the SA for Arabic datasets. However, data reduction and text summarization techniques were not given sufficient space, especially when analyzing the required time to train and test these models. Thus, this study is interested in reducing the time and size of data, where SA models are used for the data reduction.

3. The Proposed Data Reduction Techniques

In this section, we presented the proposed data reduction techniques that are developed based on The Frequency-Inverse Document Frequency (TF-IDF) [13] and BERTs Models [8, 18, 29] The proposed techniques aim to study how feature extraction methods can enhance SA tasks, especially for Arabic datasets. Hence, Figure 1 clarifies how the TF-IDF process extracts the most important features from the dataset. This work uses the classification's most important 10000, 5000, and 1000 features produced from the TF-IDF vectors. Then, the accuracy of the classifiers and the time taken by every classifier is analyzed. The classification process is done using a 5-k fold validation process. The time every classifier takes to perform the 5-K fold classification is then reported. The time taken by the vectors is also recorded and analyzed.

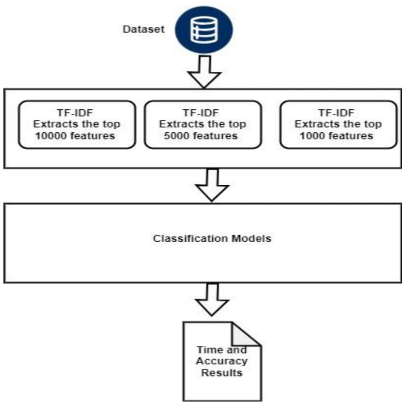


Figure 1. TF-IDF model of SA.

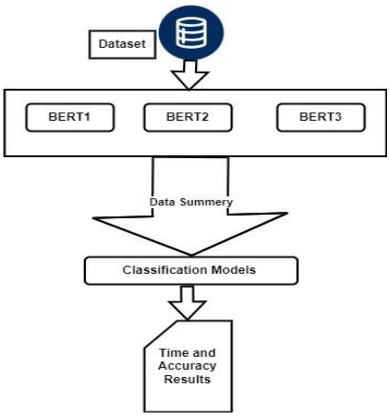


Figure 2. BERT models of SA.

On the other hand, Figure 2 shows how BERT pre-trained models are used throughout this work to extract the data features. The data is divided into batches, so every 1000 tweets are gathered in a batch and passed into the BERT tokenizer. The tokenizer prepares the data to be passed into the transformer model. The pooler layer at the core of the transformer model transforms every data instant in the dataset into a vector of features whose size is 768. These vectors of features are gathered in a list data structure and labeled to be used with the classification models later. Thus, the tweets are transformed into vectors of features; every vector length is 768, then the extracted features can be viewed as representatives for the dataset. These extracted features are used during the classification process. The accuracy and the time taken by the classifiers are then reported. A 5-k fold validation process is also used here. The time taken by the transformers is also recorded and analyzed.

3.1. Frequency-Inverse Document Frequency (TF-IDF)

This model statistically indicates the importance of a term/word in a document. TF-IDF consists of two terms: the frequency, which is the count of the times the term is repeated in the document. The inverse document frequency indicates the rareness of the term in all documents in the dataset. The less repeated terms in all documents are more helpful for classification tasks. The multiplication result of both terms indicates the importance of the term inside the document. This work used the TF-IDF value to extract the most essential words in every dataset document. After assigning TF-IDF values for all words, only the most critical n words are used in the classification process, which makes the classification rely on the most important features rather than all words in each document.

3.2. Bidirectional Encoder Representations Model (BERT)

This model is based on the transformer architecture. BERT is a state-of-the-art deep-learning architecture capable of understanding contextual data in a text. BERT can catch bidirectional context by processing words with the surrounding context in a sentence. This capability is achieved through pre-training the model using massive textual datasets. BERT learned by predicting omitted words in a sentence. Adding task-specific layers allows the model to be fine-tuned to fit with the desired language processing tasks such as sentiment analysis, question answering, and text classification.

At its core, BERT includes multiple layers of self-attention mechanisms within transformers. The attention layer in BERT is essential in comprehending contextual associations within a sequence of words. These layers enable BERT to weigh the importance of each word to the input sentence so the dependencies and relationships between words are extracted. The architecture of BERT includes an encoder that contains multiple stacked transformer layers. Every layer adds attention mechanisms that allow BERT to process information from left-to-right and right-to-left in a sentence. This bidirectional learning allows BERT to create contextual embedding for words based on their context within a sentence. The last pooling layer in BERT summarizes the data; it contains the words learned by the model. The pooler layer in BERT conducts a pooling operation across all word embedding along the sequence dimension, such as max-pooling or mean-pooling. This layer collects the information from the embedding to create a vector representation that includes the comprehensive content and context of the entire input text. This vector can then be used with classification operations.

In this work, the BERT models are utilized to extract summarization for the data so it can be used for the SA process by feeding the summary to the classification models, as discussed in the methodology section. Three BERT transformers were used throughout this work to highlight the effect of using different models in the required time to extract the features as well as the quality of the extracted features. These three models will be referred to as BERT1, BERT2, and BERT3.

1. BERT1 [8]: the model name is "BERT-base-arabertv02-twitter", it is trained with around 60 M Multi-Dialect Tweets and achieved excellent results in SA tasks for Arabic tweets.
2. BERT2 [29]: the model name is "asafaya/BERT-base-Arabic" and it is trained with around 8.2 billion words. The model was created for offensive speech identification on social media.
3. BERT3 [18]: the model we used is "CAMeL-Lab/bert-base-arabic-camelbert-da-sentiment", this model is fine-tuned for SA tasks in Arabic. The model is trained with around 5.8 billion Arabic words in Dialectal Arabic (DA). The three models were selected due to their wide usage in the literature of Arabic NLP works.

All three BERT models used in this study-AraBERTv2-Twitter, Asafaya's BERT, and CAMeL BERT follow the standard transformer architecture where the special classification token [CLS] token is prepended to each input. The sentence representation is extracted using the final hidden state of this token (CLS pooling), which is a common and effective strategy for classification tasks. Alternative pooling methods, such as mean-pooling, were not used to remain consistent with the pretraining and fine-tuning procedures of the original model developers.

3.3. The Classification Models

This work utilizes eight Machine Learning (ML) models to classify texts into positive and negative based on the sentiment carried in the text. These ML models are selected due to their reliability [7]. They set standardized methods, guaranteeing consistency and reproducibility in research findings, contributing to the robustness of the results, and increasing confidence in the conclusions drawn. Some details regarding each model are illustrated in the rest of this section.

1. The Gaussian Naive Bayes (GNB) classifier: is a probabilistic model based on Bayes' theorem. GNB assumes that the features used through the classification process are independent. It estimates the probability of a data point belonging to a particular class by estimating the conditional probabilities of each feature given the class label.
2. Logistic Regression (LR): is a linear model used for classification tasks. It calculates the probability that a given input belongs to a particular class using the logistic function. The output is usually between 0 and 1. LR is appropriate for binary classification tasks.
3. K Nearest Neighbors (KNN): is an effective classifier that classifies a data point based on the majority class of its neighbors. The similarity is usually measured using Euclidean distance between the data points and assigns a class label based on the most dominant class among its closest neighbors.
4. Linear Discriminant Analysis (LDA): is a classification method that seeks to locate linear combinations of features that represent the differences between classes. LDA calculations are based on calculating the mean and variance of each class to create decision boundaries to separate classes.
5. Adaptive Boosting, or AdaBoost: is an ensemble learning method. It creates a robust classifier by merging different weak classifiers. It trains classifiers sequentially and gives higher weight to misclassified data instances. The AdaBoost classifier used in this work consists of 100 classifiers.
6. Random Forest (RF): it is an ensemble learning procedure from different decision trees. The decisions of the independent decision trees are used to generate the classification conclusion. The model used in this work consists of 100 decision trees.
7. Support Vector Machines (SVM): find the optimal hyperplane that best splits classes in a high-dimensional space.
8. The Multi-Layer Perceptron (MLP) classifier: it is a multilayered artificial neural network. It consists of input, hidden, and output layers. The MLP model used in this work uses the Rectified Linear Unit (RelU) activation function.

4. Results and Results Analysis

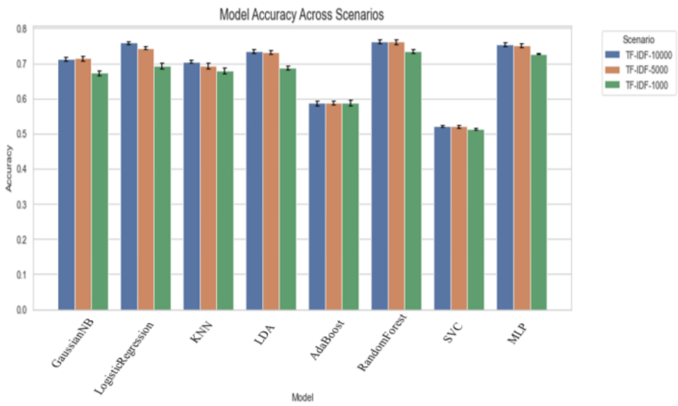
This section illustrates the results of using the feature extraction methods to enhance the SA classifiers' accuracy. Measuring the performance of an SA model requires measuring the performance of the classification method used to build the model. The accuracy metric is used to measure the classifiers' performance. The used dataset is balanced; thus, accuracy can express the classification results accurately. Besides, training and testing periods are selected to evaluate the performance of the used classifiers. A five-fold cross-validation is used to validate the measured results in the classification process. In the remainder of this section, the dataset used is explained first. Then, the results obtained from the SA models that used TF-IDF and BERT models for data extractions are illustrated.

4.1. Dataset

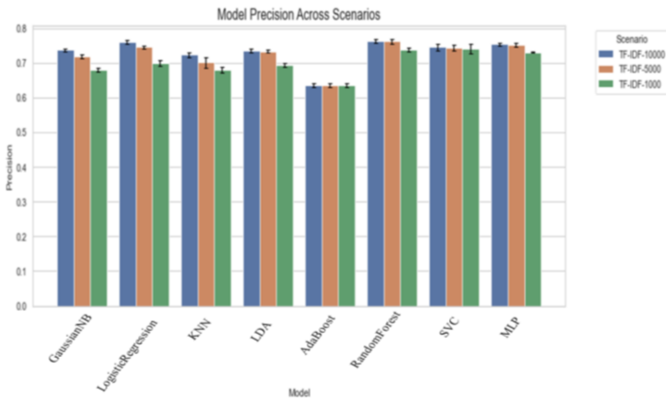
The dataset used in this work is the Arabic Sentiment Twitter Corpus (ASTC) [29]. ASTC dataset was collected to offer an Arabic sentiment corpus for the research community to build and analyze ML and deep learning methodologies regarding Arabic sentiment analysis. This dataset we organized in April 2019 is one of the most recent in this field. It contains 58K Arabic tweets (47K training, 11K test) annotated in positive and negative labels. The dataset is balanced and collected using a positive and negative emoji lexicon. This dataset has widely been used through different works such as [23, 28, 32, 35, 37]. The data went through multiple preprocessing steps in this work, such as punctuation marks and stop-word removal. After cleaning the dataset, the number of positive tweets is 22761, while the number of negative tweets is 22514.

4.2. Results of TF-IDF Feature Extraction

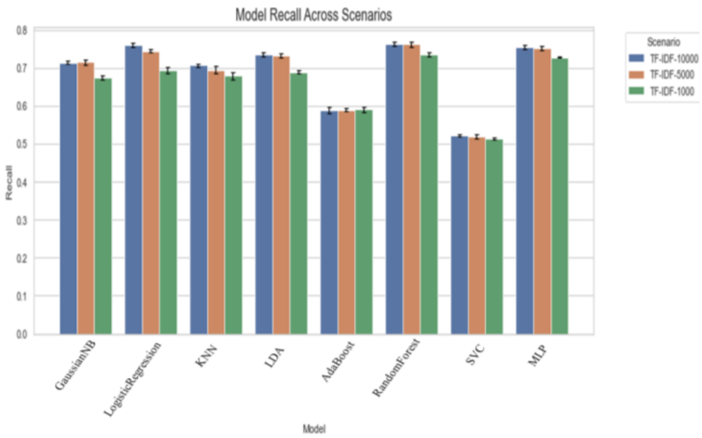
When the TF-IDF vectors were used as a feature extraction method, the vectors selected the top 10000, 5000, and 1000 features. The selected features are then input to the classifiers to perform the SA task. The top 10000, 5000, and 1000 features have been selected because these numbers reflect almost 20%, 10%, and 2% of the total words in the corpora. Figure 3 shows the performance measures of the eight investigated ML classifiers. The measures are shown for the three TF-IDF vectorization cases. The figure shows that all classifiers' performance results are the best when the top 10000 features are used. It is also clear that the drop in accuracy when the top5000 words are used is minor and reaches around 2% when the LR model is used. The decrease in the accuracy when the top 1000 words are used for classification is notable compared to the results of the top 10000 and 5000 words.



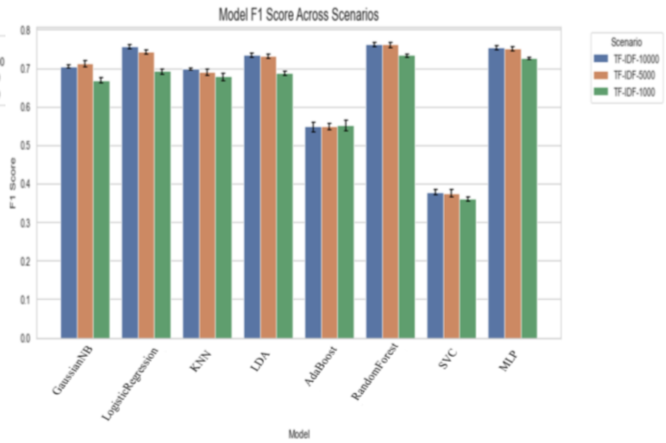
a) Accuracy.



b) Precision.



c) Recall.



d) F1 Score.

Figure 3. Accuracy, Precision, Recall, and F1 measures of top 10000, 5000, 1000 features generated by TF-ID.

Regarding the training and testing time, the classifiers taken when TF-IDF is used for feature extraction are presented in Table 1. Most models have the highest time when the top 10000 features are used. When the top 5000 features drop, and drop more when the top 1000 features are used. The notable drop in the required time to train and test models is confronted by the minor decrease in the accuracy results when fewer features are used, mainly when 20% and 10% of the features are used.

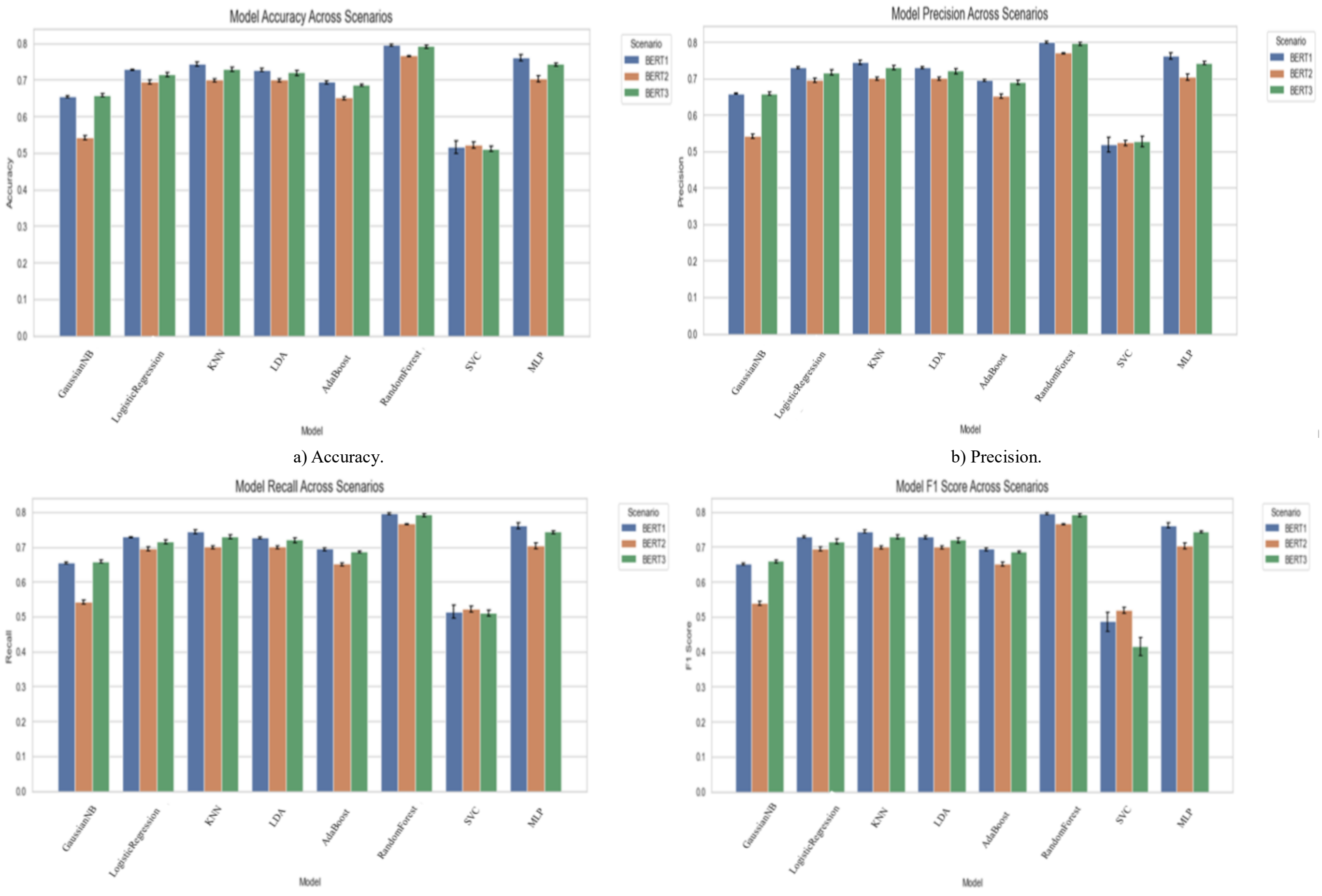
Moreover, it is clear that the RF model produces the best accuracy results in all cases. The model also requires the highest training and testing times. The GNB model, on the other hand, requires the most minor training and testing time. The SVC model produces the worst results; this is since the model classifies the data into one class only. The time taken by the TF-IDF vectors model was 2.49(s) when the top 1000 features were selected. 16.59(s) and 31.81(s) when the top 5000 and 10000 features were selected.

Table 1. Training and testing times for top features generated by TF-IDF.

Model	Train_10000	Test_10000	Train_5000	Test_5000	Train_1000	Test_1000
NV	23.9873s	5.108189s	26.03301s	6.406889s	2.831947s	0.639846s
LR	66.62291s	15.27611s	52.68916s	11.95598s	9.839882s	2.357709s
KNN	152.7666s	54.01835s	165.9282s	87.14357s	31.6071s	16.39402s
LDA	4357.14s	373.2022s	702.0165s	164.01s	23.17714s	5.73879s
AdaBoost	1917.546s	2056.956s	2962.901s	773.4175s	241.3688s	63.62087s
RF	3690.575s	2299.896s	3124.26s	930.0855s	533.0658s	142.6075s
SVC	203.2327s	54.96273s	80.59473s	21.36403s	18.85116s	4.606454s
MLP	2447.832s	576.8566s	688.2734s	163.3238s	191.3207s	47.48142s

4.3. Results of BERT Feature Extraction

The classification performance evaluation across BERT-based feature sets, in Figure 4 shows that Random Forest consistently achieves the highest accuracy, precision, recall, and F1 scores, indicating its strong adaptability to contextual embeddings. KNN, LDA, and logistic regression also function reliably across all metrics, keeping a stable margin of effectiveness regardless of the BERT variant used. Among the three BERT models, BERT1 generally generates slightly better results, indicating that its embedding structure better maintains discriminative information. Conversely, naive bayes and SVC demonstrate noticeably weaker performance, particularly under BERT2 and BERT3, likely due to their limited capacity to leverage the complexity of deep models. MLP, while not the absolute top performer, shows a solid balance between precision and recall, making it a competitive choice when considering both performance and computational feasibility. Overall, the results highlight that tree-based and neural models are more effective at utilizing BERT-derived features, with BERT1 emerging as the most robust among the tested variants.



a) Accuracy. b) Precision.

c) Recall. d) F1 Score.

Figure 4. Accuracy, precision, recall, and F1 measures when BERT models are used.

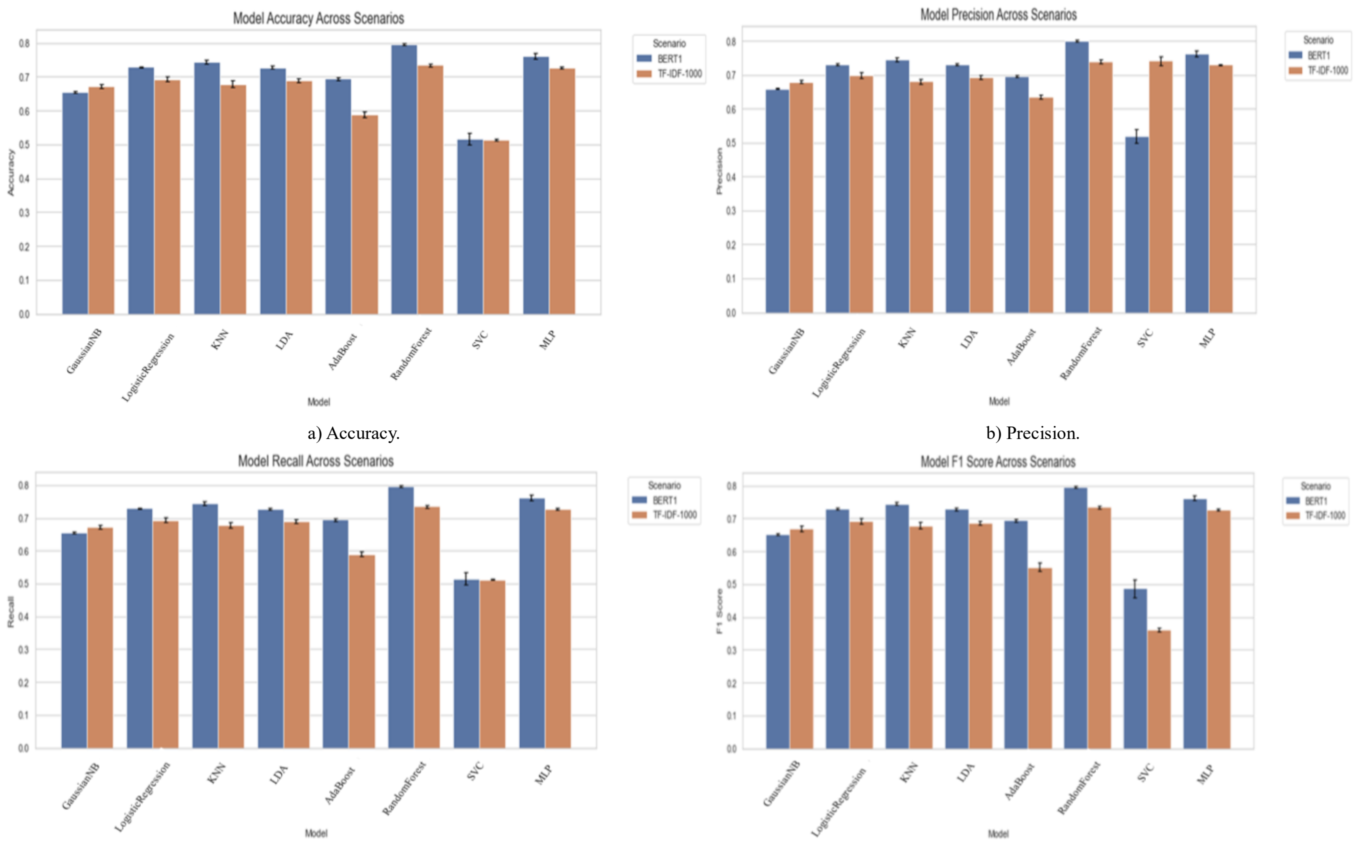
The training and testing times for the models are shown in Table 2. We notice that the time taken by the classifiers is close for the features extracted from the three BERT models. This is due to the equivalent size of the data extracted by the three BERT models and passed to the classifiers. It is also clear that the AdaBoost model took the highest required training and testing time, despite its low accuracy. The GNB model, on the other hand, took the least time. In average, extracting the features using BERT1 took 9307.95(s), While BERT2 took 10030.00(s), and BERT3 took 8996.99(s).

Table 2. Training and testing time(s) for the features generated by BERT Models.

Model	Train_BERT1	Test_BERT1	Train_BERT2	Test_BERT2	Train_BERT3	Test_BERT3
NV	2.534095s	0.702595s	2.624136s	0.698781s	2.476210s	0.647053s
LR	9.929080s	2.340627s	9.659170s	2.132352s	9.067993s	2.168705s
KNN	61.526750s	32.484020s	58.687390s	31.689780s	51.066670s	17.381920s
LDA	34.183040s	8.540174s	34.209390s	8.230220s	19.220860s	5.516998s
AdaBoost	2208.156000s	535.949200s	1929.663000s	290.627100s	1164.702000s	297.949100s
RF	314.526000s	78.372560s	318.466700s	83.724830s	334.511300s	83.937470s
SVC	13.734250s	3.560161s	14.984460s	3.850793s	14.960800s	3.693091s
MLP	424.387600s	110.063500s	437.470500s	104.887700s	415.551200s	103.355300s

4.4. Comparison Between TF-IDF and BERT

The comparison between the TF-IDF and BERT considering the best results from the TF-IDF step (when the top 10000 features were used) with the best results of the BERT phase. It is good to mention here that BERT models extract vectors of length 768 from every data instant (Tweet), indicating that the number of features used through the classification process is minor compared to when TF-IDF was used to extract the features. Figure 5 illustrates the comparison results. From the Figure, the TF-IDF feature extraction method produces results from the NV and LR classifiers with better. In contrast, the rest of the classifiers perform better with the features extracted from the BERT models.



a) Accuracy.

b) Precision.

c) Recall.

d) F1 Score.

Figure 5. Accuracy, precision, recall, and f1 measures comparison between TF-IDF and BERT.

There is a consistent advantage in favor of BERT across all evaluation metrics. Models trained on BERT1 representations generally outperform their TF-IDF counterparts in terms of accuracy, precision, recall, and F1 score. This improvement is particularly evident in complex classifiers like RF and MLP, suggesting that contextual embeddings capture more nuanced patterns in the data than frequency-based features. While TF-IDF still provides reasonable performance, especially in simpler models like naive bayes, the overall results underscore the effectiveness of transformer-based representations in enhancing classification performance.

Finally, referring to Tables 1 and 2 we can infer that the time taken by classifiers when BERT models are used is generally lower than the time taken by the models when TF-IDF is used with the top 10000 and 5000 features.

5. Conclusions and Recommendations

In this work, Arabic tweets SA was analyzed when two feature extraction methods were used prior to the classification process. The feature extraction methods are TF-IDF vectorization, when specific percentages of the most important features are extracted, and BERT transformers. Three BERT transformers are used throughout this work, and the pooler layer output of each transformer is used for classification. The accuracy of eight famous classifiers is analyzed for each reduction technique. The time taken by every classifier and reduction technique is also presented and analyzed. Extracting features using BERT models takes much more time than TF-IDF, but it can enhance the accuracy and the required time for the used ML classifiers. Based on the presented results, the following recommendations can be raised:

- BERT transformers takes more time than TF-IDF for feature extraction.
- The features extracted by BERT transformers can better represent the sentimental content of the dataset.
- The total time taken for feature extraction and classification is higher when BERT models are used. Yet, the classifiers take less time when the features extracted by BERT models are used.
- RF, AdaBoost, and MLP models take longer than other classifiers but can produce better accuracy results.
- GNB and SVC models are not recommended for SA tasks when the feature extraction process exceeds the SA process.

In the future, the work can be repeated with other datasets and feature extraction methods such as PCA. Further data preprocessing steps will be implemented to reduce the text dimensionality by grouping similar words in clusters based on semantic similarity to prove its efficiency in reducing the data dimensionality.

References

[1] Abo M., Raj R., and Qazi A., “A Review on Arabic Sentiment Analysis: State-of-the-Art, Taxonomy and Open Research Challenges,” *IEEE Access*, vol. 7, pp. 162008-162024, 2019. DOI:10.1109/ACCESS.2019.2951530

[2] Ahuja R., Chug A., Kohli S., Gupta S., and Ahuja P., “The Impact of Features Extraction on the Sentiment Analysis,” *Procedia Computer Science*, vol. 152, pp. 341-348, 2019. https://doi.org/10.1186/s40537-022-00680-6

[3] Ajith A., Gupta B., Verma S., Maurya A., Husain M., and et al., “Improvement of Translation Accuracy for the Word Sense Disambiguation System Using Novel Classifier Approach,” *The International Arab Journal of Information Technology*, vol. 21, no. 6, pp. 1124-1142, 2024. https://doi.org/10.34028/iajit/21/6/14

[4] Alayba M. and Palade V., “Leveraging Arabic Sentiment Classification Using an Enhanced CNN-LSTM Approach and Effective Arabic Text Preparation,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 9710-9722, 2021. https://doi.org/10.1016/j.jksuci.2021.12.004

[5] Alhumoud S. and Al Wazrah A., “Arabic Sentiment Analysis Using Recurrent Neural Networks: A Review,” *Artificial Intelligence Review*, vol. 55, no. 1, pp. 707-748, 2022. https://doi.org/10.1007/s10462-021-09989-9

[6] Alkaoud M., Alsaqoub M., Aljodhi I., Alqadibi A., and Altammami O., “ACLM: Developing a Compact Arabic Language Model,” *The International Arab Journal of Information Technology*, vol. 22, no. 3, pp. 535-546, 2025. https://doi.org/10.34028/iajit/22/3/9

[7] Alqarqaz M., Younes M., and Qaddoura R., “An Object Classification Approach for Autonomous Vehicles Using Machine Learning Techniques,” *World Electric Vehicle Journal*, vol. 14, no. 2, pp. 1-17, 2023. https://doi.org/10.3390/wevj14020041

[8] Antoun W., Baly F., and Hajj H., “AraBERT: Transformer-Based Model for Arabic Language Understanding,” *in Proceedings of the Workshop Language Resources and Evaluation Conference*, Marseille, pp. 9-15, 2020. https://aclanthology.org/2020.osact-1.2/

[9] Araujo M., Pereira A., and Benevenuto F., “A Comparative Study of Machine Translation for Multilingual Sentence-Level Sentiment Analysis,” *Information Sciences*, vol. 512, pp. 1078-1102, 2020. https://doi.org/10.1016/j.ins.2019.10.031

[10] Avinash M. and Sivasankar E., “A Study of Feature Extraction Techniques for Sentiment Analysis,” *in Proceedings of the Emerging Technologies in Data Mining and Information Security*, Kolkata, pp. 475-486, 2018. https://doi.org/10.1007/978-981-13-1501-5_41

[11] Awajan A., “Semantic Similarity Based Approach for Reducing Arabic Texts Dimensionality,” *International Journal of Speech Technology*, vol. 19, no. 2, pp. 191-201, 2016. https://doi.org/10.1007/s10772-015-9284-6

[12] Ayesha S., Hanif M., and Talib R., “Overview and Comparative Study of Dimensionality Reduction Techniques for High Dimensional Data,” *Information Fusion*, vol. 59, pp. 44-58, 2020. https://doi.org/10.1016/j.inffus.2020.01.005

[13] Chakraborty K., Bhatia S., Bhattacharyya S., Platos J., and et al., “Sentiment Analysis of Covid-19 Tweets by Deep Learning Classifiers-A Study to Show How Popularity is Affecting Accuracy in Social Media,” *Applied Soft Computing*, vol. 97, pp. 106754, 2020. https://doi.org/10.1016/j.asoc.2020.106754

[14] Chen X., Xue Y., Zhao H., Lu X., Hu X., and Ma Z., “A Novel Feature Extraction Methodology for Sentiment Analysis of Product Reviews,” *Neural*

Computing and Applications, vol. 31, pp. 6625-6642, 2019. https://doi.org/10.1007/s00521-018-3477-2

[15] Elhassan N., Varone G., Ahmed R., Gogate M., and et al., "Arabic Sentiment Analysis based on Word Embeddings and Deep Learning," *Computers*, vol. 12, no. 6, pp. 126, 2023. https://doi.org/10.3390/computers12060126

[16] Gaashan K. and Younes M., "Deep Learning-based Arabic Optical Character Recognition: A New Comprehensive Dataset at Character and Word Levels," *in Proceedings of the 15th International Conference on Information and Communication Systems*, Irbid, pp. 1-6, 2024. DOI:10.1109/ICICS63486.2024.10638273

[17] Ghallab A., Mohsen A., and Ali Y., "Arabic Sentiment Analysis: A Systematic Literature Review," *Applied Computational Intelligence and Soft Computing*, vol. 2020, pp. 1-21, 2020. https://doi.org/10.1155/2020/7403128

[18] Inoue G., Alhafni B., Baimukan N., Bouamor H., and Habash N., "The Interplay of Variant, Size, and Task Type in Arabic Pre-Trained Language Models," *in Proceedings of the 6th Arabic Natural Language Processing Workshop*, Kyiv, pp. 1-13, 2021. https://aclanthology.org/2021.wanlp-1.10/

[19] Iqbal F., Hashmi J., Fung B., Batool R., Khattak A., Aleem S., and Hung P., "A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction," *IEEE Access*, vol. 7, pp. 14637-14652, 2019. DOI:10.1109/ACCESS.2019.2892852

[20] Kaloub A. and Elgabar E., "Speech-Based Techniques for Emotion Detection in Natural Arabic Audio Files," *The International Arab Journal of Information Technology*, vol. 22, no. 1, pp. 139-157, 2025. https://doi.org/10.34028/iajit/22/1/11

[21] Kaur G. and Sharma A., "A Deep Learning-Based Model Using Hybrid Feature Extraction Approach for Consumer Sentiment Analysis," *Journal of Big Data*, vol. 10, no. 5, pp. 1-25, 2023. https://doi.org/10.1186/s40537-022-00680-6

[22] Kumar H., Harish B., and Darshan H., "Sentiment Analysis on IMDB Movie Reviews Using Hybrid Feature Extraction Method," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 5, pp. 108-114, 2019. DOI:10.9781/ijimai.2018.12.005

[23] Manzoor A., Rehman Z., Shaheen M., and Khan M., "Data Mining of IoT based Sentiments to Classify Political Opinions," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 36, no. 4, pp. 1-16, 2022. https://doi.org/10.1080/0952813X.2022.2093406

[24] Mhatre M., Phondekar D., Kadam P., Chawathe A., and Ghag K., "Dimensionality Reduction for Sentiment Analysis Using Pre-Processing Techniques," *in Proceedings of the International Conference on Computing Methodologies and Communication*, Erode, pp. 16-21, 2017. DOI:10.1109/ICCMC.2017.8282676

[25] Nandwani P. and Verma R., "A Review on Sentiment Analysis and Emotion Detection from Text," *Social Network Analysis and Mining*, vol. 11, pp. 81, 2021. https://doi.org/10.1007/s13278-021-00776-6

[26] Oussous A., Benjelloun F., Lahcen A., and Belfkih S., "ASA: A Framework for Arabic Sentiment Analysis," *Journal of Information Science*, vol. 46, no. 4, pp. 544-559, 2020. DOI:10.1177/0165551519849516

[27] Pandya S. and Mehta P., "A Review on Sentiment Analysis Methodologies, Practices and Applications," *International Journal of Scientific and Technology Research*, vol. 9, no. 2, pp. 601-609, 2020.

[28] Saeed R., Rady S., and Gharib T., "Optimizing Sentiment Classification for Arabic Opinion Texts," *Cognitive Computation*, vol. 13, no. 1, pp. 164-178, 2021. https://doi.org/10.1007/s12559-020-09771-z

[29] Safaya A., Abdullatif M., and Yuret D., "KUISAIL at Semeval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media," *in Proceedings of the 14th Workshop on Semantic Evaluation*, Barcelona, pp. 2054-2059, 2020. DOI:10.18653/v1/2020.semeval-1.271

[30] Singh K., Devi S., Devi H., and Mahanta A., "A Novel Approach for Dimension Reduction Using Word Embedding: An Enhanced Text Classification Approach," *International Journal of Information Management Data Insights*, vol. 2, no. 1, pp. 1-10, 2022. https://doi.org/10.1016/j.jjimei.2022.100061

[31] Styawati S., Nurkholis A., Aldino A., Samsugi S., Suryati E., and Cahyono R., "Sentiment Analysis on Online Transportation Reviews Using Word2vec Text Embedding Model Feature Extraction and Support Vector Machine (SVM) Algorithm," *in Proceedings of the International Seminar on Machine Learning, Optimization, and Data Science*, Jakarta, pp. 163-167, 2022. DOI:10.1109/ISMODE53584.2022.9742906

[32] Suleiman D., Odeh A., and Al-Sayyed R., "Arabic Sentiment Analysis Using Naive Bayes and CNN-LSTM," *Informatica*, vol. 46, no. 6, pp. 79-86, 2022. DOI:10.31449/inf.v46i6.4199

[33] Touahri I., "The Construction of an Accurate Arabic Sentiment Analysis System Based on Resources Alteration and Approaches Comparison," *Applied Computing and Informatics*, vol. 21, no. 1, pp. 1-15, 2022. DOI:10.1108/ACI-12-2021-0338

[34] Wankhade M., Rao A., and Kulkarni C., "A Survey on Sentiment Analysis Methods,

Applications, and Challenges," *Artificial Intelligence Review*, vol. 55, pp. 5731-5780, 2022. https://doi.org/10.1007/s10462-022-10144-1

[35] Younes M. and Boukerche A., "A Performance Evaluation of a Context-Aware Path Recommendation Protocol for Vehicular Ad-Hoc Networks," *in Proceedings of the IEEE Global Communications Conference*, Atlanta, pp. 516-521, 2013. DOI:10.1109/GLOCOM.2013.6831123

[36] Younes M. and Boukerche A., "SCOOL: A Secure Traffic Congestion Control Protocol for VANETs," *in Proceedings of the IEEE Wireless Communications and Networking Conference*, New Orleans, pp. 1960-1965, 2015. DOI:10.1109/WCNC.2015.7127768

[37] Younes M., Alonso G., and Boukerche A., "A Distributed Infrastructure-Based Congestion Avoidance Protocol for Vehicular Ad Hoc Networks," *in Proceedings of the IEEE Global Communications Conference*, Anaheim, pp. 73-78, 2012. DOI:10.1109/GLOCOM.2012.6503093

Remah Younisse received the B.Sc. degree in Computer Engineering from Jordan University for Technology, the M.Sc. degree in Electrical Engineering from Princess Sumaya University for Technology, Amman, Jordan. Currently a Ph.D. candidate in computer science at Princess Sumaya University for Technology. She is currently a Teaching Assistant with the Computer Science Department. Her research interests include Natural Language Processing, Machine Learning, Explainable AI, 5G-Enabled Smart Grids, Fog Computing, Cybersecurity, and Reinforcement Learning for Cyber-Physical Systems.

Arafat Awajan is the Vice President for Scientific Research at the Royal Scientific Society (RSS). He served as President of Mutah University in Jordan from 2020 to 2023 after serving as vice president of Princess Sumaya University for Technology from 2017 to 2020. He earned his Bachelor's degree in Computer Science from Orleans University-France in 1983, and his Master Degree in computer Science from Franche-Comte university-France in 1984. In 1988, He completed his Ph.D. degree in Computer Science from Franche-Comte university-France. Professor Awajan has published over 120 Journal articles and conference papers indexed in Scopus and the Clarivate Master Journal Lists and has supervised more than 30 master and PhD thesis. His research interests include Artificial Intelligence, Natural Language Processing, and Image Processing.

Maram Bani Younes received the Ph.D. degree in Computer Science from the University of Ottawa, Ottawa, ON, Canada, in 2015. Her master and bachelor studies are completed in the University of Jordan. She is currently a full Professor with the Department of Cybersecurity, American University of Madaba, Jordan, and a Research Associate with the School of Information Technology and Engineering, University of Ottawa. Her research interests Are Wireless Networks, Wireless Ad Hoc and Sensor Networks, Green Applications, Security Protocols, Vehicular Networks, and Traffic Efficiency for Vehicular Network, NLP. She has received some prestigious research prizes such as: Best paper award in Globecome 2012, Best researcher award in Paradise Research Laboratory 2015 the Excellence in scientific research at Philadelphia University 2020, and the Rising Star Award from MSWIM 2022.