Journal of Information Systems Engineering and Management
2025, 10(10s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Deep Learning Hybrid Approach for Accurate SMS Spam Identification

Adel Al-Zebari[1], Mohammed Barwary[2], Naaman Omar[3], Nechirvan Asaad Zebari[4], Dilovan Asaad Zebari[5,6*]
[1]*Akre University for Applied Sciences, Technical College of Informatics-Akre, Department of Information Technology, Duhok, Iraq;*
[2]*Pedagogical and Training Center, University of Duhok, Duhok, Iraq;*
[3]*Duhok Polytechnic University, Amedi Technical Institute, Department of Information Technology, Duhok, Iraq;*
[4]*Department of Information Technology, Lebanese French University, Erbil, Kurdistan Region, Iraq;*
[5]*Department of Computer Science, College of Science, Nawroz University, Duhok 42001, Kurdistan Region, Iraq;*
[6]*Faculty of Computing and Information Technology, Sohar University, Oman.*
Corresponding Author Email: dilovan.majeed@nawroz.edu.krd

ARTICLE INFO

Received: 10 Nov 2024

Revised: 28 Dec 2024

Accepted: 14 Jan 2025

ABSTRACT

Short messaging service (SMS) is a popular application for mobile devices. People often use SMS when they are not suitable for voice calls. Nowadays, SMS is used for commercial purposes. These SMS can sometimes be useful. But sometimes, unwanted SMS, which is called spam SMS, can disturb mobile phone users. Thus, spam SMS detection becomes an important application for mobile phone service providers. Up to now, machine-learning approaches have been used for spam SMS detection. These approaches used various supervised learning methods for detection purposes. In this paper, three deep-learning approaches are used for SMS spam detection. These approaches are Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and hybrid CNN with LSTM, respectively. The mentioned all deep learning approaches are trained in an end-to-end fashion. As these approaches use numeric input data, the text input data is initially converted to numeric data. To do that, a pre-trained word embedding network is used to convert each SMS text data to an array of word vectors. An SMS spam dataset, which is downloaded from the UCI machine learning repository, is considered in experimental works for the performance evaluation of the mentioned deep learning approaches. Four performance evaluation metrics namely accuracy, precision, recall, and F-score are used in the performance evaluation. The experimental works show that the hybrid approach produces better detection results than the CNN and LSTM approaches. Besides, the result of the hybrid method is compared with some of the published results. Comparisons show that the hybrid method outperforms the other compared methods.

Keywords: SMS spam detection, machine learning, deep learning, hybrid deep model end-to-end training

1. INTRODUCTION

One of the most popular and widely used contact systems today is the short message service, also known as SMS. From 1.46 billion in 2000 to 7.9 trillion in 2012, the number of SMS messages sent has risen dramatically [1]. The number of SMS-enabled mobile phone users had increased to 6.1 billion by 2015. A substantial increase in sales has resulted from the rise in smartphone users. Although revenue has been declining since 2017, global SMS revenue is projected to hit $83.2 billion in 2022, according to the most recent statistics [2]. Furthermore, the P2P (person-to-person) SMS industry generates about half of global SMS revenue (43 billion dollars), while the A2P (agent-to-person) SMS market generates the other half (40.2 billion dollars) (application-to-person). Bulk SMS sending services like bulksmsonline.com and bulksms.com send commercial A2P messages like authentication codes, e-commercial warnings, and express delivery updates [3]. SMS spam has increasingly focused on mobile phones in recent years. The term "SMS spam" refers to any text message that is sent through a mobile network that is not intended for the recipient. Users find that they are disruptive [4]. A poll reveals that 68 percent of people who use mobile phones receive unsolicited text messages (SMS Spam). Smishing is one example of malevolent behavior that can be found in some situations of SMS spam. Smishing is a form of cybercrime that targets mobile users and attempts to trick them

into downloading dangerous software or visiting dangerous websites by sending them unsolicited text messages (also known as spam). SMS and phishing are the two words that are combined to form smishing [5]. SMS is commonly used by attackers since it is a straightforward method to contact victims [6]. Smartphone users make up the vast majority of those who fall victim to phishing and smishing attacks [7]. By delivering a link to victims or making direct contact with them through SMS texts, the attackers seek to steal confidential information belonging to the users, such as credit card numbers, bank account details, and so on [8].

The growing popularity of short message service (SMS) among mobile users has piqued the interest of spammers, and research has indicated that approximately 33% of SMS sent in Asia are deemed to be spam. SMS spam is still a growing problem on a global scale because of the high response rate of mobile consumers, which is based on the trusted and personal services that SMS provides. Generally speaking, an unwelcome or unsolicited message is referred to as "spam." Spam can be delivered arbitrarily by an individual who does not have any relationship with the user, and it is typically transmitted with commercial, fraudulent, or malevolent intent [9]. The persistent increase in the number of unwanted text messages sent to end users as a result of network congestion caused by an overwhelming number of SMS messages at the end of mobile network providers [10] was the impetus for conducting this research study. In addition, an SMS spammer will use telemarketing to clog up a network. This creates a potential problem for the SMS gateway because the spammer will use a script to send a large number of messages through a single gateway, which will either deny mobile users service or cause them to experience a delay in receiving it. The impact of spam text messages on consumers is significantly greater than that of spam emails. This is because SMS subscribers have a greater sense of safety when utilizing this service for transferring personal information, authorizing payments, and performing a variety of other day-to-day tasks. Because of the following factors, accurate classification of SMS presents a significant challenge: Because of the restricted number of characters involved in SMS (160 characters), there is not always a sufficient SMS dataset available for use in training and testing purposes. The prevalence of idiosyncratic languages and non-standard abbreviations (such as punctuation, emoticons, and so on) has been shown to have an effect on the efficiency of existing classifiers [11].

SMS messaging charges have fallen below 0.001 US dollars in some markets and are also free in others as the service has increased in popularity. Furthermore, attackers can send malicious messages for very little money thanks to the rapid growth of text messaging and the availability of unlimited texting plans. This, coupled with consumers' tacit trust in their mobile devices, creates an environment that is ripe for attack. As a result, advertisers are increasingly using text messages to reach customers, making mobile phones the next electronic junk mail target. SMS spam refers to any unwanted text message sent to a cell phone (also known as mobile phone spam). While this is not a common practice in North America, it is very common in Asia. There is a remedy that can be applied to the above issues. It is accomplished by categorizing and filtering SMS. Some of the most popular text classification techniques include decision trees, Naive Bayes, rule induction, neural networks, nearest neighbors, and support vector machines. Nonetheless, this SMS classification varies from that of a regular document text or e-mail [2, 3] due to the very short text (maximum 160 7-bit characters), many abbreviated messages, and the propensity for SMS to be informal text. If an SMS is just a few words long, another question arises: "Does the feature differentiate between SMS spam and non-spam?" Furthermore, today's SMS types are becoming more complex, necessitating the use of a different technique to incorporate features that can distinguish between SMS spam and non-spam. Any current SMS variation, especially SMS spam, follows a similar pattern. The example can be used to support the use of the technique of words occurring simultaneously as an additional feature to distinguish spam from non-spam SMS [2].

Furthermore, spam filtering via SMS on smartphones is not as robust as spam filtering via email; however, it is supplemented by more advanced spam filtering methods [12]. The utilization of deep neural networks is one of the more recent approaches that has been shown to be beneficial in the process of resolving issues of this nature. There have been various applications of the deep neural network-based architecture, including convolutional neural networks (CNN), recurrent neural networks (RNN), and long short-term memory (LSTM), for the classification of problems involving images, videos, and texts. In addition to the more conventional machine learning techniques for clustering, automatic pattern identification can be accomplished with deep learning (unsupervised techniques). It features a hierarchical structure with multiple layers of information processing stages [4]. In order to reduce the number of incorrect classifications produced by deep learning classifiers, each level makes use of the individual components of the deep neural network. However, deep neural networks are not yet being utilized to their full potential in the process of text message classification.

To address the gap in the research, the purpose of this paper is to investigate the potential of deep neural networks for the classification of SMS spam. It leverages the CNN model. This is due to the fact that CNN is useful at capturing local and temporal features, including n-grams from the text. It was determined that the RNN, which is another type of deep neural network, was beneficial. This is possible due to the fact that RNN is designed to deal with the long-term reliance that comes with word sequences. RNN is able to assess whether or not a message is a spam by analyzing only the first few words of the message. Because short message service (SMS) typically does not impose any length constraints, RNNs that have the capacity to memorize long sequences of text could prove to be valuable. Despite this, the regular RNN has a problem called the "vanishing gradient" that hinders its performance. Because of this, an RNN offshoot called LSTM was utilized. After all, prior research that was relevant to this topic has likewise made use of LSTM. Therefore, the objective of this paper is to classify mobile text messages as Spam or Not-Spam using the CNN and the LSTM model and the main contributions of this work are:

- To investigate the most powerful deep learning models for classifying SMS spam.
- To design a hybrid deep learning model to classify SMS spam into spam and not spam.
- A comparative study has been presented between the proposed hybrid model with other machine learning and deep learning models.

The remainder of this paper is as follows. Section 2 presents recent related studies. In section 3, materials and the proposed hybrid model are presented. In Section 4, the dataset and the experimental works are given. The conclusion of the paper can be found in Section 5.

2. RELATED WORK

Research into the detection of spam is not a recent development. Since then, investigators have been looking into the matter. Detecting spam on the internet and via email had been the primary focus up until recently; now, however, the scope has expanded to include social media. Deep learning methods were the logical next step due to their ability to produce rapid and accurate results without the need for ongoing human support. This was necessary for light of the challenges that machine learning presents, such as spam drift and the manipulation of extracted features.

Roy et al. recently published a paper [4] in which they recommended the use of deep learning to categorize SMS messages as either spam or not spam. The combination of CNN and LSTM, which are both forms of deep learning, is supposed to achieve the goal of their methodologies. The purpose of this exercise is to sort through the various text messages and determine which ones are spam and which ones are not. They compared the performance of the proposed method with that of existing machine learning algorithms, such as Naive Bayes, Random Forest, Gradient Boosting, Logistic Regression, and Stochastic Gradient Descent. The goal of this comparison was to determine how well the new method performed. The findings that were collected demonstrated that the CNN and LSTM models are significantly superior to other machine learning models that were considered. The authors of [12] created a model that they referred to as the "Smishing Detector" with the intention of detecting phishing communications with a lower rate of false positives. The model that has been proposed is composed of four separate modules. Utilizing the Naive Bayes classification technique, the first module's objective is to do an analysis of the contents of text messages with the purpose of determining which messages contain malicious material. The second one is the one that is utilized to conduct an audit of the URL that is included in the messages. The third module's purpose is to conduct an investigation into the website's source code, which is linked in the messages. The final module is called an APK download detector, and its purpose is to determine whether or not a file that could be considered dangerous is downloaded when the URL is called. The accuracy of this model was determined to be 96.29% by a series of experimental tests that were carried out by the authors.

A method for detecting spam that is based on the integration of multiple modalities was proposed by Yang et al. [13], they attained an accuracy rate of 98.48% by employing the deep neural network model in their spam detection process. A method for the detection of spam that was based on the artificial immune system (ISAIS) was proposed in [14], and it was shown to have an accuracy of 98.05%. The research presented in [15] utilized naive Bayes classifiers in conjunction with an artificial neural network to make an accurate prediction of spam based on text data; however, the analysis was only performed on data in the English language. Sharaff et al. [16] presented an original model for an SMS spam filter that was based on an algorithm that was biologically inspired and was given the terms "krill herd optimization" and "dendritic cell algorithm." The researchers' experimental findings demonstrated that the proposed model produced more accurate results when compared to other machine learning classifiers, such as the NB, LR,

SVM, and XgBoost classifiers. Based on Indonesian text messages, the research presented in this article [17] proposes a methodology for classifying messages as spam, promotion, or ham. The model was trained using a total of 4,125 text messages, and it was then validated using a total of 1,260 text messages. The classifiers were evaluated using a 10-fold cross-validation method, and the findings indicate that Random Forest (94.62%), Multinomial Logistic Regression (94.57%), Support Vector Machine (94.38%), and XGBoost (94.52%) are some of the best models that can be utilized for a multiclass SMS classification. Detecting smishing messages using a feature-based technique was one of the authors' recommended methods [18]. The method utilizes the extraction of ten attributes, all of which the authors say are able to differentiate between genuine and fake communications. After that, the characteristics were applied to a dataset that had previously been used as a benchmark, and five different classification techniques were used in order to evaluate how well the suggested method worked. Based on the results of the experimental evaluation, it was determined that the model has a true positive rate of 94.20% and an accuracy rate of 98.74% overall when it comes to detecting smishing messages.

In addition to this, a novel architecture for deep learning that is based on convolutional neural networks (CNN) and long short-term memory (LSTM) has been developed [19]. The model was augmented with semantic information in the representation of words through the utilization of knowledge bases such as WordNet and ConceptNet. The utilization of these knowledge bases improved the performance by delivering a superior semantic vector representation of test data that possessed a random value due to the fact that they were not utilized in the training process. Study in [20] proposed a GCN-based anti-spam (GAS) model. This model was a large-scale anti-spam strategy that was based on graph convolutional networks (GCN) for the purpose of detecting spam advertising "Xianyu," which is China's most popular application for selling used goods. In order to capture both the local and global contexts of a comment, a heterogeneous graph and a homogeneous graph had to be joined. Experiments both offline and online were carried out, and the results showed that the technique that was proposed was successful. Study in [21] suggested a model for semi-supervised social spam detection that utilized directed social graphs and incorporated both graph convolutional networks (GCNs) and Markov random fields (MRFs). This model operated on directed social graphs. Experiments were run on two real-world Twitter datasets, and excellent results were obtained from both. The multi-filter that was created by Bosaeed et al. [22] utilized multiple ML-based classifiers using three different classification methods. These methods were naive Bayes (NB), support vector machines (SVM), and naive Bayes multinomial (NBM). The research demonstrates the versatility of numerous platforms by applying their proposed model in part and in its entirety to both mobile and server applications. This ensures that computational resource optimization is achieved.

Wei and Nguyen [23] presented a new technique for the identification of SMS spam that is based on a lightweight deep neural model. This technique is referred to as the Lightweight Gated Recurrent Unit (LGRU). The authors of this study compared their findings against over 30 distinct machine learning and deep learning classifiers so that they could demonstrate the viability of the strategy they had created. Aside from that, the proposed strategy accomplished greater results when compared to models that already exist, and the authors claimed that it also incurs less complexity in terms of the amount of time spent training. An additional intriguing piece of work was carried out by Xia and Chen [24], who presented an improved hidden Markov model for a weighted feature set and label words. According to the findings of their research, the application of weighted features to enhance HMM performs better than the LSTM when it comes to accuracy as well as computational speed. The researchers who published their findings in [25] suggested an innovative method that makes use of linguistic characteristics to identify bogus reviews. Unsupervised learning through self-organizing maps (SOM) and convolutional neural networks (CNN) were employed by the researchers in conjunction with each other to classify the reviews. Words from the reviews were arranged around a self-organizing map (SOM) grid cell in order to create visuals. These images were then used to represent the reviews. Extensive testing revealed that the proposed approach was successful in a variety of contexts, including those involving a single domain and those involving many domains. Gadde et al. [26] and Al-Bataineh and Kaur [27] are the two publications in which the authors demonstrated the application of deep learning approaches for the identification of SMS spam using LSTM. The authors of the earlier study also utilized three distinct word embedding strategies, which were based on the count, TF-IDF, and hashing vectorizer, respectively. The findings of the LSTM experiments were analyzed and contrasted with those of several cutting-edge machine-learning approaches. On the other hand, writers in the latter demonstrated the resiliency of LSTM topologies by using a clonal selection technique for text categorization. The research was assessed with the help of three different datasets and compared to a number of cutting-edge machine learning classifiers. The findings of the experiments demonstrated that the

suggested model performs better than other models in terms of accuracy, precision, recall, F1-score, and the amount of processing time required.

In this study, three deep learning approaches namely CNN, LSTM, and a hybrid CNN with LSTM are used for SMS spam detection. The training of the mentioned deep learning approaches is carried out in an end-to-end way. As CNN, LSTM, and a hybrid CNN with LSTM approaches use numeric input data, text input data is initially converted to numeric data. To do this, a pre-trained word embedding network is used to convert each SMS text data into a set of word vectors. The used CNN, LSTM, and a hybrid CNN with LSTM architectures are composed of nine, six, and fourteen layers, respectively. In experimental studies, an SMS spam dataset downloaded from the UCI machine learning repository is considered for performance evaluations. Four performance evaluation criteria are used in performance evaluation: accuracy, precision, recall, and F score. Experimental studies show that the hybrid CNN with LSTM approach produces better detection results than the CNN and LSTM approaches.

In this research, methodologies based on deep learning are utilized in order to detect spam in SMS messages. In order to detect spam in SMS messages, CNN and LSTM algorithms are utilized. Then, a CNN and LSTM-based hybrid technique is offered as a solution for the identification of spam in SMS messages. When the LSTM method is used, the text data that is input is first transformed into numeric sequences for the purpose of making it more convenient to work with the input of the LSTM approach. For the purpose of converting text to numeric sequence indices, a word encoding process is utilized. This procedure maps individual documents to sequences of numeric indices. When the CNN method is used, the text data is transformed into images at the beginning of the process.

3. PROPOSED METHOD

In this section, we will provide a comprehensive explanation of the model that we have proposed. Processing the gathered text messages (SMS) and applying a deep learning algorithm in order to categorize them and determine which ones are deemed to be spam or phishing messages is the primary goal of this detection system. In Figure 1, we show the overall structure of the model that has been proposed. The process of the suggested system kicks off with the removal of superfluous information from the text messages that have been received. After that, a pre-processing task will be applied to these messages in order to represent the textual data in a manner that is consistent with the rest of the system. Following the completion of the data preparation, the classification algorithms will be applied to these data in order to differentiate between messages that are considered spam and messages that are not considered spam. In this study, we compare the newly proposed method that we developed with several different machine learning classifications.

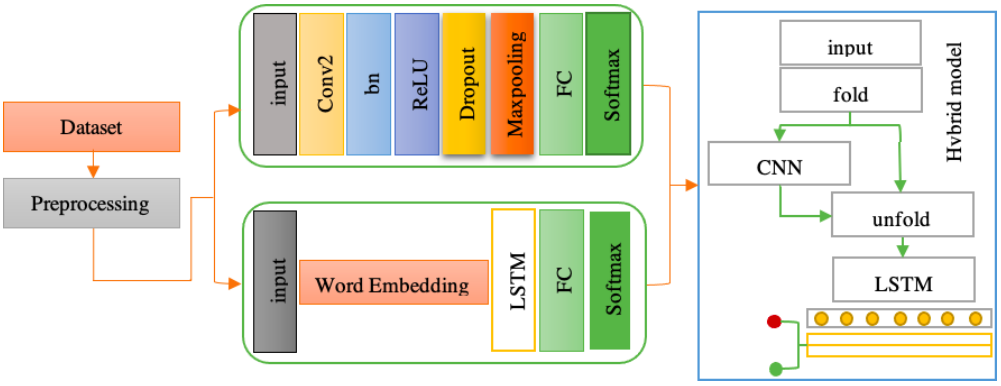

Figure 1. Proposed hybrid model

3.1 Data preprocessing

The first phase of the model that is being offered is the cleansing of the data. The performance of the machine learning model will be improved if this task is completed successfully, which will result in the removal of words and symbols from text messages that are not necessary. Additionally, all of the words should be written in lowercase once the capital letters have been removed. In addition, remove all punctuation marks, prepositions, and short-length words that have fewer than or are equal to two *(<=)* alphabets. Because we wanted to make sure that our model was accurate, we took the time to get rid of all of the extraneous punctuation, such as stopwords, exclamation points,

short words, and so on. Because most abbreviations are not standardized, it was necessary for us to make our preprocessing more flexible so that we can accommodate them.

3.2 CNN Architecture

In natural language-based applications such as text classification, sentiment analysis, email spam detection, fake news detection, and so on, deep learning models have consistently demonstrated great and significant performance. These applications range from text classification to fake news detection to email spam detection. CNN is one of the best-known examples of a deep learning model that has the capability to derive useful features from the provided data [28].

Table 1. CNN model architecture description

Layer	Output	Layer	Output
Conv_1	(None, 128, 128, 16)	max_pooling _3	(None, 16, 16, 64)
bn _1	(Batch (None, 128, 128, 16)	Conv_4	(None, 32, 32, 64)
ReLU_1	(None, 128, 128, 16)	bn_4	(Batch (None, 32, 32, 64)
max_pooling _1	(None, 64, 64, 16)	ReLU _4	(None, 32, 32, 64)
Conv_2	(None, 64, 64, 32)	max_pooling_4	(None, 16, 16, 64)
bn _2	(Batch (None, 64, 64, 32)	flatten_1 (Flatten)	(None, 16384)
ReLU _2	(None, 64, 64, 32)	dense_1 (Dense)	(None, 64)
max_pooling_2	(None, 32, 32, 32)	bn_5	(Batch (None, 64)
Conv_3	(None, 32, 32, 64)	ReLU _5	(None, 64)
bn _3	(Batch (None, 32, 32, 64)	dense_2 (Dense)	(None, 32)
ReLU _3	(None, 32, 32, 64)	dense_3 (Dense	(None, 2)

In the case of an image email, the attachment is first shrunk down to 128 × 128 pixels before being fed into the CNN model. This allows for the calculation of the likelihood that the image itself is spam. The CNN model consists of four convolutional layers; the first convolutional layer has 16 filters, the second convolutional layer contains 32 filters, the third convolutional layer contains 64 filters, and the fourth convolutional layer contains 128 filters. Different square filter kernel sizes are used for each convolutional layer, specifically 2 × 2, 3 × 3, 4 × 4, and 5 × 5, while the convolution stride and space padding are also set to one pixel. Each convolutional layer is connected to a MaxPooling window that is 2 × 2, and there is one stride between each window. The max-pooling layer's primary function is to standardize the output that is generated by the convolutional layer in order to acquire inputs that can be fed into the classification layer. This is done in order to improve the accuracy of the classification layer. In addition, the pooling layer cuts down on the dimensions that were produced by the layer that came before it. It does this while simultaneously preserving the integrity of significant features and removing a barrier to overfitting. Following that is the maximal function, which is the component of this layer that places the most emphasis on the importance. The foundation of this method is taking the highest possible value from the cell that was used to generate the window of cells. Following this step, the resulting matrices from each filter are joined in order to generate a univariate vector. Table 1 contains a concise overview of the CNN model architecture, as well as the output shapes and parameters for each layer.

Journal of Information Systems Engineering and Management
2025, 10(10s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The fully connected layer (FC), is the last step in the construction of the CNN algorithm's architecture. This layer is responsible for receiving the converted output of the pooling layer in the form of a one-dimensional feature vector. This operation is referred to as "flattening." As a result, we make use of the flattening technique in order to reduce a vector with several dimensions to a vector with a single dimension. The final step involves obtaining the categorization probability value of the image data as spam by employing three fully connected layers with a total of 64, 32, and 2 neurons, respectively. Neurons in all hidden layers are activated by a nonlinear ReLu function, and this function is followed by the neuron. CNN has expressed support for an operator for regularization known as "dropout." In most cases, dropouts are utilized to simplify the connections between nodes that are part of a densely linked layer that is fully connected. It is a user-dependent variable that can accept values between 0 and 1, and the range is from 0 to 1.

In addition, batch normalization technology is used to normalize the image dataset after each convolutional layer. This is done to prevent the distribution of the image dataset from changing while it is being trained, which helps to avoid gradient disappearance or explosion and speeds up CNN model training by reducing internal covariate shifts. The optimizer's job is to cut down on the number of mistakes made by the model as a means of making it more accurate. Table 2 displays the range of values for these hyperparameters, as well as the values that the CNN model determined to be optimal.

Table 2. The range and optimal values of hyperparameters for CNN.

Parameter	Value
epochs	32
batch size	20
learning rate	0.01
dropout	0.3
optimization	Adam

3.3 LSTM Architecture

Figure 2 provides a general representation of the LSTM model's internal structure. It is made up of three layers: one with a single word embedded in it, two LSTM layers, and one fully connected (FC) layer. The following is a list of the procedures that need to be taken when handling the text component of an email in order to acquire the classification probability value of the email: acquiring the text data of an email by first employing the preprocessing technique and then utilizing the word embedding technique in order to obtain its word vector representation [29].

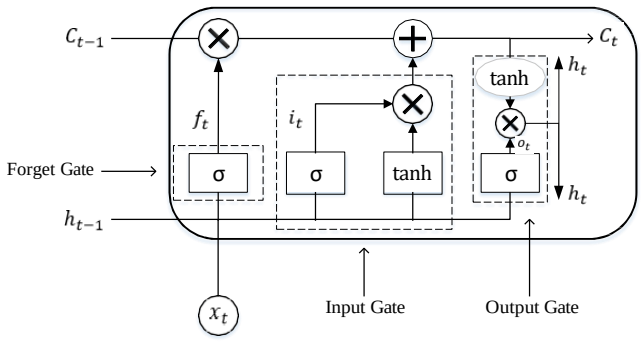

Figure 2. The representation of the LSTM unit.

Word embedding is a method in which each word is represented by a vector containing integers that indicate the word's semantic similarity to other words in a text corpus (i.e., similar words have similar representations). The word

embedding technique transforms each word into a dense vector called a "word vector" that captures the word's relative meaning within the document using the GloVe algorithm that is implemented by the embedding layer [30]. In contrast, the one-hot encoding method splits each text into a group of words and turns each word into a sequence of numbers, disregarding the word's meaning within context. This is in contrast to the word embedding technique, which transforms each word into a dense vector called the word embedding method, each message m is represented as a string of words called *w1, w2, w3, ..., wn*, and each word is shown in the form of a word vector that has a length of d. After that, all of the word vectors that make up a particular message that is, *n*-word vectors are joined together to create a word matrix that has the form $M \in Rn \times d$. At last, the word matrix is received through the input layer, where it is received, and the convolution operation is carried out.

Following this step, we apply the designed LSTM layer to the text data in order to automatically extract features from it. Finally, we apply the FC layer with the Softmax activation function in order to obtain the classification probability value of the text data as spam. The LSTM model is trained and optimized by making use of the log-likelihood function in order to minimize the loss function [31]. Please refer to the scholarly research for a more in-depth algorithm for the LSTM unit. Let's refer to the textual content of an email as "T." Insert T into the embedding stage so that it can be converted into a word vector x, where $x = (x1, x2, ...,xl)$, where $xi \in R$, n is the n-dimensional word vectors for the i-th word in the document T, and matrix $x \in R$ lxn denotes the document T, where l is the maximum length of and l is less than 500. Input sequences in the form of phrases are combined with the results of the previous LSTM unit before being introduced into the LSTM unit. This process is repeated with each new sentence that is supplied, and as a result, the LSTM units are able to continue preserving the essential characteristics. The number of LSTM units that save the most relevant features is the variable in question. Consequently, by use of the LSTM layer, the FC layer, and the Softmax activation function, we utilize the grid search optimization algorithm to determine the best possible values for the LSTM model's five hyperparameters, which are the learning rate, batch size, epochs, dropout rate, and optimization technique. These five hyperparameters are responsible for the model's overall performance. Table 3 displays the range of values for these hyperparameters, as well as the values that the LSTM model determined to be optimal.

Table 3. The range and optimal values of hyperparameters for LSTM.

Parameter	Value
epochs	32
batch size	32
learning rate	0.001
dropout	0.3
optimization	Adam

3.4 Proposed hybrid model

Figure 3 shows the third approach where the hybrid CNN and LSTM network is used for SMS spam detection. The hybrid CNN and LSTM network is composed of several layers. After the input layer, a folding layer is located. The folding layer is connected to both conv2 and unfold layers, respectively. To handle the text-to-image conversion, a padding procedure is employed for the input text data for providing a constant text length. Then, the constant-length text data is converted into sequences of word vectors of length *C* using word embedding. A pre-trained word embedding network is used to convert each SMS text data to an array of word vectors. If the constant text length is assumed to be *S*, then an SMS text is represented by a $1 \times S \times C$ image, where the height is *1, S* shows width, and *C* symbolizes the number of channels.

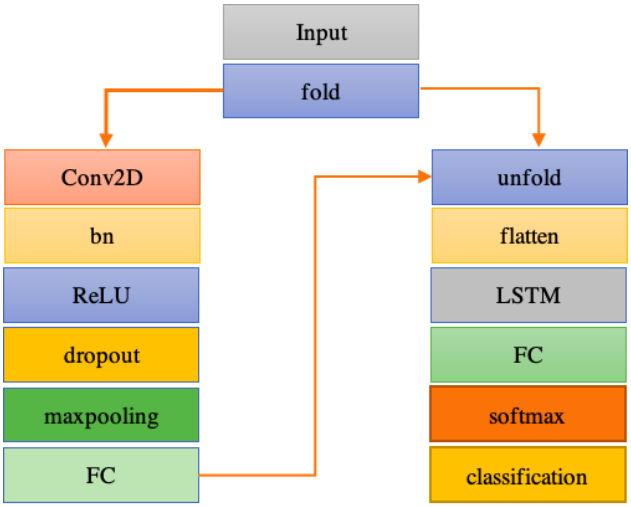

Figure 3. Representation of hybrid CNN and LSTM approach for SMS spam detection

It is worth mentioning that a pre-trained word embedding network is also used to convert each SMS text data to an array of word vectors as used in the first approach. After the conv2 layer, batch normalization (bn2), ReLu (relu2), dropout (drop2) and max pooling (max2), and fully connect (FC) layers are located. The mentioned layers are used to compose the CNN part of the hybrid CNN and LSTM network. The FC layer is connected to the unfolding layer. After the unfolding layer, a flattened layer is used for converting the output of the FC layer to the sequential data which is proper for the input of the LSTM layer. After the LSTM layer, another fully connected layer (FC2), softmax, and classification layers are located.

In CNN, the convolution operation, which is denoted as "$*$", is employed in convolutional layers. The input data and convolutional filters are used in the convolutional layers. The size of the filters can be $n \times m$ and a padding process can be used as an option in the convolutional layer for tuning the convolutional area. The main aim of the convolution operation is to extract features by finding similar local regions of the input data. The 2D convolution operation is defined as follows:

$$(X * F)(i,j) = \sum_m \sum_n F(m,n)X(i-m, j-n) \qquad (1)$$

Where, X and F indicate the input data and the filter, respectively.

In CNN architecture, the batch normalization (BN) layer is utilized to decrease training time and enhance network initialization performance. Besides, the vanishing gradient problem is reduced by using the BN layer. By using the mini-batch mean b_m and mini-batch variance b_v of the input data, the BN layer output y_i can be calculated as follows:

$$\hat{x}_i = \frac{x_i - b_m}{\sqrt{b_v^2 + \epsilon}} \qquad (2)$$

$$y_i = b\,\hat{x}_i + a \qquad (3)$$

where $\hat{x}_i$ is the normalized activation, and ϵ is a constant that was used to balance the numerical result if b_v is very small. Scale variable a and balance variable b, which are adjustable parameters are updated for the best y_i during optimization. The Rectified Linear Unit (ReLU) layer is generally used as an activation function in CNN architectures. The ReLU equation is defined as follows;

$$(x) = \max(0, x) \qquad (4)$$

According to Eq.4, the input data equals zero if it is negative, otherwise, the input is equalized to the output. In the flattened layer, 2D data conveyed from the previous layer is turned into 1D data for transmission to the FC layer. An LSTM unit contains input, output, and forgets gates, respectively. LSTM network can be seen as an improved version of a recurrent neural network (RNN) model. The LSTM unit keeps values determined in a prior time through these gates, and these gates control the data transmission in the units [17]. Besides, the LSTM layer significantly reduces

the gradient vanishing and explosion problems. The structure of the forgetting gate resembles a single-layer neural network (SLNN). According to Eq. (5), the forget gate activates if the output is equal to 1.

$$f_t = \sigma(W[x_t, h_{t-1}, C_{t-1}] + b_f) \quad (5)$$

Where, x_t represents the input of the existing LSTM unit, h_{t-1} represents the output vector of a prior LSTM unit, C_{t-1} represents the memory of prior LSTM unit, b_f represents the biased values, W represents the weighted vector, and σ represents a logistic sigmoid function. The input gate is a structure in which the existing memory is composed of an SLNN with the hyperbolic tangent function and the prior memory unit values. These computations are given in Eq. (6) and Eq. (7).

$$i_t = \sigma(W[x_t, h_{t-1}, C_{t-1}] + b_i) \quad (6)$$

$$\mathrm{C_t} = \mathrm{f_t}.\,\mathrm{C_{t-1}} + \mathrm{i_t}.\tanh([\mathrm{x_t}, \mathrm{h_{t-1}}, \mathrm{C_{t-1}}]) + \mathrm{b_c} \quad (7)$$

The data and information transmitting from the existing LSTM unit are conveyed to the output gate. The computations in the output gate are given in Eq. (8) and Eq. (9).

$$\sigma_t = \sigma(W[x_t, h_{t-1}, C_{t-1}] + b_o) \quad (8)$$

$$h_t = o_t.\tanh(C_t) \quad (9)$$

In the FC layer, all neurons of the prior and following layers are connected. The neuron values provide information about how well a value fits any class [18]. The data in the last FC layer is transmitted to the softmax layer with the class possibility scores. The dropout layer prevents overfitting by equalizing some input values to zero with a certain possibility when the training going on [19]. The softmax layer is used as the core classifier for CNNs. The softmax function is as follows:

$$S^k = \frac{e^{x^k}}{\sum_{i=1}^{n} e^x} \quad (10)$$

The output vector S^k is calculated for each input value (x^k), and the sums of all output values are equal to 1.

4. EXPERIMENTAL RESULTS

4.1 Dataset

The dataset, which contains English SMS messages, is used in the experimental works. The dataset can be downloaded from the UCI repository [20]. These SMS messages were collected from a United Kingdom public forum. The dataset is composed of a total of 5574 SMS messages where 747 SMS messages are from the Spam class and 4827 SMS messages are from the Ham class as given in Table 3. The dataset is available online at https://www.kaggle.com/code/adepvenugopal/detecting-sms-spam-using-machine-learning/data.

Table 3. The statistics related to the dataset.

Class	Number of samples	Percentages
Spam	747	13.4%
Ham	4827	86.6%
Total	5574	100%

The percentages of the dataset are also given in the third column of Table 1. While Spam SMS messages are 13.4% of the total dataset, the percentage of Ham SMS messages in total is 86.6%. Fig. 4 also shows the class distribution of the dataset.

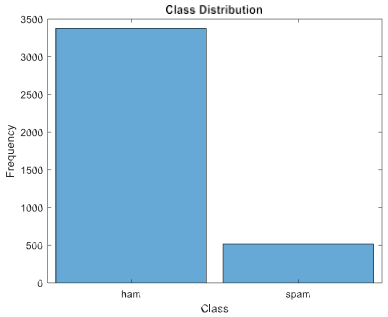

Figure. 4 Class distribution of the dataset.

4.2 Results and Discussion

All experimental works were carried out in MATLAB by using a workstation having Quadro M4000 GPU. The deep learning toolbox was used for the construction of the proposed deep models. A randomly selected 70% dataset was used in training and the rest 30% was used for performance testing of the proposed methods. Fig. 5 shows the visualization of the training text data using a word cloud.

For all deep network approaches, the sequence length, which indicated the length of numeric data that was converted from the text data, was assigned to 48. For the deep CNN approach, the number of filters in the convolutional layer was assigned to 200. The dropout probability was set to 0.2. The MiniBatch size and the number of MaxEpoches were chosen as 128 and 15, respectively. For the LSTM approach, the word embedding dimension was set to 50 and the number of neurons in the LSTM layer was set to 80. The MiniBatch size, GradientThreshold, and the number of MaxEpoches were selected as 16, 2, and 3, respectively. Lastly, for the CNN+LSTM approach, the number of filters in convolutional and LSTM layers were assigned to 200 and 100, respectively. The dropout probability was also set to 0.2. Besides, the MiniBatch size and the number of MaxEpoches were chosen as 128 and 3, respectively.

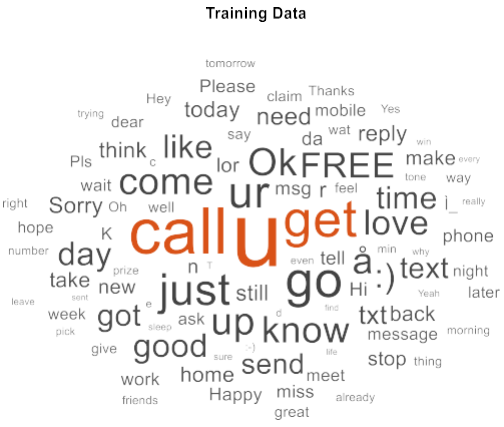

Figure 5. The visualization of the training text data using a word cloud.

Fig. 6 shows the training and testing progress of the CNN network. As seen in Fig 6, while the blue curve indicates the training progress, the black dashed curve shows the testing process. The CNN was trained in 450 iterations, which elapsed time was 55 seconds. The initial performance of the CNN model was around 20% and it increased toward 98% at 50 iterations. The obtained classification accuracy was 98.75%.

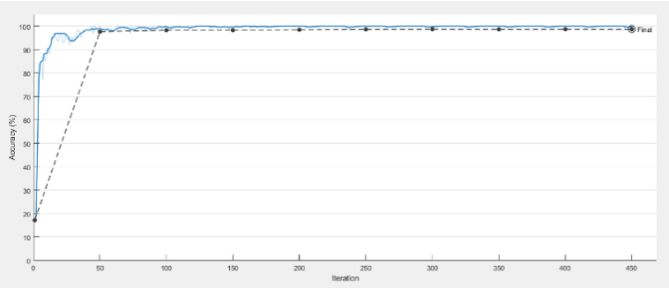

Figure 6. The training and testing progress of the CNN

Fig. 7 shows the confusion matrix that was obtained via the CNN model. While the columns of the confusion matrix show the true classes, the rows show the predicted classes. From Fig. 7, it is seen that 2 Ham samples were classified as Spam and 19 Spam samples were misclassified.

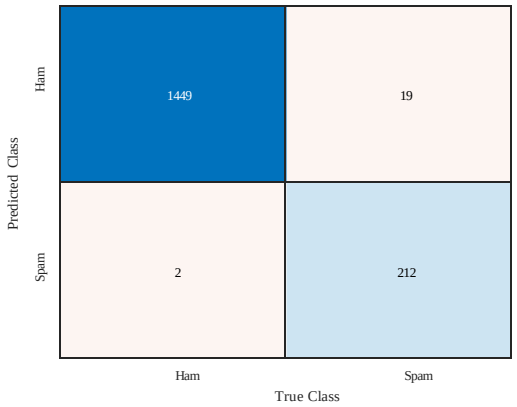

Figure 7. Confusion matrix for obtained CNN

Table 4 shows the calculated evaluation metrics namely, accuracy, precision, recall and F-score, for the CNN model. The rows of Table 2 show the Ham and Spam classes and the columns show the evaluation metrics. As mentioned earlier, the classification accuracy for the CNN model was 98.75%. The precision values for Ham and Spam classes were 99.86% and 91.17%, respectively. Besides, the recall scores for Ham and Spam classes were 98.71% and 99.07% and lastly, the F-scores for Ham and Spam classes were 99.28% and 95.28%, respectively.

Table 4. Evaluation metrics for CNN performance

Class	Accuracy	Precision	Recall	F-score
Ham	98.75%	99.86%	98.71%	99.28%
Spam	98.75%	91.17%	99.07%	95.28%

Fig. 8 shows the training and testing progress of the LSTM network. The training progress was completed at 729 iterations. The initial testing accuracy of the LSTM network was around 87% and it reached its final accuracy score of 98.69 % at 3 epochs. Similarly, the loss value was around 0.7 at the beginning of the training progress of the LSTM network and its final score was lower than 0.1.

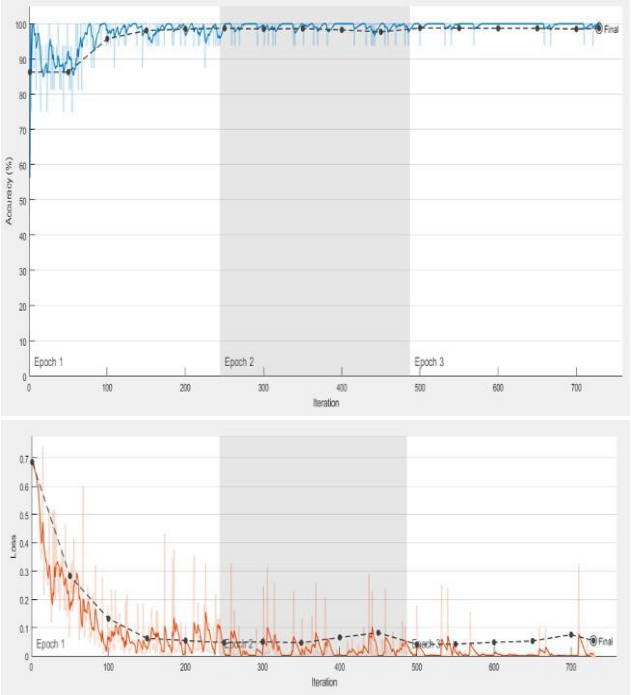

Figure 8. The training and testing progress of the LSTM

Fig. 9 shows the confusion matrix that was obtained by using the LSTM network. From Fig. 9, it is seen that the samples from the Ham class were classified with as 100% accuracy score and 22 samples from the Spam class were classified as Ham.

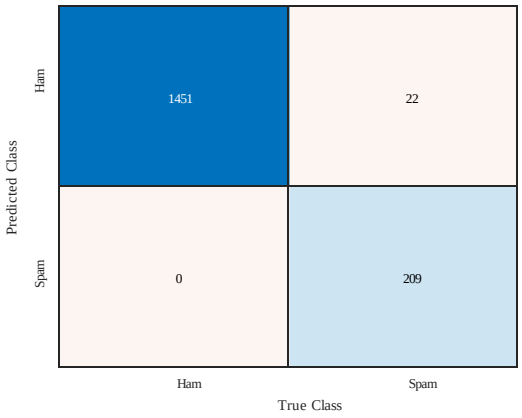

Figure 9. Confusion matrix for obtained LSTM

The calculated accuracy, precision, recall and F-score values for the LSTM network were given in Table 5. The obtained classification accuracies were 98.69% for both Ham and Spam classes. The precision values for Ham and Spam classes were 100% and 90.48%, respectively. Besides, the recall scores for Ham and Spam classes were 98.51% and 100% and lastly, the F-scores for Ham and Spam classes were 99.25% and 95.00%, respectively.

Table 5. Evaluation metrics for LSTM performance

Class	Accuracy	Precision	Recall	F-score
Ham	98.69%	100%	98.51%	99.25%
Spam	98.69%	90.48%	100%	95.00%

Finally, Fig. 10 shows the training and testing progress of the CNN+LSTM network. The training progress was completed at 180 iterations. In other words, the training progress was completed in 37 seconds. The initial testing accuracy of the CNN+LSTM network was around 80% and it reached its final accuracy score of 98.81% at 6 epochs. Similarly, the loss value was around 0.7 at the beginning of the training progress of the CNN+LSTM network and its final score was close to 0.

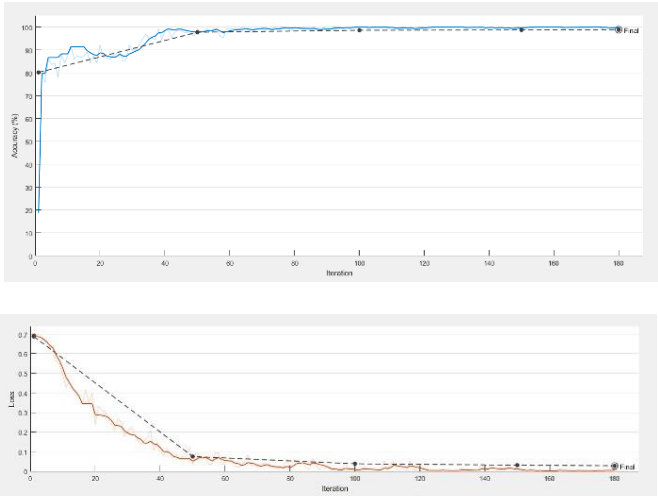

Figure 10. The training and testing progresses of the hybrid model of CNN and LSTM

Figure 11 shows the confusion matrix that was obtained by using the CNN+LSTM network. As seen in Fig. 9, 4 Ham samples were classified as Spam and 16 samples from the Spam class were classified as Ham.

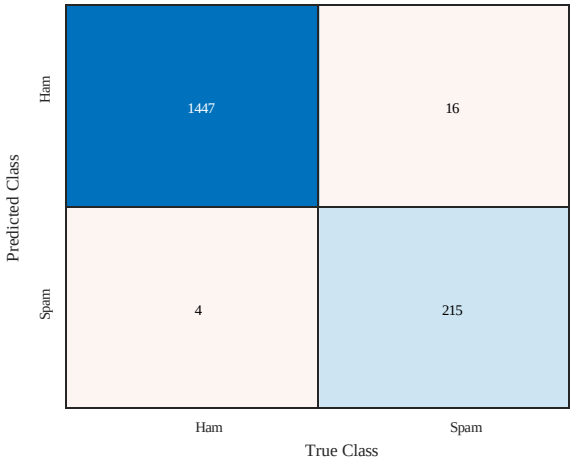

Figure 11. Confusion matrix for obtained model of CNN and LSTM

The calculated accuracy, precision, recall, and F-score values for the CNN+LSTM network were given in Table 6. The obtained classification accuracies were 98.81% for both Ham and Spam classes. The precision values for Ham and Spam classes were 99.72% and 93.07%, respectively. Besides, the recall scores for Ham and Spam classes were 98.91% and 98.17% and lastly, the F-scores for Ham and Spam classes were 99.31% and 95.56%, respectively.

Table 6. Evaluation metrics for hybrid model based CNN and LSTM performance

Class	Accuracy	Precision	Recall	F-score
Ham	98.81%	99.72%	98.91%	99.31%
Spam	98.81%	93.07%	98.17%	95.56%

As the obtained results were compared, it was seen that the hybrid CNN and LSTM produced the highest accuracy score among all investigated approaches. Besides, the highest precision, recall, and F-score values for the Ham class were obtained by LSTM, CNN, and the hybrid model-based CNN and LSTM, respectively. The highest precision, recall, and F-score values for the Spam class were produced by the hybrid proposed model. Furthermore, we compared the obtained highest accuracy score with some of the existing methods. The comparisons were given in Table 6. In comparison, for the Naïve Bayes (NB), support vector machine (SVM), and latent Dirichlet allocation (LDA) methods, the overall evaluation scores were given and for the Hidden Markov Model (HMM) method, the class-based evaluation scores were given. As a comparison between accuracy scores was considered, it was seen that the proposed method produced the highest accuracy score. The second-best accuracy score of 95.90% was produced by the HMM approach. SVM and LDA methods produced 93.60% and 90.40% overall accuracy scores, respectively. The NB approach produced the worst accuracy score.

Table 7. Comparison of the proposed method with some of the other methods

Method	Class	Accuracy	Precision	Recall	F-score
NB [21]	Overall	84.20%	95.00%	97.20%	87.00%
SVM [21]	Overall	93.60%	**97.00%**	97.70%	94.00%
LDA [21]	Overall	90.40%	96.00%	97.60%	92.00%
HMM [1]	Ham	95.90%	89.20%	81.60%	85.20%
	Spam	95.90%	**96.90%**	98.30%	**97.60%**
CNN+ LSTM	**Ham**	**98.81%**	**99.72%**	**98.91%**	**99.31%**
	Spam	**98.81%**	**93.07%**	**98.17%**	**95.56%**
	Overall	**98.81%**	96.39%	**98.54%**	97.43%

5. CONCLUSIONS

In this paper, three deep learning approaches namely, CNN, LSTM, and proposed hybrid model-based CNN and LSTM were used for SMS Spam detection. Among these approaches, the proposed hybrid approach produced better results than the CNN and LSTM approach. A further comparison of the hybrid approach with some of the existing machine-learning approaches was also carried out in the presented work. The comparison showed that the hybrid approach outperformed the compared machine learning approaches. In the hybrid approach, there were several parameters which were needed to be tuned for high performance. These parameters were adjusted manually in this work, which can be seen as a drawback. A metaheuristic optimization approach can be used for solving the mentioned problem in future works. Besides, the length of the input numeric sequence data, which was constructed from the text data, needed to be investigated for performance improvement. These all issues will be explored in our future works. Finally, deeper hybrid architecture will be investigated in future works.

REFERENCES

[1] Xia, T., & Chen, X. (2020). A discrete hidden Markov model for SMS spam detection. *Applied Sciences*, 10(14), 5011.

[2] Ali, S. S., & Maqsood, J. (2018). Net library for SMS spam detection using machine learning: A cross platform solution. *In 2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)* (pp. 470-476). IEEE.

[3] Yadav, K., Kumaraguru, P., Goyal, A., Gupta, A., & Naik, V. (2011). SMSAssassin: crowdsourcing driven mobile-based system for SMS spam filtering. *In Proceedings of the 12th Workshop on Mobile Computing Systems and Applications* (pp. 1-6).

[4] Zebari, D. A., Haron, H., Sulaiman, D. M., Yusoff, Y., & Othman, M. N. M. (2022, December). CNN-based Deep Transfer Learning Approach for Detecting Breast Cancer in Mammogram Images. In *2022 IEEE 10th Conference on Systems, Process & Control (ICSPC)* (pp. 256-261). IEEE.

[5] Ghourabi, A., Mahmood, M. A., & Alzubi, Q. M. (2020). A hybrid CNN-LSTM model for SMS spam detection in arabic and english messages. *Future Internet*, 12(9), 156.

[6] Goel, D.; Jain, A. Smishing-Classifier: A Novel Framework for Detection of Smishing Attack in Mobile Environment. In Proceedings of the Smart and Innovative *Trends in Next Generation Computing Technologies (NGCT 2017)*, Dehradun, India, 30–31 October 2017; pp. 502–512.

[7] Goel, D., & Jain, A. K. (2018). Mobile phishing attacks and defence mechanisms: State of art and open research challenges. *computers & security*, 73, 519-544.

[8] Jain, A. K., Yadav, S. K., & Choudhary, N. (2020). A novel approach to detect spam and smishing SMS using machine learning techniques. *International Journal of E-Services and Mobile Applications (IJESMA)*, 12(1), 21-38.

[9] Sheikhi, S., Kheirabadi, M. T., & Bazzazi, A. (2020). An effective model for SMS spam detection using content-based features and averaged neural network. *International Journal of Engineering*, 33(2), 221-228.

[10] Kondamudi M. Classifying and predicting spam messages using text mining in SAS® Enterprise minerTM. *South Central SAS Users Group (SCSUG 2017)*; 2017. https://www.sas.com/content/dam/SAS/support/en/sas-global /2650-2018.pdf

[11] Abayomi-Alli, O., Misra, S., & Abayomi-Alli, A. (2022). A deep learning method for automatic SMS spam classification: Performance of learning algorithms on indigenous dataset. *Concurrency and Computation: Practice and Experience*, e6989.

[12] Mishra, S.; Soni, D. Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis. *Future Gener. Comput. Syst*. 2020, 108, 803–815.

[13] Yang, H., Liu, Q., Zhou, S., & Luo, Y. (2019). A spam filtering method based on multi-modal fusion. *Applied Sciences*, 9(6), 1152.

[14] Saleh, A. J., Karim, A., Shanmugam, B., Azam, S., Kannoorpatti, K., Jonkman, M., & Boer, F. D. (2019). An intelligent spam detection model based on artificial immune system. *Information*, 10(6), 209.

[15] Mardi, V.; Kini, A.; Sukanya, V.M.; Rachana, S. Text-Based Spam Tweets Detection Using Neural Networks. *In Advances in Computing and Intelligent Systems;* Springer: Singapore, 2020; pp. 401–408.

[16] Sharaff, A., Kamal, C., Porwal, S., Bhatia, S., Kaur, K., & Hassan, M. M. (2021). Spam message detection using Danger theory and Krill herd optimization. *Computer Networks*, 199, 108453.

[17] Theodorus, A., Prasetyo, T. K., Hartono, R., & Suhartono, D. (2021, April). Short Message Service (SMS) Spam Filtering using Machine Learning in Bahasa Indonesia. *In 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)* (pp. 199-203). IEEE.

[18] Jain, A.K.; Gupta, B.B. Feature Based Approach for Detection of Smishing Messages in the Mobile Environment. *J. Inf. Technol. Res*. 2019, 12, 17–35.

[19] Jain, G., Sharma, M., & Agarwal, B. (2019). Spam detection in social media using convolutional and long short term memory neural network. *Annals of Mathematics and Artificial Intelligence*, 85(1), 21-44.

[20] Li, A.; Qin, Z.; Liu, R.; Yang, Y.; Li, D. Spam review detection with graph convolutional networks. *In Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Beijing China, 3–7 November 2019.

[21] Wu, Y., Lian, D., Xu, Y., Wu, L., & Chen, E. (2020, April). Graph convolutional networks with markov random field reasoning for social spammer detection. *In Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 01, pp. 1054-1061).

[22] Bosaeed, S., Katib, I., & Mehmood, R. (2020, April). A fog-augmented machine learning based SMS spam detection and classification system. *In 2020 fifth international conference on fog and mobile edge computing (FMEC)* (pp. 325-330). IEEE.

[23] Wei, F., & Nguyen, T. (2020, October). A lightweight deep neural model for sms spam detection. *In 2020 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1-6). IEEE.

[24] Xia, T., & Chen, X. (2021). A weighted feature enhanced Hidden Markov Model for spam SMS filtering. *Neurocomputing*, 444, 48-58.

[25] Neisari, A., Rueda, L., & Saad, S. (2021). Spam review detection using self-organizing maps and convolutional neural networks. *Computers & Security*, 106, 102274.

[26] Hirway, C., Fallon, E., Connolly, P., Flanagan, K., & Yadav, D. (2022, December). A Deep Learning Approach for Minimizing False Negatives in Predicting Receipt Emails. *In 2022 International Conference on Computer and Applications (ICCA)* (pp. 1-7). IEEE.

[27] Al Bataineh, A., & Kaur, D. (2021). Immunocomputing-based approach for optimizing the topologies of LSTM networks. *IEEE Access*, 9, 78993-79004.

[28] Zebari, D. A., Sadiq, S. S., & Sulaiman, D. M. (2022, March). Knee Osteoarthritis Detection Using Deep Feature Based on Convolutional Neural Network. *In 2022 International Conference on Computer Science and Software Engineering (CSASE)* (pp. 259-264). IEEE.

[29] Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630*.

[30] Zebari, N. A., Alkurdi, A. A., Marqas, R. B., & Salih, M. S. (2023). Enhancing brain tumor classification with data augmentation and densenet121. *Academic Journal of Nawroz University*, *12*(4), 323-334.

[31] Mohammed, H. J., et al. (2022). ReID-DeePNet: A hybrid deep learning system for person re-identification. *Mathematics*, *10*(19), 3530..