IJWR

Persian SMS Spam Detection using Machine Learning and Deep Learning Techniques

Roya Khorashadizadeh[a], Somayyeh Jafarali Jassbi*[a], Alireza Yari[b]

[a]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran; roya.khorashadi@gmail.com, S.jassbi@srbiau.ac.ir*

[b]Iran Telecom IT Research Faculty, ICT Research Institute Tehran, Iran; a_yari@itrc.ac.ir

ABSTRACT

Spams are well-known examples of unsolicited text or messages which are sent by unknown individuals and cause issues for smartphone users. The inconvenience imposed on users, the loss of network traffic, the rise in the calculated cost, occupying more physical space on the mobile phone, and abusing and defrauding recipients are but a few of their downsides. Consequently, the automated identification of suspicious and spam messages is undoubtedly vitally important. Additionally, text messages which are smartly composed might be difficult to recognize. However, the present methodologies in this subject are hindered by the absence of adequate Persian datasets. A huge body of research and experiments has revealed that techniques based on deep and combined learning are superior at identifying unpleasant text messages. This work sought to develop an effective strategy for identifying SMS spam through utilizing combining machine learning classification algorithms together with deep learning models. After applying preprocessing on our gathered dataset, the suggested technique applies two convolutional neural network layers, the first of which being an LSTM layer, and the second one which is a fully connected layer to extract the data characteristics, thereby implementing the suggested deep learning approach. As part of the Machine Learning methodologies, the vector support machine makes use of the data and features at hand to determine the ultimate classification. Results indicate that the suggested model is implemented more effectively than the existing techniques, and an accuracy of 97.7% was achieved as a result.

Keywords: ***SMS Spam, Spam Detection, Support Vector Machine, Convolutional Neural Network ,LSTM.***

1. Introduction

Any nonrelevant text message is called spam SMS, which are normally sent by mobile networks. These users share unwanted content, in which there are messages that take up space in the device memory and defraud individuals by Scamming [1]. Some of these SMS Spam messages deceit consumers, resulting in account information loss and privacy breaches. Multiple approaches have been suggested to identify such websites, emails, and SMSs [2]. Yet, the frequency of attacks is still on the rise which has turned them into a pervasive problem. Consequently, it is undeniably a necessary requirement to develop systems by which spams can be detected.

Due to the usage of linguistic properties, the language used in text messages is among the significant obstacles to spam detection. As a result, spam detection methods vary by language, with the majority of prior research focusing on English while other languages have not been investigated and Persian is no exception. This indicates that Persian has fewer standard datasets. Other obstacles include the detection of SMS spam cleverly created. This means that systems that find spams by taking into account the presence or absence of certain phrases in these texts, such as winner, gift, click, etc. may be readily tricked by spammers.

After reviewing previous research conducted on the identification of SMS spams in Persian, our aim was to provide a novel hybrid model on the basis of deep and machine learning techniques in the present work. The general characteristics of the suggested technique will be discussed in what follows.

In this research, while taking into account the various studies for spam detection, specifically SMS Spams in Persian, we will strive to offer a new multi-part model which is consist of deep and machine learning methods. Therefore, a general picture of this approach will be presented in this article.

2. Related Works

Roy et al. [3] introduced a deep learning method in order to distinguish annoying text messages from non-annoying ones. Researchers have examined the function of two deep learning algorithms (convolutional networks and LSTM) on a standard dataset. This dataset contains 747 spam data and 4827 non-spam SMS. Analysis of the results in the accuracy evaluation index indicates that LSTM with 99.4% accuracy has better performance than convolutional networks.

Gadde et al. [4] they examined the function of a wide variety of machine learning and deep learning algorithms to facilitate the process through which spams are detected. The standard set of data published in the UCI database is used for this purpose. The results show that these studied methods of LSTM algorithm with 98.5 accuracy have the best performance in detecting annoying SMS.

doi ijwr.2022.348824.1126 20.1001.1.26454335.2022.5.1.8.1

Citation. R. Khorashadizade, S. Jafarali Jassbi, and A. Yari, " Persian SMS Spam Detection using Machine Learning and Deep Learning Techniques", *International Journal of Web Research*, vol.5,no.1, pp.56-66, 2022.

**Coressponding Author*

Article History: Received:24 June 2022; Revised: 10 August 2022; Accepted: 21 August 2022

Tekerek et al. [5] tried to distinguish SMS Spam from Non-Spam by using data mining algorithms. For this purpose, they evaluated the performance of algorithms such as Bayesian, K Nearest Neighbors, Random Forest, Decision Tree, and Support Vector Machine. Dataset that used in this research includes 747 annoying text messages and 4827 non-annoying text messages. The results show that the SVM algorithm was more accurate than other algorithms in detecting SMS Spam.

Ballı et al. [6] suggested a novel technique for identifying spam that combines deep learning methodologies and simple classification algorithms. In this method, textual properties associated with messages are retrieved by the aid of the Word2Vec algorithm and Convolutional networks, and the diagnosis is made using fundamental classification methods. The findings demonstrate that the suggested method achieved an accuracy of 99.6%, hence better results in comparison with other algorithms.

3. Proposed Methodology

In this study, we suggested a strategy that combines deep learning and machine learning to identify SMS spam. CNN and LSTM extract features, followed by classification using the SVM method. CNN and LSTM networks serve the purpose of extracting optimum features, and resulting data demonstrates that the proposed method has proven to be superior.

Each phase of the proposed technique is seen in Fig. 1, and is described below. In the present research, a hybrid approach, consist of deep learning and machine learning has been offered to detect SMS Spams. CNN and LSTM are charged with the responsibility for feature extraction, then classification with SVM algorithm. CNN and LSTM networks are used to take out optimal representation, and the outcomes reveal that our proposed complex model yields better performance.

Figure 1 shows the steps and modules of our method; each phase is demonstrated in details.

3.1 Pre-processing:

In this stage, through preprocessing activities, the obtained data is altered so that it can be analyzed. Basic preprocessing consists of:

- ***Punctuation Removal:***

In this step, unnecessary text symbols such as emojis, hyperlinks, punctuation, numbers and Whitespace are removed. Because these signs and symptoms do not help identify the content of spam.

- ***Text Normalizing:***

Sometimes the characters used in two identical words are different; this causes that words with different spellings to be considered as two different words when counting words. To prevent this problem, we need a tool to standardize the text database. For example, the prefix "می" and the suffix "ها" at the beginning and end of words, respectively, may cause a word to appear in three different ways. Which is shown in Tables 1 and 2.

- ***Tokenization:***

All text processing methods require identifying sentence boundaries to distinguish them.

Most of the time, sentence boundaries are determined by examining the separating marks. Symbols used to define sentence boundaries in this article are «.» ،«؟» ،«!» ،«؛» ، «?»And ":". Also identify words by examining symbols such as space, tab, new hyphen, ",", ",", ">", "<", "]", "[", "-", "_" and "/" will be done.

- ***StopWords Removal:***

Stop words are words that are repetitive but do not matter. These words, despite appearing in most documents and their many repetitions, lack semantic information such as: از, با, در, به, برای, اگر and so on. In most text applications, deleting these words greatly improves processing results and reduces computational load and speeds up. For this reason, these words are often removed in the preprocessing phase.

- ***Lemmatizing:***

At this stage, the separated words become the root of the word. In the Lemmatization operation, as opposed to Stemming, the word is always converted to the base form and the correct spelling. To achieve this, the basic form of the word is extracted based on the relevant reference dictionary. Converting similar words to the root reduces the complexity of the problem and achieves higher final accuracy.

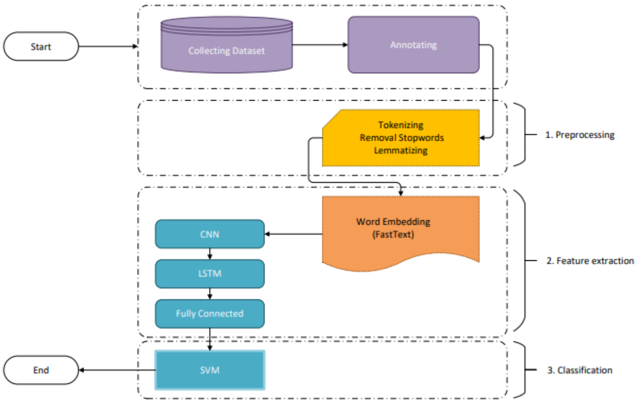


Figure. 1. The chematic of the proposed method

Table 1. Various Writings for a Word in Persian

Stick together	***Separated by space***	***Separated by half space***
کتابها	کتاب ها	کتاب‌ها
میرود	می رود	می‌رود

Table 2. An Example of Different Writing Modes in Persian

The certain word	***First state***	***Second state***
مسئول	مسؤول	مسوول
مجموعه ی	مجموعۀ	مجموعه
پاییز	پائیز	-

3.2 Word Embedding:

Turning to the dataset at hand, word vectors are initially embedded, meaning each line is a representation associated with a specific word. The aforementioned vector could be a one-hot representation of a certain word, or word embedded by well known word embedding models, such as word2vec, GloVe or FastText.

In this research, the FastText method has been used, which is less sophisticated in terms of computations compared with other approaches [7].

In the current dataset, the word vectors are first introduced, with each line having a vector representing a thoroughly distinct word. This vector may be a one-hot representation or a word with fewer dimensions using word2vec, GloVe, or FastText.

FastText is a character-level while word2vec is a word-level method. In word2vec, the embedded vector is extracted for each individual word, but in FastText, vectors are from N-gram characters, which we used 5 and 7 gram.

Having investigated the performance metrics of this algorithm in various languages and tasks, the length of the word which embeds the vector for each word in the FastText algorithm is considered 300 [8].

In this work, the length of the word embedding vector for each word in the FastText method is assumed to be 300, based on an evaluation of the algorithm's performance in various texts and languages [8].

Two well-known methods for extract embedding vectors are skip gram and CBOW methods. In each of these methods, the representation of one or more words is considered as input. The error is calculated with the error function, and the word weights are updated.

In the CBOW method, the goal is to obtain the central word by knowing the words before and after. While in the skip gram method, we have a word and try to reconstruct the words before and after. We used Skip-gram to extract Word embedding, which is more accurate than CBOW.

The difference between the two methods mentioned in the Figure 2 is significant.

3.3 Feature Extraction:

- ***Convolutional Neural Networks (CNN):***

Convolution networks has shown a descent performance in feature extraction; and this is why the embedding vector of text messages is provided to the convolution network so that the most effective features can be extracted (Figure 3). Two CNN layers are chosen in the proposed method, since utilizing any more than two of these layers exacerbates the model's complexity without enhancing the outcomes. Additionally, a network with less than two layers cannot extract the best properties.

Length of text message embedding vectors are 300; to view the vectors in a two-dimensional space, they are divided into 10 rows and 30 columns. It has to be noted that the matrix dimensions must be compatible with the number of layers to which convolution filters are to be applied; since max pooling applies to each layer of convolution and reduces the matrix to half of its initial size.

A further course of action to take with regard to text messages is padding (Figure 4). Padding operation means introducing a number of additional layers with zero value around the feature vector. In this method, 64 filters with a size of 3 × 3 are made for convolutional networks. A thorough investigation of the extensive work done in this field has revealed that lower values for the number of aforementioned filters reduces the number of extraction features and model efficiency [9]. Furthermore, the higher concentration of filter kernels increase the number of channels, leading to more computational complexity of the model [10]. Once the filters are provided with the convolution operator, which is applied to the output of the Max pooling layer, the main purpose of this operation is to keep the number of parameters lower and prevent Overfitting.

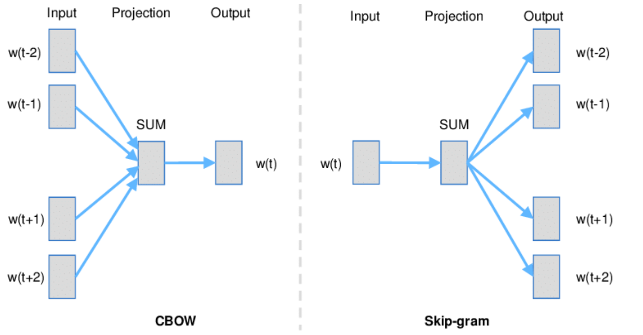


Figure. 2. CBOW vs Skip-gram

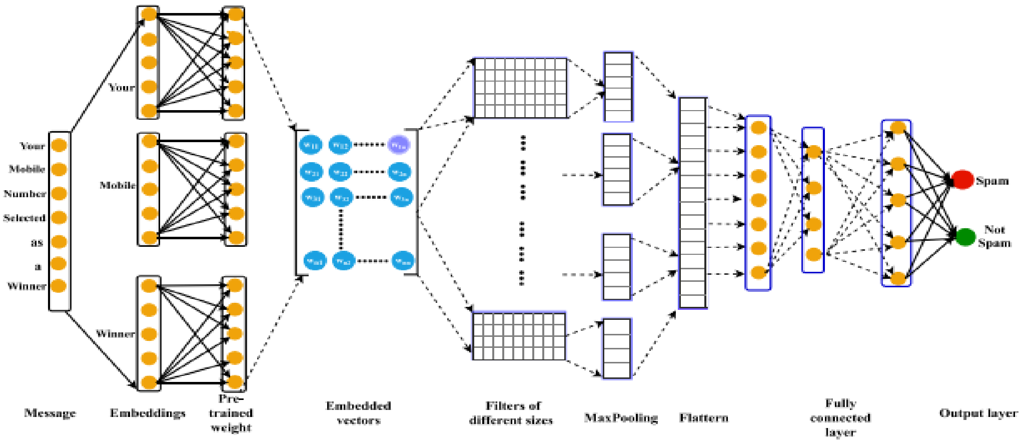


Figure. 3. A framework of convolutional neural network[3]

Padding is another procedure performed on text messages (Figure 4). To perform a padding operation, extra layers with zero value are added in the vicinity of the feature vector. In the proposed method, convolutional networks are given with 64 filters of size 3×3. A study of prior research demonstrates that a lower value for the number of mentioned filters has an impact on the number of extraction features and the model's efficiency [9]. In addition, having more filters increases the number of features, thereby making the model more complex [10]. As soon as the convolution operator is given to the filters, it is then applied to the output of the Max pooling layer; this operation is conducted to limit the number of parameters and avoid Overfitting.

Polling operations can also be carried out with other operators such as averaging, but maximization has had the most success [11].

For example, Figure.5 show a 4×4 matrix as input and it is divided into 4 parts of 2×2, the maximum value of each which is set as output.

- ***Long short term memory network layer (LSTM)***

In the next stage, the feature vector is taken out from the convolutional networks given as input to the LSTM network. In other words, two CNN and LSTM networks have been made use of to extract the feature. The LSTM layer is able to record the most important information of previous states thanks to its memory. This boosts the efficiency of the network in order to the benefit of the weights. In addition to the current state, the previous state of the network is also taken into account. We utilized an LSTM layer containing 150 LSTM units. This layer uses recursive relationships for the extraction of proper features from the text. The LSTM layer can store information from previous states owing to its memory. Thus the LSTM layer will increases the efficiency of the network because of its native property to model time dependencies. It’s important to notice the role of the previous LSTM state, while calculating the current state and output (Figure.6).

Subsequently, the feature vector is extracted from the input convolutional networks and introduced to the LSTM network. This means that we extracted the feature by using two CNN and LSTM networks. As it has a memory, the LSTM layer may keep information about prior states. This step increases the network's efficiency for the purpose of modifying the weights. In addition to the current state, its previous state of the network is also examined. We used single LSTM layer with 150 units. This layer uses recursive connections to extract relevant text characteristics. Due to its memory, the LSTM layer may store information from prior states. This step improves the network's performance since it change the weights. Regarding the present condition of the network, the previous state is also evaluated (Figure 6).

- ***Fully Connected layer***

It has to be noted that through using fully connected layer, the rate of extracted features utilized for categorization drops. The LSTM layer’s output is thus transmitted to the fully connected layer. Also, there are 25 neurons in the layer of complete connections. Looking at the prior research in the field of deep learning, it can be seen this number of neurons in the fully connected layer may effectively achieve accuracy; moreover, the aforementioned rate of neurons does not inflict a large computational burden [12].

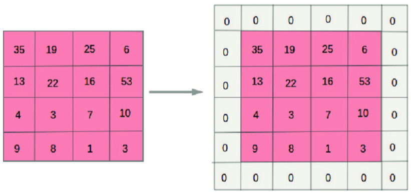
In the model proposed in this article, a convolutional neural network is constructed on our dataset two times, and the results are then transferred to long short-term memory networks. During this stage, the fully connected layer is utilized to subsequently minimize the number of extracted features used for categorization.

Figure 7 depicts how the deep learning component of the suggested technique is structured.

In the following stage, the Adam algorithm is employed to optimize the weights make them up-to-date. It has to be added that other methods to rectify errors, including RMSprop, SGD, NAG, AdaGrad, AdaDelta, AdaMax, etc. have been presented [13]. Adam's method is a popular variant with superior error optimization performance compared to the other algorithms listed.

3.4 Classification:

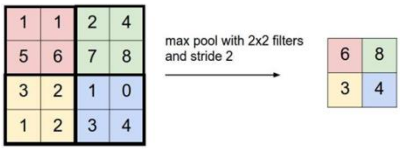
After the procedure of extracting features is completed using the techniques outlined in the preceding phase, the



35	19	25	6
13	22	16	53
4	3	7	10
9	8	1	3

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Figure. 4. Zero padding in the proposed method



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters and stride 2

6	8
3	4

Figure. 5. Max pooling

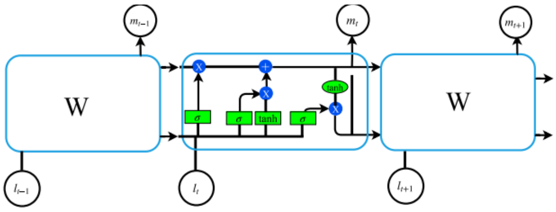


Figure. 6. A model of LSTM network[3]

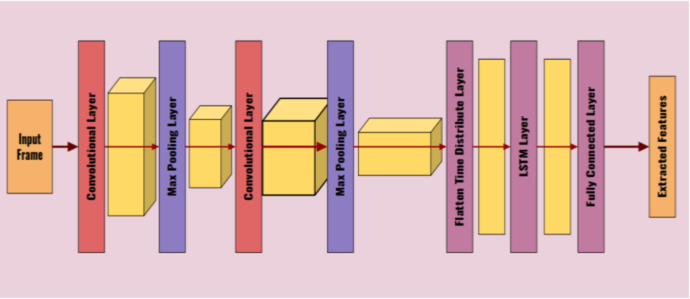


Figure. 7. The structure of the deep learning model used in proposed method

classifier was provided with these characteristics to execute the classification. As described earlier, the classification task refers to the binary classification of SMS spam and Non- spam classes.

The Vector Machine method is based on the categorization of linear data; therefore, in distinguishing data, it attempts to choose a line with a higher success rate. QP approaches, renowned methods for the purpose of solving finite problems [14], are used to solve the equation and obtain the best line for the data.

The proposed approach for identifying SMS spam employs the support vector machine classifier, which obtains a pattern from the training set and then executes the detection procedure based on the it. In the categorization part, a fully connected layer is used draw comparisons. In this layer, the activatation function is Softmax.

The whole vector is introduced into the Softmax function and a k-dimensional vector containing real values, including z, is accepted as input. Furthermore, the next k-dimensional vector containing integers in the range [0,1] is taken as the output, so that the sum of these numbers is precisely equal to 1. This means that this function assigns probability values to the input numbers.

This operation is represented mathematically as Eq.(1):

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_k^K e^{Z_k}} \qquad (1)$$

Where j = 1, 2, 3, ... K in conjunction with the denominator of the fraction normalize the numbers.

4. Experimental Analysis

Bearing in mind the description of the process in the previous section, CNN and LSTM were used to extract the optimum features, followed by the SVM algorithm for classification. CNN and LSTM networks are used to extract the best features, and the outcome demonstrate that the suggested technique is superior.

Due to Python's diverse libraries and easy syntax, the suggested model is emulated in its software environment. Facebook's FastText libraries and the Numpy library were used to generate the semantic vector. Also, Keras is used to implement the portion on deep learning, whereas Scikit Learn is used to implement the piece on machine learning.

4.1 Evaluation Metrics:

- ***Confusion Matrix:***

Before introducing evaluation criteria, it is necessary to introduce the confusion matrix. Which is commonly used for supervised learning classification algorithms. In the field of artificial intelligence, the confusion matrix is called a matrix that is formed based on the output of the classifier.

Table 3 shows the possible states for the output of a classification. For example, it shows the number of samples that are actually members of the positive class and are properly classified in the positive class (TP).

In fact, this matrix provides comprehensive information on how classifier works.

The performance metrics of accuracy, precision, Recall, and F score are individually analyzed.

Accuracy: Accuracy is a statistic that represents the performance of the model across with respect to all classes. This criterion is the most frequently used assessment criterion for classification algorithms, indicating the degree to which a classifier is accurate while carrying out the classification. It's mathematical equation is as follows in Eq.(2).

$$Accuracy = \frac{TP + TN}{Tp + TN + Fp + FN} \qquad (2)$$

Precision: A descriptive criterion of inherent measurement errors indicates how near the findings are to the real value. It is a fraction with its numerator being the number of real positives and denominator being the overall number of positive predictions as indicated in Eq.(3).

$$precision = \frac{TP}{TP + FP} \qquad (3)$$

Recall: It is the proportion of positive samples accurately categorized as positive relative to the overall number of positive samples. The recall reflects the model's accurate detection (Eq.(4)).

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{Tp} + \mathrm{FN}} \qquad (4)$$

F1-Score: It is the harmonic mean of the precision and recall (Eq. (5)).

$$F1 = 2 \times \frac{precision + Recall}{precision + Recall} \qquad (5)$$

- ***Dataset:***

Due to regulatory prohibitions in the majority of nations, mobile operators cannot disclose consumer SMS. As a result, the database was compiled with the assistance of individuals from various backgrounds and occupations using online forms and backups of users' text messages, resulting in very diverse text message content. The bank of data in this research comprises 6317 Persian text messages, of which 4411 are not spam and 1906 are spam. In addition, this dataset was annotated by five individuals with diverse preferences who were instructed to label non-spam text messages with the number 1 and spams with the number 0.

The train set and test set are selected at random from the dataset, at a ratio of 80 percent for training/confirmation and 20 percent for examination (Table.4).

Table 3. Confusion Matrix

Predict classification	***Real classification***	
	Real class	***Fake class***
Positive class	RP (Real Positive)	FP (Fake positive)
Negative class	FN (Fake Negative)	RN (Real Negative)

In this method, the procedure through which train and test data are found is random at each iteration. Therefore, the experiment is repeated 10 times, and the average of the data is analysis and comparison with other techniques.Table.5.

• *Efficiency of the proposed method in Accuracy index:* Examination of the results in the accuracy evaluation index shows that the proposed method has been able to find annoying SMS with an average accuracy of 97.7%.

According to the obtained results, the highest detection accuracy rate was 99% and the lowest was 97%.

• *Efficiency of the proposed method in Precision index:* Precision evaluation index shows that the proposed thesis method able to spot SMS Spam with an average Precision of 95.4%. The highest diagnostic was 97% and the lowest 94%.

• *Efficiency of the proposed method in Recall index:* The method at hand spots unwanted text messages with Recall factor of 96.3% on average. The outcome of this research indicates that the highest and lowest detection rates were 98% and 95%, respectively. The high Recall metric relative to the precision indicates that the suggested technique has successfully detected unpleasant text messages.

• *Efficiency of the proposed method in F-score index:* The results in the evaluation index F show that the proposed method has been capable of spotting unwanted text messages at an average rate of 95.4%. The obtained results show the highest and lowest rates were 97% and 94%, respectively.

Figure 8 depicts the overall outcomes of this investigation.

5. Result Analysis

In this section, three testing phases are devised to confirm the effectiveness of the combined method.

In step one, we review the Deep learning section of the proposed method and the second phase, the machine learning section is tested and in the third phase, the idea of combining deep learning and machine learning in this method is evaluated.

5.1 Comparison of Deep learning section:

Like previous deep learning methods in the simulation process, the deep learning component of this technique requires the specification and quantification of model parameters. The parameters and suggested values that are used in our study are shown in Table.6.

In the suggested technique, the model parameter values shown in Table.5 are considered. However, to demonstrate its efficacy, the model was also examined with different parameter values. Visualize the outcomes of these trials using in tables from 7 to 10.

To demonstrate the efficacy of the deep learning component of our technique, its performance is compared with that of other modes in this section (various values of the parameters). Each mode of the model has been applied 10 times, and the average performance of each mode has been recorded using a variety of performance measures. The number of filters, the size of the convolution layer filters, and the length of the word embedding vector in the FastText technique were examined with alternative values.

For the parameter regarding the number of filters in our model, the value 64 is evaluated, as well as the values 32 and 128. For the filter size parameter in the suggested approach, a size of 3 * 3 and a size of 5 * 5 are considered and tested, respectively. In addition, a length of 300 is considered for the word embedding vector parameter in the model, while a length of 100 is also evaluated here.

• *Assessing the performance of the deep learning section of present method with Accuracy index:* As shown in the Table 11, the optimum value for the number of filters was 64 which yielded more accurate results than 32 and 128. In the filter size parameter, the considered value 3*3 has a higher accuracy than the 5*5 size.

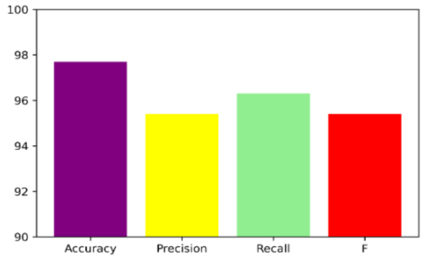


Figure. 8. Assessment of the efficiency in the proposed method

Table 4. Statistic of the Dataset

	Number of messages	*% of messages*	*Training and validation set*	*Testing set*
Spam	1906	30.1%	80%	20%
Non-Spam	4411	69.9%		
Total	6317	100%		

Table 5. The Performance Metrics of The Proposed Method in Various Evaluation Indicators

Evaluation indicators	*Exe 1*	*Exe 2*	*Exe 3*	*Exe 4*	*Exe 5*	*Exe 6*	*Exe 7*	*Exe 8*	*Exe 9*	*Exe 10*	*Avg*
Accuracy	97	98	99	97	98	97	97	97	99	98	97.7
Precision	94	95	97	95	95	98	94	97	94	97	95.4
Recall	95	96	96	98	96	96	95	98	96	97	96.3
F-Score	94	95	96	96	95	95	94	97	95	97	95.4

It is also observed that for the parameter of embedding vector length, the considered length 300 has a higher accuracy than 100. In general, it has been found from these experiments that the values considered for the model parameters with a value of 97.7 have a higher accuracy than all other cases.

- ***Evaluate the performance of the deep learning section of the current method with Precision index:***

The results of the experiments related to the efficiency of each of the states mentioned in the precision index are shown in the Table.12. As can be seen, the value for the number of filters 64 is higher Also, the 3*3 value for the filter size parameter is higher than the other value. It is further observed that the length considered for the embedding vector parameter 300 also has a higher precision index. Generally, it has been found from these experiments that the values considered for the model parameters have a higher precision of 95.4 than others.

- ***Evaluate the performance of the deep learning section of the suggested method with Recall and F-score index:***

From these experiments, it has been determined that the values considered for the model parameters with the value of 96.3 have Recall equal to or greater than other exposition. (Table.13)

These tests also show that the values considered for the model parameters with a value of 95.4 have a better performance in the F-score than other conditions. (Table.14)

The results show that the values considered for the parameters have better performance in accuracy, precision, Recall and F-score.

Figure 9 displays the results of the first phase trials in the form of a bar graph.

5.2 Comparison of Machine Learning section

In this work, two alternative classification methods of the area of machine learning were employed instead of the support vector machine to categorize the text messages, and the results were compared using K-nearest neighbors and decision tree. In this paper, following the prior research, each algorithm was tested 10 times, and the average results were evaluated throughout the comparison phase. The outcomes of these studies are shown in Table 15. The findings reveal that the proposed strategy outperforms the alternatives across all performance criteria. Figure 10 illustrates the outcomes of these tests.

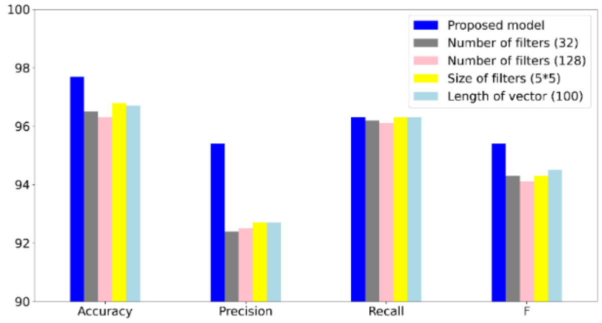


Figure. 9. The efficiency of the deep learning section in the present method in various assessment metrics

Table 6. Deep Learning Parameters of the Proposed Method

Parameter	*Type / Value*
Word Embedding function	FastText
Length of Word Embedding vector	300
Number of filters	64
Size of filters	3*3
Optimizer	Adam

Table 7. Evaluation of the Performance of the Deep Learning Section of the Proposed Method with a Filter Size of 3 * 3 and a Vector Length of 100

Filters	*Accuracy*	*Precision*	*Recall*	*F-Score*
32	96.9	93.5	96.2	94.7
64	96.7	92.7	96.3	94.5
128	96.9	93.5	96.2	94.6

Table 8. Evaluation of the Performance of the Deep Learning Section of The Proposed Method with a Filter Size of 3 * 3 and a Vector Length of 300

Filters	*Accuracy*	*Precision*	*Recall*	*F-Score*
32	96.5	92.4	96.2	94.3
64	97.7	95.4	96.3	95.4
128	96.3	92.5	96.1	94.1

Table 9. Evaluation of the Performance of the Deep Learning Section of the Proposed Method with a Filter Size of 5 * 5 and a Vector Length of 100

Filters	*Accuracy*	*Precision*	*Recall*	*F-Score*
32	96.6	92.6	96.2	94.3
64	97	93.6	96.3	94.7
128	96.6	92.6	96.2	94.3

Table 10. Evaluation of the Performance of the Deep Learning Section of the Proposed Method with a Filter Size of 5 * 5 and a Vector Length of 300

Filters	*Accuracy*	*Precision*	*Recall*	*F-Score*
32	97	93.5	96.2	94.6
64	96.8	92.7	96.3	94.3
128	96.9	93.5	96.2	94.6

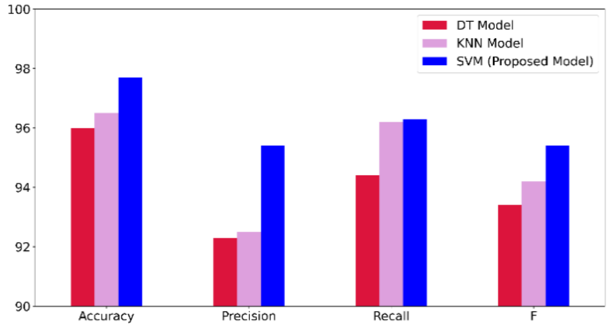


Figure. 10. The efficiency of the machine learning section evaluation in proposed method in various evaluation metrics

Table 11. `The Accuracy of the Model for Different Values of Parameters

Evaluation indicators	***parameter***	***Exe 1***	***Exe 2***	***Exe 3***	***Exe 4***	***Exe 5***	***Exe 6***	***Exe 7***	***Exe 8***	***Exe 9***	***Exe 10***	***Avg***
Number of filters	**32**	96	96	97	97	97	97	97	96	96	97	96.5
	64	97	98	99	97	98	97	97	97	99	98	**97.7**
	128	97	96	96	97	97	96	96	96	96	96	96.3
Size of filters	**3*3**	97	98	99	97	98	97	97	97	99	98	**97.7**
	5*5	96	97	97	97	97	97	97	97	97	96	96.8
Length of Word Embedding vector	**100**	97	97	96	97	96	97	96	97	97	97	96.7
	300	97	98	99	97	98	97	97	97	99	98	**97.7**

Table 12. The Precision of the Model for Different Values of Parameters

Evaluation indicators	***parameter***	***Exe 1***	***Exe 2***	***Exe 3***	***Exe 4***	***Exe 5***	***Exe 6***	***Exe 7***	***Exe 8***	***Exe 9***	***Exe 10***	***Avg***
Number of filters	**32**	92	92	93	93	92	93	92	92	92	93	92.4
	64	94	95	97	95	95	95	94	97	94	97	**95.4**
	128	93	92	92	92	93	93	92	93	93	92	92.5
Size of filters	**3*3**	94	95	97	95	95	95	94	97	94	97	**95.4**
	5*5	92	93	93	92	92	93	93	93	93	93	92.7
Length of Word Embedding vector	**100**	93	92	93	93	92	93	92	93	93	93	92.7
	300	94	95	97	95	95	95	94	97	94	97	**95.4**

Table 13. The Recall of the Model for Different Values of Parameters

Evaluation indicators	***parameter***	***Exe 1***	***Exe 2***	***Exe 3***	***Exe 4***	***Exe 5***	***Exe 6***	***Exe 7***	***Exe 8***	***Exe 9***	***Exe 10***	***Avg***
Number of filters	**32**	96	96	97	96	96	96	96	96	96	97	96.2
	64	95	96	96	98	96	96	95	98	96	97	**96.3**
	128	97	96	96	96	96	96	96	96	96	96	96.1
Size of filters	**3*3**	95	96	96	98	96	96	95	98	96	97	**96.3**
	5*5	96	96	96	97	97	96	96	96	97	96	96.3
Length of Word Embedding vector	**100**	96	96	96	97	96	96	97	96	96	97	96.3
	300	95	96	96	98	96	96	95	98	96	97	**96.3**

Table 14. The F-Score of the Model for Different Values of Parameters

Evaluation indicators	***parameter***	***Exe 1***	***Exe 2***	***Exe 3***	***Exe 4***	***Exe 5***	***Exe 6***	***Exe 7***	***Exe 8***	***Exe 9***	***Exe 10***	***Avg***
Number of filters	**32**	94	94	95	94	94	95	94	94	94	95	94.3
	64	94	95	96	96	95	95	94	97	95	97	**95.4**
	128	95	94	94	94	94	94	94	94	94	94	94.1
Size of filters	**3*3**	94	95	96	96	95	95	94	97	95	97	**95.4**
	5*5	94	94	94	95	94	94	95	94	95	94	94.3
Length of Word Embedding vector	**100**	95	9494	95	94	95	95	94	94	95	95	94.5
	300	94	95	96	96	95	95	94	97	95	97	**95.4**

The results of the experiments reveal that the Support Vector Algorithm promises better performance in all individual tests than other methods. Figure 10 shows the results of these experiments.

5.3 Comparison of the proposed method with other methods

In this part, we will assess the concept of combining deep learning and machine learning into the suggested technique. The first technique is the deep learning model, which is comparable to the proposed method's deep learning component. Once two convolutional layers in this model are applied, one layer of long short-term memory network and one fully connected layer are investigated by using them on the output of another fully connected layer in order to identify text messages.

Number of categorization classes represented by the number of neurons in the second fully connected layer (spam and non-spam). Consequently, in this technique, text messages are identified without the use of the machine learning algorithm (support vector machine) and just via deep learning.

The second model strategy relies solely on machine learning. In this approach, only the base of the support vector machine is used to divide the data in a sensible manner.

The third way is the suggested method, which consists of two convolution layers followed by a long short-term memory network layer. The output is next applied to a fully connected layer, and the results are sent to the support vector machine.

Every methods has been executed 10 times, and Table 16 presents some information with regard to their performance.

In terms of accuracy, the hybrid model had the best performance with a score of 97.7. In addition, its precision surpasses all other approaches by 95.4%. With F-Score value of 95.4. It is ultimately better than the other two approaches. Figure 11 illustrates this advantage with the results of the third phase of studies.

6. Conclusion

The suggested technique is implemented in the Python and its performance is assessed using four evaluation metrics: precision, recall, F score, and accuracy. The findings indicate that the suggested technique can distinguish between SMS spam and non-spam with 97.7% accuracy, 95.4% precision, 96.3% recall, and a 95. 4% F score.

In addition, several studies have been devised and carried out to demonstrate the efficacy of each model component. In these trials, the suggested approach was compared against other methods; the three parts that follow describe the most significant findings.

- The first is the use of deep learning, which can also extract textual features from text messages. In this procedure, specifying parameters such as the length of the embedding vector in the FastText method, the number of convolution layer filters, and their size is a bare necessity. To demonstrate their efficacy in the first round of trials, the parameters of the deep learning section were evaluated with varying values, and it was determined that they show more promising results than the other methods.
- The following conclusion relates to the suggested technique for detecting and categorizing SMS. In the classification portion, the approach of support vector machines is used. The findings presented in Section V demonstrate that the support vector machine method outperforms all other techniques.
- The final result is that combining deep learning models with machine learning techniques may increase the precision of recognizing unpleasant text messages. The concept of integrating deep learning with machine learning was examined in Section V. This approach has been compared with methods on the basis of deep learning and machine learning to demonstrate its efficacy. The findings of this section indicate that the combination of deep learning and machine learning yields superior outcomes than each of the techniques alone.

7. Future Work

- The feature vector extracted with the Word2vec approach for words with the same text and different meanings is not done correctly and extracts similar vectors, so the use of transformers based on attention method can solve this problem and extract different vectors for these words. For this reason, we suggest

Table 15. Comparison of Machine Learning Performance with other Methods in Different Evaluation Metrics

Methods	*Accuracy*	*Precision*	*Recall*	*F Score*
Decision Tree	96	92.3	94.4	93.4
K Nearest Neighbors	96.5	92.5	96.2	94.2
SVM	97.7	95.4	96.3	95.4

Table 16. The Efficiency of The Proposed Method Evaluation with Models Based only on Deep Learning and Machine Learning in Different Evaluation Indicators

Methods	*Accuracy*	*Precision*	*Recall*	*F Score*
Deep learning	96.3	92.5	95.7	94
Machine learning (SVM)	96	92	95.1	94
Proposed Method	97.7	95.4	96.3	95.4

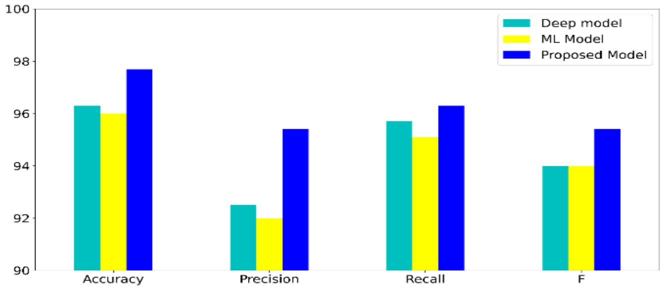


Figure. 11. The efficiency of the hybrid proposed method with machine learning only and deep learning only methods in various evaluation metrics

the use of transformers in future work to extract more accurate embedding word vectors. Also Transformers with longer term memory can memorize the meanings of consecutive sentences over longer intervals than LSTM network, therefore have more accurate results.

- By collecting larger and more comprehensive datasets that include different types of SMS Spam and Non-Spam, Can train a generalized model to be used in operational applications.
- By converting binary classification to multiple classes, we can have a more accurate classification of SMS Spam by increasing the types of spam, such as advertising spam and phishing spam.

Declarations

Funding

This research did not receive any grant from funding agencies in the public, commercial, or non-profit sectors.

Authors' contributions

RK: Study design, acquisition of data, implementing proposed model, interpretation of the results, statistical analysis, drafting the manuscript;

SJ: Study design, Supervision, revision of the manuscript, interpretation of the results, statistical analysis;

AY: Study design, Supervision, drafting the manuscript, revision of the manuscript, interpretation of the results, statistical analysis;

Conflict of interest

The authors declare that there is no conflict of interest.

References

[1] P. Navaney, G. Dubey, and A. Rana, "SMS spam filtering using supervised machine learning algorithms," in 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) ,2018 ,pp.43-48: IEEE.

[2] N. N. A. Sjarif, N. F. M. Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, "SMS spam message detection using term frequency-inverse document frequency and random forest algorithm," Procedia Computer Science, vol. 161 ,pp.509-515, 2019.

[3] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter SMS Spam," *Future Generation Computer Systems,* vol. 102 ,pp. 524-533, 2020.

[4] S. Gadde, A. Lakshmanarao, and S. Satyanarayana, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," in *7th International Conference on Advanced Computing and Communication Systems (ICACCS),* 2021, vol. 1, pp.358-362: IEEE.

[5] A. Tekerek, "Support vector machine based spam SMS detection," *Politeknik Dergisi,* vol. 22, no. 3, pp. 779-784, 2019.

[6] S. Ballı and O. Karasoy, "Development of content-based SMS classification application by using Word۲Vec-based feature extraction," IET Software, vol. 13, no. 4, pp. 295-304, 2019.

[7] I. Santos, N. Nedjah, and L. de Macedo Mourelle, "Sentiment analysis using convolutional neural network with fastText embeddings," in 2017 IEEE Latin American conference on computational intelligence (LA-CCI), 2017, pp. 1-5: IEEE.

[8] P. Mojumder, M. Hasan, M. F. Hossain, and K. A. Hasan, "A study of fasttext word embedding effects in document classification in bangla language," in International Conference on Cyber Security and Computer Science, 2020, pp. 441-453: Springer.

[9] Y. Li, Z. Hao, and H. Lei, "Survey of convolutional neural network," Journal of Computer Applications, vol. 36, no. 9, pp. 2508-2515, 2016.

[10] Y. Li, Z. Hao, and H. Lei, "Survey of convolutional neural network," Journal of Computer Applications, vol. 36, no. 9, pp. 2508-2515, 2016.

[11] S. Sony, K. Dunphy, A. Sadhu, and M. Capretz, "A systematic review of convolutional neural network-based structural condition assessment techniques," Engineering Structures, vol. 226, p. 111347, 2021.

[12] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," in 2017 *International Conference on Communication and Signal Processing (ICCSP),* 2017, pp. 0588-0592: IEEE.

[13] A. S. Vyas, H. B. Prajapati, and V. K. Dabhi, "Survey on face expression recognition using CNN," in 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 102-106: IEEE.

[14] R. Mu and X. Zeng, "A review of deep learning research," KSII Transactions on Internet and Information Systems (TIIS), vol. 13, no. 4, pp. 1738-1764, 2019.

[15] Scholkopf, Bernhard, and Alexander J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. Adaptive Computation and Machine Learning Series, 2018.

Roya khorashadizadeh was born in 1991 in Mashhad. she received her B.Sc. in Information Technology Engineering from mashhad islamic azad University in 2015. Currently, she is an M.Sc. student in Information Technology at Science and Research Branch, islamic azad university. her research interests include Text processing and machine learning.

Sommayeh Jafarali jassbi was born in Tehran,Iran, in 1982. She received the M.Sc degree in computer architecture engineering in 2007, and the Ph.D. degree in computer architecture engineering in 2010 from the Islamic Azad Univeriry Science and Research Branch. In 2010, she joined the Department of computer engineering, Islamic Azad University Science and Research Branch. She became an associate professor in 2011. Her interests are cloud computing, internet of things, wireless sensor network and computer architecture and cryptography. She was head of computer department in 2012.Now she is selected as a head of computer department again.She was also an active member of young researcher club from 2004. She has written, translate and published several professional books and paper in her fields.

Alireza Yari received his B.Sc. degree in control system engineering in 1993 from the University of Tehran, Iran, and M.Sc. and Ph.D. degree in System engineering in 2000 from Kitami institute of technology, Japan. He is currently doing research in Information Technology research faculty of Iran Telecom Research Center (ITRC). His research interests include cloud computing and data centers. He is also working on application of cloud computing in data intensive application, such as web search engine.