Hindawi
Scientific Programming
Volume 2021, Article ID 6508784, 11 pages
https://doi.org/10.1155/2021/6508784

Research Article

Machine Learning-Based Detection of Spam Emails

Zeeshan Bin Siddique,[1] Mudassar Ali Khan ,[1] Ikram Ud Din ,[1] Ahmad Almogren ,[2] Irfan Mohiuddin,[2] and Shah Nazir [3]

[1]*Department of Information Technology, The University of Haripur, Haripur, Pakistan*
[2]*Chair of Cyber Security, Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11633, Saudi Arabia*
[3]*Department of Computer Science, University of Swabi, Swabi, Pakistan*

Correspondence should be addressed to Ikram Ud Din; ikramuddin205@yahoo.com, Ahmad Almogren; ahalmogren@ksu.edu.sa, and Shah Nazir; shahnazir@uoswabi.edu.pk

Received 29 September 2021; Revised 26 November 2021; Accepted 27 November 2021; Published 17 December 2021

Academic Editor: Zeeshan Pervez

Social communication has evolved, with e-mail still being one of the most common communication means, used for both formal and informal ways. With many languages being digitized for the electronic world, the use of English is still abundant. However, various native languages of different regions are emerging gradually. The Urdu language, coming from South Asia, mostly Pakistan, is also getting its pace as a medium for communications used in social media platforms, websites, and emails. With the increased usage of emails, Urdu's number and variety of spam content also increase. Spam emails are inappropriate and unwanted messages usually sent to breach security. These spam emails include phishing URLs, advertisements, commercial segments, and a large number of indiscriminate recipients. Thus, such content is always a hazard for the user, and many studies have taken place to detect such spam content. However, there is a dire need to detect spam emails, which have content written in Urdu language. The proposed study utilizes the existing machine learning algorithms including Naive Bayes, CNN, SVM, and LSTM to detect and categorize e-mail content. According to our findings, the LSTM model outperforms other models with a highest score of 98.4% accuracy.

1. Introduction

The Internet has become an inseparable part of human lives, where more than four and half billion Internet users find it a convenient to use it for their facilitation. Moreover, emails are considered as a reliable form of communication by the Internet users [1]. Over the decades, e-mail services have been evolved into a powerful tool for the exchange of different kind of information. The increased use of the e-mail also entails more spam attacks for the Internet users. Spam can be sent from anywhere on the planet from users having deceptive intentions that has access to the Internet. Spams are unsolicited and unwanted emails sent to recipients who do not want or need them. These spam emails have fake content with mostly links for phishing attacks and other threats, and these emails are sent in bulk to a large number of recipients [2]. The intention behind them is to steal users' personal information and then use them against their will to gain materialistic benefits [3]. These emails either contain malicious content or have URLs that lead to malicious content. Such emails are also sometimes referred to as phishing emails.

Despite the advancement of spam filtering applications and services, there is no definitive way to distinguish between legitimate and malicious emails because of the ever-changing content of such emails. Spams have been sent for over three or four decades now, and with the availability of various antispam services, even today, nonexpert end-users get trapped into such hideous pitfall [4]. In e-mail managers, spam filters detect spam and forward it to a dedicated space, spam folder, allowing the user to choose whether or not to access them. Spam filtering tools such as corporate e-mail systems, e-mail filtering gateways, contracted antispam services, and end-user training can deal with spam emails in English or any other language [4]. However, they are ineffective at filtering spam emails in other languages that recently have been digitized, such as Urdu Language. The proposed study exploits the existing artificial intelligence models to detect spam emails written in Urdu. This article describes how machine learning (ML) and deep learning (DL) models such as Support Vector Machine (SVM), Naive Bayes, Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), a recurrent neural network, can be trained to detect Urdu spam emails. Moreover, as there is no dataset for spam emails, this article also explains its creation and training of various machine learning models.

Precision, recall, and f-measure are considered key evaluating measures to compare Naive Bayes and SVM, while the evaluation parameters, i.e., Model Loss and ROC-AUC, are calculated for deep learning models such as CNN and LSTM. Finally, a comparison is made between all models for the best accuracy and values of evaluation parameters obtained by DL and ML models [5]. As we all know, a lot of work has been done in the field of e-mail spam detection in English or any other foreign language. There is a notable work done in roman Urdu script, which is quite different from Urdu writing script. We write Roman Urdu using English alphabets, but Urdu script is adapted from Persian language, and its writing script is derived from Arabic alphabets. For example, we have a sentence” We are going to Karachi” in Roman Urdu, and it will be written as” hum Karachi ja rhy hein” and in Urdu script, it will be written as” . . . “. There may be some research work done in the field of Urdu scripted spam e-mail detection, but we encountered only one approach in our literature review [4]. They provided very little information about the dataset they utilized, and their results are quite lower than ours. The proposed research is purely based on the identification of Urdu scripted e-mail spam. First, we gathered dataset from online available resources such as “kaggle” and “UCI repository” and then converted it into Urdu using Google Trans Ajax API in CSV format. The proposed work is unique in such a way that that dataset used by us is purely based on Urdu script. Utilized dataset is solely created and deployed by us for Urdu e-mail spam detection. The suggested study is focused on the detection of Urdu scripted e-mail spam. Our contribution to this study is that we used our own Urdu scripted dataset to test machine learning and deep learning methods. We also deployed CNN and LSTM deep learning models. These models have been rarely applied in previous approaches of the same context. They have delivered excellent results for Urdu spam e-mail detection. The highest accuracy was obtained by LSTM 98.4% and CNN obtained 96.2%. Because our situation differs from other approaches, we are unable to make a comparison with earlier approaches. Our dataset is written in Urdu, which is not the same as roman Urdu or any other linguistic script. The Urdu spam e-mail detection (USED) process is illustrated in Figure 1.

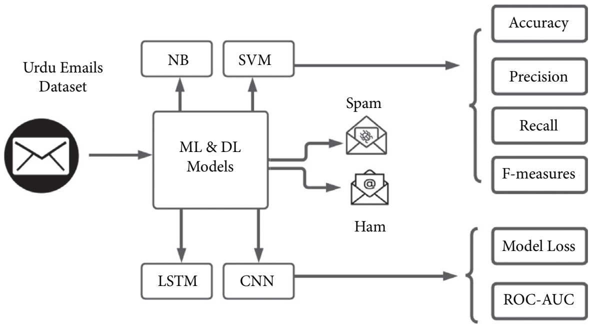


Figure 1: Detection of SPAM and HAM using machine learning approaches (adapted from [1]).

2. Literature Review

Before implementing a spam detection model using machine learning or deep learning for e-mail written in Urdu, existing studies were studied, regardless of the language in which the e-mail content was written. A comparative summary of the same is also explained in Table 1.

The authors in [6] gathered 1463 tweets written in Roman Urdu and categorized 1038 of them as ham and 425 of them as spam. On that data, they used discriminative multinomial Naive Bayes techniques. They got 95.12% with DMNBText and 95.42% with NB. The techniques were used with a numerical sequence of words that did not take into account domain or linguistic details. Linguistic techniques, such as those that take into account the contextual characteristics of important terms in Roman Urdu literature, are expected to improve classification results.

Backpropagation neural network (BPNN) was used in [7] to filter spam emails. They gathered 200 spam-based emails and labelled half of them as spam and half as ham. They used the K-Means Clustering Algorithm to preprocess the dataset. Using the k-mean clustering approach in preprocessing and backpropagation neural networks (BPNN) in the learning stage of the model, they got a maximum accuracy of 95.42%. The suggested work had limitations in that the models take a long time to train and test.

The efficiency of hybrid feature selection in the classification of emails was evaluated in [8]. They collected 169 emails, 114 of which were spam and 55 of which were ham. Using hybrid feature selection, they were able to achieve the highest accuracy rate. They employed Hybrid Feature Selection (TF-IDF) with a total of four reducts and achieved an accuracy of 84.8%. The work’s shortcoming is that the Malay language dataset was not appropriate with TD-IDF and rough set theory, and it did not provide the highest level of accuracy for the planned study.

In [9], the authors implemented Naive Bayes and J48 (Decision Tree) algorithms in a machine learning-based hybrid bagging approach for spam e-mail detection. They gathered 1000 emails, half of which were spam, and the other half were ham. For the training of both models, the dataset was divided into two sections. They employed Naive Bayes, J48, and a hybrid bagging strategy to classify spam and ham emails, with J48 providing the highest accuracy, which is 93.6%.

The boosting strategy replaces the flawed classifier’s learning characteristics with those of the base classifier, improving the overall system efficiency. The concept of

Table 1: Summary of existing approaches.

Ref.	Algorithms	Dataset	Evaluation parameter(s)	Performance
[6]	DMNBText, Liblinear, NB, J48	1463 Roman Urdu tweets	Accuracy, ROC-AUC	Obtained Max accuracy 95.42% using DMNB
[7]	K-Means clustering	200 spam-based emails	Accuracy, recall	They obtained 98.42% accuracy
[8]	NB, SVM	English and Malay emails	Accuracy	Max accuracy achieved is 86.40%
[9]	Hybrid bagging, Approach, NB, J48	1000 Spam base emails	Accuracy, Precision, recall, F-Measure	HB approach has obtained max accuracy
[1]	AdaBoost bagging, SVM, KNN, RF	5573 emails dataset	Accuracy	Max accuracy achieved is 98%
[10]	Boosting, bagging, KNN, SVM, RF, ensemble	5674 labelled dataset	Accuracy, precision, recall, F-measure	Max accuracy achieved is 97.5% using SVM
[11]	Integrated NB, PSO	Spam base emails dataset	Accuracy, precision, recall, F-measure	Max accuracy achieved is 95.5% using INB

boosting technique could be used for additional study in order to improve the system's outcomes.

The authors in [1] used machine learning algorithms to detect spam emails. They compiled a dataset using online tools such as 'kaggle' and others. They have collected 5573 emails and used that data to train seven machine learning models. The greatest result is 98.5% accuracy with Multinomial Nave Bayes; however, it has obvious limitations as class-conditional dependency, which causes the system to misidentify some data items. On the other hand, ensemble approaches have been shown to be effective since they use many learners to predict categories.

In [10], the authors have also made significant contributions to the field of spam e-mail detection. They used kaggle and the UCI machine learning repository to collect 5674 emails and define them as spam or ham. They estimated accuracy using six machine learning classifiers. They explored a variety of ml algorithms; however, it was discovered that Ensemble Filter produces more remarkable outcomes and has accuracy of 98.5%, which is higher than the other learners, as well as faster testing. The article's limitation is that testing was done on an e-mail sample without taking into account evolving trends in the mails, which could impair a classifier's effectiveness.

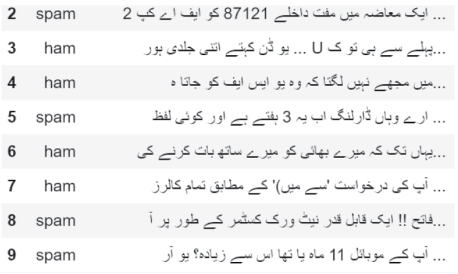
For the filtration of spam emails, an integrated Naive Bayes algorithm along with particle swarm optimization (PSO) is defined in [11]. They used NB to train and classify emails and PSO for swarm behavior property distribution. Finally, they used the proposed integrated concept NB and PSO to achieve evaluation steps. They employed a combined NB and PSO method. PSO is utilized to optimize the parameters of the NB technique. Naive Bayes is employed as a separator among spam and ham emails based on the keywords. They achieved a maximum accuracy of 96.42% after using an integrated NB method. It would be better if the Naive Bayes approach was used in combination with ant colony or artificial bee colony optimization.

The authors in [4] implemented four machine learning algorithms from the pool of algorithms. For classifying spam/ham e-mail detection in Urdu, they chose Naive Bayes, SVM, KNN, and RF. They generated their own dataset for Urdu emails but did not provide any information about it. Using NB, they were able to attain the greatest accuracy of 89%. The study's drawbacks include that these machine learning techniques are only successful for limited, labelled datasets. These algorithms take a long time to train, and the results they provide are also mediocre. The suggested approaches for spam e-mail detection are summarized in Table 1.

3. Methodology

In the following sections, creating of dataset, training of learning models, and data preprocessing are explained.

3.1. Dataset. For this study, the raw data collected is obtained from the online resource kaggle, which will be used to train machine learning models. The data was originally available in English language, and it was obtained in the comma separated values (CSV) format [12]. Further, the dataset obtained was translated using the Googletrans python library in URDU, which uses the Google Translate Ajax API. After this, a manual correction of the translated data was performed by the authors. We have used our own Urdu translated Urdu scripted dataset, which includes 5000 spam and ham emails. We created our own dataset, because Roman Urdu is written using English alphabets, and Urdu script is based on Arabic alphabets. Urdu scripts are distinct from Roman Urdu scripts. In At the end of this process, we obtained a total of 5000 emails, listed in two columns. The first column, labelled as 'type' having the two possible values as spam or ham, was meant to be used to classify emails. The second column is labelled 'e-mail Text' and contained a variety of e-mail content. It was decided that up to 80% of the emails will be used to train the models (approximately 4000 emails), whereas the remaining 20% will be used to test models individually (1000 emails). Figure 2 is a preview of Urdu spam e-mail dataset. This Urdu scripted dataset has been posted to a GitHub repository for future use. Any researcher who wants to study Urdu scripted emails will have no difficulty doing so now because the dataset is published publicly. Here is a link to the GitHub repository: https://github.com/zeeshanbinsiddique/Urduemaildataset. The dataset is further defined in Tables 2 and 3.



2	spam	ایک معاضہ میں مفت داخلے 87121 کو ایف اے کپ 2 ...
3	ham	پہلے سے ہی تو ک U ... یو ڈن کہتے اتنی جلدی ہور...
4	ham	میں مجھے نہیں لگتا کہ وہ یو ایس ایف کو جاتا ہ...
5	spam	ارے وہاں ڈارلنگ اب یہ 3 ہفتے ہے اور کوئی لفظ ...
6	ham	یہاں تک کہ میرے بھائی کو میرے ساتھ بات کرنے کی...
7	ham	آپ کی درخواست 'سے میں)' کے مطابق تمام کالرز ...
8	spam	فاتح !! ایک قابل قدر نیٹ ورک کسٹمر کے طور پر آ...
9	spam	آپ کے موبائل 11 ماہ یا تھا اس سے زیادہ؟ یو آر ...

Figure 2: Preview of the translated dataset for spam and ham emails before preprocessing.

3.2. Data Preprocessing. In machine learning (ML), the preprocessing phrase refers to organizing and managing of raw data before using it to train and test different learning models. In simplistic words, preprocessing is a ML data mining approach that turns raw data into a usable and resourceful structure [6].

The very first step in the construction of a ML model is preprocessing, in which data from the actual world, typically incomplete, imprecise, and inaccurate owing to flaws and deficient, is morphed into a precise, accurate, and usable input variables and trends [4].

The below mentioned subsection will highlight each step that is involved in a the data preprocessing phase [13], which is also beautifully illustrated in Figure 3 as the USED architecture.

3.2.1. Import Data. The first stage is to import the dataset, which is downloaded from 'Kaggle' and then converted to CSV format in Urdu [4]. The dataset containing 5000 emails were already classified as spam and ham. The data was obtained while being written in the English Language. As explained before, unlike the usual approach, we have translated the data set into Urdu, to achieve our goal of Urdu spam e-mail detection. We created our own dataset, because Roman Urdu is written using English alphabets, and Urdu script is based on Arabic alphabets. Urdu scripts are distinct from Roman Urdu scripts.

3.2.2. Tokenization. Being a critical phase of preprocessing, in this step, all the words from emails are gathered, and the number of times each word appears and location of appearance are counted [14]. With the aid of Count Vectorizer, we were able to find the repetition of words in our dataset. Each word is given a unique number, and hence, they are called tokens, also depicting their occurrences and quantity of occurrences. The token includes one of a kind feature values that will later help in the creation of feature vectors. In a tokenization phase, every word is assigned a unique token. Figure 4 shows tokens and unique numbers allotted to every token in a dataset. It is a screenshot of tokens taken after tokenization of dataset with the help of tokenizer. For better understanding, Figure 4 depicts some tokens taken from Urdu spam e-mail dataset after the tokenization process [15].

Table 2: Extracted dataset definition report.

Dataset feature	Value
Number of variables	2
Number of observations	5000
Missing cells	0
Missing cells %	0.0%
Duplicate rows	238
Duplicate rows %	4.8%
Total size in memory %	78.2 KB
Average record size in memory %	16.0 B

Table 3: Dataset pandas profiling report.

Attribute	Value	Token of Urdu e-mail	Value
Distinct	4664	. . .	26
Distinct %	93.3%	. . .	11
Missing	0	. . .	10
Missing %	0.0%	. . .	4
Memory size	39.2 KB	. . .	4
		Others	4659

3.2.3. Stop Word Removal. Once the dataset has been transformed into unique tokens, the next step is to delete every unnecessary word with no significance, e.g., white spaces, commas, full stops, colons, semicolons, and punctuation marks [16]. Stop words elimination is the name given to the method of eliminating unnecessary words. *Python* has built-in library known as natural language toolkit (NLTK), which has been widely used in language processing. Here, we used NLTK toolkit for stop words removal process for elimination of unnecessary words and spaces (Figure 5).

3.2.4. Stemming. After the tokens have been created, the next step is to stem them. Stemming is the method of converting the dataset's derived terms back to their original forms [14]. First, the base terms are stripped of prefixes and suffixes. Next, both modified or misspelled words are changed into their base or stem words using the stemming algorithm. For this step as well, we used NLTK python library to perform a perfect stemming process. After stemming of the content emails, spam words can be easily identified [5]. The following are some examples of stemmed words shown in Figure 6.

3.2.5. Feature Extraction and Selection. Feature extraction is the process of converting a large raw dataset into a more manageable format. Any variable, attribute, or class can be extracted from the dataset during this step, depending on the original dataset [17].

Feature extraction is a crucial step in training of the model, which helps in producing more reliable and accurate results. During the feature extraction process, out of the possible many attributes, the method of selecting some key variables that properly characterize data is called feature selection [9]. The model is then constructed using these selected attributes or variables [18, 19]. If feature selection is performed properly, in return, the model construction will

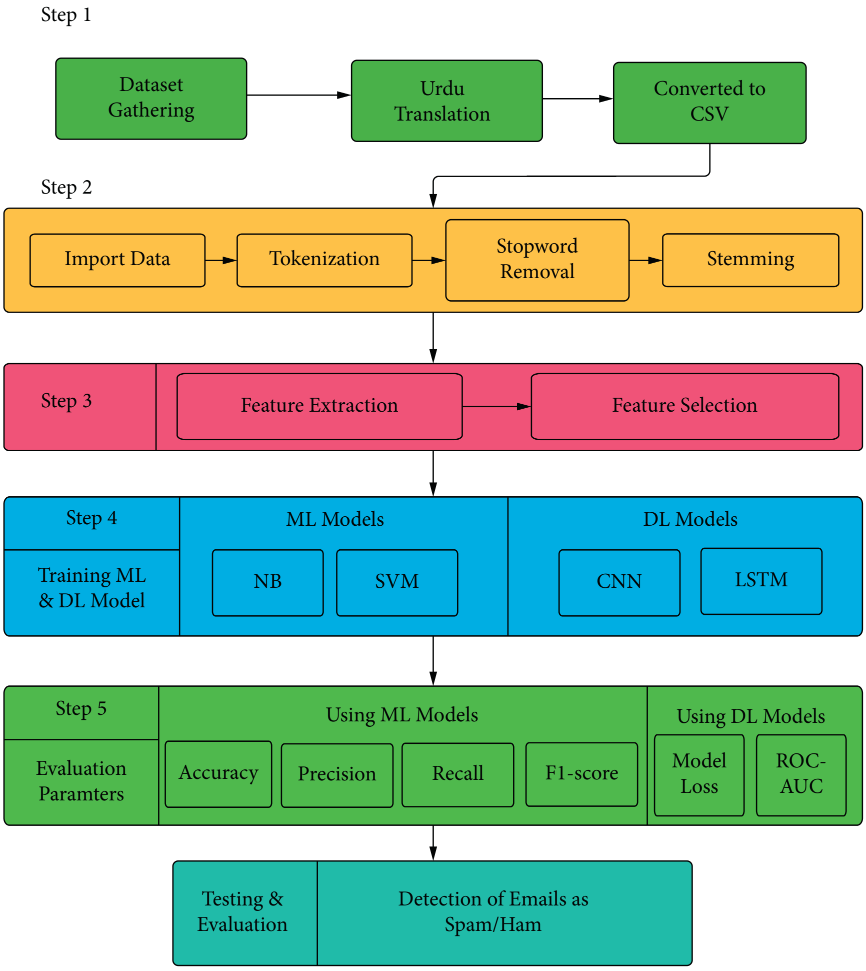
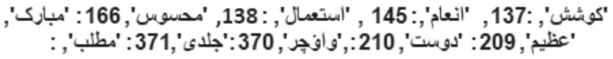
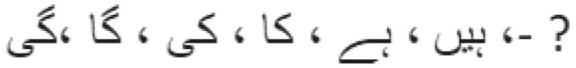


Figure 3: Urdu spam e-mail detection (USED) architecture highlighting different phases.



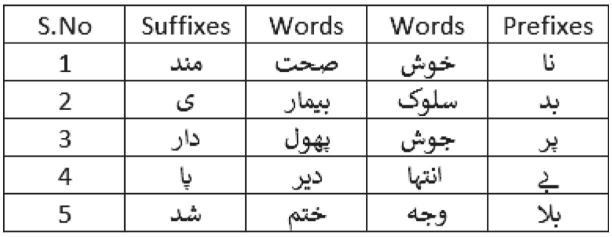
'کوشش',:137, 'انعام',:145 , 'استعمال', :138, 'محسوس', 166: 'مبارک',
'عظیم', 209: 'دوست', 210:,'واؤچر', 370:'جلدی', 371: 'مطلب', :

Figure 4: Unique numbers allotted to every token in an Urdu spam e-mail.



? -، ہیں ، ہے ، کا ، کی ، گا ،گی

Figure 5: Common stop words occurred in Urdu language.



S.No	Suffixes	Words	Words	Prefixes
1	مند	صحت	خوش	نا
2	ی	بیمار	سلوک	بد
3	دار	پھول	جوش	پر
4	پا	دیر	انتہا	بے
5	شد	ختم	وجہ	بلا

Figure 6: Stemming examples for various words extracted from spam emails.

take less time. Figure 7 explains the overall working of the machine learning models.

3.3. Step by Step Algorithm's Demonstration

Step 1. Pick a random mail from the collection for testing purposes.

Step 2. The e-mail in question is in its unprocessed state. E-mail must be preprocessed before the feature extraction and classification procedure can begin. Tokenization, stemming, and stop word elimination are all steps in the preprocessing process:

(1) To begin, split down the e-mail into distinct words and tokenize it. Tokenization separates each word into its own token.

(2) Eliminate all punctuation marks from the characters you obtained through tokenization.

(3) Stemming is done with the tokens earned in the previous stage. The stemming process decreases the size of a word to its base word. For stemming, a predetermined range of available words is examined, as well as the irrespective stem words.

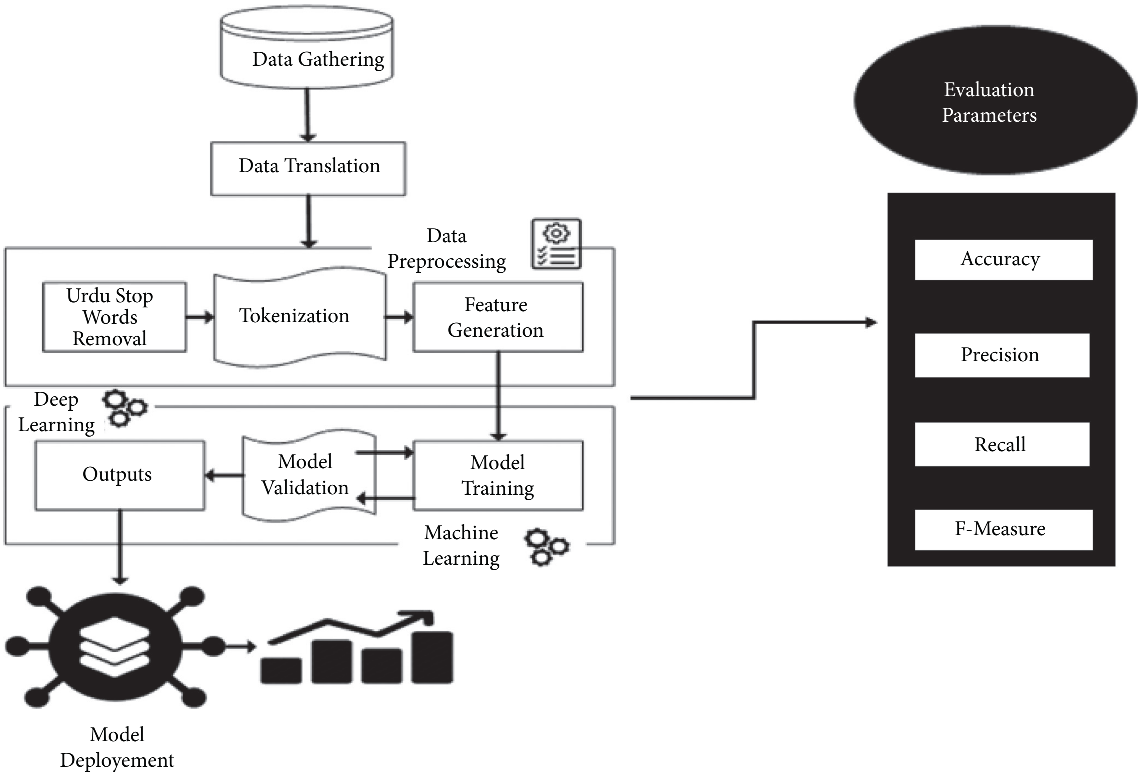


Figure 7: Workflow diagram for the Urdu spam e-mail detection (used).

(4) For stemming, a list of suffixes keywords is maintained in an array with their base words.

(5) Check to see whether there are any tokens available in the base input text.

(6) Stem the phrase to the proper base word from of the array list if the test token's suffixes are true.

(7) Otherwise, stemming is unnecessary. Word has already been converted to its root word format. Therefore, proceed to the next token.

Step3. To use the feature extraction technique, select suitable attribute words from the validation set. Just the set of features that is most nearly connected to the category is selected.

Step4. Use extracted features and created tokens to train ML and DL models. That model can easily distinguish between spam and ham emails.

Step5. Tokens are classified as spam or ham based on their feature similarity as ML models determines.

Step 6. Finally, the likelihood of spam or ham tokens in a sentence is evaluated for final classification:

(1) The mail is regarded spam if the significance level of spam tokens is higher than zero

(2) Otherwise, e-mail is regarded as ham e-mail

Step7. Mark the e-mail as spam or ham and proceed with the rest of the emails.

Figure 8 depicts processes of USED.

4. ML and DL Models Used for Experiment

4.1. Naive Bayes. Since 1998, Naive Bayes has been used in supervised machine learning to identify spam [20]. It primarily relies on the chance to differentiate between different entities based on predefined characteristics. Naive Bayes senses a word or event that happened in a previous context and calculates the likelihood of that word or event occurring again in the future [21]. For example, if a word appears in a spam e-mail but not in a ham e-mail, the algorithm would most likely classify it as spam.

$$\begin{gathered} P(c/x) = (P(x/c)P(c))/(P(x)), \\ P(x) = \sum_{y} P(x/c)P(c). \end{gathered} \tag{1}$$

Here, X denotes a set of function vectors, C stands for a class variable with multiple outcomes, P (c/x) denotes the likelihood of something happening in the future, P(x/c) P(c) stands for prior likelihood, and P(x) denotes the proof based on function variables.

4.2. Support Vector Machine (SVM). The Support Vector Machine (SVM) is another supervised machine learning algorithm. It only works for datasets that have been classified. For training purposes, SVM often uses both positive and negative

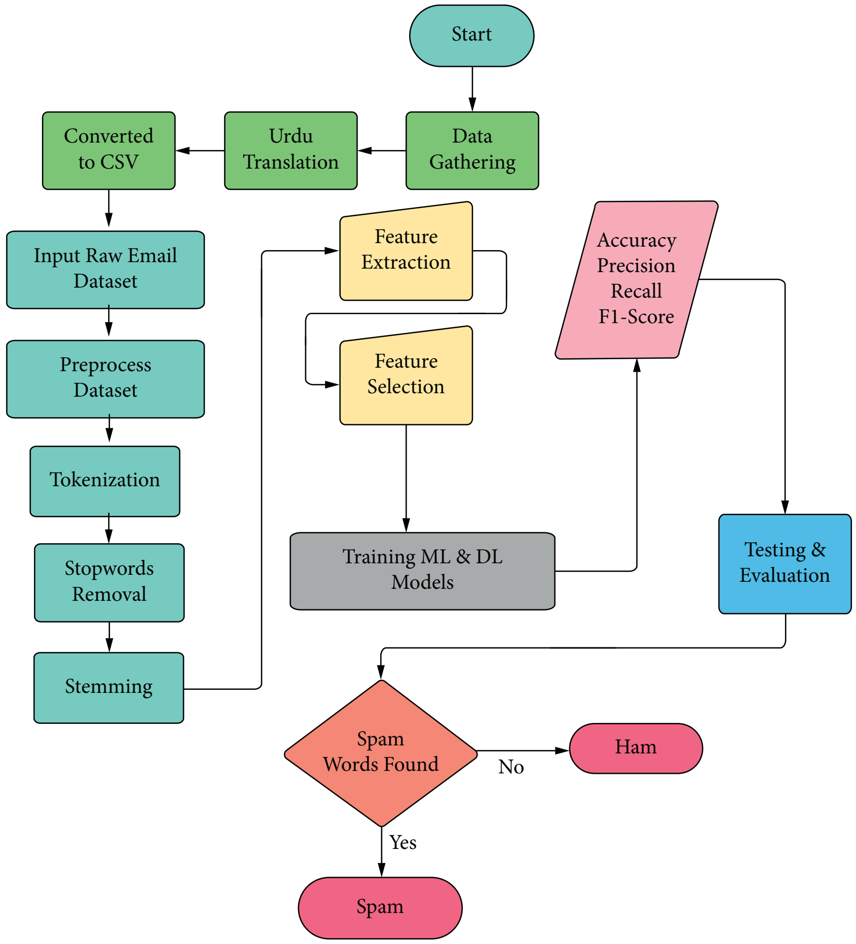


Figure 8: Flow diagram depicting processes of USED.

datasets. Negative datasets are not used in any other machine learning model's preparation. SVM is the most commonly used classification and regression model [22]. For the classification of data, it is more reliable than any other model. SVM is the fastest and most reliable classification model when we only have a small amount of labelled data. The SVM model employs a hyperplane to separate positive and negative values (spam and ham) from the dataset. Then, figure out the values are near enough to the decision surface [23]. SVM is represented in Figure 9.

$$y1 = \sum +b. \tag{2}$$

4.3. Deep Learning Models. Neural networks are used in deep learning, which is a new technology [10]. The DL models have a collection of hidden layers with weights that can be modified [2]. The DL models are given an input, which is then processed inside hidden layers to make a prediction using adjustable weights.

4.3.1. LSTM. One of the deep learning models is long short-term memory (LSTM). The LSTM architecture is a recurrent neural network (RNN) [25]. It is made up of feedback links and can handle both entire data sequences and single data points [26]. Unsegmented data and data based on time series work well with LSTM for classification and prediction. The workflow of LSTM is displayed in Figure 10.

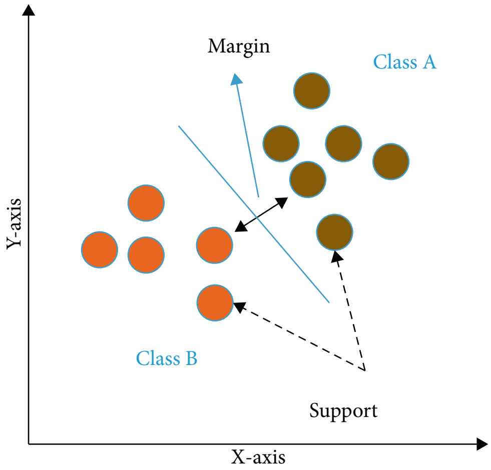


Figure 9: Support vector machine (adapted from [24]).

4.3.2. CNN. Convolutional Neural Networks (CNN) are a commonly used deep learning model class. It is made up of artificial neural networks that are space invariant (SIANN)

[25]. Their architecture is focused on mutual weights. CNNs are also referred to as multilayer networks or fully connected networks. Each neuron in the next layer is linked to a single neuron in the first layer and is a completely connected network [27]. Convolutional layers, which conduct convolutions, are among the hidden layers in CNN. Tensors are fed into these convolutional layers, which extract features. The tensors were then moved on to the next layer, which resulted in a prediction. The CNN model diagram is presented in Figure 11.

5. Evaluation Parameters

Precision, recall, f-measure, and classification accuracy are used to test the proposed algorithm's efficiency. True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values can be used to measure these parameters.

The following parameters are calculated using Naive Bayes and SVM models.

5.1. Accuracy. It refers to how much data from the whole dataset is correctly estimated. Thus, it depicts the overall accuracy of a classifier's prediction of data.

5.2. Precision. It is the metric by which a classifier's efficacy is measured. In other words, the total number of real true values classified by a classifier is called precision.

5.3. Recall. The measurement of a classifier's prediction's correctness is known as recall.

5.4. F-Measure. It indicates the accuracy of the classifier's prediction.

The following parameters are calculated using DL models, i.e., CNN and LSTM.

5.4.1. ROC-AUC. Receiver Operating Characteristics (ROC) are a metric that shows the accuracy of a classifier. The ROC curve represents the probability of a prediction. Thus, it is a reliable indicator of classifier efficiency. AUC stands for area under the ROC curve, and it indicates how well the classifier distinguishes between the two classes. The higher the AUC value, the more accurate the prediction.

5.4.2. Model Loss. Model loss denotes the classifier's failure to predict bad results for every case. The perfect model assumes that there will be no loss. Otherwise, if the model does not predict well, the loss would be greater.

6. Results and Analysis

The results and comparisons of different classifiers after data training and testing are presented in this section. We gathered 5000 emails from the online resource 'kaggle' and translated them into Urdu using the python library Googletrans, which uses the Google Translate Ajax API. Four thousand emails were used to train various ML and DL models. One thousand emails were used for testing in order to quantify accuracy and assessment metrics. As explained about evaluation measures in section 5, we have evaluated accuracy, precision, recall, and f-measures that are evaluation measures measured using SVM and Naive Bayes. CNN and LSTM are used to measure ROC-AUC and model loss values. Finally, using various graphs, a comparison of models is presented below. The findings in Table 4 show that the deep learning algorithm (LSTM) is a stronger method for detecting Urdu spam emails, with high accuracy of 98.4%.

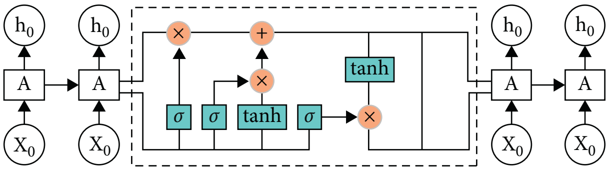


FIGURE 10: Long short-term memory (LSTM) (adapted from [2]).

In the mentioned Table 4, we have compared the accuracy of four different ML and DL models. We can see that the DL model (LSTM) is the most accurate among all the models, but it takes a long time to train. ML models like SVM and Naive Bayes are around the same accuracy percentage lower than LSTM/CNN, which is also a DL model and has the lowest accuracy percentage. Figure 12 shows accuracy comparison of ML and DL models.

ML models (i.e., SVM and Naive Bayes) are used to calculate evaluation parameters such as precision, recall, and f-measures, which are described in Table 5. In terms of recall and f-measures, we found that Naive Bayes is more successful and produces better results; however, SVM produces the highest precision percentage when compared to Naive Bayes. The results of the comparative analysis provided in Table 5 show that Naive Bayes achieves better results in terms of recall and f-measure, while SVM achieves better results with respect to precision.

The comparison of the results obtained by Naive Bayes and SVM is depicted visually in Figure 13.

The evaluation measures calculated for DL models (CNN and LSTM) are ROC-AUC and model loss. As shown in Table 6, when compared to CNN, LSTM has a greater percentage of ROC-AUC and a lower model loss rate. Finally, the entire findings were compared. We found that the LSTM has a greater accuracy and ROC-AUC value and a very low model loss rate. Compared to CNN, the comparative study findings show that LSTM produces better ROC-AUC and model loss results. The LSTM has a lower model loss of 5% and a high ROC-AUC of 99%.

Figure 14 shows a graphical representation of the results obtained by CNN and LSTM.

The following graphs (Figures 15 and 16) demonstrate the model loss for CNN and LSTM for each epoch. The graph line is obviously decreasing as the epochs increase, as can be seen. When the number of epochs is increased, the model loss rate decreases.

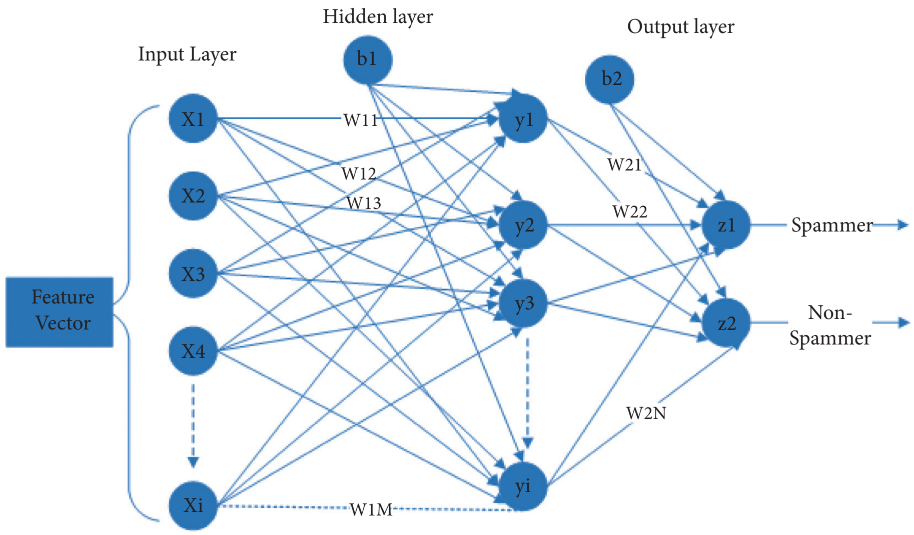


Figure 11: Convolutional neural network (CNN) (adapted from [7]).

Table 4: Accuracy of different models.

Models	Accuracy (%)
LSTM	98.4
CNN	96.2
Naive Bayes	98.0
SVM	97.5

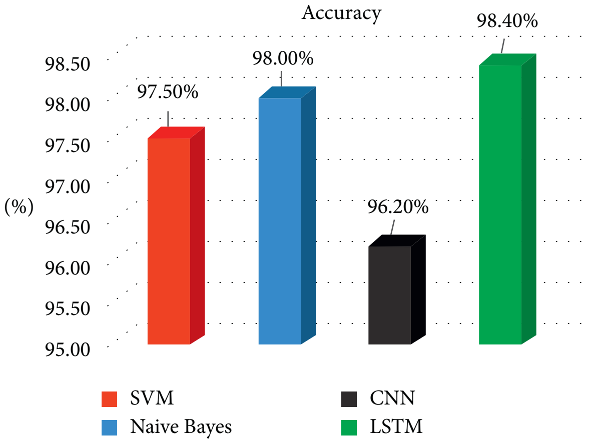


Figure 12: Accuracy graph building comparison between various DL/ML Models.

Table 5: Evaluation parameter values of ML models.

ML models	Precision (%)	Recall (%)	F-measure (%)
Naive Bayes	96.5	95.0	96.0
SVM	97.0	92.0	95.0

This depicts that DL models are particularly good at detecting and classifying Urdu spam emails, as they produce more accurate and precise detection.

In this study, we used existing models for detection of Urdu spam emails, and more training and better detection were also explained for SVM, Naive Bayes, CNN, and LSTM.

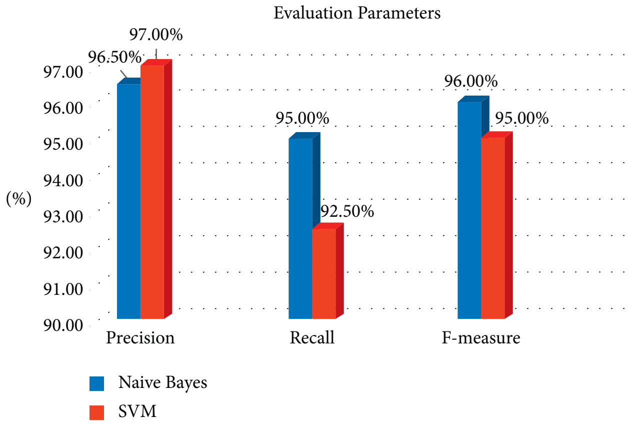


Figure 13: Comparison of Naive Bayes and SVM for precision, recall, and F-measure.

Table 6: ROC-AUC and model loss values of ML models.

ML models	ROC-AUC (%)	Model loss (%)
LSTM	99.0	5.0
CNN	98.0	11.0

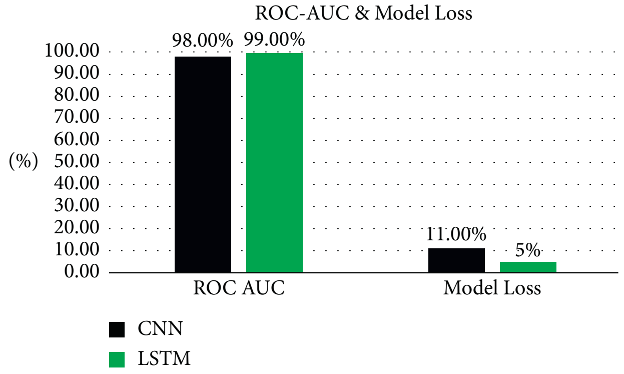


Figure 14: ROC-AUC and model loss comparison graph.

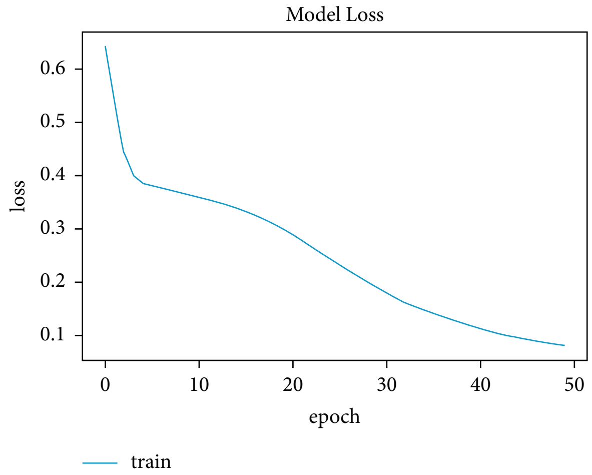


Figure 15: Model loss graph for convolutional neural networks using the proposed dataset for USED.

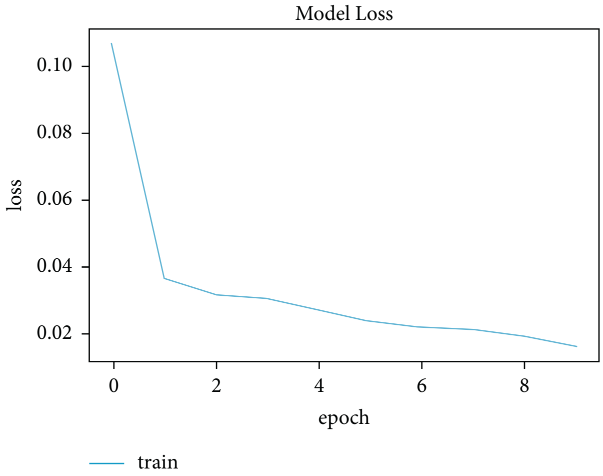


Figure 16: Model loss graph for LSTM model using our dataset for USED.

Furthermore, the accuracy of each model was calculated, and evaluation measures such as precision, recall, and f-measure for SVM and Naive Bayes and for CNN and LSTM as well as the measures ROC-AUC and model loss were used for comparative evaluation. According to the findings, the LSTM model obtained higher accuracy than the other models with a score of 98.4%.

7. Conclusion

With the increase usage of emails, this study focuses on using automated ways to detect spam emails written in Urdu. The study uses various machine learning and deep learning algorithms to detect them. In the study, a translated emails dataset including spam and ham emails is generated from Kaggle, which is preprocessed for various approaches. Accuracy, precision, recall, F-measure, ROC-AUC, and model loss are used as comparative measures to examine performance. The study concludes that deep learning models are more successful in classifying Urdu spam emails. Comparatively, LSTM algorithm has a high accuracy rate of around 98% with low model loss rate of 5%. Even though LSTM takes a little longer to train than CNN, SVM, or Naive Bayes, its efficiency and accuracy rate are far better than those of the other approaches. The creation of an actual dataset of Urdu emails can be considered as a viable future task. In addition, more recent artificial intelligent approaches may also be considered to detect spams.

Data Availability

The dataset is downloaded from ‘Kaggle’ and then converted to CSV format in Urdu.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia, through the Vice Deanship of Scientific Research Chairs: Chair of Cyber Security.

References

[1] N. Kumar, S. Sonowal, and Nishant, “Email spam detection using machine learning algorithms,” in *Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 108–113, IEEE, Coimbatore, India, July 2020.

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

[3] F. Masood, G. Ammad, A. Almogren et al., “Spammer detection and fake user identification on social networks,” *IEEE Access*, vol. 7, pp. 68140–68152, 2019.

[4] A. Akhtar, G. R. Tahir, and K. Shakeel, “A mechanism to detect Urdu spam emails,” in *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 168–172, IEEE, New York, NY, USA, Oct 2017.

[5] H. Drucker, D. Donghui Wu, and V. N. Vapnik, “Support vector machines for spam categorization,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

[6] H. Afzal and K. Mehmood, “Spam filtering of bi-lingual tweets using machine learning,” in *Proceedings of the 2016 18th International Conference on Advanced Communication Technology (ICACT)*, pp. 710–714, IEEE, PyeongChang, Korea (South), Feb 2016.

[7] S. K. Tuteja and N. Bogiri, “Email spam filtering using bpnn classification algorithm,” in *Proceedings of the 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pp. 915–919, IEEE, Pune, India, Sep 2016.

[8] M. Mohamad and A. Selamat, “An evaluation on the efficiency of hybrid feature selection in spam email classification,” in *Proceedings of the 2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, pp. 227–231, IEEE, Kuching, Malaysia, Apr 2015.

[9] P. Sharma, U. Bhardwaj, and U. Bhardwaj, "Machine learning based spam e-mail detection," *International Journal of Intelligent Engineering and Systems*, vol. 11, no. 3, pp. 1–10, 2018.

[10] S. Suryawanshi, A. Goswami, and P. Patil, "Email spam detection: an empirical comparative study of different ml and ensemble classifiers," in *Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC)*, pp. 69–74, IEEE, Tiruchirappalli, India, Dec 2019.

[11] K. Agarwal and T. Kumar, "Email spam detection using integrated approach of naïve bayes and particle swarm optimization," in *Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 685–690, IEEE, Madurai, India, June 2018.

[12] A. Iyengar, G. Kalpana, S. Kalyankumar, and S. GunaNandhini, "Integrated spam detection for multilingual emails," in *Proceedings of the 2017 International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 1–4, IEEE, Chennai, India, February 2017.

[13] K. Kandasamy and P. Koroth, "An integrated approach to spam classification on twitter using url analysis, natural language processing and machine learning techniques," in *Proceedings of the 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science*, pp. 1–5, IEEE, Bhopal, India, March 2014.

[14] X.-l. Chen, P.-y. Liu, Z.-f. Zhu, and Y. Qiu, "A method of spam filtering based on weighted support vector machines,"vol. 1, pp. 947–950, in *Proceedings of the 2009 IEEE International Symposium on IT in Medicine & Education*, vol. 1, pp. 947–950, IEEE, Jinan, China, Aug 2009.

[15] H. Kaur and A. Sharma, "Improved email spam classification method using integrated particle swarm optimization and decision tree," in *Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 516–521, IEEE, Dehradun, India, Oct 2016.

[16] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," *IEEE Access*, vol. 7, pp. 168261–168295, 2019.

[17] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos, *An Evaluation of Naive Bayesian Anti-spam Filtering*, arXiv preprint cs/0006013, 2000.

[18] N. G. M. Jameel and L. E. George, "Detection of phishing emails using feed forward neural network," *International Journal of Computer Applications*, vol. 77, no. 7, 2013.

[19] S. K. Trivedi and S. Dey, "A study of ensemble based evolutionary classifiers for detecting unsolicited emails," in *Proceedings of the 2014 conference on research in adaptive and convergent systems*, pp. 46–51, ACM, New York, NY, United States, October 2014.

[20] H. Kaur and P. Verma, "E-mail spam detection using refined mlp with feature selection," *International Journal of Modern Education and Computer Science*, vol. 9, no. 9, 2017.

[21] T. Kumaresan and C. Palanisamy, "E-mail spam classification using s-cuckoo search and support vector machine," *International Journal of Bio-Inspired Computation*, vol. 9, no. 3, pp. 142–156, 2017.

[22] P. Parveen and P. Halse, "Spam mail detection using classification," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 5, pp. 347–349, 2016.

[23] S. Prof and T. Verma, "E-mail spam detection and classification using svm and feature extraction," *International Jouranl Of Advance Reasearch, Ideas and Innovation In Technology*, vol. 3, no. 3, 2017.

[24] Z. S. Torabi, M. H. Nadimi-Shahraki, and A. Nabiollahi, "Efficient support vector machines for spam detection: a survey," *International Journal of Computer Science and Information Security*, vol. 13, no. 1, p. 11, 2015.

[25] A. Zamir, H. U. Khan, W. Mehmood, T. Iqbal, and A. U. Akram, "A feature-centric spam email detection model using diverse supervised machine learning algorithms," *The Electronic Library*, vol. 38, no. 3, 2020.

[26] S. Rana, S. Jasola, and R. Kumar, "A review on particle swarm optimization algorithms and their applications to data clustering," *Artificial Intelligence Review*, vol. 35, no. 3, pp. 211–222, 2011.

[27] D. Puniškis, R. Laurutis, and R. Dirmeikis, "An artificial neural nets for spam e-mail recognition," *Elektronika ir Elektrotechnika*, vol. 69, no. 5, pp. 73–76, 2006.