NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 2025; 14(2), 688-700

Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences

Araştırma makalesi / Research article

www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

Malsmsdetector: Malicious text message detector with hybrid feature vector and stacked ensemble model: a comparative study

Malsmsdetector: Hibrit özellik vektörü ve istiflenmiş topluluk modeli kullanan kötücül mesaj tespit aracı: karşılaştırmalı bir çalışma

Recep Sinan Arslan[1,*]

[1] *Kayseri University, Computer Engineering Department, 38000, Kayseri, Türkiye*

Abstract

In recent years, the emergence of telecommunication systems has led to an increase in global electronic messaging traffic. Most of this traffic contains unwanted content for the user. In this study, an approach is proposed in which feature vectors generated using DBOW and PV-DM techniques are used for classification as a hybrid for spam SMS detection. In the training and testing of the proposed method, four different datasets (UCI, BEC, Big NUS and DITNUS) that are widely used are combined and used. This dataset is tested with 10 different machine learning algorithms and then a unique stacked ensemble model is proposed to increase the performance. In the tests using the model, accuracy, precision, recall, F-score and AUC values are 98.38%, 98.39%, 98.39%, 98.37% and 96.81%, respectively. When 10-fold cross validation is applied to the obtained results, the standard deviation value is 0.004. The analysis time per sample is 0.087 milliseconds.

Keywords: Mobile spam filtering, Short message service (SMS), Bag of Words (BOW), Doc2Vec, hybrid feature vector (HFV)

Öz

Son yıllarda telekomünikasyon sistemlerinin ortaya çıkması, küresel elektronik mesajlaşma trafiğinde (SMS veya e-posta) artışa yol açmıştır. Bu trafiğin çoğu, kullanıcı için istenmeyen içerikler içermektedir. Bu çalışmada, spam SMS tespiti için DBOW ve PV-DM teknikleri kullanılarak üretilen öznitelik vektörlerinin hibrit olarak sınıflandırma için kullanıldığı bir yaklaşım önerilmiştir. Önerilen yöntemin eğitim ve testlerinde yaygın olarak kullanılan dört farklı veri kümesi (UCI, BEC, Big NUS ve DIT NUS) birleştirilerek kullanılmıştır. Bu veriseti 10 farklı makine öğrenmesi algoritması ile test edilmiş daha sonra başarımı artırmak için özgün bir yığılmış topluluk modeli önerilmiştir. Model kullanılarak yapılan testlerde doğruluk, kesinlik, geri çağırma, F-puanı ve AUC değerleri sırasıyla %98.38, %98.39, %98.39, %98.37 ve %96.81 olmuştur. Elde edilen sonuçlara, 10 katlı cross validation yapıldığında elde edilen standart sapma değeri 0,004'tür. Örnek başına analiz süresi 0.087 milisaniyedir. Testler sonucunda hibrit özellik vektörünün kullanımının SMS spam tespiti için başarılı sonuçlar sağladığı ve sistem performansının iyileştirilmesine katkıda bulunduğu gösterilmiştir.

Anahtar kelimeler: Mobil spam filtreleme, Kısa mesaj servisi (SMS), Kelime Torbası (BOW), Doc2Vec, hibrit özellik vektörü (HFV)

1 Introduction

Technological development makes people's lives easier. This development is a continuous and rapid process. The use of these technological infrastructures in many fields, such as biomedicine, information technology and communication [1], has now become mandatory [2]. Technological developments have responded well to this need, solving many problems with high efficiency and performance over the last 30 years [3]. For this reason, it is used in all areas of life and facilitates opportunities. Nowadays, people are becoming more and more dependent on computers every day, and they use computers that they think are safe to perform their tasks. Improvements have also been made in the field of communication and it has helped people to connect with the outside world by telephone [4]. Over the past 10 years, mobile technology applications have been an exciting area for researchers and developers. By 2021, there will be 5.27 billion individual mobile phone users worldwide, which is a high proportion of the world's population, approximately 67.1% [5]. In addition to normal telephone communication, the introduction of smartphones has led to the creation of applications that support many new features. Smartphones and tablets are used for many daily activities, such as internet research, entertainment, some mobile payments, access to personal data, banking services and, more recently, distance learning [6, 7].

Security structures in mobile devices are not as advanced and diversified as computers. The main reason for this is that the infrastructure in mobile devices has limited resources and users do not care enough about the security mechanism. However, these devices can store many personal data that is not available in traditional computer systems. This contradiction is an issue that needs to be worked on because of the situation [8]. Hence, mobile devices are becoming a

* Sorumlu yazar / Corresponding author, e-posta / e-mail: sinanarslanemail@gmail.com (R.S. Arslan)
Geliş / Received: 09.10.2024 Kabul / Accepted: 17.03.2025 Yayımlanma / Published: 15.04.2025
doi: 10.28948/ngumuh.1563906

much more inviting target for cyber attackers [9]. Spam or unwanted messages are one of the easiest ways to reach these targets [10]. These messages can be a medium for malware types, such as phishing scams, worms, backdoors, and key loggers. As a result, operational and financial losses may occur. Through the software that might come with these messages, cyber attackers can perform some malicious activities, such as calling certain numbers without the users' consent, stealing the contact data of people, and joining fraud or botnet groups [11]. Mobile devices offer many textual communication modes to communicate with people for personal or commercial purposes. These modes can be in the form of instant communication or short message services. One of the biggest problems of these communication methods is the increase in the rate of unwanted messages [12]. According to the report published by DataProt in 2021, 122.33 billion messages are sent daily around the world. Approximately 85% of these are spam messages [13]. Spam messages can be unsolicited adult material, violent material, a waste of resources, or a hotspot for various security breaches.

Short message service, or SMS, is very popular among mobile phone users [14]. SMS is used as a medium to send purchase notifications or advertisements because it is the most cost-effective way to reach large audiences. By 2021, approximately 98% of the 272 million smartphone users in the US will be using short message services, and this usage rate is increasing by 6.2% each year. Statistics show that 98% of SMS messages are opened and 95% are read and replied to within 3 minutes of receipt [15]. This increased use of SMS makes it a useful and convenient tool for attackers. SMS is one of the methods used by cyber attackers because of the high response rate of text messages, the low cost of sending them in bulk, and the difficulty of detecting the malicious URL addresses that accompany the messages [16]. Most of the existing approaches to spam filtering are for email spam filtering [17]. The so-called frequency-inverse document frequency method, in which emails are weighted by examining them one by one, is the most basic approach used for this problem [18]. Statistical or machine learning (ML) models are then used to determine whether a message is spam or benign using training data [19]. The modelling takes into account not only the content of spam messages, but also the style characteristics. In addition, although spam emails and spam SMS messages have many similarities, SMS messages are shorter, contain less contextual information, and use a lot of abbreviations and jargon [20]. The official restrictions in countries have a problematic structure for content-based classification because of phonetic abbreviations, lack of or poor punctuation, use of emotional symbols, etc. [21].

In the future, the next generation of mobile technologies will have a much higher security risk. Rule-based or content-based systems are recommended to prevent or filter the proliferation of large volumes of spam SMS messages. Many of the available techniques work with spam filters. In this technique, different parts of the messages are examined to detect spam and decide whether they are spam or not. In particular, rule-based systems are used by many software companies that provide security services [22]. Similarly, studies on rule-based classification have been carried out by brief researchers [23, 24] in news services. The rule-based approach is an efficient method, but it is necessary to constantly create and update large amounts of rules to maintain high filtering accuracy. As the number of rules increases, it becomes more difficult to check a text message against all the rules, and this takes more and more time each day. This aspect will reduce the efficiency of the developed system [25]. Therefore, it is necessary to replace or support rule-based systems with content-based systems to filter malicious messages. This will improve both the classification accuracy and the efficiency of the system [22]. On the other hand, content-based security systems have attracted much attention in recent years and are preferred as the main method for detecting spam SMS [26]. Recently, several hybrid methods have been proposed for SMS spam detection [27, 28]. However, the accuracy is still relatively low and research is still needed to propose different computational and classification approaches using new features.

The aim of this study is to create a dataset with more samples in SMS classification, to improve the performance of the existing SMS classification and to reduce the analysis time per sample. To this end, the main research questions are as follows:

1- Is it possible to create a dataset collection with more samples for SMS classification?

2- Does a feature vector with more features contribute positively to the classification?

3- Can the performance of SMS classification be further improved with a new classifier to be designed?

4- Can the analysis time per application be reduced?

The results of the studies carried out according to these objectives and questions are presented in this article. It presents a comparative analysis of the hybrid feature vector and stacking ensemble model, created by combining feature vectors extracted by different methods, with machine learning models and standard feature extraction methods. First, 4 different benchmark datasets were analysed and combined. Feature vectors were then extracted from these datasets using PVDM and DBOW networks. The resulting feature vectors were used in the training and testing processes of the proposed stacking ensemble model. Thus, the most successful feature generation and classification method for SMS classification with unbalanced datasets is revealed. The contributions of this work focus on addressing the above issues by further increasing the accuracy.

The contributions of Hybrid Feature Vector Design and Stacking Ensemble Model are as follows:

- A larger dataset built from 4 benchmark ensembles: BEC, Big NUS, DIT Nus, SMS Spam Collection.

- All necessary preprocessing steps (text cleaning, lemmatisation, stemming, stop word removal, spelling correction, rare and common word removal, etc.) have been applied.

- A proposed hybrid feature vector is obtained by combining the feature vector obtained by training the DBOW and PVDM models.
- A proposed stacked ensemble model with effective optimisation is designed for higher classification performance.
- The maximum classification value is 99.22% with 0.04 standard deviation, which is one of the highest classification values in the literature. A performance increase of 4% was achieved.
- The analysis time per SMS is only 0.087 milliseconds.
- The proposed model is language independent.

The rest of this paper is organised as follows: In Section 2, studies on binary classification of SMS are summarised. In section 3, the dataset, text embedding approaches (PVDM, DBOW), training and testing parameters, materials and methodology are explained. In section 4, the obtained results are presented and discussed in comparison with other studies. In the last section, the proposed study is summarised and information about future studies is given in the light of the limitations encountered during the study.

2 Related works

A substantial corpus of research has been dedicated to the analysis of SMS classification, with the findings being disseminated in the form of systematic reviews. These reviews encompass a wide range of classification methods, feature extraction and selection approaches [29]. The BoW model, a conventional approach employed in SMS spam detection, is deficient in its treatment of word order. The limitation imposed by SMS systems on the number of characters per message renders the distinction of word importance in frequency systems, which count the frequency of a word in the text, impractical. The study by Xia et al. [30] proposes a new method based on the HMM model to solve the low-frequency problem in SMS spam detection using word order information. Tests are performed on the University of California, Irvine (UCI) spam dataset and a separate Chinese SMS spam dataset containing 2000 messages. The experimental results demonstrate that the proposed model can effectively detect spam SMS with high accuracy, irrespective of the language. The proposed method is a language-insensitive SMS classification method that uses HMM. In tests on the Chinese SMS dataset, the proposed model demonstrated an accuracy of 95.9%. Xia et al. [22] proposed a Hidden Markov Model (HMM)-based SMS classifier design. The method utilises weighted features of tag words in SMS. In tests on the UCI dataset, which is one of the datasets utilised in this study, the HMM was shown to outperform the LSTM model and achieve classification performance comparable to that of the CNN. Convolutional neural networks (CNNs) and long short-term memory (LSTM) models have been proposed by Roy et al. [31] for highly accurate spam SMS detection. The proposed model is based on text data and extracts the feature set itself. The efficacy of the proposed model was evaluated through a series of tests conducted on 5574 messages, resulting in a classification rate of 97.7%, signifying its notable performance in accurately identifying spam SMS messages.

Sjarif et al [32] used TF-IDF features to extract frequencies in SMS messages and performed classification with RF. Fewer features and smaller messages were analysed. Several classifiers are combined with the voting model. With this hybrid classifier including RF, 97.5% success was achieved. Liu et al [33] proposed a transformer model to classify SMS spam messages. ML models were used in the tests using the SMS Spam Collection v.1 dataset. In the tests, the Transformer model achieved 98.92% success, while the F-score value was 96.13. Ebadati et al [34] proposed a more efficient spam detection system with fewer processes, while selecting features that reduce classification performance. The tests were carried out on samples from Hewlett-Packard (HP) laboratories. Using Bayes' theorem, a genetic algorithm was used to select features without specifying a number. Benign and spam classification was then performed using NB. Similar results were obtained using NB and a hybrid genetic algorithm (GA)-NB. Zhao et al [35] proposed a heterogeneous infrastructure to improve the impact of class imbalance on spam detection in social networks. A cost-sensitive learning model was built using a deep neural network. Different costs of misclassification were specified and the prediction results were dynamically adjusted. Experimental results showed that the proposed platform significantly improved the spam detection rate in unstable datasets. Hong et al [36] proposed a multimodal architecture based on model fusion to filter out spam emails hidden in text or sent openly as images. A hybrid structure was created by combining the CNN and LSTM models. The text parts were classified with the LSTM and the image parts with the CNN and a binary fusion model was used. Hyperparameter optimisation was used with grid search and validation was done with k-fold cross validation. The result was spam detection with a higher accuracy than traditional models. Jain et al [37] used the LSTM architecture, which is a type of recurrent neural network (RNN) used for spam classification. The feature generation has the ability to learn abstract features, not manually. The vectors needed for classification were generated using word2vec, wordnet and conceptnet. Classification results were compared with SVM, NB, ANN, k-NN and random forest. Experiments were conducted on the SMS spam collection and the Twitter dataset. The results showed that the LSTM model was more successful in detecting spam than the traditional methods. Srinivasarao et al [38] processed the dataset with word2vec for SMS classification and started the processing with data augmentation. Then six different feature selection methods were fed into an equilibrium optimisation. A hybrid classifier was designed using KNN and SVM. Rat Swarm Optimisation (RSO) was used to parameterise the network and AFINN and SentiWordNet were used for sentiment analysis. After this long process, high performance in SMS classification was achieved. Jain et al [39] proposed an NLP technique using a semantic CNN network and Word2vec word vectors. The concept network searches for similar words in the data for a given word. The model has a success rate of 86.5% on the SMS spam dataset. Sharaff et al [40]

proposed an SMS filtering algorithm inspired by krill swarm optimisation and the biological dendritic cell algorithm. Tests were performed with different ML classifiers such as NB, LR and SVM and the results were compared. The results show that the proposed model is more successful than standard ML models. Gazal [41] proposed a model using a high-level filter, a fuzzy logic-based second-level evaluation layer, and a low-level classifier. The model performs majority voting classification for RF and has high performance. There are also some commercial applications that provide services in this area. Apache SpamAssassin is a platform designed to classify and block spam messages. It is a successful and typical example of rule-based systems [42].

In almost all studies using UCI data, either smote (data balancing) was performed or the performance was too low for spam classification. This is why there is a difference between the accuracy and f-score values. In this study, despite the significant imbalance in the amount of data, no data balancing was performed and the results were designed to be high for both raw and spam messages. High performing models tend to use either a complex classifier design or a combination of attributes. In addition, the studies are mostly based on text classification and NLP techniques. Text-based problems such as spam Twitter messages and spam emails have been studied. The model proposed in this study achieves the highest performance on the UCI dataset compared to the literature.

3 Material and methods

The proposed model for classifying SMS as spam and ham is described in this section. The proposed model structure includes (1) dataset acquisition, (2) preprocessing, (3) feature engineering with text embedding, (4) hybrid feature vector design, (5) machine learning approaches, and (6) three different ensemble model designs (hard voting ensemble, soft voting ensemble, and stacking ensemble). All these steps are visually illustrated in Figure 1. In addition, the details of the proposed ensemble model structures are given in Figure 2.

The flowchart of the model proposed in this study is shown in Figure 1. In the first stage of the proposed model, a large dataset consisting of different datasets was created. By combining 4 different datasets, the aim is to create a broad and homogeneously distributed dataset that contains more samples. Although there is sample imbalance between classes, as in many studies, a SMOTE-like data balancing process was not performed. This is because the reliability of tests with synthetically generated data is reduced. The next step was to perform a series of pre-processing operations on the data. In this step, a number of operations that can be applied to natural language processing problems, such as text cleaning, stemming, lemmatisation, tokenisation, and removal of common and rare words, can be applied. For the model proposed in this study, the operations indicated by the green arrow in the figure provide an improvement in performance, while the operations indicated by the red arrow cause a decrease in performance. The next step is to convert the texts into numerical vectors. This makes it possible to train and test machine learning models. In the literature, feature extraction algorithms such as Bag of Words (BOW), N-gram, TF-IDF, Word2vec (CBOW) and Doc2vec (PVDM) have been used [43, 44]. In general, these features can be used individually or as a hybrid. In this study, a hybrid vector was obtained by combining the feature vectors of the DBOW model and the PVPM model, which achieved the highest performance for the proposed model. Thus, the aim is to obtain more features for each sample text and to capture distinctive features. For this process, training and test data were trained separately with DBOW and PVDM models.

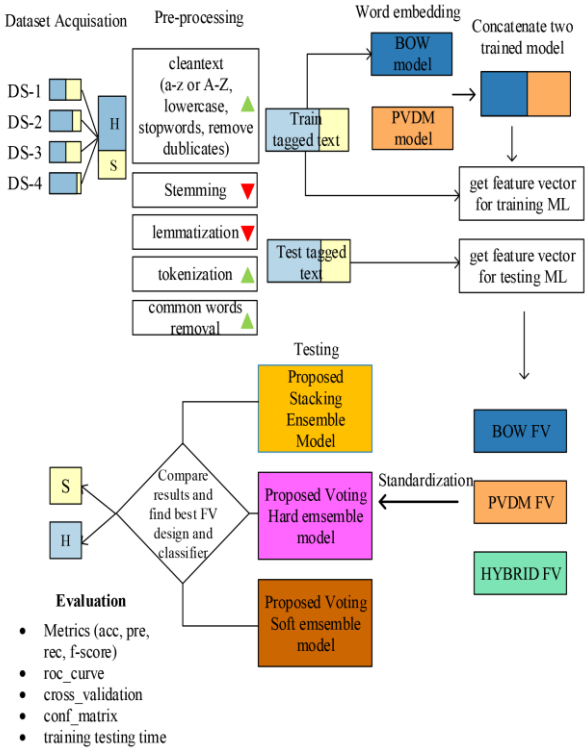

Figure 1. Methodology diagram of the proposed SMS malware detection model

Machine learning or deep learning models can be used to classify the resulting 3 different feature vectors. For this problem, 3 different ensemble models were designed and tested. The structure of the 3 different ensemble models used to solve the problem is shown in Figure 2. A total of 12 different classifiers, of which 3 are boosting ensembles and 2 are bagging ensembles, are used in the design of 3 different ensemble models. Both comparative results were obtained and analysed and the most successful model was revealed. The predictions made with 12 different classifiers in the first layer were subjected to a final evaluation in the second layer and the final decision was determined.

In the voting soft and hard models, all classifiers are included in the voting. In the stacking ensemble model, SVC is used as the decision maker (layer 1). Since the proposed model in this study aims to classify SMS with the highest performance, a large number of comparative tests and hybrid feature structures were used. The test results with the model with the highest performance are presented in the next section.

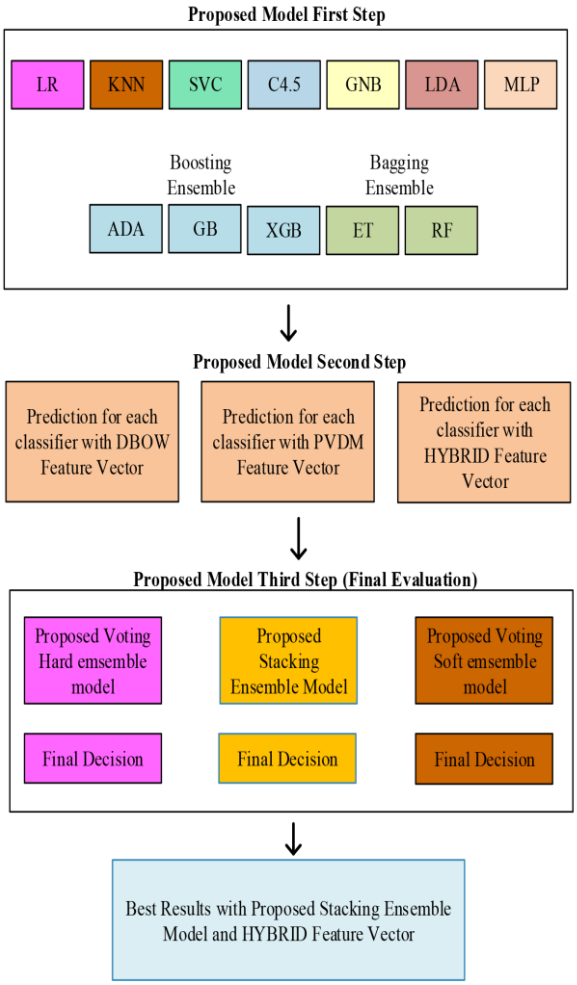

Figure 2. Systematic representation of the proposed ensemble models

3.1 Datasets

SMS classification studies are generally tested using the UCI SMS dataset. The main difference between the studies comes from the feature extraction and classifier design. Accordingly, different classification performances have been obtained. The UCI dataset contains 747 spam and 4827 raw messages. The total number is quite small, and the number of samples belonging to the spam and ham classes is quite unbalanced. In order to find a solution to this problem and to prepare a dataset for further examples, 4 different benchmark datasets in this area, shown in Table-1, were discussed and analysed. The datasets are stored in a text file where each line represents a message. Each row contains a message and information about the class it represents. The issue of user privacy is the most important factor affecting the collection of SMS data and the creation of large datasets worldwide. Most existing SMS datasets are based on the compilation and combination of small SMS datasets. These different datasets, which contain little data, were converted into a common format by performing the necessary pre-processing for this study. This allowed the proposed model to be trained and tested on more comprehensive data.

Table 1. List of datasets

Corpus Name	Number of spam messages	Number of ham messages
DIT (Dublin Institute of Technology) [21]	1351	1353
UCI SMS Spam Collection [45]	747	4827
Corpus v.0.1 big NUS [46]	322	1002
British English Corpus (BEC) [47]	425	450
TOTAL	2845	7632
TOTAL (after remove duplicates)	1811	6660

When the datasets shown in Table-1 were combined into a single dataset, a more comprehensive dataset was created containing 2845 spam and 7623 raw messages. However, when these datasets were examined, it was found that some data was found in more than one dataset. This situation led to data repetition in the dataset and caused the problem of working with repetitive data in the training and testing processes of learning models. To overcome this, it was necessary to clean up the duplicated data and create a dataset in which all samples are unique. As a result of the study carried out for this purpose, a dataset of 1811 spam and 6660 raw messages was created. All testing was carried out on this original dataset.

3.2 Pre-processing

In order to obtain better data processing and results, it is useful to remove some unnecessary information from the SMS. In this study, the pre-processing process included cleaning the dataset, tokenisation, and removal of stop words.

3.2.1 Cleaning Text

Prior to the utilisation of the resulting dataset for classification purposes, a Python library was employed to eliminate extraneous elements such as spaces, new lines, missing values, punctuation marks, special characters, and XML and HTML tags. Given the case sensitivity of certain Python libraries, it was necessary to convert all characters to lower case. Duplicate words were removed. This process resulted in a reduction of both the size of the data and the time required for classification. Stemming and lemmatisation operations, which are frequently employed in NLP studies, were not performed as they have been shown to cause degradation in performance metrics [48]. The binary categorical classification scores, in the form of both raw and spam, were converted to a numerical form using the label-encoding module in the scikit-learn library.

3.2.2 Stop words removal

Stop word removal is one of the most common approaches used in natural language processing [49]. Stop words are considered to be the most common words in any language and in many cases they have a negative impact on text classification and may not contain any semantic information. Removing these words and punctuation from

the text is considered useful for NLP. Informal words, spoken language texts, emoji's, abbreviations, acronyms and many unintelligible words are called stop words. In the tests conducted in this study, it was found that removing stop words had no positive impact on classification, while it had a positive impact on training and testing time. In line with the goal of faster classification, stop word removal was applied to the database using the nltk library in the preprocessing phase.

3.2.3 Tokenization

Tokenisation is the process of extracting words from SMS text [41]. It is a kind of data normalisation process. The Doc2Vec network models used in this study are language independent. In the tokenisation process, there is no problem with western languages such as English and Turkish, as the separation into words is done according to the space character. However, since the sentence structure is different in languages such as Chinese and Japanese, some pre-processing may be required before tokenisation [22]. The pre-processing should be in the form of separation into words and use of punctuation. Separating the text into words and preserving the original order is critical in the tokenisation process. In this study, classification of SMS messages of one word or less was not used. Since the classification was based on the context according to the paragraph, the context could not be determined in one-word messages. The input features were converted to numerical values using the Keras tokeniser. The resulting result was divided by 70%-30% and used in the training and testing process.

3.3 Text representation methods

3.3.1 Paragraph Vector PV-DM (Doc2Vec)

Doc2vec takes the whole sentence as input and transforms it into vectors using the structure shown in Figure 4, where a feature representing the concept of the SMS replaces the word. In the training process, the word vector and the sentence vector are trained. This model is called the PV-DM model. This model structure was generated using the Gensim library for hybrid feature vector generation and benchmarking in our study [50].

3.3.2 DBOW Model (Distributed Bag of Words)

Bag of words (BOW) is one of the most basic approaches to representing data. When training the model, words are scored independently of their position in the text. BOW differs from PV_DM in that it forces the prediction of selected words from a sentence without considering the context [50].

4 Results

4.1 Experimental environment and hyper-parameters

In order to evaluate the effectiveness of the proposed model, the most frequently used 5 datasets were used. It has been shown that the traditional BOW models have some problematic points in the classification of SMS, which contain short texts and are full of idioms, symbols and abbreviations; therefore, their performance may decrease. A similar situation occurs in applications with a character limit, such as instant messaging systems like WhatsApp or Bip, and social networks or forums such as Twitter, Facebook, etc. All possible types of concatenation have been identified in the tests and the tests have been performed with each combination. These algorithms are currently the best available techniques [34]. All the codes for testing the proposed model were written in Python version 3.8.1. The codes handled the reading of the SMS data, the tokenisation, the generation of feature vectors with the Word2Vec network and the classification with 10 different algorithms. All methods are available in the scikit-learn library. The results presented in this study were computed on a desktop computer with an Intel Core i7-9700 CPU 3.0 GHz processor and 8 GB memory.

While the majority of parameters in machine learning models are learned by extracting them directly from the data, it is not possible to automatically learn hyper-parameters in this way, and the state of these parameters directly affects the classification performance, regardless of the complexity of the model. It is difficult to find the most optimal combination of multiple hyper- parameters because there is a wide range of values that multiple parameters can take. The grid search algorithm is a common approach that performs tuning for multiple parameters with a predefined range. This ensures that the best parameters are found. The grid search algorithm was used to select the parameters that would achieve the highest performance for 12 different machine learning models used in this study. After running the algorithm, the parameter details for each classifier and the design of the stacked ensemble model proposed for this study are shown in Table 2.

4.2 Classification results and evaluation

In this study, (1) hybrid feature vector generation and (2) stacking ensemble classifier design are proposed. The aim is to improve the performance of SMS classification independent of the data set. In line with this goal, tests were conducted with a total of 15 different classifiers, including 2 different text embedding approaches and 3 ensemble models, and the results are presented in detail in this section. When analysing the test results given in Table 3 with the feature vector generated with both DBOW and PVDM embedding models, it can be seen that the proposed 3 different ensemble models have a higher performance than the standard machine learning models. This performance is valid for all metrics, including accuracy and AUC. The best values are obtained with the stacking ensemble model structure proposed in this study. This proves that the stacking ensemble model contributes positively to the classification.

Table 2. Hyper parameters of classifiers and details of ensemble models

Classifier	Model Details
LR	C=1000.0, random_state=0, others parameters with default value
KNN	n_neighbors=5, p=2, metric='minkowski', others parameters with default value
SVC	kernel='linear', C=1.0, random_state=0, probability=True, shrink=true, verbose=false, others parameters with default value
C4.5	random_state=0, others parameters with default value
GNB	parameters with default value
LDA	solver=svd, n_components=None, others parameters with default value
MLP	max_iter=1000,activation='relu', alpha= 0.5, hidden_layer_sizes= (10, 20, 10), learning_rate= 'adaptive', solver= 'adam'
Ada Boosting	parameters with default value
GB	loss=logloss, learning_rate=0.01, max_depth=None, others parameters with default value
RF	criterion=entropy, n_estimators=10, random_state=1, others parameters with default value
ET	max_depth=8, others parameters with default value
XGB	earning_rate =0.01, n_estimators=1000, max_depth=5
Proposed Voting ensemble(soft)	Estimator models: LR, RF, DT, MLP, SVC, XGB, GB, GNB, LDA, voting=soft, weights:None, verbose=false, transform=true
Proposed Voting ensemble(hard)	Estimator models: LR, RF, DT, MLP, SVC, XGB, GB, GNB, LDA voting=hard, weights:None, verbose=false, transform=true
Proposed Stacked Ensemble Model LO:L1	LO estimator models: LR, RF, DT, MLP, SVC, XGB, GB, GNB, LDA L1 final estimator model: SVC

Table 3. Classification results with DBOW and PV-DM feature vectors

		DBOW				PVDM			
Type	Algorithm	Acc (%)	Pre (%)	Rec (%)	F-score (%)	Acc (%)	Pre (%)	Rec (%)	F-score (%)
Regression	LR	95.7	95.6	95.7	95.6	95.6	95.6	95.6	95.5
Instance-based	KNN	95.2	95.2	95.2	95.0	94.5	94.7	94.5	94.3
	SVC	95.8	95.8	95.8	95.7	95.7	95.7	95.7	95.6
Decision Tree	C4.5	92.8	92.7	92.8	92.8	90.8	90.7	90.8	90.7
Bayesian	GNB	86.2	88.5	86.2	86.8	90.6	90.8	90.6	89.9
LDA	LDA	94.8	95.1	94.8	94.6	91.8	92.5	91.8	91.1
Artificial Neural Network	MLP	94.8	94.8	94.8	94.8	95.1	95.0	95.1	95.0
Ensemble	ADA	95.0	94.9	95.0	95.0	94.3	94.3	94.3	94.1
	GB	95.5	95.5	95.5	95.4	94.7	94.8	94.7	94.5
	RF	95.8	95.8	95.8	95.7	94.1	94.5	94.1	93.8
	ET	95.0	95.2	95.0	94.8	93.9	94.3	93.9	93.5
	XGB	95.1	95.0	95.1	95.0	95.4	95.5	95.4	95.2
	Proposed Voting ensemble (soft)	96.0	96.0	96.0	95.9	95.4	95.6	95.4	95.3
	Proposed Voting ensemble (hard)	95.9	95.9	95.9	95.8	95.2	95.5	95.2	95.0
	Proposed Stacked Ensemble Model LO:L1	96.1	96.1	96.1	96.1	96.3	96.4	96.3	96.2

Table 4 shows the results obtained for the hybrid feature vector, where 2 different feature vectors are combined to increase the performance even further. While the performance value for accuracy increases by an average of 4%, the increase in the AUC value is about 5%. Thus, with the DBOW+PVDM hybrid feature vector and the designed stacking ensemble classifier, a success rate of 98% and above is achieved.

Table 4. Classification results with proposed hybrid tors

Feature Vector Type	Type	Classification Algorithm	Acc (%)	Pre (%)	Rec (%)	F-score (%)	ROC - AUC
	Regression	LR	98.00	97.99	98.00	97.98	96.10
	Instance-based	KNN	93.87	94.29	93.87	93.49	85.96
		SVC	97.68	97.67	97.68	97.66	95.71
	Decision Tree	C4.5	95.32	95.27	95.32	95.28	92.30
	Bayesian	GNB	89.62	89.30	89.62	89.38	82.59
	Artificial Neural Network	MLP	97.48	97.47	97.48	97.46	95.32
	LDA	LDA	97.37	97.43	97.37	97.31	94.07
Proposed Hibrit FV	Ensemble	Adaboost	97.33	97.31	97.33	97.31	95.28
		GB	97.68	97.69	97.68	97.65	95.31
		RF	97.44	97.49	97.44	97.39	94.38
		ET	96.93	97.05	96.93	96.85	92.94
		XGB	96.11	96.10	96.11	96.03	92.42
		Proposed Voting ensemble(soft)	98.23	98.25	98.23	98.21	96.12
		Proposed Voting ensemble(hard)	98.27	98.29	98.27	98.25	96.28
		Proposed Stacked Ensemble Model LO:L1	98.39	98.39	98.39	98.37	96.81

In order to better evaluate these successful results, ROC curves including all classifiers are obtained for all three models and are shown in Figure 3. According to this, while all machine learning models in general achieve a high AUC value, the stacking ensemble model guarantees the highest value for all three feature vector structures. This is because, as can be seen in the graph, for DBOW the best result is obtained with the gradient boosting classifier, except for the ensemble models, while for PVDM the same machine learning model has a very low value and the best value is obtained with SVM and MLP. On the other hand, for the hybrid feature vector, the best result was obtained with LR. This causes instability in the results obtained and in the choice of classifier. With the stacking ensemble structure, this problem is solved and it is possible to guarantee the best classification success for the whole feature vector.

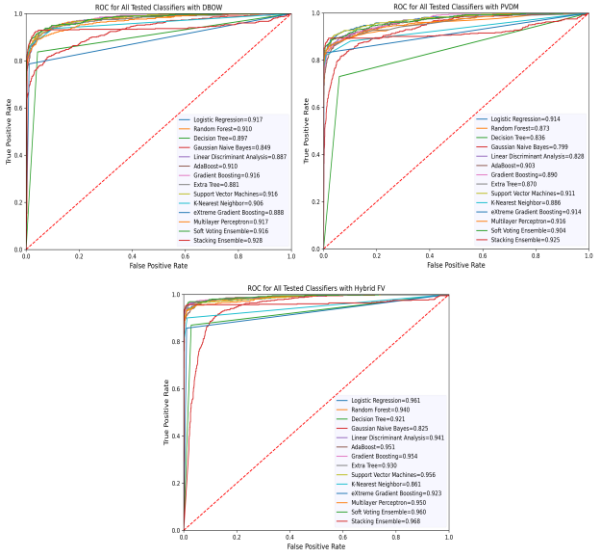

Figure 3. ROC with DBOW, PV-DM and HYBRID feature vectors

The cross-validation graphs of the obtained results are shown in Figure 4, where it is ensured that the results are not random and the results of the stacking ensemble model are repeatedly evaluated for different sample selections (k=10). The lowest standard deviation and the highest average performance are obtained with the proposed hybrid feature vector and the standard deviation is only 0.04.

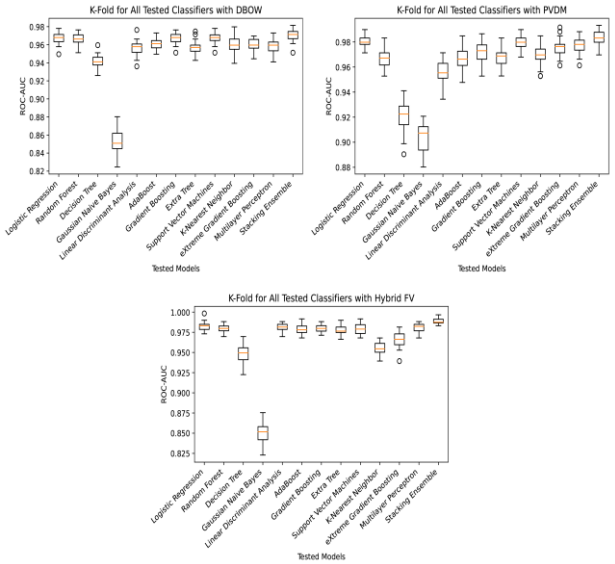

Figure 4. Cross validation with 10 folds for all classifiers and proposed stacking model

The results presented so far in this section show average values for the binary classification problem. Due to the data imbalance between the classes (ham =6660; spam =1811), the results obtained should be evaluated separately on a class basis. For this purpose, the confusion matrices obtained with the stacking ensemble classifier for all three feature vectors are shown in Figure 5. According to this, for the hybrid feature vector, 41 false detections were made in a total of

2544 test samples and the false positives were mostly due to spam sms being detected as ham. It also shows that ham messages can be detected with a very low FP value. When comparing the three vectors, the lowest FP and FN values were obtained with the hybrid feature vector.

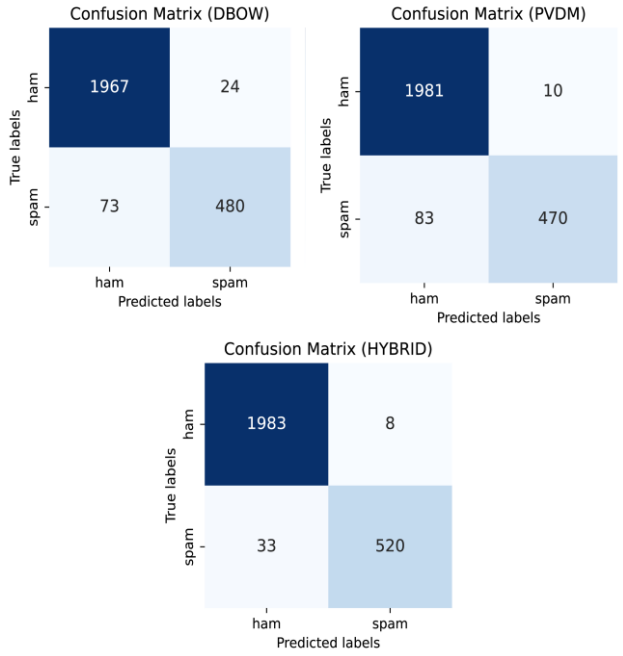

Figure 5. Confusion matrix for proposed stacking model

4.3 Speed evaluation results of the DBOW+DMM

The results presented so far in this section show average values for the binary classification problem. Due to the data imbalance between the classes (ham =6660; spam =1811), the results obtained should be evaluated separately on a class basis. For this purpose, the confusion matrices obtained with the stacking ensemble classifier for all three feature vectors are shown in Figure 5. According to this, for the hybrid feature vector, 41 false detections were made in a total of 2544 test samples. And the false positives were mostly due to spam messages being detected as ham. It also shows that ham messages can be detected with a very low FP value. When comparing the 3 vectors, the lowest FP and FN values were obtained with the hybrid feature vector.

It is clear that the result of this study will be a much larger vector, using feature vectors generated by more than one network. For this reason, it is necessary to avoid an additional load on the system in terms of process and memory, while achieving an increase in performance. Therefore, the values shown in Table 5 are crucial for the model proposed in this study. As a result, very low training and testing times per sample were required. Considering that these tests were carried out on a desktop computer, this is significant. It is assumed that these processes will be completed in a much shorter time, as it is expected that in reality filtering systems will be run on mail or web servers. As a result, although the ensemble model appears to use proportionally more processing time, in terms of time, it requires negligibly less SMS/classification time.

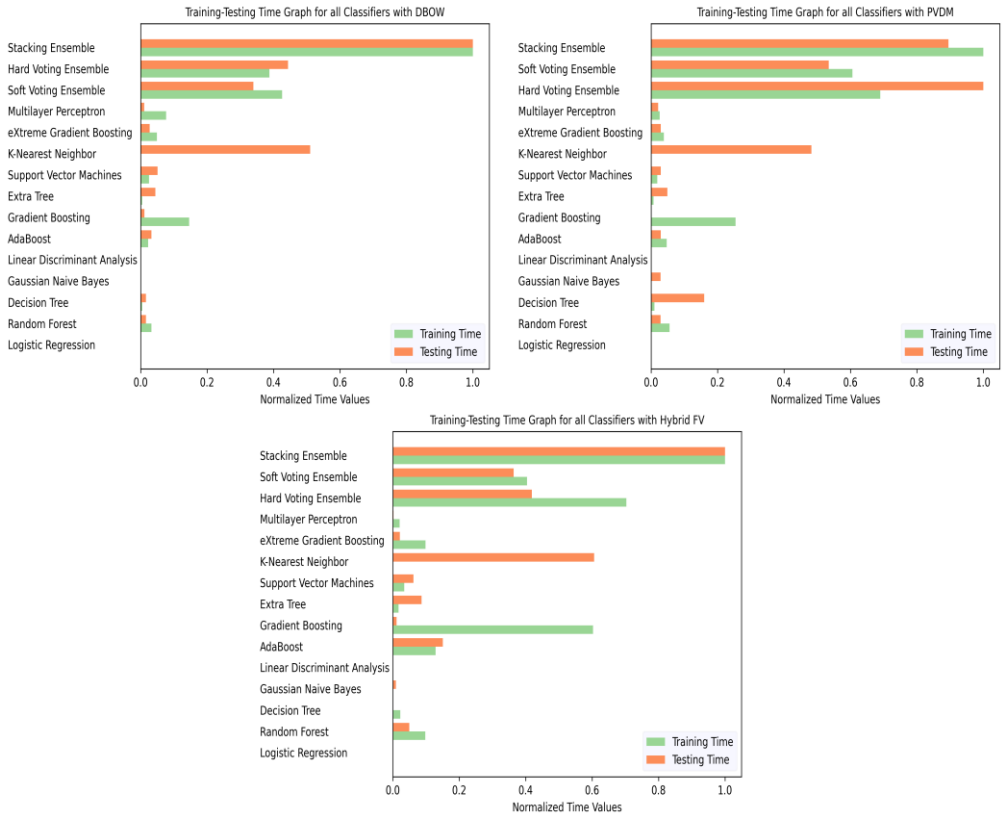

Figure 6. Training-testing time graph for all classifiers with proposed stacking model

Table 5. Execution time for hybrid feature vector and stacking ensemble model

Dataset	Training Time per SMS (millisecond)	Testing time per SMS (millisecond)
BEC Dataset [47]	0.102	9.29
Corpus v.0.1. Big NUS dataset [46]	0.087	8.94
DIT NUS Concatenated Dataset [21]	0.093	8.73
SMS Spam Collection Dataset (UCI) [45]	0.129	10.64
Original Dataset (Collection)	0.087	7.85

5 Discussion

The ML outputs obtained from the experiments performed with 300 different combinations were compared with different metrics. The proposed method generated a more comprehensive vector by combining data from several feature-generating network structures (DBOW, PVDM, HYBRID) and using it for classification. The aim was to achieve a high success rate. As SMS is a text-based communication method that has been used on mobile devices for many years, many security studies have been conducted in this area. The SMS Spam Dataset (UCI) has been used in most of these studies. Table 6 shows some of the studies carried out with this dataset according to the model, type of classification and results using different metrics. Different algorithms were used in the model designs in each study. In the tests carried out on four different datasets, great results were obtained with an average success rate of 98% and above. In addition, the classification success rate increased to 99.22% in the tests performed with the UCI dataset only.

When the studies in the literature are examined, it is seen that deeper neural networks are used as classifiers instead of machine learning in order to achieve high performance in SMS classification. In addition, it is seen that statistically based natural language processing approaches are used in some studies to extract features from SMS texts, while deep learning structures that can directly obtain features are used in most of the remaining studies. While this situation allows high performance to be achieved in terms of accuracy in terms of classification performance, similar values cannot be obtained in terms of other metrics. This situation shows that the models converge to certain classes. Since it is more difficult to determine spam values in terms of SMS than to determine raw values, models tend to include SMSs in the raw class. This situation is seen in almost every study in the literature except our study. In this study, we used the stacking structure to overcome this problem and solved the problem of convergence that different classifiers would make to certain classes alone within the ensemble structure. In this way, we achieved equal and high performance in all metrics together with the accuracy value.

Table 6. Comparison of classification results of papers in literature and proposed stacking ensemble model

Work	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
LSTM[31]	96.7	89.6	84.2	86.8
1CNN [31]	98.2	97.5	89.2	93.1
2CNN[31]	98.2	98.7	89.0	93.3
3CNN[17]	98.6	97.6	91.8	94.6
Bİ-GRU[16]	99.3	99.2	96.2	97.7
TF-IDF+NB[32]	97.0	97.0	97.0	-
TF-IDF+KNN[32]	91.1	92.0	89.0	-
TF-IDF+RF[32]	97.5	98.0	97.0	-
TF-IDF+DT[32]	96.5	96.0	97.0	-
Xia [30]	96.9	93.6	85.0	-
HMM [22]	95.9	89.2	81.6	-
LSTM[37]	99.0	98.7	99.3	99.2
BiLSTM[52]	98.6	96.9	91.7	94.2
TF-IDF+SMOTE+RF[4]	99.0	99.0	95.0	97.0
CNN[31]	97.9	98.8	85.8	92.2
LSTM[31]	95.3	84.9	77.7	81.1
SmishingDetector+NB[20]	91.6	93.0	92.0	92.0
SmishingDetector+RF[20]	82.3	88.0	82.0	83.0
SmishingDetector+DT[20]	88.2	91.0	88.0	89.0
SGNN[53]	98.01	-	-	-
NB[54]	93.9	89.0	84.0	-
Proposed Model with Original Dataset	**98.39**	**98.39**	**98.39**	**98.37**
Proposed Model with Sms spam dataset (similar to literature papers)	**99.22**	**99.22**	**99.22**	**99.22 (0.004 std)**

Although the proposed method has better accuracy, F-score value and AUC value compared to the studies in the literature, it has inaccurate predictions, especially for the spam class. This is due to the removal of words from some messages due to some pre-processing (especially common word removal), which contributes to efficiency and performance. In this case, spam messages are classified as ham. For this reason, assessing the message context for spam messages is considered to contribute to performance, and context-based machine learning models will be used in the future. Another shortcoming of our model is that although a larger sample set was obtained by combining four different datasets, unlike similar studies, tests were still performed

with a limited dataset. In addition, a long preprocessing time is required due to the need for a number of preprocessing steps. In future work, we will focus on applying more practical and less computationally intensive data augmentation methods to solve the class imbalance instead of the SMOTE techniques used in similar work. We will also work on creating a new set of synthetic SMS for a more advanced testbed.

6 Conclusion

SMS is one of the essential communication tools for mobile devices. Therefore, SMS service providers need to design fast and reliable security systems to ensure the security of SMS sending services. In this study, different feature vectors for SMS messages were generated using different text embedding techniques. These vectors were concatenated and used for classification. By concatenating different feature vectors, the aim was to generate more features related to SMS messages and thus increase the classification success. The concatenated vector was tested separately with the UCI, Big NUS, DIT NUS, BEC datasets for different classifiers and the results were evaluated. Furthermore, the results were compared with other studies using the ML technique. In the tests performed with the proposed hybrid feature vector structure and stacking ensemble classifier, an accuracy value of 98.39% is obtained, while the precision, recall and f-score values have the same value. In the UCI dataset, this value increases to 99.22%. It was found that the proposed model has higher accuracy and f-score values than similar studies with very low analysis time per application. Therefore, it is suitable for use on either the client or server side. The model was successful in detecting samples belonging to both spam and benign classes. This work solves important consistency problems in SMS classification with high performance and low analysis speed. The details of the experiments and all the results obtained are presented in the results section. A comprehensive discussion section presents the advantages and disadvantages of the proposed model, along with suggestions for future work.

Future work will focus on faster execution of training and testing and improved classification performance. The speed penalty of working with large feature vectors due to the concatenation of more than one feature set will also be addressed. Experiments will be conducted to ensure the balance between accuracy and speed, and feature selection tests will be performed on feature sets. In addition, other artificial intelligence techniques will be considered along with ML algorithms as classifiers. Research will also be carried out on the pre-processing of interlanguage conversion in studies with languages whose basic structure is not similar to Western languages and whose sentence structure is different. In addition, an attempt will be made to obtain more attributes of SMS with context-based analysis, and classification will be performed with deep learning techniques.

Acknowledgement

This work has been supported by Kayseri University Scientific Research Projects Coordination Unit under grant number #FKB-2022-1092.

Conflict of interest

The authors declare that there is no conflict of interest.

Similarity rate (iThenticate): 12%

References

[1] M. Fallgren, T. Abbas, S. Allio, J. Alonso, G. Fodor, L. Gallo, A. Kousaridas, Y. Li, Z. Li, Z. Li, J. Luo, T. Mahmoodi, T. Svensson and G. Vivier, Multicast and broadcast enablers for high-performing cellular V2X systems. IEEE Transactions on Broadcasting, 65 (2), 454-463, 2019. https://doi.org/10.1109/tbc.2019.2912619.

[2] M. A. Abid, M. F. Mushtaq, U. Akram, B. Mughal, M. Ahmad and M. Imran, Recommending domain specific keywords for Twitter. Advances in Intelligent Systems and Computing, 253-263, 2019. https://doi.org/10.1007/978-3-030-36056-6_25.

[3] G. M. Duc, L. Manh and D. H. Tuan, A novel method to improve the speed and the accuracy of location prediction algorithm of mobile users for Cellular Networks. Journal of Research and Development on Information and Communication Technology, 2 (36), 113, 2017. https://doi.org/10.32913/mic-ict-research-vn.v2.n36.357.

[4] M. A. Abid, S. Ullah, M. A. Siddique, M. F. Mushtaq, W. Aljedaani and F. Rustam, Spam SMS filtering based on text features and supervised machine learning techniques. Multimedia Tools and Applications, 81(28), 39853–39871, 2022. https://doi.org/10.1007/s11042-022-12991-0.

[5] Digital around the world, Global digital insights. https://datareportal.com/global-digital-overview, Accessed 6 May 2023.

[6] R. S. Arslan, E. Ölmez and O. Er, AFWDroid: Deep Feature Extraction and Weighting for Android Malware Detection. Dicle University Journal of Engineering, 12 (2), 237-245, 2021. https://doi.org/10.24012/dumf.875036.

[7] M. Tasyurek and R. S. Arslan, RT-Droid: A novel approach for real-time Android application analysis with Transfer Learning-based CNN Models. Journal of Real-Time Image Processing, 20(3), 2023. https://doi.org/10.1007/s11554-023-01311-w.

[8] R. S. Arslan, FG-Droid: Grouping Based Feature Size Reduction for Android malware detection. PeerJ Computer Science, 8:e1043, 2022. https://doi.org/10.7717/peerj-cs.1043.

[9] E. M. El-Alfy and A.A. AlHasan, Spam Filtering Framework for multimodal mobile communication based on dendritic cell algorithm. Future Generation Computer Systems, 64, 98-107, 2016. https://doi.org/10.1016/j.future.2016.02.018.

[10] S. Ballı and O. Karasoy, Development of content-based SMS classification application by using word2vec-

based feature extraction. IET Software, 13(4), 295–304, 2019. https://doi.org/10.1049/iet-sen.2018.5046.
[11] R. S. Arslan, AndroAnalyzer: Android Malicious Software Detection based on Deep Learning. PeerJ Computer Science, 7:e533, 2021. https://doi.org/10.7717/peerj-cs.533.
[12] G. Waja, G. Patil, C. Mehta and S. Patil, How AI can be used for governance of messaging services: A study on spam classification leveraging multi-channel convolutional Neural Network. International Journal of Information Management Data Insights, 3(1), 100147, 2023. https://doi.org/10.1016/j.jjimei.2022.100147.
[13] Nikolina Cveticanin, What's on the other side of your inbox - 20 spam statistics for 2023. https://dataprot.net/statistics/spam-statistics, Accessed 6 May 2023.
[14] O. Karasoy and S. Ballı, Spam SMS detection for Turkish language with deep text analysis and deep learning methods. Arabian Journal for Science and Engineering, 47(8), 9361–9377, 2021. https://doi.org/10.1007/s13369-021-06187-1.
[15] S. Mannheimer, USA text message statistics updated for 2023, SMS Comparison. https://www.smscomparison.com/mass-text-messaging/2021-statistics/, Accessed 6 May 2023.
[16] T. Xia and X. Chen, Category-learning attention mechanism for short text filtering. Neurocomputing, 510, 15–23, 2022. https://doi.org/10.1016/j.neucom.2022.08.076.
[17] S. Rao, A. K. Verma and T. Bhatia, Hybrid ensemble framework with self-attention mechanism for social spam detection on Imbalanced Data. Expert Systems with Applications, 217, 119594, 2023. https://doi.org/10.1016/j.eswa.2023.119594.
[18] W. H. Park, I. F. Siddiqui, C. Chakraborty, N. M. Qureshi and D. R. Shin, Scarcity-aware spam detection technique for Big Data Ecosystem. Pattern Recognition Letters, 157, 67–75, 2022. https://doi.org/10.1016/j.patrec.2022.03.021.
[19] R. Kiran, P. Kumar and B. Bhasker, OSLCFIT (organic simultaneous lstm and cnn fit): A novel deep learning based solution for sentiment polarity classification of reviews. Expert Systems with Applications, 157, 113488, 2020. https://doi.org/10.1016/j.eswa.2020.113488.
[20] S. Mishra and D. Soni, SMISHING detector: A security model to detect smishing through SMS content analysis and URL behavior analysis. Future Generation Computer Systems, 108, 803–815, 2020. https://doi.org/10.1016/j.future.2020.03.021.
[21] S. J. Delany, M. Buckley and D. Greene, SMS spam filtering: Methods and Data. Expert Systems with Applications, 39(10), 9899–9908, 2012. https://doi.org/10.1016/j.eswa.2012.02.053.
[22] T. Xia and X. Chen, A weighted feature enhanced Hidden Markov model for spam SMS filtering. Neurocomputing, 444, 48–58, 2021. https://doi.org/10.1016/j.neucom.2021.02.075.
[23] S. Abiramasundari, V. Ramaswamy and J. Sangeetha, Spam filtering using Semantic and Rule Based model via supervised learning. Annals of the Romanian Society for Cell Biology, 25(2), 3975-3992, 2021.
[24] S. Bhatnagar and A. Kumar, A rule-based classification of Short Message Service Type. 2018 2nd International Conference on Inventive Systems and Control (ICISC), pp. 1-4, Coimbatore, India, 2018. https://doi.org/10.1109/icisc.2018.8398982.
[25] T. Xia, A constant time complexity spam detection algorithm for boosting throughput on rule-based filtering systems. IEEE Access, 8, 82653–82661, 2020. https://doi.org/10.1109/access.2020.2991328.
[26] M. Novo-Lourés, D. Ruano-Ordás, R. Pavón, R. Laza, S. Gómez-Meire and J. R. Méndez, Enhancing representation in the context of multiple-channel spam filtering. Information Processing & Management, 59(2), 102812, 2022. https://doi.org/10.1016/j.ipm.2021.102812.
[27] S. D. Gupta, S. Saha and S. K. Das, SMS SPAM detection using machine learning. Journal of Physics: Conference Series, 1797(1), 012017, 2021. https://doi.org/10.1088/1742-6596/1797/1/012017.
[28] V. Gupta, A. Mehta, A. Goel, U. Dixit and A. C. Pandey, Spam Detection Using Ensemble Learning. Harmony Search and Nature Inspired Optimization Algorithms, 661–668, 2018. https://doi.org/10.1007/978-981-13-0761-4_63.
[29] O. Abayomi-Alli, S. Misra, A. Abayomi-Alli and M. Odusami, A review of soft techniques for SMS SPAM classification: Methods, approaches and applications. Engineering Applications of Artificial Intelligence, 86, 197–212, 2019. https://doi.org/10.1016/j.engappai.2019.08.024.
[30] T. Xia and X. Chen, A discrete hidden Markov model for SMS spam detection. Applied Sciences, 10(14), 5011, 2020. https://doi.org/10.3390/app10145011.
[31] P. K. Roy, J. P. Singh and S. Banerjee, Deep learning to filter sms spam. Future Generation Computer Systems, 102, 524–533, 2020. https://doi.org/10.1016/j.future.2019.09.001.
[32] N. N. A. Sjarif, N. F. M Azmi, S. Chuprat, H. Sarkan, Y. Yahya and S.M. Sam, SMS SPAM message detection using term frequency-inverse document frequency and random forest algorithm. Procedia Computer Science, 161, 509–515, 2019. https://doi.org/10.1016/j.procs.2019.11.150.
[33] X. Liu, H. Lu and A. Nayak, A Spam Transformer Model for SMS Spam Detection. IEEE Access, 9, 80253-80263, 2021. https://doi.org/10.1109/ACCESS.2021.3081479.
[34] O. M. Ebadati and F. Ahmadzadeh, Classification spam email with elimination of unsuitable features with hybrid of ga-naive Bayes. Journal of Information and Knowledge Management, 18(01), 1950008, 2019. https://doi.org/10.1142/s0219649219500084.
[35] C. Zhao, Y. Xin, X. Li, Y. Yang and Y. Chen, A heterogeneous ensemble learning framework for SPAM detection in social networks with Imbalanced

Data. Applied Sciences, 10(3), 936, 2020. https://doi.org/10.3390/app10030936.
[36] Y. Hong, Q. Liu, S. Zhou and Y. Luo, A spam filtering method based on multi-modal fusion. Applied Sciences, 9(6), 1152, 2019. https://doi.org/10.3390/app9061152.
[37] G. Jain, M. Sharma and B. Agarwal, Optimizing Semantic LSTM for spam detection. International Journal of Information Technology, 11(2), 239–250, 2018. https://doi.org/10.1007/s41870-018-0157-5.
[38] U. Srinivasarao and A. Sharaff, Machine intelligence based hybrid classifier for spam detection and sentiment analysis of SMS messages. Multimedia Tools and Applications, 82, 31069-31099, 2023. https://doi.org/10.1007/s11042-023-14641-5.
[39] G. Jain, M. Sharma and B. Agarwal, SPAM detection on social media using semantic convolutional neural network. International Journal of Knowledge Discovery in Bioinformatics, 8(1), 12–26, 2018. https://doi.org/10.4018/ijkdb.2018010102.
[40] A. Sharaff, C. Kamal, S. Porwal, S. Bhatia, K. Kaur and M. M. Hassan, SPAM message detection using danger theory and krill herd optimization. Computer Networks, 199, 108453, 2021. https://doi.org/10.1016/j.comnet.2021.108453.
[41] Gazal and K. Juneja, Two-phase fuzzy feature-filter based hybrid model for Spam Classification. Journal of King Saud University - Computer and Information Sciences, 34(10), 10339–10355, 2022. https://doi.org/10.1016/j.jksuci.2022.10.025.
[42] SpamAssassin, Apache Software Foundation. https://spamassassin.apache.org/, Accessed 01 January 2025.
[43] R. S. Arslan, Kötücül web sayfalarının tespitinde doc2vec modeli ve makine öğrenmesi yaklaşımı. European Journal of Science and Technology, 27, 792-801, 2021. https://doi.org/10.31590/ejosat.981450.
[44] R. S. Arslan, Kötücül URL filtreleme için derin öğrenme modeli tasarımı. European Journal of Science and Technology, 29, 122-128, 2021. https://doi.org/10.31590/ejosat.1011961.
[45] T. A. Almeida, J. M. G. Hidalgo and A. Yamakami, Contributions to the study of SMS Spam Filtering. Proceedings of the 11th ACM Symposium on Document Engineering, pp. 259-262, California, USA, 2011. https://doi.org/10.1145/2034691.2034742.
[46] G. V. Cormack, J. M. G. Hidalgo and E. P. Sánz, Spam filtering for short messages. Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 313-320, New York, USA, 2007 . https://doi.org/10.1145/1321440.1321486.
[47] M. T. Nuruzzaman, C. Lee and D. Choi, Independent and personal SMS spam filtering. 2011 IEEE 11th International Conference on Computer and Information Technology, pp. 1-7, Paphos, Cyprus, 2011. https://doi.org/10.1109/cit.2011.23.
[48] S. Rao, A. K. Verma and T. Bhatia, A review on Social Spam Detection: Challenges, open issues, and Future Directions. Expert Systems with Applications, 186, 115742, 2021. https://doi.org/10.1016/j.eswa.2021.115742.
[49] O. N. Akande, O. Gbenle, O. C. Abikoye, R. G. Jimoh, H. B. Akande, A. O. Balogun and A. Fatokun, SMSPROTECT: An automatic smishing detection mobile application. ICT Express, 9(2), 168–176, 2023. https://doi.org/10.1016/j.icte.2022.05.009.
[50] Q. Le and T. Mikolov, Distributed Representaions of Sentences and documents, International conference on machine learning, pp. 32(2):1188-1196, Beijing, China, 2014.
[51] M. V. C. Aragão, E. P. Frigieri, C. A. Ynoguti and A. P. Paiva, Factorial design analysis applied to the performance of SMS anti-spam filtering systems. Expert Systems with Applications, 64, 589–604, 2016. https://doi.org/10.1016/j.eswa.2016.08.038.
[52] O. Abayomi-Alli, S. Misra, A. Abayomi-Alli and M. Odusami, A deep learning method for automatic sms spam classification: Performance of Learning Algorithms on Indigenous Dataset. Concurrency and Computation: Practice and Experience, 34(17), 1-15, 2022. https://doi.org/10.1002/cpe.6989.
[53] W. Pan, J. Li, L. Gao, L. Yue, Y. Yang, L. Deng and C. Deng, Semantic Graph Neural Network: A conversion from spam email classification to Graph Classification. Scientific Programming, 2022:6737080, 1–8, 2022. https://doi.org/10.1155/2022/6737080.
[54] H. H. Mansoor and A. P. Shaker, Using classification techniques to SMS spam filter. International Journal of Innovative Technology and Exploring Engineering, 8(12), 1734–1739, 2019. https://doi.org/10.35940/ijitee.l3206.1081219.