





## Proactive MiDLAF: A Novel Mining MinHash-Deep Learning Approach for Advanced Spam Email Filtering



Fryal Jassim Abd Al-Razaq<sup>1</sup>, Ali Kadhim Bermami<sup>2,3</sup>, Ali Khalid Ali<sup>4</sup>, Mehdi Ebady Manaa<sup>5,6\*</sup>

<sup>1</sup> Department of Software, College of Information Technology, University of Babylon, Babil 51001, Iraq

<sup>2</sup> College of Information Technology, University of Babylon, Babil 51001, Iraq

<sup>3</sup> Computer techniques Engineering Department, College of Engineering and Technologies, Al-Mustaqbal University, Babil 51001, Iraq

<sup>4</sup> Computer Center, Al Qasim Green University, Babylon 51013, Iraq

<sup>5</sup> Department of Intelligent Medical Systems, College of Sciences, Al-Mustaqbal University, Babil 51001, Iraq

<sup>6</sup> Department of Information Networks, Information Technology College, University of Babylon, Babylon 51013, Iraq

Corresponding Author Email: [mahdi.ebadi@uomus.edu.iq](mailto:mahdi.ebadi@uomus.edu.iq)

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijse.150305>

### ABSTRACT

**Received:** 13 January 2025

**Revised:** 14 March 2025

**Accepted:** 20 March 2025

**Available online:** 31 March 2025

#### Keywords:

*DDoS attacks, min-hash, DL, spam emails, SVM*

Spam email filtering has recently become the most important task helping in maintaining secure and efficient communication systems. As spam emails lead to security leach, reduced productivity, and increased storage costs, this paper is intended to present a proactive approach to spam email classification, leveraging the advanced techniques to increase detection accuracy and efficiency. The proposed work consists of the three steps. The preprocessing step introduces MinHash which provides a small signature matrix for fast approximation based on a k-shingle technique that generates overlapping sequences of k word, effectively capturing the context and nuances of the spam email text. The second step uses the advanced techniques of machine learning (ML) Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), Multi-Layer Perceptron (MLP), Logistic Regression, and K-Nearest Neighbors (KNN), and Long Short-Term Memory (LSTM) for deep learning (DL) to classify ham and spam emails. The outcomes illustrate that combining the k-shingle, MinHash with advanced text for feeding ML and DL results in high accuracy rate compared with the other works where the SVM classifiers achieves accuracy rate of 98.95% highlighting its effectiveness in distinguishing between ham and spam emails. Other ML shows competitive performance, With MLP 98.25%, RF 95.6%, Logistic regression 98%, DT 93.3%, and lowest accuracy with KNN 70.1%. DL satisfies a high accuracy rate up to 96.1%. This work contributes to the development of a scalable and reliable solution for spam filtering in modern communication systems.

## 1. INTRODUCTION

Digital communication systems pose a significant challenge due to the spread of spam emails, causing a lack in of ability to maintain the integrity and efficiency of these systems. In light thereof, spam email filtering becomes a crucial issue in the process of filtering out unwanted and harmful emails. Spam emails do not only involve unsolicited messages, but also contain non-genuine content sent to many recipients to overwhelm their inboxes, leading to several issues and security breaches. They are spread in a short time, typically by advertising, spreading malware, and phishing [1, 2].

The spam email detection advancement has transitioned from rule-based systems to sophisticated machine learning (ML) methods which involve a computational model and the extraction of valuable insights from raw data through analyzing spam emails using three primary techniques. Classification clustering and association rules are employed to identify spam emails. Classification models examine different

email attributes such as specific keywords message structure and sender details to categorize emails, as legitimate or spam [3].

Those techniques, which are one of the effective approaches that classify pattern into many classes, fall into many types such as Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), and Multi-Layer Perceptron (MLP). Notably, ML techniques produce decision-making for large organizations to predict future decisions, increase reputation, and minimize the security threats. Alternatively, deep learning (DL) such as Long Short-Term Memory (LSTM) uses enhanced capabilities by learning complex patterns directly from spam emails data [4].

Spam emails filtering using ML and DL comes to play an integral role to protect the information for many organizations and increase efficiency. Classification of spam emails is one of the open challenges to protect organization from phishing attacks and emails containing malware. Then, it is critical to improve the network performance, cost saving, and to reduce

the email servers load by filtering spam emails for these organizations. Using spam filtering techniques have been extended to a large scale due to its functionality in minimizing the security threats arising from growth of information technology to increase productivity. It evolves many steps of spam emails data in the pre-processing step such as removing noise, tokenizing the text, and encoding the text into a suitable format for model training. K-shingle is a technique of tokenizing the text into k-words. In addition to tokenization, other pre-processing methods such as stemming, lemmatization, and the removal of stop words are used to normalize the text. These steps are essential to ensure that the pre-processed data fed into ML and DL in a manageable way and to increase the accuracy of spam email detection [5, 6]. In addition, the MinHash (Min-wise Independent Permutations) approach is a technique used to efficiently estimate the similarity between two sets. It is particularly useful for applications in large-scale data mining, such as near-duplicate detection in documents, clustering, and spam detection. MinHash technique uses many hash functions to generate the characteristics matrix for each set of tokens. The characteristics matrix is then converted to a Signature matrix by applying multiple hash functions and storing the minimum value from each to find out the compact representation of spam emails sets [7].

The aim of this work is to classify spam emails using ML and DL to show the effectiveness of the proposed system protection against the spam malicious detection. There are two main steps used in this work: utilize the spam email data using pre-processing steps and k-shingle to convert the pre-processed text into many k words tokenization. The other step is to apply MinHash for the pre-processed text to generate similarity estimation and signature matrices. ML and DL algorithms are used classify the handled spam/ham emails into ham and spam classes. In plain words, ham emails in the same class should be classified as possible, from the spam class emails with minimum error and high accuracy rate. The remaining of this paper is organized as follows. The related works are discussed in Section 2. The proposed system is illustrated in Section 3. The results and discussion are shown in Section 4. Finally, Discussion of the conclusion is presented in Section 5 of the paper.

## 2. RELATED WORK

In their publication, Hadi et al. [8] implemented the Trigonometric Words Ranking (TWRM) for spam emails classification. They presented an efficient approach by using a word ranking strategy in comparison with MinHash and Vector Space Model (VSM) techniques. The main criteria of TWRM are low time complexity due to its efficient word ranking, which reduces the time required of class query message prediction. The results show a good performance for TWRM outperforms both MinHash and VSM in terms of processing time. The proposed method is considered more suitable for real-time spam detection in online communication systems, where speed and accuracy are critical.

Doshi et al. [9] proposed a model for enhancing the detection of phishing and spam email using dual-layer architecture. The proposed model addresses the critical issue of email-Based cyber-attacks, which steal sensitive information by copying legitimate sources. The authors depended on combining the features from both email header

and content during model training, studying the limitation of previous studies that focus on either email content or email body. The results show good performance in accuracy utilizing DL models such as Artificial Neural Network (ANN), Recurrent Neural Network (RNN), and Convolution Neural Networks (CNN) to classify the input data into phishing and spam. Experimental evaluations illustrate high performance metrics, including an accuracy of 99.51%, recall of 99.68%, precision of 99.5%, and F1-score of 99.52%. The main contribution of dual-layer approach is handling imbalance data issues and enhancing the detection of malicious emails which lead to improve cybersecurity in digital communication.

A novel approach for spam emails classification using agglomerative hierarchical clustering and a topic-based methodology was proposed Jáñez-Martino et al. [10]. The proposed system addresses the problem of spam email, which contains malicious content such as malware and phishing attacks. The authors develop two datasets: SPEMC-15K-E and SPEMC-15K-S with a total of 15000 emails each in English and Spanish. The clustering approach labels the datasets into 11 distinct class, while the spam email classification uses four classifiers (SVM, Naïve Bayes, Random Forest (RF) and Logistic Regression) combining with four text representational and tokenization (Bag of Words, TF-IDF, Word2Vec, and BERT). The results indicate that the high performance is achieved using TF-IDF with logistic regression, where the F1 score of 0.953 and accuracy of 94.6% for the English dataset. This approach provides a proactive model for spam email detection, which is crucial for improving email security and digital communication.

The presented work by Zavrak and Yilmaz [11] introduced a novel approach to spam email detection using a hierarchical Attention Hybrid Neural Network (HAN). The model combined two DL Convolutional Neural Network, and Gated Recurrent Units (GRUs) with attention mechanisms. The combined approach improved the classification of email content. The combined mode is designed to focus on essential parts of the email text during the training step, which enhances the contextual understanding, where CNN are used to capture features from the email text, while GRUs are used to model the sequential samples. In addition, the attention mechanism refines this process by focusing on the relevant sections of the text. The hierarchical approach enhancing the classification accuracy to spam and ham emails, but it does not address the time complexity of the proposed approach, where the depth of CNN layers, the size of GRUs, and the nature of the attention mechanism are main factors that affect time. The results demonstrated superior performance in spam detection using various datasets by achieving high accuracy, precision, recall and F1 scores.

A Hybrid correlation-based DL model for classifying spam emails using a fuzzy interface system was proposed by Ayo et al. (2024). They contribute to addressing the low detection rate and high false alarm in spam emails detection. The proposed method uses combining of rule-based hybrid feature selection with DL models. Two techniques of combining correlation-based feature selection subset (CfsSubSetEval) and Rule-based Genetic Search (RBGS) are used to identify the important features with class from a pre-processed spam dataset. The selected features are then fed to deep neural networks for spam classification. Additionally, fuzzy logic is employed for reducing false alarm by categorizing each spam email into many severity levels. The results demonstrate a superior performance with F-score of 96.5 % for the spam and

94.6% for ham test set. Also, the proposed work shows an accuracy of 94%, error rate 5.99%, processing time of 0.5 seconds. The results obtained show an efficiency of spam email detection compared with the other works [12].

Nicholas and Nirmalrani [6] presented a proactive mechanism using a DL technique combined with a bio-inspired algorithm for spam email detection. Their model consists of many steps: the first step uses rigorous pre-processing techniques, including lemmatization and tokenization for dataset set processing. The second step leverages hybrid approaches for feature extraction and classification approach by using both Bag of Words (BoW) techniques and the Novel Sand Cat Swarm Optimization (SCSO) algorithm. The proposed model addresses the ‘Cures of Dimensionality’ by using n-gram features and employing optimal feature selection to improve spam email detection. The results obtained show good performance for accuracy of 92.5% with 10 epochs and minimum computation time. The model extends its capabilities to identify phishing attacks in email security [13].

The suggested work by Miranda García and his colleagues in 2024 explored how DL methods can be applied in tasks, like identifying spam emails detecting malware and filtering adult content in online. They utilized LSTM along with Deep Neural Network (DNN) to effectively filter out spam messages and achieve results with an Area Under the ROC curve (AUC) greater than 0.94 for spam detection. Moreover, the DNN neural network proves to be highly accurate, in spotting malware threats. CNN and transfer learning methods are used to filter content and showcase the advantages of utilizing trained models, for image classification assignments. The outcomes in terms of cost and efficiency are attained through learning techniques that offer an effective solution, for cybersecurity identification and categorizing spam emails [14, 15].

A DL models with feature selection techniques, especially leveraging BERT and Grey Wolf Optimization were proposed by Nasreen et al. [15]. The combined novel email spam detection method demonstrated significant improvement in spam detection accuracy, where they used the Lingspam dataset which handled high-dimensional data. The results achieved 99.14% accuracy using the hybrid approach. Also, the performance hybrid method is much better than the existing algorithm in terms of accuracy, speed, and space complexity. The optimization algorithm reduced the execution time and increased the classification algorithm using advanced DL.

A DL models with dealing of spam emails unbalanced data using Generative Adversarial Networks (GANs), especially leveraging word embedding to feed for ML was proposed in 2024. The combined approach uses GANs and BERT embedding algorithms to improve spam emails detection and to solve the fundamental problem of the data augmentation techniques. The proposed approach focuses on addressing the challenges of unbalanced datasets using BETT embeddings, and then using GANs to generate synthetic data which serves to augment the minority class for the spam message in the datasets. Machines learning methods include LSTM, Bi-LSTM, SVC and others, were trained and tested using the augmented dataset. The results show good performance using accuracy, precision, recall, and F1-score. The Bi-LSTM model outperformed by achieving precision score of 100%, a recall of 95.3%, and the highest f-score of 97.5945%, with an accuracy of 99.3722%.

These results show that the Bi-LSTM is highly effective in detection spam with less false positive [16]. The methodology involves pre-processing the text data, converting it into high-dimensional vector representations using BERT embeddings, and then using GANs to generate synthetic spam messages. Various ML models, including LSTM, Bi-LSTM, SVC, and others, are trained and evaluated on this augmented dataset.

The recent study discussed many approaches for spam emails detection. Jamal et al. [17] presented (IPSDM) an improved phishing spam detection transformer model based on fine-tuning and the BERT family of models. The results are better classified the emails into spam/ham for balanced and imbalanced datasets. Performance metrics in term of classification accuracy, precision, recall, and F1-score outperformed the baseline models. The validation accuracy results for RoBERTA balanced da-tasets were 30.28%, compared to the proposed IPSDM 51.32%, for RoBERTA imbalanced datasets, the ac-curacy was 43.78%, whereas the proposed IPSDM achieves 66.97%, compared with many baseline models for balance and unbalanced datasets. The accuracy continued to improve after the optimization process and addressing the overfitting issue, until it reached the satisfied rate and highlight the critical role of data balancing in enhancing the model performance.

### 3. THE PROPOSED SYSTEM AND METHODOLOGY

The system outlined in Figure 1 shows the cases of the process of categorizing email spam, across phases. The initial phase includes pre-processing steps like tokenization, punctuation removal stops word, lemmatization and k shingle creation for raw email information readiness. The second stage applies the MinHash method to spot similarities within email content. Lastly, the third stage involves the use of ML models, for spam categorization. The system’s effectiveness is assessed using measurements such, as Accuracy and F score along with Precision and Recall among other relevant performance metrics utilized in the evaluation process of spam detection mechanisms integrating DL methods, for improved classification accuracy. The sections below are the main steps of the proposed system.

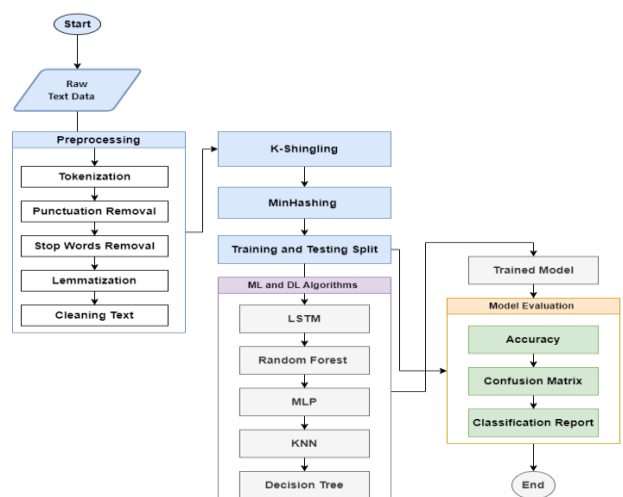


Figure 1. The main steps of the proposed system

### 3.1 Pre-processing steps

The first step is to analyse spam emails. The k-shingle approach uses the k character to tokenize the document into sets of words after the pre-processing of stop word and lemmatization. The following steps illustrate the k-shingle process:

- a) Pre-process the spam text by removing the punctuation and adjusting the white space.
  - b) K-shingle Generation and MinHash Steps: Choose the (K) words to divide the data spam emails into sets of words.
- For example, to apply (a) and (b) steps, we have the following:

**Spam Email 1:**

Subject: Win a Free iPhone!  
 Dear User,  
 You have been selected to win a brand new iPhone. Click on the link below to claim your prize.  
 www.fakewebsite.com.  
 Congratulations!

**Spam Email 2:**

Subject: You Won a Lottery!  
 Hi there,  
 You have won a lottery of \$1,000,000. Please provide your bank details to receive the money.  
 www.anotherfakewebsite.com  
 Best regards,  
 Lottery Team

A). *Tokenization:*

**Email 1:** ['Win', 'a', 'Free', 'iPhone', 'Dear', 'User', 'You', 'have', 'been', 'selected', 'to', 'win', 'a', 'brand', 'new', 'iPhone', 'Click', 'on', 'the', 'link', 'below', 'to', 'claim', 'your', 'prize', 'www.fakewebsite.com', 'Congratulations']

**Email 2:** ['You', 'Won', 'a', 'Lottery', 'Hi', 'there', 'You', 'have', 'won', 'a', 'lottery', 'of', '\$1,000,000', 'Please', 'provide', 'your', 'bank', 'details', 'to', 'receive', 'the', 'money', 'www.anotherfakewebsite.com', 'Best', 'regards', 'Lottery', 'Team']

B). *Punctuation Removal:*

**Email 1:** ['Win', 'a', 'Free', 'iPhone', 'Dear', 'User', 'You', 'have', 'been', 'selected', 'to', 'win', 'a', 'brand', 'new', 'iPhone', 'Click', 'on', 'the', 'link', 'below', 'to', 'claim', 'your', 'prize', 'wwwfakewebsitecom', 'Congratulations']

**Email 2:** ['You', 'Won', 'a', 'Lottery', 'Hi', 'there', 'You', 'have', 'won', 'a', 'lottery', 'of', '1000000', 'Please', 'provide', 'your', 'bank', 'details', 'to', 'receive', 'the', 'money', 'wwwanotherfakewebsitecom', 'Best', 'regards', 'Lottery', 'Team']

C). *Stop Words Removal:*

**Email 1:** ['Win', 'Free', 'iPhone', 'Dear', 'User', 'selected', 'win', 'brand', 'new', 'iPhone', 'Click', 'link', 'claim', 'prize', 'wwwfakewebsitecom', 'Congratulations']

**Email 2:** ['Won', 'Lottery', 'Hi', 'You', 'won', 'lottery', '1000000', 'Please', 'provide', 'bank', 'details', 'receive', 'money', 'wwwanotherfakewebsitecom', 'Best', 'regards', 'Lottery', 'Team']

D). *Lemmatization:*

**Email 1:** ['Win', 'Free', 'iPhone', 'Dear', 'User', 'selected', 'win', 'brand', 'new', 'iPhone', 'Click', 'link', 'claim', 'prize', 'wwwfakewebsitecom', 'Congratulations']

**Email 2:** ['Win', 'Lottery', 'Hi', 'You', 'win', 'lottery', '1000000', 'Please', 'provide', 'bank', 'detail', 'receive', 'money', 'wwwanotherfakewebsitecom', 'Best', 'regards', 'Lottery', 'Team']

The final cleaned tokens for email 1 and email 2 after pre-processing.

### 3.2 K-shingle

For example, with k=3, we have the following shingles:

**Email 1:** ['Win Free iPhone', 'Free iPhone Dear', 'iPhone Dear User', 'Dear User selected', 'User selected win', 'selected win brand', 'win brand new', 'brand new iPhone', 'new iPhone Click', 'iPhone Click link', 'Click link claim', 'link claim prize', 'claim prize wwwfakewebsitecom', 'prize wwwfakewebsitecom Congratulations'].

**Email 2:** ['Win Lottery Hi', 'Lottery Hi You', 'Hi You win', 'You win lottery', 'win lottery 1000000', 'lottery 1000000 Please', '1000000 Please provide', 'Please provide bank', 'provide bank detail', 'bank detail receive', 'detail receive money', 'receive money wwwanotherfakewebsitecom', 'money wwwanotherfakewebsitecom Best', 'wwwanotherfakewebsitecom Best regards', 'Best regards Lottery', 'regards Lottery Team'].

### 3.3 MinHash concepts

Generate the characteristic matrix (S), If two emails E1 and E2 are used as an example. The document E1 consists of the sentence “Win Free iPhone” with k=3. Table 1 shows the characteristic matrix (S) that includes the tokens of shingles in column (1) and its existence in either email 1 or email 2. The two other emails E1 and E2 are tokenized in the same process.

**Table 1.** Characteristic matrix based on k-shingle for email n

Shingles	E1	E2	.. En
Win Free iPhone	1	0	1
Free iPhone Dear	1	0	0
Win Lottery Hi	0	1	0
...			
N shingles from emails	0/1	0/1	0/1

#### 3.3.1 MinHash algorithm

The idea is to convert large data sets shingling to small group signatures. These signatures are used to measure similarity between emails. The general formula of this technique is illustrated in Eq. (1):

$$h(x) = (ax + b) \bmod p \tag{1}$$

The following terms are used in Eq. (1): x refers to the tokens (shingles) in the original characteristic matrix (S). a, b are random numbers which are less or equal to the prime number (p). p, is a prime number (slightly larger than the total number of shingles sets). Algorithm (1) shows the main step of this technique [18].

#### Algorithm 1: MinHash Steps

<b>Input</b>	Characteristic Matrix M, Hash Functions $h_1, h_2, h_3 \dots h_n$ .
<b>Output</b>	Signature Matrix (S)
<b>Begin</b>	
<b>1.</b>	Picking n randomly hashing functions $h_1, h_2, h_3 \dots h_n$ .

2. Create the signature Matrix S using the Matrix M by assigning each row (indexed by i) as a hash function and each column (indexed by c as a document). Then designate SIG(i,c) representing the element, in the signature matrix, for the ith hash function hi(r). Column c.
  - 2.1: Convert the long bit vector into short signatures. For each column c in documents, do the following
    - a. if c has 0 in both documents rows r, do nothing.
    - b. if row has 1, then, for each i=1,2, ....., n set SIG(i,c) to the smaller value of the current value of SIG(i,c) and hi(r)
  - 2.2 Then  $\Pr[h_r(c1)=h_r(c2)]=\text{sim}(c1, c2)$

END

The output of algorithm (2) is a signature matrix which shows the signature matrix using two hash functions in the form (hash of single, D1, D2, ..., Dn). The signature matrix that is generated is converted in Table 2 using two hash functions in rows and documents in columns.

### 3.3.2 ML and DL algorithms

ML is discussed below in subsection (A).

#### A. ML

In today's world progress, ML systems like RF, SVMs, DTs, and Logistic Regression play key role in fields like healthcare imaging analysis and ecological surveillance efforts due to their ability to handle intricate tasks effectively. RF stands out as a method that combats overfitting issues by combining DTs together for better accuracy, on complex and noisy datasets. Recent studies have demonstrated that SVM performs well in situations because it can determine the best boundaries between classes effectively and accurately. On the one hand, DTs are simple to understand and offer insights quickly; however, they may overfit if not pruned correctly. MLPs utilize layers in order to detect patterns in data which is especially useful for datasets that are nonlinear, like those found in image and speech recognition tasks. Logistic Regression is commonly used for classification due to its nature and ease of interpretation when dealing with linear connections between variables. On the hand, KNN can be powerful for tasks classification when data points are closely situated to their neighbors in lower dimensional spaces despite of its heavy computational requirements in higher dimensions. Studies that compare these two algorithms highlight benefits and performance fluctuations based on factors such as size, dimensionality and the specific application at hand. In researches conducted previously RF and SVM have been proven as choices for applications in remote sensing and biomedical data because of their effectiveness, in managing noise and intricate data structures [19, 20].

#### B. LSTM algorithm

It is quite probable that the error-incorporated signals propagating backward in time would vanish or explode because of back-propagation with real-time recurrent learning or time; the magnitude of the error-incorporated signal's temporal shifts depends heavily on the weight sizes. In the event of a burst, the weights will probably begin to oscillate, and in the event of disappearance, either the time required to learn bridging is too high or, worse, it doesn't function. In 1991,

Sepp Hochreiter and Jurgen Schmidhuber introduced the LSTM method, a new kind of recurrent neural network, to address the problems above with error back-propagation. This approach outperformed the previous systems. This long-short-term memory algorithm's original implementation merely used cells, input gates, and output gates [21].

Table 2. Signature matrix of Table 1

Hash Function	E1	E2	...En
h1	1	1	1
h2	1	1	2

The input layer, one hidden layer, and the output layer are the three main parts of the LSTM architecture. A collection of recurrently linked units that make up the hidden layer's single-cell blocks. The input vector  $xb$  is added to the network at time  $t$ . Eqs. (2)-(7) outline the properties of each block [21].

$$f_t = \sigma(W_f xt + U_f h_{t-1} + b_f) \quad (2)$$

$$i_t = \sigma(W_i xt + U_i h_{t-1} + b_i) \quad (3)$$

$$o_t = \sigma(W_o xt + U_o h_{t-1} + b_o) \quad (4)$$

$$\bar{c}_t = \tanh(w_c xt + U_c h_{t-1} + b_c) \quad (5)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \bar{c}_t \quad (6)$$

$$h_{t=0_t} \odot \tanh(C_t) \quad (7)$$

In each LSTM block, the forget, input, and output gates are described by Eqs. (2)-(7), where  $f_t$ , and  $i_t$  are the corresponding variables. At time  $t$ , the relevant values are updated by the input gate, information can be forgotten and discarded by the forget gate, and the output gate and the block output select the outgoing data. The block input at time  $t$ , denoted as  $C_t$  in Eq. (5), is a tanh layer. Together with the input gate, the two determine the new data that needs to be stored in the cell state. Eq. (6) updates the cell state at time  $t$ , which  $C_t$  represents. Finally, the output of the  $h_t$  is block is at time  $t$ . The bias vector is denoted by  $b$ , while the weight matrices are  $W$  and  $U$ . The  $\odot$  sign is obtained by multiplying two vectors point-wise. The functions  $\sigma$  and  $\tanh$  represent hyperbolic tangent activation and point-wise non-linear logistic sigmoid, respectively.

MinHash is used to generate compressed and similarity features vectors, which are then passed to the DL model (e.g., LSTM) for spam email classification. The hybrid model enhances the accuracy and the ability to generalize similar inputs. A step-by-step pseudocode between the input vector features for the DL is illustrated in algorithm (2).

#### Algorithm 2: Spam email Detection using ML and DL based MinHash Technique

**Input** Spam/ham emails E1, E2, E3, ..., En

**Output** Classification of Spam/Ham emails

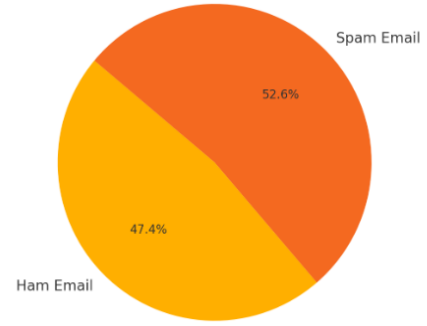
**Begin**

1. Preprocess

2. Preprocessing Emails Spam.

Set  $k$ =integer, then construct from each emails a set of  $k$ -shingle.

3. Generate vector features Signature Matrix S from Algorithm (1), feed the signature matrix vector features to the ML, and DL.
  4. Generate Signature Matrix S from Algorithm (1), if the signature matrix is short then stop.
  5. Using ML and DL to classify the spam email into spam/ham emails (algorithm (2)).
  6. Optionally, go to emails and check the similarity ratio.
  7. Evaluation Step:
    - a) accuracy b) f-score measure c) Silhouette Index
- End**



**Figure 2.** Distribution of spam/ham emails in original dataset

**Table 3.** Spam/ham emails dataset description

Aspect	Description
Dataset Origin	Kaggle
Dataset Origin	The dataset combined 2007 TREC Public Spam Corpus, and Enron Spam Dataset
Dataset Language	English
Time Range	1999-2002 for Enron, and 2005-2007 for TREC Spam Corpus
Class Balancing	Random undersampling of ham class to balance class distribution
Text Cleaning	Yes, by removing HTML tags and special characters
Text Normalization	Yes, by lowercasing and stopword removing

Table 3 shows the main description of the *2007 TREC Public Spam Corpus* and *Enron-Spam Dataset*.

#### A). Accuracy

The accuracy metric calculates the percentage of identified instances (including positives and true negatives) relative, to the total number of instances analyzed. Accuracy is calculated by using Eq. (8).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

TP: True Positives are instances that are accurately predicted as positive.

TN: True Negatives are instances where negative predictions are accurately made.

FP: False Positives are instances where something is inaccurately identified as positive.

FN: False Negative are instances where something is inaccurately identified as negative.

#### B). Precision

Precision is calculated as the ratio of identified cases, to all cases identified as positive as shown in Eq. (9).

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

#### C). Recall

Recall calculates the ratio of predicted cases to all real positive cases.

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

The main steps of pseudo-code for algorithm (2) is shown below

```
# Step 1: Preprocessing
Input: Raw text email, E1, E2, ... En
Output: Classification of Emails into Ham and Spam
Begin
tokens=preprocess_text(emails)

# Step 2: Shingling
k_shingles=generate_k_shingles(tokens)

# Step 3: MinHashing
Generate minhash_signature=apply_minhash(k_shingles)

# Step 4: Feature Normalization
normalized_vector=normalize(minhash_signature)

# Step 5: DL/ML Input
prediction=Apply ML (SVM, DT, LR, KN, and ANN)/LSTM_Model(normalized_vector)

# Step 6: Classification Output
Confusion_matrix=CM(prediction_model, testing)
return "Spam" or "Ham"

# Step 6: Performance Output
return "Performance Classification Report: Accuracy, Precision, Recall, and F-score"
```

## 4. RESULTS AND DISCUSSION

### 4.1 Dataset collection

This work is implemented using the python language with laptop specification: CPU Intel(R) 12<sup>th</sup> Gen Intel(R) Core (TM) i5-12500H 2.50GHz, 8GB memory size and Windows 11 as the operating system. The email data is gathered from Kaggle dataset [22]. The CV file containing of 83446 records from spam and ham emails, where label "1" indicate that the email is classified as spam, and "0" denotes that the email is ham. It is formed by combining the 2007 TREC Public Spam Corpus and Enron-Spam Dataset original link [23]. The proposed model is implemented using 20,000 records from the total of 83446. There are different types of emails size ranging on the available datasets. Emails dataset are downloaded and then pre-processed to be classified in the proposed system using ML and DL, the ratio of training and testing is 70:30 for the snap. Figure 2 shows the distribution of original spam/ham emails.



D). F-measure

The F score calculates a value taking into consideration both precision and recall by using their mean which accounts for inaccuracies, in both directions. False positives and false negatives are effectively captured in one metric, for analytic convenience.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (11)$$

The results indicate that SVM and RF exhibit performance levels; SVM demonstrates near precision and recall rates along with an F Measure score of 99% achieving the highest test accuracy at 98.95% establishing itself as the leading performer in the study. RF trails with notable precision (99%) recall

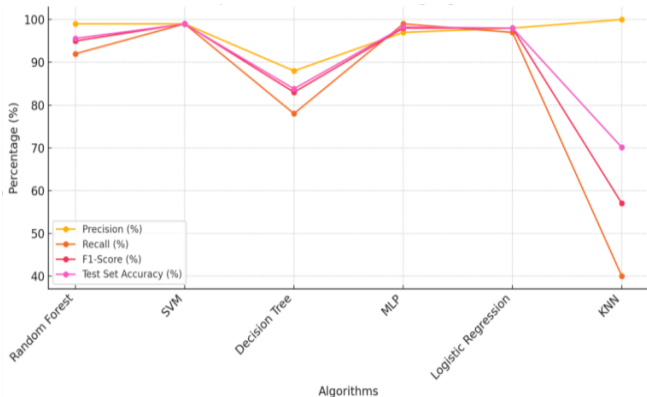
(92%) and F Measure (95%) scores alongside a solid test accuracy of 95.6% striking a balance, between performance and resilience. MLP also achieves accuracy at 98% along with balanced performance showing 97% precision and 99 % recall rates. True to its name Logistic Regression demonstrates metrics with an accuracy score of 98% implying a linear distinction within the dataset. In addition, Tree stands out for its interpretability falls short of generalization with an accuracy rate of 83.8, a decrease in recall to 78%, at overfitting issues. In conclusion, K nearest neighbors (KNN) exhibits the performance comparable to models specifically in terms of recall at 40% and F measure at 57%. This indicates that KNN faces challenges in generalizing to the dataset due to its sensitivity to high dimensional spaces or imbalances in class distribution as illustrated in Table 4.

**Table 4.** The main results of ML algorithm

Algorithms	Precision (%)	Recall (%)	F1-Score (%)	Test Set Accuracy (%)
RF	99	92	95	95.6
SVM	99	99	99	98.95
DT	88	78	83	83.8
MLP	97	99	98	98.25
Logistic Regression	98	97	98	98
KNN	100	40	57	70.1

Figure 3 shows the main comparison between these algorithms.

LSTM learning model has undergone 10 training epochs and achieved an accuracy of 96.1%. The accuracy level is, at 96.1%, Precision 95%, Recall 96%, F1-Score 95.5% and the final test accuracy 96.1%.



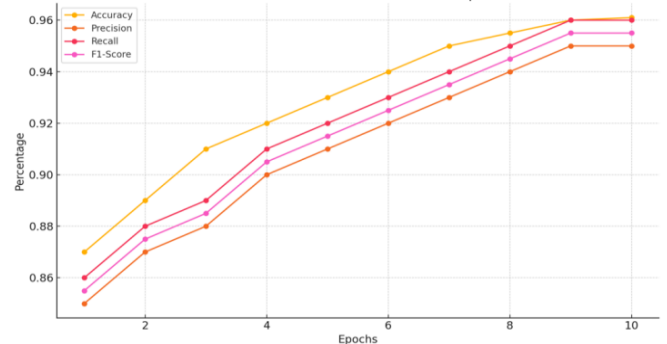
**Figure 3.** Comparisons of ML

**Table 5.** Performance metrics for LSTM model

Epoch	Accuracy	Precision	Recall	F1-Score
1	0.87	0.85	0.86	0.855
2	0.89	0.87	0.88	0.875
3	0.91	0.88	0.89	0.885
4	0.92	0.9	0.91	0.905
5	0.93	0.91	0.92	0.915
6	0.94	0.92	0.93	0.925
7	0.95	0.93	0.94	0.935
8	0.955	0.94	0.95	0.945
9	0.96	0.95	0.96	0.955
10	0.961	0.95	0.96	0.955

These numbers are standard for a LSTM in tasks such as classification rate for time series of spam emails accurately by striking a balance between accuracy and completeness of

results. The F score demonstrates how well the model balances precision and recall while the Test Set Accuracy of 96.1% shows that it can perform well with data. For the 10,000-snap dataset, we have the first 10 epochs based MinHash in Table 5.



**Figure 4.** LSTM performance for the first 10 epochs

**Table 6.** Compassion with related works using the same datasets

Model	Accuracy (%)	Dataset
SVM (proposed work)	98.95	TREC+Enron datasets
MLP (proposed work)	98.25	TREC+Enron datasets
Logistic Regression (proposed work)	98	TREC+Enron datasets
RF (proposed work)	95.6	TREC+Enron datasets
DT (proposed work)	93.3	TREC+Enron datasets
KNN (proposed work)	70	TREC+Enron datasets
LSTM (proposed work)	96.1	TREC+Enron datasets

ANN (Alsuwit et al. [24])	98	TREC+Enron datasets
LR, RF, NB (Jamal et al. [17])	97	TREC+Enron datasets
DistilBERT (Jamal et al. [17])	98.67	TREC+Enron
RoBERTa (Shazad et al. [25])	98.92	TREC+Enron
BERT (Shazad et al. [25])	98	Enron
SVM (Champa et al. [26])	96	TREC+Enron
RF (Champa et al. [26])	95	TREC+Enron

Figure 4 shows the LSTM performance model for the testing accuracy over 10 Epochs.

The effectiveness of the proposed model was compared with the published related works using the same dataset (TREC, Enron, and combined TREC and Enron). As shown in Table 6, the proposed SVM approach achieved 98.95% outperformed several state-of-the-art models including ANN (98% and BERT 98%). In addition, the proposed MLP (98.25%) and Logistic Regression (98%) showed highly competitive performance. The RoBERTa (98.92) and DistilBERT (98.67) closely to the proposed SVM, indicating that traditional ML model, when combine with preprocessing, can still compete closely with transformer-based models in spam classification.

## 5. CONCLUSIONS

The analysis of distinguishing spam emails using ML and DL based on MinHash technique produced some findings. The DL models, like LSTM and the traditional ML models such as SVM and RF showed results with LSTM reaching an accuracy of 96.1. SVM topping at 98.95%. These models demonstrated their ability, in identifying patterns within the dataset and managing both nonlinear connections effectively. SVMs, with their outstanding outcomes displaying scores in precision and recall as well as F1 score metrics alike indicate their suitability for tasks involving the classification of spam emails wherein pinpoint identification of spam and reducing false negatives are deemed crucially important aspects to be considered for successful detection of spam messages. Spam detection tasks is the commendable performance offered by RF which boasts an impressive overall accuracy rate of 95%. This can be attributed to its capability of effectively managing datasets and intricate feature relationships thereby solidifying its status as another option, for undertaking such a challenging undertaking.

In the land of DL technology lies LSTM-known for its ability in handling data and demonstrating competitiveness by reaching accuracy levels comparable with SVMs. Its knack for grasping relationships, within email text sequences proves pivotal in the realm of spam filtering systems; especially when it comes down interpreting word patterns as they evolve over time. While DTs and KNN show results in their performance metrics comparable to advanced models such, as RF and SVM, they fall short in terms of accuracy and robustness when dealing with complex and high dimensional text-based email data due to overfitting issues noted in DTs 83.8% Accuracy. KNNs rate of 70.1%. On the other hand, Logistic Regression and MLP demonstrate performance with accuracies hovering around 98% making them reliable options, for situations where computational simplicity is a key consideration.

To sum up the discussion, both ML and DL models utilizing MinHash techniques prove effective, in categorizing spam content. DL models like LSTM and sophisticated ML models

such as SVM and RF have shown performance in this domain. These models strike a balance between precision, recall and accuracy; which makes them well suited for scenarios where misclassification of spam emails can have consequences. The selection of a model should be based on the task requirements, including factors like resources, interpretability and the trade off, between precision and recall. Also, the hybrid model increased the computational cost for the training step, but for the testing step, the time is close to zero after generating the Min-hash based features vectors. The generalization ability of the proposed method on small dataset remains effective and validated. As part of the future work, a lightweight model and optimized MinHash variants should be implemented to reduce computational overhead by applying the future framework on real-time streaming environments. This enhancement will increase the scalability and integration of the proposed system.

## REFERENCES

- [1] Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., Alegre, E. (2023). A review of spam email detection: Analysis of spammer strategies and the dataset shift problem. *Artificial Intelligence Review*, 56(2): 1145-1173. <https://doi.org/10.1007/s10462-022-10195-4>
- [2] Sajwan, M.S., Singh Mahar, A., Rawat, P., Sharma, K., Das, P. (2024). Email spam filtration with machine learning. In *2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)*, Gwalior, India, pp. 1361-1366. <https://doi.org/10.1109/AIC61668.2024.10731098>
- [3] Agarwal, R., Dhoot, A., Kant, S., Bisht, V.S., Malik, H., Ansari, M.F., Afthanorhan, A., Hossaini, M.A. (2024). A novel approach for spam detection using natural language processing with AMALS models. *IEEE Access*, 12: 124298-124313. <https://doi.org/10.1109/ACCESS.2024.3391023>
- [4] Sarker, I.H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3): 160. <https://doi.org/10.1007/s42979-021-00592-x>
- [5] Ahmed, N., Amin, R., Aldabbas, H., Koundal, D., Alouffi, B., Shah, T. (2022). Machine learning techniques for spam detection in email and IoT platforms: Analysis and research challenges. *Security and Communication Networks*, 2022(1): 1862888. <https://doi.org/10.1155/2022/1862888>
- [6] Nicholas, N.N., Nirmalrani, V. (2024). An enhanced mechanism for detection of spam emails by deep learning technique with bio-Inspired algorithm. *E-Prime-Advances in Electrical Engineering, Electronics and Energy*, 8: 100504. <https://doi.org/10.1016/j.prime.2024.100504>
- [7] Mitzenmacher, M., Pagh, R., Pham, N. (2014). Efficient estimation for high similarities using odd sketches. In *Proceedings of the 23rd International Conference on World Wide Web*. New York, NY, USA, pp. 109-118. <https://doi.org/10.1145/2566486.2568017>
- [8] Hadi, S.M., Alsaedi, A.H., Al-Shammari, D., Alkareem Alyasseri, Z.A., Mohammed, M.A., Abdulkareem, K.H., Nuiiaa, R.R., Jaber, M.M. (2022). Trigonometric words ranking model for spam message classification. *IET*



- Networks, 11(6): 249-260. <https://doi.org/10.1049/ntw2.12063>
- [9] Doshi, J., Parmar, K., Sanghavi, R., Shekokar, N. (2023). A comprehensive dual-Layer architecture for phishing and spam email detection. *Computers & Security*, 133: 103378. <https://doi.org/10.1016/j.cose.2023.103378>
- [10] Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., Alegre, E. (2023). Classifying spam emails using agglomerative hierarchical clustering and a topic-Based approach. *Applied Soft Computing*, 139: 110226. <https://doi.org/10.1016/j.asoc.2023.110226>
- [11] Zavrak, S., Yilmaz, S. (2023). Email spam detection using hierarchical attention hybrid deep learning method. *Expert Systems with Applications*, 233: 120977. <https://doi.org/10.1016/j.eswa.2023.120977>
- [12] Ayo, F.E., Ogundele, L.A., Olakunle, S., Awotunde, J.B., Kasali, F.A. (2024). A hybrid correlation-Based deep learning model for email spam classification using fuzzy inference system. *Decision Analytics Journal*, 10: 100390. <https://doi.org/10.1016/j.dajour.2023.100390>
- [13] Gibson, S., Issac, B., Zhang, L., Jacob, S.M. (2020). Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. *IEEE Access*, 8: 187914-187932. <https://doi.org/10.1109/ACCESS.2020.3030751>
- [14] Miranda-García, A., Rego, A.Z., Pastor-López, I., Sanz, B., Tellaeche, A., Gaviria, J., Bringas, P.G. (2024). Deep learning applications on cybersecurity: A practical approach. *Neurocomputing*, 563: 126904. <https://doi.org/10.1016/j.neucom.2023.126904>
- [15] Nasreen, G., Khan, M.M., Younus, M., Zafar, B., Hanif, M.K. (2024). Email spam detection by deep learning models using novel feature selection technique and BERT. *Egyptian Informatics Journal*, 26: 100473. <https://doi.org/10.1016/j.eij.2024.100473>
- [16] Filali, A., Merras, M. (2024). Enhancing spam detection with GANs and BERT Embeddings: A novel approach to imbalanced datasets. *Procedia Computer Science*, 236: 420-427. <https://doi.org/10.1016/j.procs.2024.05.049>
- [17] Jamal, S., Wimmer, H., Sarker, I.H. (2024). An improved transformer-Based model for detecting phishing, spam and ham emails: A large language model approach. *Security and Privacy*, 7(5): e402. <https://doi.org/10.1002/spy2.402>
- [18] Leskovec, J., Rajaraman, A., Ullman, J.D. (2020). *Mining of Massive Data Sets*. Cambridge University Press.
- [19] Thanh Noi, P., Kappas, M. (2017). Comparison of random forest, k-Nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. *Sensors*, 18(1): 18. <https://doi.org/10.3390/s18010018>
- [20] Muñoz-Zavala, A.E., Macías-Díaz, J.E., Alba-Cuéllar, D., Guerrero-Díaz-de-León, J.A. (2024). A literature review on some trends in artificial neural networks for modeling and simulation with time series. *Algorithms*, 17(2): 76. <https://doi.org/10.3390/a17020076>
- [21] Rather, A.M. (2021). LSTM-based deep learning model for stock prediction and predictive optimization model. *EURO Journal on Decision Processes*, 9: 100001. <https://doi.org/10.1016/j.ejdp.2021.100001>
- [22] Singhvi, P. Spam Email Classification Dataset. <https://www.kaggle.com/datasets/purusingshvi/email-spam-classification-dataset>, accessed on Nov. 8, 2024.
- [23] Dada, E.G., Bassi, J.S., Chiroma, H., Abdulhamid, S.I.M., Adetunmbi, A.O., Ajibuwa, O.E. (2019). Machine learning for email spam filtering: Review, approaches and open research problems. *Heliyon*, 5(6): e01802. <https://doi.org/10.1016/j.heliyon.2019.e01802>
- [24] Alsuwit, M.H., Haq, M.A., Aleisa, M.A. (2024). Advancing email spam classification using machine learning and deep learning techniques. *Engineering, Technology & Applied Science Research*, 14(4): 14994-15001. <https://doi.org/10.48084/etasr.7631>
- [25] Shazad, A., Chaudhry, M.N., Abid, M.K., Aslam, N. (2024). Spam email detection using transfer learning of BERT model. *Journal of Computing & Biomedical Informatics*, 5(1): 1-7. <https://jcbi.org/index.php/Main/article/view/349>
- [26] Champa, A.I., Rabbi, M.F., Zibran, M.F. (2024). Curated datasets and feature analysis for phishing email detection with machine learning. In *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*. Mt Pleasant, MI, USA, pp. 1-7. <https://doi.org/10.1109/ICMI60790.2024.10585821>