Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2023, Article ID 6648970, 16 pages
https://doi.org/10.1155/2023/6648970

Research Article

An Intelligent Framework Based on Deep Learning for SMS and e-mail Spam Detection

Umair Maqsood,[1] Saif Ur Rehman,[1] Tariq Ali,[1] Khalid Mahmood,[2] Tahani Alsaedi,[3] and Mahwish Kundi[4]

[1]*University Institute of Information Technology, PMAS-Arid Agriculture University, Rawalpindi 46000, Pakistan*
[2]*Institute of Computing and Information Technology, Gomal University, D.I. Khan, KP, Pakistan*
[3]*Applied College, Taibah University, Madina 42353, Saudi Arabia*
[4]*University of Leicester, Leicester, UK*

Correspondence should be addressed to Saif Ur Rehman; saif@uaar.edu.pk

Received 30 June 2023; Revised 31 July 2023; Accepted 3 September 2023; Published 20 September 2023

Academic Editor: Jyotismita Chaki

Copyright © 2023 Umair Maqsood et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The use of short message service (SMS) and e-mail have increased too much over the last decades. 80% of people do not read e-mails while 98% of cell phone users daily read their SMS. However, these communication media are unsafe and can produce malicious attacks called spam. The e-mails that pretend to be from a trusted company to provide "financial or personal information" are phishing e-mails. These e-mails contain some links; users might download malicious software on their computers when they click on them. Most techniques and models are developed to automatically detect these "SMS and e-mails" but none of them achieved 100% accuracy. In previous studies using machine learning (ML), spam detection using a small dataset has resulted in lower accuracy. To counter this problem, in this paper, multiple classifiers of ML and a classifier of deep learning (DL) were applied to the SMS and e-mail dataset for spam detection with higher accuracy. After conducting experiments on the real dataset, the researchers concluded that the proposed system performed better and more accurately than previously existing models. Specifically, the support vector machine (SVM) classifier outperformed all others. These results suggest that SVM is the optimal choice for classification purposes.

1. Introduction

People are increasingly communicating via cell phone text messaging. SMS has grown in popularity over the previous decades. From 2012 to 2022, the average number of SMS sent each month rose by a stunning 7700%. Other than e-mails, simple messages are most successful for businesses. The reason is that, although 98% [1] of mobile users have read their SMS before the day's end, around 80% of e-mails stay unread (comparison of SMS, 2020). Therefore, it is evident why the short service message has grown into a billion-dollar industry [2]. Both domestic and foreign senders send these messages. According to a report, 68% of cell phone users are affected by SMS spam, with teens being the most affected group [3, 4].

Furthermore, unlike e-mails, which benefit from robust spam filtering, SMS spam filtering is currently inadequate. Most studies that categorize SMS spam are limited by inefficient human-engineered features [5, 6]. Prior knowledge and subject experience are required for identifying potential characteristics for correct categorization. Even yet, the feature selection must be reevaluated based on parameters such as information gain and significance graph.

An iterative time-consuming, error, and trial approach is to be expected [7–9]. Using deep neural networks is one way to avoid this wasteful feature engineering process. Deep learning (DL) is a machine learning (ML) approach that uses multiple layers of information processing to automatically categorize patterns. It involves unsupervised learning to extract features and iterative self-training to reduce categorization errors. The increased use of social networks and Internet online transmission has set off an important section of our everyday lives. E-mail is a popular medium for formal and commercial communication due to its ease of use, speed, and dependability. Spam e-mails were rapidly increasing in popularity as e-mail became more popular. Detection of automatic spam is not a current issue [10]; businesses and firms are constantly looking for methods to improve their users' experiences to protect their PCs against potential damage if viruses were included in spam e-mails. Moreover, it helps in the conservation of network resources and time. Communities of ML and natural language processing (NLP) have been attracted to solve this problem and have provided various detections of spam datasets and train models to analyze the complete record [11]. Although because it is built on the Internet, security issues will automatically arise at several levels, which hackers and malevolent groups might use to achieve various goals in terms of theft of identity and financial revenue. This system is vulnerable to various attacks due to its weaknesses and characteristics, which makes it a target for commercial spammers worldwide. According to a study conducted in 2014, spam e-mail is a significant threat to e-mail systems, as it generates data traffic that is expected to account for nearly 90% of all e-mail traffic [12]. Furthermore, almost 236 billion e-mails were sent in 2018 daily [13] with 53.5% of them being spam. Every day, over 320 billion spam e-mails are generated, and this medium is used to spread 94% of malware. The total loss was predicted to be twelve billion dollars because of spam e-mails sent to business e-mail subscribers [14]. Figure 1 explains that according to the Statista reportof 16th January 2023, within one day worldwide country, a huge number of spam e-mails were transferred to the U.S. around 8.6 billion and Czechia and the Netherlands ranked second and third, with 7.7 and 7.6 billion, respectively.

There are five main categories of spam [16]: mobile spam, messaging spam, e-mail spam, SEO spam, and social networking spam. Figure 2 describes common e-mail spam types. According to a virus analysis, 94% of malware was sent via e-mail. Most spam e-mails have an attachment, and 45% of those attachments are Office doc files. Windows programs came in second, with 26%, as another method of virus transmission via spam e-mail.

Communication media such as SMS and e-mail facilities play an important character in facilitating information sharing and access in the 21st century. As a result, the mobile phone, a wonderful technical creation for transmission, has become a vital part of people's lives, and it is made up of numerous components that serve as a means of communication. SMS is one of its most often utilized components nowadays. It enables customers of the Global System for Mobile Communication Association (GSMA), Time Division Multiple Access (TDMA), and Code Division Multiple Access (CDMA) mobile networks to transfer messages of up to one sixty characters using a "save, forward" protocol [18]. Other transmission mechanics, such as cell phone banking apps, special apps, networking social apps, and health of initiatives apps, have all communicated with app users via e-mail and text messages. According to the GSMA projection study [19], 3.8 billion people worldwide utilize the smartphone Internet.

According to another poll [20], the most common actions by cell phone owners 91% and 90%, respectively, are obtaining SMS and e-mail messages. The widespread use of SMS and e-mail for communication has made them vulnerable to malicious attacks, such as spam. To combat this, various artificial intelligence (AI) algorithms, including DL, are used to identify and analyze potential threats. Trustworthy third-party cash physical unclonable function (PUF) protocols also support AI to enhance security, increase productivity, and isolate digital currency [21]. Spam detection has become an essential challenge in the rapidly evolving digital world. Previous studies focused solely on ML or DL models for spam detection, without comparing their efficiency. In contrast, this study explores classifiers from both ML and DL to identify the most effective model for detecting SMS and e-mail spam. This paper utilizes machine learning classifiers such as random forest (RF), multinomial Naïve Bayes (NB), support vector machine (SVM), and the deep learning algorithm convolutional neural network (CNN). These classifiers are applied to SMS and email datasets for performance assessment based on chosen metrics. TensorFlow, a library of Python, was applied to conduct the proposed research. We hope that this comparative analysis will be helpful to practitioners and scholars who are trying to improve spam detection in a wide range of digital communications.

The remaining research work is planned as Section 2 provides a review of existing research on the detection of spam in SMS and e-mail messages. Section 3 outlines the methods and materials used in this study to achieve the results. Section 4 presents the findings and offers a detailed discussion. The paper concludes with Section 5.

2. Literature Review

The following image data provide a comprehensive overview of the global spam landscape. Data on spam problems across various communication channels are presented in this article. With Cameroon leading the list of nations affected, spam SMS is still a serious issue. There are now 6.648 billion cell phone users worldwide or 83% of the world's population. Russia emerges as the top source of outgoing spam, with billions of spam e-mails being sent every day. In addition, China has the most live spam problems, which demonstrate the need for more effective antispam efforts in impacted areas. Figure 3 explains that according to the Truecaller report, the world's top twenty countries are affected by spam SMS in 2022, and Cameroon tops the list with the most average spam SMS per user per month. Most of these countries are in Africa, indicating that spam SMS is a significant problem across the African continent.

Figure 4 describes the total number of cell phone users worldwide. In accordance with statistics, the present cell phone global user population is almost 6.648 billion, which shows that 83% of the world population has a cell phone. Since "2016," this statistics has increased when only 3.668

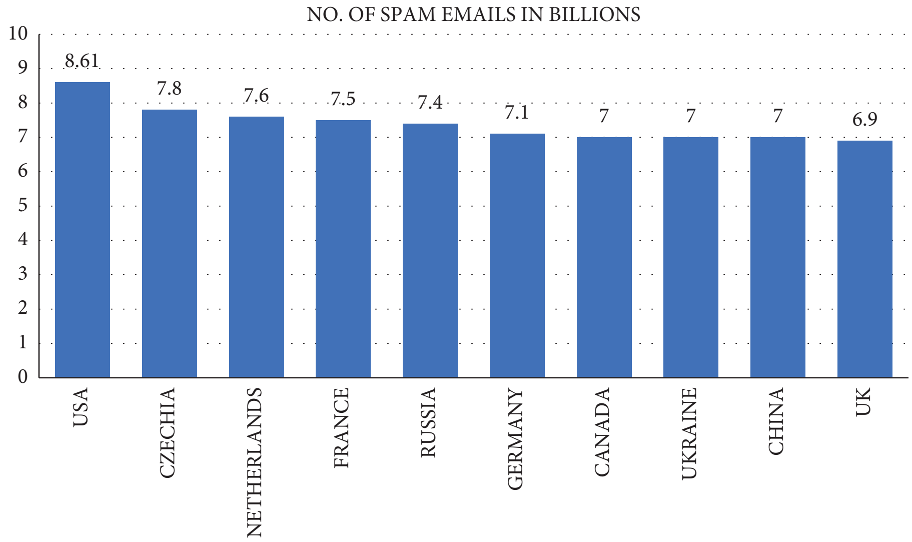

Figure 1: Worldwide daily spam e-mails [15].

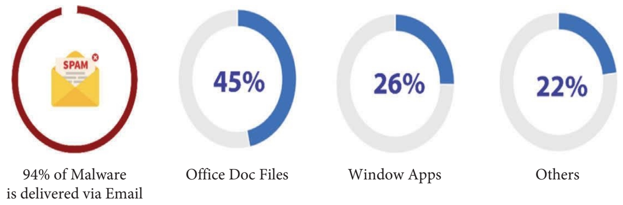

Figure 2: Common spam e-mail types [17].

billion users were there, accounting for 49% of the population of the world at that time.

Figure 5 describes that according to the secure list report, leading countries of outgoing spam, Russia tops the list with the largest source of spam with 23.5%, Germany second with 11%, and the U.S. third with 10.85%.

Figure 6 describes that from the 2021 spam has report regarding spam, it is indicated that China has the highest number of live spam issues. The issue of spamming becomes a huge problem when a country or state fails to take any action to address it. Simply, we can say that antispam issues are inadequate or nonexistent in these countries.

A few years back, scientists of computers were given various machine learning methods to separate spam from nonspam [24–36]. To mobile phone text messages, these works are not only limited but also include web spam e-mail [37] and spam on social media networks such as Facebook and Sina Weibo [26, 38–41]. A detailed literature review of previous studies is described below.

In a recent study presented by the authors in [42], the Naïve Bayes approach, K-nearest neighbor, and reverse DBSCAN algorithms are used to identify text and image-based spam e-mails. Before performing algorithms, the Enron corpus' e-mail dataset content is preprocessed using several feature extraction approaches such as Black and Whitelisting and utilizing the Tesseract Open Source Library developed by Google. The accomplishment of these three algorithms is dependent upon 4 factors that are correctness, accuracy, sensitivity, and specificity, with all algorithms achieving good results. These provided methods allow text in special fonts only. Researchers in [43] created a spam e-mail separation model using a high-level integration algorithm and then used SPEMC-11K to isolate the combination of BoW encodings and term frequency-inverse document frequency (TF-IDF) with support vector machine (SVM) classifiers, logistic regression (LR), and Naïve Bayes (NB). The experienced results describe that TF-IDF with NB scans e-mails at 2.13 ms for the fastest spam classification and term frequency-inverse document frequency that is associated with SVM for best micro *F*1 performs 95.39%.

In another work created to identify shocking messages, the authors in [44] presented a model called "S-detector." The given system consists of four modules. The first module is a determinant to classify and block. The second module is an analyzer to identify the short message service content. The third one is text messages of spam. The latter is an SMS data storage website. The partitioning algorithm used in this model is NB. Researchers have introduced a law-abiding classification system to identify phishing SMS. Their method analyzes nine laws that can detect and distinguish SMS phishing from it. The authors revealing a 92% true level and 99% negative rating [45] carried out the following results. In a recent work [46], a feature-based method for spam message detection was described. This method removes 10 characteristics that the author claims can discriminate between fake and ham signals. Following that, the features were applied to a benchmarked dataset using five grouping methods to evaluate the performance of the suggested technique. According to the experimental tests, the model can detect stamped messages and can obtain a total accuracy of 98.74% with a positive accuracy rate of 94.20%.

In another study on the same problem, the authors in [47] introduced a model called "SmiDCA" which was for detecting spam messages using machine learning methods. The authors choose to apply correlation methods in the model. They used four different techniques to get the 39

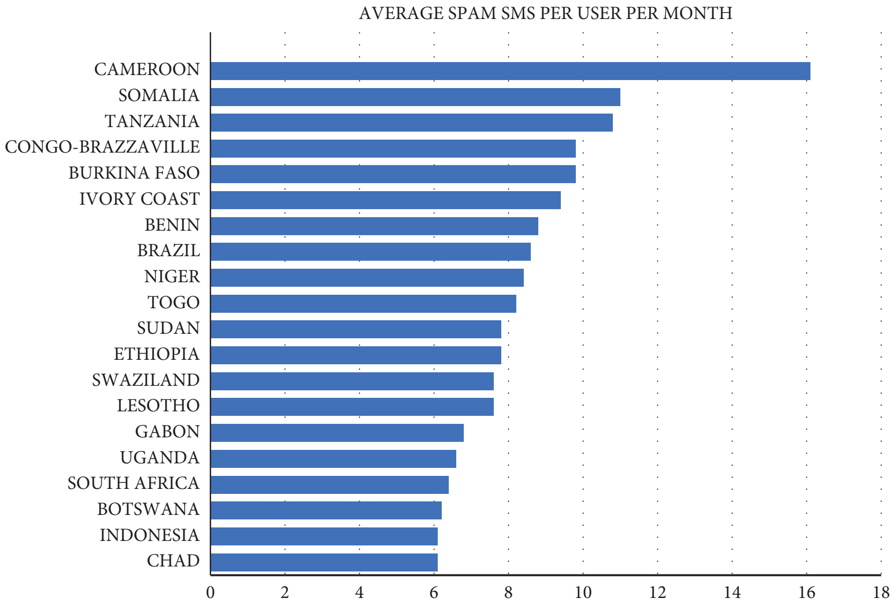

Figure 3: Topmost 20 countries affected by spam SMS in 2022 [22].

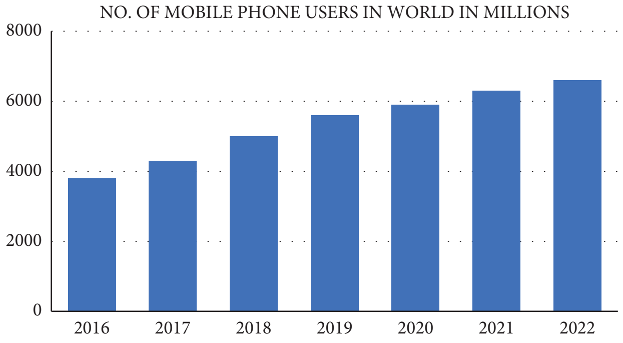

Figure 4: The statistics of mobile phone users globally [23].

most important elements from smishing messages, and machine-learning separators are used to test the efficiency of their model. The four classifiers were as follows: SVM, random forest, decision tree, and AdaBoost are all examples of machine learning algorithms. The experiential evaluation of this model's accuracy, using the random forest classifier, was 96.4%. Another relevant study [48] suggested a model-based vector support system to distinguish between phishing and nonphishing e-mails. The author aimed to increase the model's performance by identifying the optimal parameter configuration, as SVM's performance is highly dependent on its parameters. A parameter search technique was developed specifically for SVM to achieve this objective. SVM was used in this work since it can provide improved predicted accuracy with a small number of samples. They have concentrated all their efforts on simply improving the detection accuracy. In the training phase, this system has time complexity. A detection system should be concerned not just with prediction accuracy but also with timely performance.

To identify striking messages with a reduced level of the false positive rate, researchers in [49] introduced a model named "Smishing Detector." Spam is a new type of cybercrime, and its name is a portmanteau of SMS and phishing. The presented model consists of 4 modules. The system described in the study consists of four modules, each with a specific function. The first module utilizes the NB classification algorithm to detect harmful content and identify the message's text. The second module identifies URLs within the messages, while the third module focuses on identifying the source code of websites linked in the messages. The researchers report that the system got an accuracy of 96.29% in their analysis.

The authors proposed a spam detection approach based on support vector machines (SVMs) that achieved 94% accuracy [50]. To reduce the number of textual features, they employed a semantic feature selection methodology that used WordNet ontology and several semantic-based algorithms and measures. They also used analysis of principal components and correlation feature selection (CFS) to identify the most informative features. Compared to other classifiers such as J48, NB, RF, and radial basis function networks, logistic regression yielded the highest accuracy. In a recent study presented in [51], deep learning was proposed as an effective technique for differentiating between spam and nonspam messages. The proposed approach involved combining two deep learning architectures, namely, CNN and LSTM, with the aim of accurately categorizing messages and distinguishing between spam and nonspam messages. They evaluated the suggested approach's performance to that of several ML methods such as stochastic gradient descent, logistic regression, gradient boosting, RF, and NB. The findings demonstrated that the LSTM and CNN models perform best when it is balanced with other models of machine learning.

This study [52] used the term frequency-inverse document frequency approach in conjunction with a classifier of RF and got 97.5% accuracy. This approach quantifies the words in a document by combining two measures: inverse

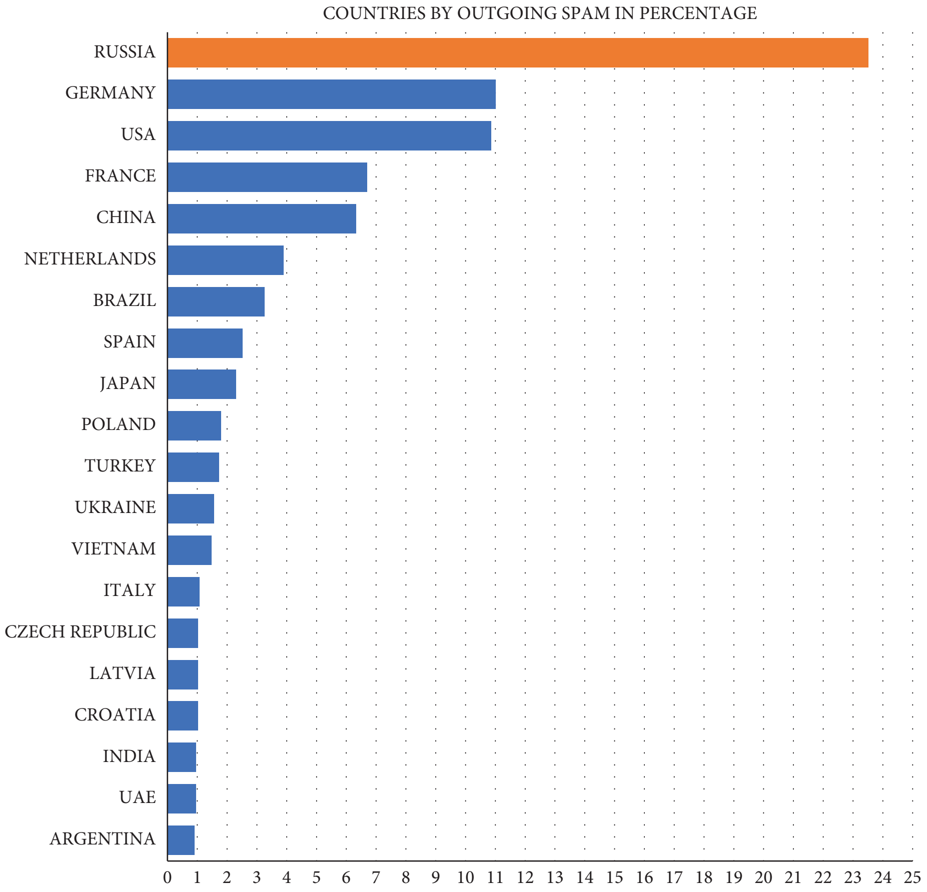

Figure 5: Topmost countries by outgoing spam [15].

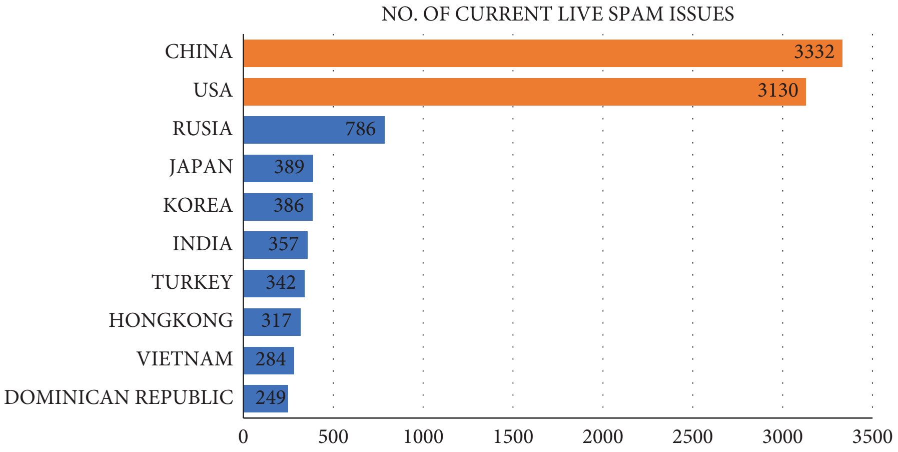

Figure 6: Countries with a huge number of live spam issues [17].

document frequency and term frequency. In the study in [53], the authors used four machine learning classifiers for e-mail spam filtering: decision trees, LR, NB, and RF, and obtained an accuracy of 97% with the RF classifier. Researchers developed various machine learning strategies and obtained 97% accuracy with SVM [54]. The authors in [55] introduced a hybrid classifier for sentiment analysis and SMS spam classification. The approach involves preprocessing the datasets and using Word2vec data augmentation to extract features. Authors used six different feature selection methods and equilibrium optimization (EO) to select the optimal set of features, which were then used to train a hybrid classifier based on KNN and SVM. To enhance accuracy, the optimization algorithm rat swarm optimization (RSO) was used to optimize the network parameters. AFINN and SentiWordNet were used for the sentiment analysis. The performance of the proposed framework was evaluated on three benchmark datasets, and it achieved a high spam detection accuracy of 99.82% on the spam dataset.

Multiple deep neural network models for spam message classification using Tiago's dataset have been introduced in the study in [56]. The initial step involves preprocessing the messages in the dataset by lowercasing the text, tokenizing, lemmatizing, and removing numbers, punctuations, and stop words and also utilizing two deep learning models with simpler architectures, namely, CNN and a hybrid model, that combine the CNN with long short-term memory network (LSTM) for classification. To improve the accuracy of these models, they integrated two-word embedding techniques, binary unique number of word (BUNOW) method and glove, which are commonly used in the sentiment analysis but can also be applied to text classification. The maximum accuracy of 98.44% was achieved by CNN, LSTM, and BUNOW models after 15 iterations, with a 70%–30% train-test split.

The authors in [57] proposed an ensemble system for detecting e-mail spam using bagging and boosting techniques in machine learning. The Ling-Spam Corpus dataset was used for experimentation. The system uses a combination of the multinomial Naïve Bayes (MNB), J48 decision tree classifiers, and then applies the AdaBoost algorithm to convert weak classifiers into strong ones. The study conducted three distinct trials and compared their results. In the first trial, individual classifiers were employed, while the second trial utilized a bagging approach, and the third experiment implemented a boosting approach. Evaluation metrics were used to determine the effectiveness of the ensemble methods in comparison to the individual classifiers. Overall, the study successfully presented a method for detecting e-mail spam.

Researchers in [58] presented a novel framework for identifying spam images. The study focused on analyzing two categories of images: spam images containing unwanted or harmful content and ham images that were free from such content. The proposed approach used various pretrained deep learning models, including InceptionV3, DenseNet121, ResNet50, VGG16, and MobileNetV2, to detect and filter out spam images. Using common test datasets as the Dredze dataset, the image spam hunter (ISH) dataset, and the improved dataset, the performance of the suggested technique was evaluated. They enhanced accuracy by specifically substituting a support vector machine (SVM) classifier for the fully connected layer of pretrained models. With 99.87% accuracy, 99.88% area under the curve (AUC), 99.98% sensitivity, 99.79% precision, and a 98.99% $F1$-score, the experimental findings showed that the ResNet50 model performed the best and the ISH dataset's computational testing took between one and two seconds.

The authors in [59] presented a new approach to spam classification that utilizes passive-aggressive spectrum algorithms with genetic optimization. The proposed algorithm was compared to existing methods for spam classification and is shown to be robust. The study also investigated the impact of hyperparameters on classification accuracy. The spam SMS dataset, spam review dataset, and Twitter spam dataset were used to evaluate the proposed algorithm, with 80% of each dataset used for training and 20% for testing. The experimental results show that the suggested method outperformed benchmark standard algorithms for spam classification in terms of accuracy, precision, and recall scores.

The authors in [60] presented a model based on the back propagation algorithm and its variant with momentum to identify spam. To increase classification performance, the researchers used SGO to tune the model. For both "clustering and classification purposes," neural networks are used to deal with any sort of data such as audio pictures and text. Back propagation has one downside in that it requires more repetitions, which increment computation time. The authors made no mention of reducing outliers, even though outliers have a significant influence on textual datasets, and normalization techniques frequently increase variance among data points. They just added two secret layers. Hidden layers with fewer numbers always do not give better performance.

A unique technique for screening spam SMS is based on two techniques of data mining. The first one is Naïve Bayes and the second one is frequent pattern (FP) growth [61]. The FP growth method is used to extract frequent item sets from text messages, while a classifier Naïve Bayes is used to categorize the messages and remove spam. The experiential results that were carried out by researchers on this technique yielded an average accuracy of 98.5%.

Table 1 summarizes the literature review based on the techniques used, the dataset, and with results. In most of the approaches, machine learning techniques have been used. Most of the techniques have demonstrated results on private datasets, whereas few approaches use benchmark datasets. Accuracy for most of the techniques is good due to the limited size of the dataset. The comparative study of the literature is described as follows.

3. Proposed Methodology

This study follows a structured methodology consisting of five distinct phases to investigate the effectiveness of a classification model on a given dataset. The first phase involves data collection, where relevant data are gathered from a reliable source in an appropriate format. The second phase is data preprocessing, which involves cleaning and preparing the data for analysis by handling missing data, removing duplicates, and transforming the data into a consistent format for feature extraction. The third phase is feature extraction, where the most important features are identified and transformed into a suitable format for the classification model. In the fourth phase, the dataset is split into training and testing sets to evaluate the model's performance, while in the fifth and final phase, a suitable classification algorithm is chosen and trained using the selected features and the training data to optimize its performance in predicting the target variable. The resulting output of this methodology is a well-trained classification model that accurately predicts the target variable based on the selected features.

Table 1: Comparative study of literature.

Objective approach	Methods/algorithms	Dataset type	Accuracy/results
The text identifies spam e-mails as image based [42]	KNN, reverse DBSCAN, and NBB	Enron Corpus'	Not mentioned
To classify the combination of BoW encodings and TFiDF with SVM classifiers, logistic regression, and Naïve Bayes [43]	Hierarchical clustering algorithm	Private dataset	95.39% accuracy
Analyzed spam messages and identified SMS [44]	NB	Private dataset	Made a model called S-detector
Analyzed spam messages and identified SMS [45]	Classification of rule based	Repository of ML (UCI): SMS spam collection of datasets	92% true level and 99% negative rating
Analyzed spam messages and inspected comprised uniform resource locator [46]	SVM, neural network, NB, RF, and feature-based technique	Smishing message images (Pinterest) repository of ML (UCI): spam collection of SMS dataset	94.20% accuracy
Analyzed spam messages and identified SMS [47]	SVM, RF, decision tree, and AdaBoost	ML repository (UCI): spam SMS collection dataset	94.20% accuracy
To distinguish between spam and nonspam e-mails [48]	SVM	Private dataset	Not mentioned
Analyzed spam messages and inspected comprised URL [49]	NB	Smishing message images (Pinterest) ML repository (UCI): spam SMS collection dataset	96.29% accuracy
Semantic feature selection methodology [50]	Radial basis function networks, SVM, J48, RF, and NB	Enron spam dataset	94% accuracy
Analyzed spam messages and identified [51] SMS	LSTM and CNN	ML repository (UCI): spam SMS collection dataset	Not mentioned
Term frequency-inverse document approach in conjunction [52]	RF classifier	Private dataset	97.5% accuracy
Used 4 machine learning classifiers for e-mail spam filtering [53]	Decision trees, logistic regression, RF, and NB	Private dataset	97% accuracy
Developed various machine-learning strategies [54]	SVM	Private dataset	97% accuracy
The hybrid classifier was proposed for SMS spam classification [55]	KNN, SVM	Private dataset	99.82% accuracy
Deep neural network models for spam message classification using Tiago's dataset [56]	CNN LSTM, BUNOW	Private	99.84% accuracy
Ensembled system for detecting e-mail spam using bagging and boosting techniques [57]	(MNB) and J48 decision tree classifiers	Ling-Spam Corpus dataset	Best accuracy
A novel framework for identifying spam images [58]	InceptionV3, DenseNet121, ResNet50, VGG16, and MobileNetV2	Dredze dataset, image spam hunter (ISH) dataset, and improved dataset	99.87% accuracy
Investigated the impact of hyperparameters on classification accuracy [59]	Passive-aggressive spectrum algorithms	SMS spam dataset, spam review dataset, and Twitter spam dataset	Achieved higher accuracy, precision, and recall scores than standard benchmark algorithms for spam classifications
To tune, the model used SGO (social group optimization) [60]	Clustering and purpose neural networks classification	Audio, picture, and text	Not mentioned
Analyzed spam messages and identified SMS [61]	NB and FP growth	Repository of ML (UCI): dataset of spam SMS collection	98.5% accuracy

3.1. Data Collection. A spam-based SMS and e-mails from Russia were used in this study. The dataset was taken from Kaggle [62]. 58 attributes may be found in the machine-learning repository where one feature (nine) is a dependent attribute and 57 independent attributes. The SMS dataset for the study is divided into two groups: spam and nonspam. Spam messages have been validated and are guaranteed to be spam (750). 4250 messages in the nonspam group are certainly not spam. Similarly, spam e-mails are 600 in number, and nonspam e-mails are 4400. These e-mails are not spam and they do not have any suspicious spam signatures. There are a total of 5000 text messages and 5000 e-mails, comprising roughly a spam ratio of 31% determined from a public corpus [63]. Tables 2 and 3 summarize the SMS and e-mail spam dataset that was taken from Kaggle. In Table 2, the dataset contains 750 spam and 4250 nonspam messages. The dataset is split into 67.67% instances as training and the remaining as a test set. In Table 3, the dataset contains 600 spam and 4400 nonspam messages. The dataset is split into 65.67% instances as training and the remaining as a test set.

3.2. Data Preprocessing. Data preprocessing is a fundamental stage in the development of a machine-learning model that involves preparing raw data for use. Real-world data are often noisy, contain missing values, and are not in a suitable format for direct use in machine-learning models. Therefore, data preprocessing is a necessary step that involves cleaning and transforming the data to enhance the efficiency and accuracy of the machine-learning model.

Figure 7 represents the preprocessing with all steps, such as tokenization, lowercasing, normalization, and stemming, that were involved. The whole process mentioned in the figure starts when the spam data are added to the database and then the database starts preprocessing stepwise. The data are purified after each stage, and when the procedure is complete, it is prepared for the next step. All the functions of the proposed model are performed on these data.

3.3. Feature Extraction. All humans can use feature selection and feature extraction procedures. For learning algorithms, however, it is an issue of feature extraction in machine learning and selecting a subset of input variables on which to focus while disregarding the rest. In other words, it has an impact on feature extraction techniques' dimensionality reduction. ML is a valuable tool for conducting a study, and ensuring the accuracy of the ML model depends on meticulous inspection and preprocessing of the data fed into the algorithm. The bag-of-words (BoW) and term frequency-inverse document frequency (TF-IDF) procedures are two of the most common text preprocessing methods. Feature extraction approaches BoW and TF-IDF are both defined as follows.

3.3.1. BoW. A bag-of-words is a text representation that simply describes the occurrence of words within a document. The phrase "bag" refers to how words are thrown together in a bucket without regard for their structure in a sentence. The bag-of-words method completely disregards the relevance of a single word within a phrase, which is a significant disadvantage in the field of NLP. Mathematically, the BoW representation of a document can be represented as a vector as shown in the following equation:

$$V = [w_1, w_2, \ldots, w_m], \tag{1}$$

where V is the text's vector representation. The individual words in the vocabulary are represented by the letters w_1, w_2, and w_m (m being the total number of unique words in the vocabulary). The words' counts in the text are represented by the elements of the vector V (1, 2, . . ., m).

3.3.2. TF-IDF. The "TF-IDF" approach is an alternative to the bag-of-words method, and it is based on the assumption that a document's ability to be useful is reduced if it contains more frequent words. Each word is given a different value by the approach to reflect the term's relevance in its context. Simply multiplying the term frequency and the inverse document frequency yields the so-called TF-IDF value. The TF-IDF weight of term t in document d is given by the product of its term frequency tf (t, d) and inverse document frequency idf (t), as represented as follows:

$$\mathrm{tf} - \mathrm{idf}\,(\mathrm{t}, \mathrm{d}) = tf\,(t, d) * \mathrm{idf}\,(t). \tag{2}$$

The term frequency tf (t, d) of term t in document d is a measure of how frequently the term t appears in document d. It is calculated as the number of times term t appears in document d divided by the total number of terms in document d. The inverse document frequency idf (t) of term t measures how important the term t is in the collection of documents. It is calculated as the logarithm of the total number of documents in the collection divided by the number of documents that contain the term t. Multiplying the TF-IDF weighting scheme gives higher weights to terms that appear frequently in a document but rarely in the collection of documents and lower weights to terms that appear frequently in both the document and the collection.

3.4. Data Splitting. In this step, the SMS dataset is divided into a 67.66% training set and a 32.33% testing set. This stage divides the e-mail dataset into a 65.66% training set and a 34.33% testing set. The model will be trained using the training set, and its effectiveness will be validated using the testing set.

3.5. Classification. The training and testing sets are given the classification algorithms to determine if an e-mail is spam or not and to categorize the message as Spam or Not-Spam different classifiers such as NB, RF, SVM, and CNN are used. Following are the details of the architecture of the models that we used in our study.

Table 2: Statistics of SMS spam dataset [62].

	No. of messages	Percentage of messages (%)	Training set (%)	Testing set (%)
Spam	750	14.5	67.67	32.32
Nonspam	4250	85.5		
Total	5000	100		

Table 3: Statistics of mail spam dataset [62].

	No. of messages	Percentage of messages (%)	Training set (%)	Testing set (%)
Spam	600	12.5	65.67	34.32
Nonspam	4400	87.5		
Total	5000	100		

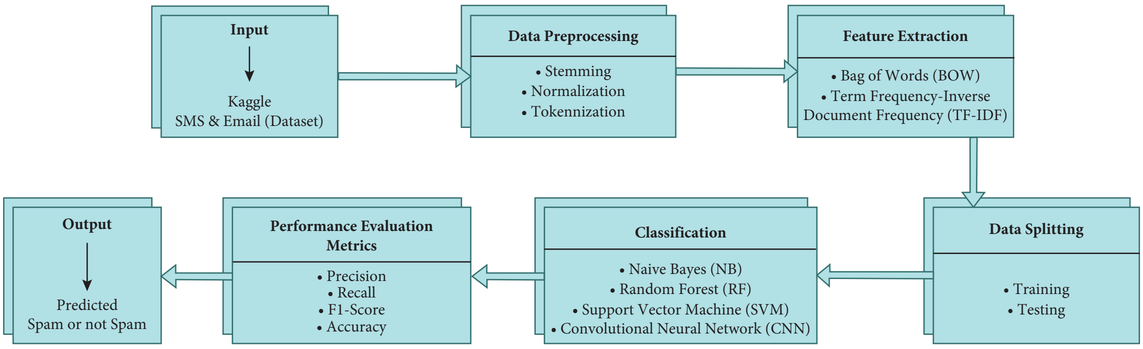

Figure 7: Proposed model.

3.5.1. *SVM Classifier*

(1) Margin-Based Classification. The SVM binary classification algorithm aims to identify the best hyperplane that maximizes the margin between data points belonging to various classes.

(2) Kernel Trick. Using kernel functions which automatically transfer the input data into a high-dimensional space where linear separation is achievable, the SVM can be expanded to handle nonlinearly separable data.

(3) C-Support Vector Classification. In this study, we used the C-Support Vector Classification version, which allows a soft margin, allowing certain misclassifications to strike a compromise between maximizing the margin and minimizing classification mistakes.

(4) SVM Equation. To discriminately classify data points, this classifier seeks to find a hyperplane in an N-dimensional space. As a result, they can be characterized in high-dimensional feature space using a linear function. The SVM is a classification and regression prediction approach that employs ML theory to improve predictive accuracy while preventing data overfitting [64]. The following is an example of how to express the SVM's mathematical classification:

$$y(x) = w^{Tx} + b, \tag{3}$$

where for a given input feature vector x, $y(x)$ is the output (prediction) of the SVM. The weight vector of the hyperplane, w, is parallel to it. X is the input feature vector, while T stands for the transpose of the vector w. The hyperplane's offset from the origin is represented by the bias term or b.

3.5.2. *NB Classifier*

(1) Probability-Based Classification. Naïve Bayes is a probabilistic classifier built on the Bayes theorem, making it especially effective for classifying texts.

(2) Multinomial Distribution. For discrete data, such as word frequencies in textual texts, the Multinomial Naïve Bayes version is utilized.

(3) Probability Estimation. Based on the frequency of occurrence of terms in the training data, the model calculates the probabilities of each class (spam and nonspam).

(4) Conditional Independence Assumption. To make probability calculations easier, the "Naïve" assumption in Naïve Bayes assumes that each word's occurrence is independent of other words given in the class label.

(5) NB Equation. The NB algorithm is a supervised learning technique that is both convenient and simple, allowing for the rapid development of machine-learning models that produce accurate predictions. The NB algorithm is a probabilistic classifier that generates predictions based on the probability of the input data. It is commonly used for tasks such as article classification, spam filtering, and sentiment analysis. The mathematical equation for the Naïve Bayes classifier can be expressed in the following equation:

$$p\frac{c}{d} = \frac{(P(c) * P(d \mid c))}{P(d)}, \tag{4}$$

where the conditional probability of event c occurring given that event d has already happened is represented by $P(c \mid d)$. Before taking into account any evidence, $P(c)$ represents the prior probability of event c occurring. $P(d \mid c)$ is the conditional possibility of event d happening given that event c has already happened. The possibility of event d occurring is expressed as $P(d)$ (evidence).

3.5.3. RF Classifier

(1) Ensemble of Decision Trees. Random forest is an ensemble-learning technique that combines various decision trees to produce predictions. It is an ensemble of decision trees.

(2) Decision Tree Architecture. To support a variety among the individual trees, the random forest builds each decision tree using a different subset of the training data and attributes.

(3) Random Feature Selection. A random subset of characteristics is taken into consideration for splitting at each node of the decision tree, lowering the possibility of overfitting and enhancing generalization.

(4) Voting Mechanism. A majority voting mechanism is often used to aggregate the predictions of various decision trees to get the final prediction of the random forest.

(5) RF Equations. Random forests are an ensemble technique for classification. To evaluate the importance of variables in a classification task, random forests are utilized [65]. For classification, the output of the random forest is a probability distribution over the classes, where the probability of each class is given by the fraction of trees that vote for that class. For a given input vector x, let Y be the output variable to be predicted, and let the random forest consist of M decision trees. The output of the random forest for classification is represented in the following equation:

$$p\left(Y = \frac{k}{x}\right) = \left(\frac{1}{M}\right) * \operatorname{sum}_{\{i=1\}}^{M} \left[p_i\left(Y = \frac{k}{x}\right)\right], \tag{5}$$

where $p\_i\ (Y = k \mid x)$ is the probability assigned to class k by the i-th decision tree, and the sum is taken over all decision trees in the forest. For regression, the output of the random forest is the mean prediction of the individual trees as shown in the following equation:

$$f(x) = \left(\frac{1}{M}\right) * \operatorname{sum}_{\{i=1\}}^{M\left[f_{i(x)}\right]}, \tag{6}$$

where $f\_i(x)$ is the prediction made by the ith decision tree, and the sum is taken over all decision trees in the forest.

3.5.4. Convolutional Neural Network

(1) Overview of the Architecture. The CNN utilized in this study is a feed forward neural network that was created primarily to process sequential input, making it appropriate for text-based tasks such as SMS and e-mail spam detection.

(2) Input Layer. The CNN's input layer accepts sequences of tokens, each of which stands for a word or letter in the text. The length of the tokenized sequences is correlated with the dimensionality of the input layer.

(3) Convolutional Layers. The CNN is made up of several convolutional layers, each of which is made up of filters (kernels) that move over input sequences to extract feature maps. These filters capture local patterns and textual features. The CNN processes data by filtering it and is distinguished by its capacity to adjust the filters during training. Even when dealing with massive datasets, the results may be fine-tuned in real-time.

(4) Activation Functions. After each convolutional layer, we used rectified linear unit (ReLU) activation functions to bring nonlinearity into the model, allowing it to learn complicated representations.

(5) Pooling Layers. We employed max pooling layers to minimize dimensionality and computational complexity by extracting the maximum value from each feature map, downsampling the representations.

(6) Fully Connected Layer. Following the pooling and convolutional layers, we added one or more fully connected layers to process the recovered features and produce final predictions.

(7) Output Layer. The CNN's output layer generates classification results, showing the likelihood of the input being spam or nonspam (ham) using a sigmoid activation function.

4. Results and Discussion

In this section, we have used the Kaggle dataset containing 5000 SMS and 5000 total e-mail instances. For experimentation, the proper steps are followed as previously mentioned in the figure of the proposed model for this research. Firstly, the two datasets taken from Kaggle were passed through a preprocessing phase to remove unnecessary outliers to normalize the data. Later, certain techniques such as BoW and TF-IDF feature extraction were done. Furthermore, the extracted features from the datasets were passed onto different models of ML (SVM, NB, and RF) and deep learning (CNN). A model trained on each classifier has given different results regarding their performance and efficiency. After the model training phase, the evaluation of the model upon each classifier is done using certain performance evaluation metrics (accuracy, precision, recall, and $F1$-score).

4.1. Performance Evaluation. We employed well-known classification metrics such as precision (P), recall (R), $F1$-score ($F1$), and accuracy to evaluate the performance of the proposed model. Precision (P) is defined as the percentage of times the right spam SMS is returned in a certain situation, whereas true positive (TP) indicates that the model successfully predicted the correct type. When a model predicts the correct type incorrectly, it is said to be false positive (FP). The calculated precision is shown as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{7}$$

Recall (R) is a metric used to measure the percentage of actual spam SMS that a model accurately predicts. False negative (FN) demonstrates that the model accurately anticipated the mistaken sort. It can be defined mathematically using the following equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{8}$$

$F1$-score ($F1$) is defined as the harmonic mean of precision and recall. The harmonic mean gives more weight to low values as shown in the following equation:

$$F1 - \text{Score} = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}. \tag{9}$$

The $F1$-score is particularly helpful in situations when there is an imbalance between the amount of positive and negative instances in the dataset because it uses the harmonic mean. It is a balanced metric that considers both false positives and false negatives because it considers both precision and recall. The $F1$-score has a range of 0 to 1, with 1 denoting perfect precision and recall and 0 denoting zero precision or recall. In conclusion, the $F1$-score offers a single metric for evaluating the model's performance, allowing both precision and recall and is particularly helpful when the dataset contains classes with imbalanced representations.

Accuracy is a metric used in machine learning to measure the proportion of correctly predicted spam messages among all messages. Equation (10) is commonly used to calculate accuracy.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{10}$$

where the number of instances that were accurately identified as positive (in this case, correctly anticipated spam messages) is known as the true positive (TP) rate. The number of instances that were accurately identified as negative (in this case, correctly anticipated nonspam SMS messages) is known as the true negative (TN) rate. The number of instances where nonspam messages were mistakenly labeled as spam is known as false positives (FP). The number of instances where spam messages were mistakenly labeled as nonspam is known as false negatives (FN).

4.2. Results Using ML and DL Approaches. The results of implementing ML and DL algorithms on e-mail and SMS datasets are presented in this section. The results of different ML and DL classifiers are depicted below through graphical and pictorial representation for spam and nonspam classes on both these datasets. Tables 4–7 give the detailed performance evaluation corresponding to each classifier, and Figures 8–11 show the overall confusion matrix for SMS and e-mail spam detection of both the datasets separately.

On the SMS dataset, Table 4 illustrates the performance of machine-learning classifiers. The dense SVM outperforms all other machine-learning classifiers, as evidenced by the three measures utilized in this study. Similarly, Table 5 shows the results based on the e-mail spam dataset using different ML and DL classifiers on the unbalanced dataset. Due to the frequently unbalanced distribution of classes in real-world data, it is typically not possible to have a balanced dataset during testing operations. In the majority of cases, it is crucial for the model to gain knowledge from the actual data distribution it would come across in practical applications. Testing on a balanced dataset might not be a good representation of the difficulties the model would encounter in use.

Finally, Tables 6 and 7 show that the given machine-learning strategy for this study outperforms other current spam detection techniques.

4.3. Comparison Analysis of Algorithms. Figure 12 describes the accuracy comparison of algorithms, and it shows that the SVM has the best accuracy both in SMS and e-mail spam detection. The SVM has given better accuracy because it can give better accuracy for certain types of problems due to its ability to handle nonlinear separation, be robust to outliers, have a regularization parameter, and maximize the margin between the decision boundary and support vectors. The SVM algorithm has outperformed in terms of clear class separation, which can be attributed to the factors mentioned earlier.

4.4. Comparison Analysis with Existing Models. Here is the short comparison analysis of our paper that how it is better than other models. Figure 13 shows the comparative analysis of others with our model. In some papers, most researchers have achieved a good accuracy but they have a very short dataset, also they have not performed experiments on both spam SMS and e-mail datasets or they did not use both algorithms of ML or DL. Some performed experiments only on spam SMS dataset and some performed experiments only on spam e-mail datasets and some have not even defined whether they are performing experiments on spam SMS or

Table 4: Unbalanced dataset results using several classifiers on spam SMS.

Classifier	Class	Precision (%)	Recall (%)	*F*1-score (%)
SVM	Spam	198.3	81.9	89.5
	Nonspam	183.8	98.4	91.2
CNN	Spam	195.2	76.2	85.2
	Nonspam	181.4	97.0	88.1
NB	Spam	155.0	65.0	59.7
	Nonspam	155.1	47.0	50.7
RF	Spam	158.0	45.0	50.8
	Nonspam	155.0	67.7	60.8

Table 5: Unbalanced dataset results using several classifiers on spam e-mail.

Classifier	Class	Precision (%)	Recall (%)	*F*1-score (%)
SVM	Spam	97.3	82.9	88.5
	Nonspam	82.8	97.4	92.2
CNN	Spam	96.2	75.2	84.2
	Nospam	82.4	96	87.1
NB	Spam	65	75	69.7
	Nonspam	65.1	57	60.7
RF	Spam	68	49	51.8
	Nonspam	65	68.7	62.8

Table 6: Results using several classifiers on spam SMS.

Classifier	Class	Prediction (%)		Accuracy (%)
		Spam (%)	Nonspam (%)	
SVM	Spam	31	10.9	99.6
	Nonspam	27	11.3	
NB	Spam	24	11.6	75
	Nonspam	21	11.9	
RF	Spam	21	11.9	68
	Nonspam	21	11.9	
CNN	Spam	20	25	66.4
	Nonspam	20	25	

Table 7: Results using several classifiers on spam e-mail.

Classifier	Class	Prediction (%)		Accuracy (%)
		Spam (%)	Nonspam	
SVM	Spam	41	11.9%	95
	Nonspam	37	12.3%	
NB	Spam	34	12.6%	72
	Nonspam	31	12.9%	
RF	Spam	12.1	21.9	80.5
	Nonspam	12.1	21.9%	
CNN	Spam	12	35%	78
	Nonspam	12	35%	

spam e-mail. Some have defined their datasets but most of them did not define their datasets. Previous studies focused solely on ML or DL models for spam detection, without comparing their efficiency. In contrast, this study explores

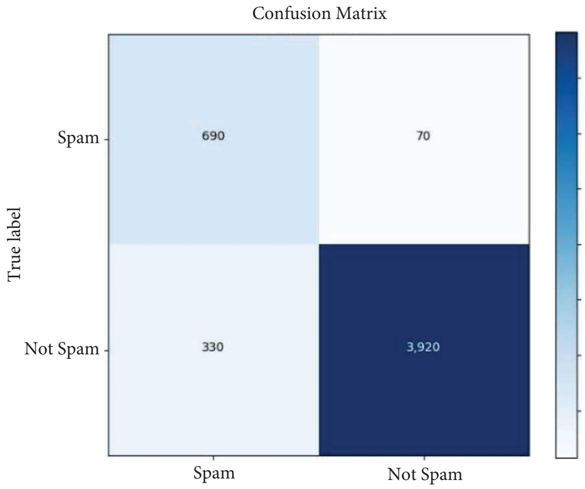

Figure 8: Overall confusion matrix for detection of spam and nonspam SMS on dataset-I.

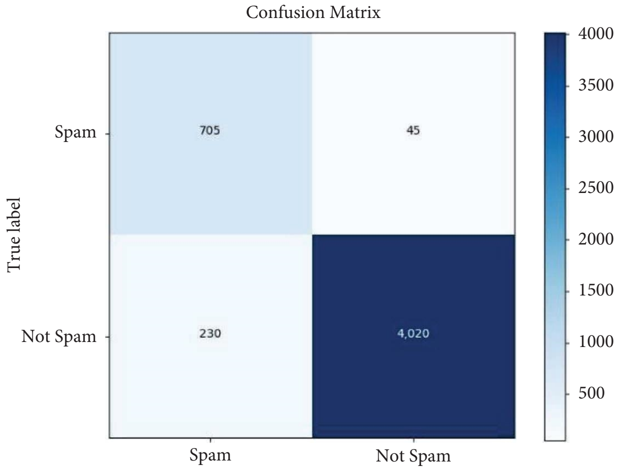

Figure 9: Overall confusion matrix for detection of spam and nonspam e-mail on dataset-I.

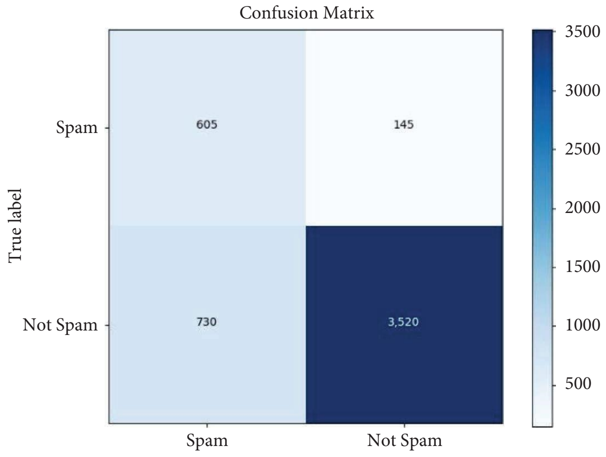

Figure 10: Overall confusion matrix for detection of spam and nonspam SMS on dataset-II.

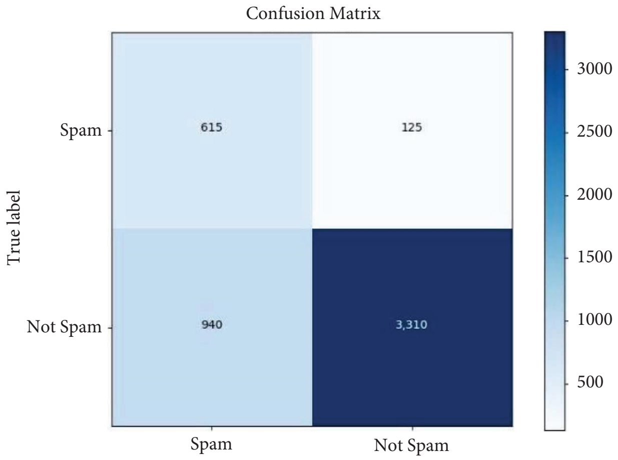

Figure 11: Overall confusion matrix for detection of spam and nonspam e-mail on dataset-II.

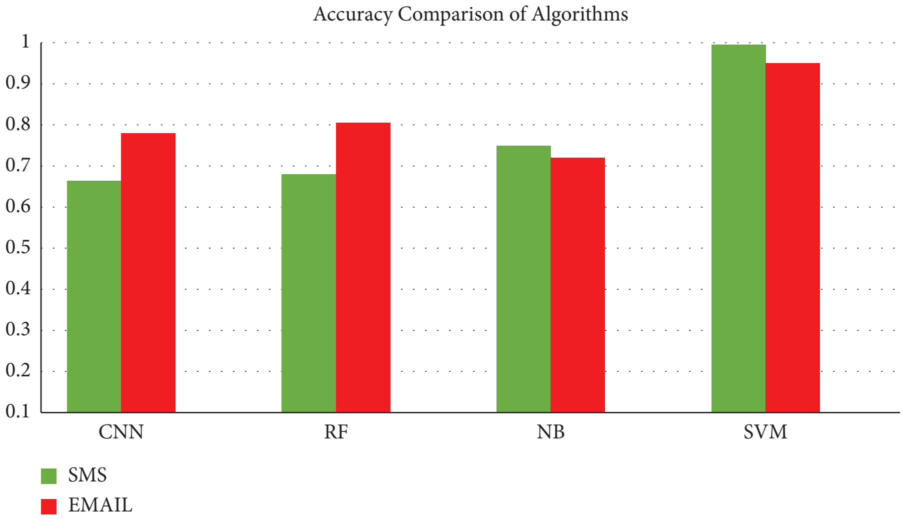

Figure 12: Accuracy comparison of ML and DL algorithms.

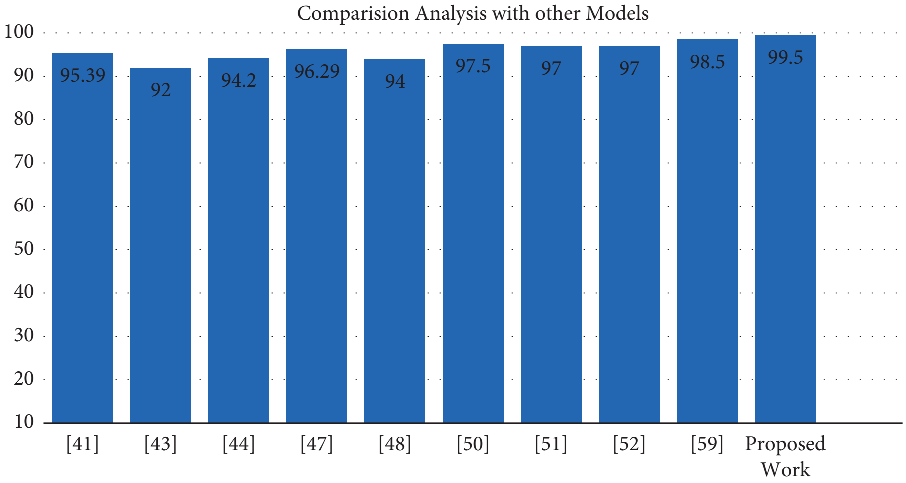

Figure 13: Comparison analysis with other models.

classifiers from both ML and DL to identify the most effective model for detecting SMS and e-mail spam. This paper applies classifiers of ML such as random forest (RF), multinomial Naïve Bayes (NB), support vector machine (SVM), and DL algorithm convolutional neural network (CNN) on both SMS and e-mail datasets and manipulates these classifiers on the two datasets by using evaluation of known metrics. Our dataset contains 5000 SMS spam messages and 5000 e-mail spam messages, and we have achieved the accuracy of 99.6% and 95% on SMS and e-mail spam.

5. Conclusion, Limitations, and Future Work

On e-mail and SMS datasets, the researchers used ML and DL classifiers such as NB, SVM, RF, and CNN. The study's findings revealed that in both datasets, the SVM outperforms machine-learning classifiers in detecting spam due to its robustness to outliers and margin maximization factors. The experimental results revealed that the SVM model worked best in terms of filtering messages with an accuracy of 99.6% and 95% on SMS and e-mail spam. The study we suggest highlights some of the limitations that the proposed model was unable to address. One of the major pitfalls includes that the model was trained just to classify the messages of the English language. So due to less versatile data, model accuracy may vary. In the future, this research can be expanded to include versatile data to make the suggested model more accurate in terms of increasing accuracy. Furthermore, the performance of the model could be improved and enhanced by training the model on deep learning algorithms other than the CNN as well to classify spam and nonspam messages that are written in different languages as well. Furthermore, class imbalances in the training data can affect the model's accuracy, and approaches for dealing with imbalanced datasets, such as oversampling or utilizing class weights, may be helpful. This paper further emphasizes the importance of improving the model's contextual understanding of evolving spam tactics and techniques, which might be accomplished by adding the use of recurrent neural networks or transformer-based models. In addition, efforts to improve model clarity and real-time performance via interpretability techniques and model optimization are suggested to improve the proposed spam detection methodology. Overall, fixing these constraints would strengthen the model and allow for future improvements to spam detection research.

Data Availability

The data used in this research are spam SMS/e-mail benchmark datasets that are taken from Kaggle website [62] and are openly available.

Conflicts of Interest

The authors have no affiliations with or involvement in any organization or entity with any financial interest or nonfinancial interest in the subject matter or materials discussed in this manuscript.

References

[1] C. Sms, *The Real Value of Sms to Businesses*, 2019, https://www.Smscomparison.co.uk/sms-gateway-uk/2019-statistics.

[2] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: new collection and results," in *Proceedings of the 11th ACM Symposium on Document Engineering*, pp. 259–262, ACM, Mountain View CA USA, September 2019.

[3] C. Wang, Y. Zhang, X. Chen et al., "A behavior-based sms antispam system," *IBM Journal of Research and Development*, vol. 54, no. 6, pp. 1–3, 2010.

[4] V. Gupta, A. Mehta, A. Goel, U. Dixit, and A. C. Pandey, "Spam detection using ensemble learning," in *Harmony Search and Nature Inspired Optimization Algorithms*, pp. 661–668, Springer, Berlin,Germany, 2019.

[5] Z. Chen, Q. Yan, H. Han et al., "Machine learning based mobile malware detection using highly imbalanced network traffic," *Information Sciences*, vol. 433-434, pp. 346–364, 2018.

[6] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos, "An evaluation of naive bayesian antispam filtering," in *Proceedings of the Workshop on Machine Learning in the New Information Age*, pp. 9–17, 11th European Conference on Machine Learning, West Point, NY, USA, June 2018.

[7] G. Jain, M. Sharma, and B. Agarwal, "Optimizing semantic lstm for spam detection," *International Journal on Information Technology*, vol. 11, no. 2, pp. 239–250, 2019.

[8] D. T. Nguyen, K. A. A. Mannai, S. Joty, H. Sajjad, M. Imran, and P. Mitra, "Robust classification of crisis-related data on social networks using convolutional neural networks," in *Proceedings of the Eleventh International AAAI Conference on Web and Social media*, Quebec, Canada, August 2017.

[9] S. Saumya, J. P. Singh, and Y. K. Dwivedi, "Predicting the helpfulness score of online reviews using convolutional neural network," *Soft Computing*, vol. 24, no. 15, pp. 10989–11005, 2019.

[10] X.-L. Wang, "Learning to classify email: a survey, in 2015 International conference on machine learning and cybernetics," *IEEE*, vol. 9, pp. 5716–5719, 2015.

[11] O. Saad, A. Darwish, and R. Faraj, "A survey of machine learning techniques for spam filtering," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 12, no. 2, p. 66, 2017.

[12] Statista, *Number Of E-Mail Users Worldwide From 2017 To 2023*, Statista Global Business Data Platform, United States, Hamburg, Germany, 2019.

[13] C. Monitor, *The Shocking Truth about How many Emails*, 2019.

[14] O. A. Okunade, "Manipulating e-mail server feedback for spam prevention," *Arid Zone Journal of Engineering, Technology and Environment*, vol. 13, pp. 391–399, 2017.

[15] statista, "Daily number of spam emails sent worldwide as of january 2023, by country(in billions)," 2023, https://www.statista.com/statistics/1270488/spam-emails-sent-daily-by-country.

[16] firms, "Spam statistics," 2023, https://99firms.com/blog/spam-statistics/#gref.

[17] firms, "99 firms," 2023, https://99firms.com/blog/spam-statistics/#gref.

[18] G. L. Bodic, *Mobile Messaging Technologies And Services Sms, Ems And Mms*, John Wiley & Sons Ltd, Hoboken, NJ, USA, 2nd edition, 2002.

[19] Gsma, “The state of mobile internet connectivity 2020,” 2020, https://www.gsma.com/r/wpcontent/uploads/2020/09/GSMA-State-of-Mobile-Internet-ConnectivityReport-2020.pdf.

[20] Salesforce Marketing Cloud, “Mobile behavior report,” 2014, https://textlocal.com/mobile-consumer-behaviour-report/.

[21] G. Fragkos, C. Minwalla, J. Plusquellic, and E. E. Tsiropoulou, “Artificially intelligent electronic money,” *IEEE Consumer Electronics Magazine*, vol. 10, no. 4, pp. 81–89, 2021.

[22] truecaller, “2021 global spam scam report,” https://www.truecaller.com/blog/insights/top-20-countries-affected-by-spam-calls-in-2021.

[23] statista, “Number of smartphone users worldwide from 2013 to 2028,” https://www.statista.com/forecasts/1143723/smartphone-users-in-the-world.

[24] M. Abdullahi and M. A. Ngadi, “Symbiotic organism search optimization based task scheduling in cloud computing environment,” *Future Generation Computer Systems*, vol. 56, pp. 640–650, 2016.

[25] M. A.-Z. Ala’, H. Faris, and M. A. Hassonah, “Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts,” *Knowl-Based Syst.*, vol. 153, pp. 91–104, 2018.

[26] C. Chen, S. Wen, J. Zhang et al., “Investigating the deceptive information in twitter spam,” *Future Generation Computer Systems*, vol. 72, pp. 319–326, 2017.

[27] Y. Cohen, D. Gordon, and D. Hendler, “Early detection of spamming accounts in large-scale service provider networks,” *Knowl-Based Syst.*, vol. 142, pp. 241–255, 2018.

[28] P. P. Chan, C. Yang, D. S. Yeung, and W. W. Ng, “Spam filtering for short messages in adversarial environment,” *Neurocomputing*, vol. 155, pp. 167–176, 2015.

[29] G. Faulkner, “A new and nasty way to flood networks with spam,” *Computers & Security*, vol. 7, pp. 622-623, 1997.

[30] B. Hancock, “Fighting spam in europe,” *Computers & Security*, vol. 20, p. 18, 2001.

[31] S. Hinde, “Spam, scams, chains, hoaxes and other junk mail,” *Computers & Security*, vol. 21, pp. 592–606, 2002.

[32] S. Jeong, G. Noh, H. Oh, and C. K. Kim, “Follow spam detection based on cascaded social information,” *Informing Science*, vol. 369, pp. 481–499, 2016.

[33] C. C. Lai, “An empirical study of three machine learning methods for spam filtering,” *Knowl-Based Syst.*, vol. 20, pp. 249–254, 2007.

[34] L. Li, B. Qin, W. Ren, and T. Liu, “Document representation and feature combination for deceptive spam review detection,” *Neurocomputing*, vol. 254, pp. 33–41, 2017.

[35] C. Vorakulpipat, V. Visoottiviseth, and S. Siwamogsatham, “Polite sender: a resource-saving spam email countermeasure based on sender responsibilities and recipient justifications,” *Computers & Security*, vol. 31, pp. 286–298, 2012.

[36] C. C. Wang and S. Y. Chen, “Using header session messages to anti-spamming,” *Computers & Security*, vol. 26, pp. 381–390, 2007.

[37] A. Makkar and N. Kumar, “Cognitive spammer: a framework for pagerank analysis with split by over-sampling and train by under-fitting,” *Future Generation Computer Systems*, vol. 90, pp. 381–404, 2019.

[38] I. Ahmed, R. Ali, D. Guan, Y. K. Lee, S. Lee, and T. Chung, “Semi-supervised learning using frequent itemset and ensemble learning for sms classification,” *Expert Systems with Applications*, vol. 42, pp. 1065–1073, 2015.

[39] Q. Fu, B. Feng, D. Guo, and Q. Li, “Combating the evolving spammers in online social networks,” *Computers & Security*, vol. 72, pp. 60–73, 2018.

[40] K. Lee, J. Caverlee, and S. Webb, “Uncovering social spammers: social honeypots+ machine learning,” in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 435–442, New York, NY, USA, July 2010.

[41] S. Liu, Y. Wang, J. Zhang, C. Chen, and Y. Xiang, “Addressing the class imbalance problem in twitter spam detection using ensemble learning,” *Computers & Security*, vol. 69, pp. 35–49, 2017.

[42] A. Harisinghaney, A. Dixit, S. Gupta, and A. Arora, “Text and image-based spam email classification using knn, naive bayes and reverse dbscan algorithm,” in *Proceedings of the 2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, pp. 153–155, Faridabad, India, February 2014.

[43] 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.

[44] J. W. Joo, S. Y. Moon, S. Singh, and J. H. S. D. Park, “An enhanced security model for detecting Smishing attack for mobile computing,” *Telecommunication Systems*, vol. 66, pp. 29–38, 2017.

[45] A. K. Jain and B. Gupta, “Rule-based framework for detection of smishing messages in mobile environment,” *Procedia Computer Science*, vol. 125, pp. 617–623, 2018.

[46] A. K. Jain and B. B. Gupta, “Feature based approach for detection of smishing messages in the mobile environment,” *Journal of Information Technology Research*, vol. 12, pp. 17–35, 2019.

[47] G. Sonowal and K. S. S. D. C. A. Kuppusamy, “An anti-smishing model with machine learning approach,” *The Computer Journal*, vol. 61, pp. 1143–1157, 2018.

[48] S. Olatunji, “Improved email spam detection model based on support vector machines,” *Neural Computing & Applications*, vol. 31, no. 3, pp. 691–699, 2017.

[49] S. Mishra and D. S. Soni, “A security model to detect smishing through SMS content analysis and URL behavior analysis,” *Future Generation Computer Systems*, vol. 108, pp. 803–815, 2020.

[50] E. M. Bahgat, S. Rady, W. Gad, and I. F. Moawad, “Efficient email classification approach based on semantic methods,” *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 3259–3269, 2018.

[51] F. Janez-Martino, E. Fidalgo, S. Gonzalez-Martınez, and J. VelascoMata, “Classification of spam emails through hierarchical clustering and supervised learning,” 2020, https://arxiv.org/abs/2005.08773.

[52] N. F. M. Azmi, “SuriayatiChuprat, “SMS spam message detection using term frequenct-inverse document frequency and random forest algorithm,” *Procedia Computer Science*, vol. 161, pp. 509–515, 2019.

[53] A. Lakshmanarao, K. Chandra Sekhar, and Y. Swathi, “An efficient spam classification system using ensemble machine learning algorithm,” *Journal of Applied Science and Computations*, vol. 5, no. 9, 2018.

[54] G. D. PavasNavaney, “Ajay Rana, “SMS spam filtering using supervised machine learning algorithms,” in *Proceedings of the 8th International Conference on Cloud Computing*, Noida, India, January 2018.

[55] U. Srinivasarao and A. Sharaff, “Machine intelligence based hybrid classifier for spam detection and sentiment analysis of

SMS messages," *Multimedia Tools and Applications*, vol. 10, pp. 1–31, 2023.

[56] S. Giri, S. Das, S. B. Das, and S. Banerjee, "SMS spam classification–simple deep learning models with higher accuracy using BUNOW and GloVe word embedding," *Journal of Applied Science and Engineering*, vol. 26, no. 10, pp. 1501–1511, 2023.

[57] U. Bhardwaj and P. Sharma, "Email spam detection using bagging and boosting of machine learning classifiers," *International Journal of Advanced Intelligence Paradigms*, vol. 24, no. 1-2, pp. 229–253, 2023.

[58] W. M. Salama, M. H. Aly, and Y. Abouelseoud, "Deep learning-based spam image filtering," *Alexandria Engineering Journal*, vol. 68, pp. 461–468, 2023.

[59] P. Naravajhula and A. Naravajula, "Spam classification: genetically optimized passive-aggressive approach," *SN Computer Science*, vol. 4, no. 2, pp. 1–12, 2023.

[60] S. Sinha, I. Ghosh, and S. Satapathy, "A study for ANN model for spam classification," *Advances in Intelligent Systems and Computing*, vol. 17, pp. 331–343, 2020.

[61] D. Delvia Arifin, Shaufiah, and M. A. Bijaksana, "Enhancing spam detection on mobile phone short message service (SMS) performance using FP-growth and naive bayes classifier," in *Proceedings of the 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 80–84, Bandung, Indonesia, September 2016.

[62] kaggle, "Spam sms/email classification 98% accuracy," 2020, https://www.kaggle.com/code/karanchinchpure/spam-sms-email-classification-98-accuracy.

[63] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in *Proceedings of the 11th ACM Symposium on Document Engineering*, Mountain View, CA, USA, September 2011.

[64] S. Bosaeed, I. Katib, and R. Mehmood, "A fog-augmented machine learning based SMS spam detection and classification system," in *Proceedings of the 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 325–330, Paris, France, April 2020.

[65] M. Arulprakash, "Eshort message service spam detection and filtering using machine learning approach," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 9, pp. 721–727, 2021.