Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 19(2): 1926–1943.
DOI: 10.3934/mbe.2022091
Received: 02 October 2021
Accepted: 28 November 2021
Published: 22 December 2021

Research article

Evading obscure communication from spam emails

Khan Farhan Rafat[1], **Qin Xin**[2,*], **Abdul Rehman Javed**[1,*], **Zunera Jalil**[1] **and Rana Zeeshan Ahmad**[3]

[1] Department of Cyber Security, Faculty of Computing and AI, Air University, PAF Complex, E-9, Islamabad, Pakistan

[2] Faculty of Science and Technology, University of the Faroe Islands, Vestarabryggja 15, FO 100, Torshavn, Faroe Islands

[3] Department of Information Technology, University of Sialkot, Pakistan

* **Correspondence:** Email: abdulrehman.cs@au.edu.pk, QinX@setur.fo.

Abstract: Spam is any form of annoying and unsought digital communication sent in bulk and may contain offensive content feasting viruses and cyber-attacks. The voluminous increase in spam has necessitated developing more reliable and vigorous artificial intelligence-based anti-spam filters. Besides text, an email sometimes contains multimedia content such as audio, video, and images. However, text-centric email spam filtering employing text classification techniques remains today's preferred choice. In this paper, we show that text pre-processing techniques nullify the detection of malicious contents in an obscure communication framework. We use *Spamassassin* corpus with and without text pre-processing and examined it using machine learning (ML) and deep learning (DL) algorithms to classify these as ham or spam emails. The proposed DL-based approach consistently outperforms ML models. In the first stage, using pre-processing techniques, the long-short-term memory (LSTM) model achieves the highest results of 93.46% precision, 96.81% recall, and 95% F1-score. In the second stage, without using pre-processing techniques, LSTM achieves the best results of 95.26% precision, 97.18% recall, and 96% F1-score. Results show the supremacy of DL algorithms over the standard ones in filtering spam. However, the effects are unsatisfactory for detecting encrypted communication for both forms of ML algorithms.

Keywords: Email classification; spam; ham; machine learning; stenography; text pre-processing

1. Introduction

Today, the internet and social media have emerged as the fastest ways to access data [1]. Aided with advanced technologies such as e-mail, e-fax, and scanning, extracting evocative information such as appraisals, views, feedbacks, messages, and commendations from such data using several Natural Language Processing (NLP) practices is now within reach [2]. NLP, a division of Artificial Intelligence (AI), uses computers and natural language to render valuable information. NLP effectively uses text classification tasks such as spam uncovering and sentiment analysis.

Email is a cheap [3], operative [4], and speedy way to exchange messages [5, 6] over the internet and hence lure users for its continual usage [7–9]. Because of the aptness, emails have become a preferred choice of the vicious to trap their targets by sending malicious email content called phishing, forcing them to take actions that help transfer escalated privileges of user systems for their evil usage [10, 11]. Spam, thus, is an unsolicited, unpackaged email sent to manifold recipients who did not request those. It is the unsolicited and unpackaged aspects that stimulate the threat. Despite substantial cyber security advances and continuous development, spam email and malware damage caused by those can avert communication, besides amplifying many folds of the internet traffic and waste users' time by requiring them to delete those personally [12, 13]. Unsolicited emails cost businesses and people millions of dollars per annum. Despite the introduction and development of several models and auto-spam detection techniques, none has shown 100% predictive accuracy. However, recent research centered around NLP filtering techniques [14] has harnessed the accuracy of machine learning models.

Cryptography and steganography techniques have been in use for a long time for secure communication [15,16]. Cryptography makes the content unintelligible [17] for an onlooker, whereas steganography hides the existence of contents [18]. Steganography stands ahead as its usage does not raise suspicion and has broader use in recent cyber attacks [19, 20]. Steganography is usually a preferred choice in spam filtering techniques. We took the message contents other than English words as pre-processed steganography embedded text for concept depiction.

Ease of implementation, less overhead, and high detection accuracy are the primary justification in using the standard ML algorithm as the preferred method to filtering spam [21]. However, the recent emergence of Graphical Processing Unit (GPU) architecture proficient in supporting Neural Networks (NN) [22] has led to using NN in spam uncovering and filtering [23]. This research exertion attempts to contrast the standard (Gaussian Naïve Bayes) and DL algorithms LSTM in segregating spam and ham email messages. Extending it further, we examined the performance of both forms of ML in detecting stealthily encoded messages sent in emails.

Sending an encrypted email seems like a whistleblower because anyone who sees it knows something is being sent in a veil. Hence, as an alternative, secret messages can be hidden in plain sight, that is, to hide those in innocuous-looking spams. However, such content often gets easily identified by freely available online tools [24]. Because of NLP text pre-processing techniques, the trick is to encode the secret message as space, punctuation, figures, and other attributes that the spam detectors bypass during the classification process. The email recipient can then extract those and decode the embedded message. Therefore, using text pre-processing techniques does facilitate the spam filtering process but nullifies the detection of malicious contents in the veiled communication framework posing a more significant risk in compromising the user's privacy and system's security.

This research paper makes the following contributions:

- We introduce an AI-based approach for detecting harmful content in an obscure communication framework. Since traditional text pre-processing approaches nullify malicious content detection, the proposed approach employs the *SpamAssassin* corpus with and without pre-processing to test its efficiency.

- Propose a practical ML and DL-based approach and conduct a comparison study with and without pre-processing techniques to see how well the proposed approach compares to traditional ML and DL approaches.
- Experiments show that the proposed approach outperforms traditional ML and DL algorithms in terms of *precision*, *recall*, and *f1- score*.

The remainder of the paper is organized as follows. We present related work in Section 2. The dataset used for testing is presented in Section 3. The proposed approach for detecting spam and ham emails is explained in Section 4. The experimental setup and results are discussed in Section 5. Section 6 presents the conclusion and future work.

2. Related literature

Authors in [25] demonstrated that online social networking sites such as Twitter had become a well-known way for web surfers to associate with messages known as tweets [26]. Its popularity, however, has made it a lucrative target for spammers to feed spam posts. Hence, quite a few spam detection techniques have found their way in deterring spam attacks against Twitter. The authors believe that the available ML algorithms failed to effectively identify spammers on Twitter because of reasonable information constraints for unsought clients to explore the arena. The authors presented an embryonic tactic based on a DL technique leveraging a text-predicated feature for detecting spammers. Their novel architecture comprised a one-dimensional dimension reduction initiation module with LSTM alongside an attention layer. Within the suggested model, the initiation module disentangles the features of the vector after GloVe word embedding [25]. After that, LSTM serves to get the context representations. The attention layer contemplates the LSTM outputted data. Finally, the sigmoid classifier labels the tweets as spam or otherwise, comparing the proposed approach against four ML-based and two DL-based methods. An F1-score of 95.74 with an accuracy of 95.75 and precision of 95.58 of our proposed course outperform the rest.

Communication over email has become popular because of being cheap and easy to use for vital information over the internet [27]. However, spam messages often constitute a significant part of the user inbox that wastes resources and valuable user time. Hence, the authors felt the need to have an efficient technique to identify the news as spam or ham and suggested a new model for detecting spam messages built on the sentiment analysis of the email message contents. Incorporating Word-Embeddings and a Bidirectional LSTM network was introduced to analyze texts' sentimental and sequential properties.

[28] stated that insider threats significantly impact businesses, governments, and the military alike, where authorized users have now assumed the potential source as insider threats. Recent insider threats detection approaches are not robust, such as the rule-based approach, which relies on expert knowledge. Above limitations in view, the authors propose blending anomaly detection and email user behavior algorithms by constructing an email content grounded on the IT administrator role using the CERT r6.2 dataset and employing natural language pre-processing components. Modeling was done by generating a vector space over the dataset, which contributed to anomaly detection algorithms detecting malicious email contents. The proposed model showed an 89% detection rate over the reference point model. To [29], phishing attacks serve as a standard tool for spiteful actors to access systems or extort people. Most counter schemes focus on awareness programs and detection models. According to the authors, methods such as detection models employing ML techniques and banned lists and other email

features like URLs, email addresses, and domain names help battle the problem [30–32]. However, it also calls for further research for accurate and precise detection of phishing scams in scenarios where phishing email data can significantly contribute. In recent times, phishing attacks exploit human weaknesses by getting specific to human emotions, such as anxiety, distress, and horror, to trick users into compromising their details. Taking the emotional exploitation situations framed in phishing emails and combining those with current uncovering arrangements could pave the way for improved detection. The authors explored the linkage between the subjugated emotion and the sender's email sphere. It also collected and analyzed the numerous phishing email types and their accompanying emotional contents in evolving restored detection methods.

[33] pointed exploitation of the internet in delivering malware and indirectly facilitating child sex reprobates to connect, host, and disseminate child sexual contents and extending that to quest new victims through evading detection. The authors elucidated the significance of steganography applications. Still, they showed apprehensions that such tools have become problematic for law enforcement officials. Since the prospect of a forensic investigator realizing illegitimate data hidden by steganography tools diminishes with the rise in usage of such tools. The situation has confronted the Law enforcement officers with the challenge of exploring direct and low-cost detection tools to confirm the steganography usage and uncover and extract the concealed data. With sophisticated applications, such as Cellebrite and EnCase, the authors emphasized investigators' understanding of steganography in discovering and identifying left behind features of the steganography artifacts.

[34] asserted that email spam classification is a significant threat to an organization's security and ML stratagems. Spam messages are even riskier to data security that can take data for vindictive purposes. For example, if compromised, account logins or credit cards information may be subjected to financial deception. The author used Multinomial Naïve classifier in segregating ham and spam emails and feared that text pre-processing is sternly a disregarded topic. In this regard, the author attributed the noticeable inconsistencies due to the absentia of NLP text pre-processing techniques.

[23] viewed that email is preferred in official matters even though spam emails are a central issue on the internet. It is easy to send an email containing spam messages by the spammers. Spam fills the inbox with numerous irrelevant emails that help steal sensitive information from devices such as files and contact. It is challenging to detect spam emails even with state-of-the-art technology. The authors proposed a Term Frequency-Inverse Document Frequency (TFIDF) approach using the Support Vector Machine (SVM) algorithm in this research. The comparison metric included the confusion matrix, accuracy, and precision. The suggested approach rendered an accuracy of 99.9% on the training dataset and 98.2% on the testing dataset, respectively. The method proposed by [35] accomplished better results than other conventional methods that showed a word embedding effect in DL for email spam detection. [36] evolved THEMIS, a learning model that used an improved Recurrent Convolution Neural Network (RCNN) to detect phishing messages considering the email header and the email body at the character and the word levels. A reported accuracy of 99.84% for THEMIS was higher than both LSTM and a Convolutional Neural Network (CNN).

3. Preliminaries

The following elucidates the groundwork supporting our assertion that state-of-the-art spam filtering algorithms founded on NLP are likely to fall prey to even rudimentary steganography algorithms.

3.1. Dataset

The need for spam and phishing email parting is quite noticeable in email datasets. For example, the Enron dataset contains legitimate ham emails. In contrast, the University of California, Irvine (UCI) machine learning repository contains a dataset for spam emails and the Nazario dataset stores phishing emails. However, **the 'SpamAssassin' dataset has ham, hard Ham (a bit trickier contents), and spam emails [37] and hence remained a preferred choice as a dataset for the author**. The details about the dataset appear in Table 1, which are downloaded from the SpamAssassin website.

Table 1. Dataset details.

e-mail type	Description
spam	500 spam messages
easy_ham	2500 non-spam messages
hard_ham	250 non-spam messages which are closer in many respects to typical spam
easy_ham_2	1400 recent non-spam messages
spam_2	1397 recent spam messages
Total	6047 messages having around 31% spam ratio

3.2. Natural language text pre-processing techniques

The necessary stages in the mining of data from email messages have the following categorization in the text pre-processing framework:

1) **Tokenization and Segmentation:** It divides the given text into minor tokens, where words, punctuation marks, numbers, and others constitute tokens. [38].

2) **Normalization:** These are the steps needed for translating text from human language (English) to machine-readable format for further processing [39]. The process starts with:

   - changing all alphabets to lower or upper case
   - expanding abbreviations
   - excluding numbers or changing those to words
   - removing white spaces
   - removing punctuation, inflection marks, and other circumflexes
   - removing stop words, sparse terms, and particular words
   - text canonicalization

3) **Noise Removal:** It is a task-specific section that can occur before or after the tokenization and normalization or in between. Examples include removing HTML Tags, CSS, and similar other traits. Dimensionality reduction creates a new set of attributes, whereas feature selection methods include and exclude details present in the data without alteration [40].

4) **Feature selection:** Following the pre-processing stage is the feature selection. Unlike dimensionality reduction, feature selection effectively typifies email message fragments as a compressed feature vector. The technique comes in handy for large datasets.

3.3. Preferred algorithms

1) **Naïve Bayes:** Strong independence assumptions between the features exist for Naïve Bayes. They have a simple design, easy to implement and train. This algorithm needs small training data to estimate the parameters required for classification purposes. This classifier show aptness to real-life situations. The math behind the Naïve Bayes algorithm is as follows:

$$\begin{aligned}
P(A \mid B) &= \frac{P(A \cap B)}{P(B)} \\
P(A \cap B) &= P(A \mid B)P(B) \\
Likewise, P(B \mid A) &= \frac{P(B \cap A)}{P(A)} \\
P(B \cap A) &= P(B \mid A)P(A) \\
P(A \mid B)P(B) &= P(B \mid A)P(A) \\
P(A \mid B) &= \frac{P(B \mid A)P(A)}{P(B)}
\end{aligned} \tag{3.1}$$

2) **Gaussian Naïve Bayes:** A variant of Naïve Bayes (NB), Gaussian Naïve follows Gaussian normal distribution and differs in provisioning support for continuous type data [41]. The prospect of the features is assumed to be:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{3.2}$$

An approach to creating a model is finding the mean and standard deviation of the points inside each label.

3) **LSTM Recurrent Neural Network (RNN):** are devoid of short-term memory, as shown in Figure 1 [*]. They can not synchronize with previous time steps, such as text processing, for extended sequences of information. They suffer from the waning gradient problem during backpropagation, where gradients are values that update the weights of a neural network. The vanishing gradient problem occurs when the gradient shrinks while propagating back through time and loses contribution in learning for minimal values.

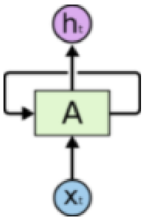


Figure 1. Illustration of recurrent neural network.

[*] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM offers a solution to the short-term memory problem of RNN with a specialized internal mechanism called gates (forget, input and output) to regulate information flow, as shown in Figure 2 [†]. Over time, these gates learn the significance of each data sequence to decide on its retention and pass pertinent information down the long chain of sequences to make predictions. LSTM may become an essential ingredient in text generation, speech synthesis and recognition, and most importantly, generating captions for videos.

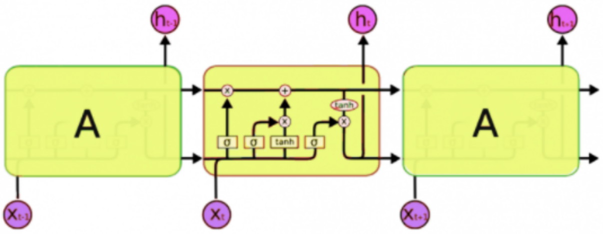


Figure 2. Illustration of LSTM recurrent neural network.

Word Cloud provides a first-rate option to analyze the text data through visualization in tags or word form. The frequency associated with each word is expressed in the size of that word and illustrates the word's significance.

4. Proposed methodology

An email message primarily contains two main components: the header and the body. The header part contains broader content-related information such as the subject, sender, recipient email addresses, and timestamp. The heart of the email is its body comprising variable contents that vary from message to message. For example, it may include a web page, files, analog data, images, audio, video, and HTML markup, among other innards [42]. The header line usually commences with a ”From” field. The email undergoes some alteration as it moves from one server to another through intermediate servers. Users may inspect headers to view the email adhered's routing path. The contents may go through pre-processing before they serve as an input to the classifier. We extract email messages for classification purposes. Uniform Resource Locator (URLs), Hypertext Markup Language (HTML), and cascading style sheets (CSS), JavaScript code, special symbols are removed first from the email message, leaving unformatted text only in the stage called pre-processing.

We first proceed to the standard ML algorithm. For text-based classification, the NLTK library tokenized the message text into terms where each term got quantized using the TFIDF technique. Within the proposed framework, the model extricates the features using TFIDF Vectorizer followed by Gaussian Naïve Bayes (comes with the sklearn python library) for prediction. The case of deep machine learning is a special one. In this part, the tokenization layer maintains a dictionary that maps a word to an index. The embedding layer internally maintains a lookup table, which maps the index/token to a vector. It is the vector that represents the word in the complex dimensional space. The process appears as email $\rightarrow Token \rightarrow lookuptable \rightarrow vector$.

We use components such as Embedding and Bidirectional LSTM Layers (Bi-LSTM) and GRU in building our network. The purpose of using bidirectional is that the situational information often comes

[†] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

at the end of the sentence sometimes. Without using this information, uncertainty might ascend. The first LSTM network feeds in the input sequence as per usual. The second LSTM network reverses the input sequence and feeds that into the LSTM network. After that, the merger of these two networks served as input to the next layer. Of the two embedding options, including 1) train from scratch the Embedding Layer and 2) Employee some weight embedding pre-trained open source, we preferred the Glove word embedding from Stanford NLP Group. To apply the GloVe embeddings, we first converted email message text to sequences. Keras then defined a vocabulary where each word had an inimitable index. We padded shorter sentences to the max length (most extended length rendered after pre-processing). After that, LSTM served to get the context representations. We further extended the testing phase by processing the emails by including and excluding digits/figures from their main body, respectively. Table 2 gives the hardware specifications of the machine used for computation.

Table 2. Hardware specification.

Parameters	description
Processor	Intel(R) Core(TM) i7-4500U CPU,1.80GHz
Video Card	Intel(R) HD Graphics Family
RAM	16.0 GB
Operating System	64-bit Operating System (OS) Windows 10

5. Experimental results and discussion

As stated earlier, the same dataset is processed twice: first using the natural language text processing techniques and then by excluding those. The intent is to compare standard ML and DL for the two scenarios and compare the better amongst the two approaches. In the first scenario, we apply ML and DL algorithms with preprocessing techniques, and on the other hand, for the second scenario, we apply ML and DL algorithms without pre-processing techniques. Next, which is more sensitive, is to ascertain the impact of text processing techniques on detecting veiled communication for standard and DL algorithms. First, we split the data into two windows; training and testing, using 80% of the data for training and 20% for testing. secondly, we further subdivided the training data into training and validation. 20% of the training data is used for the validation.

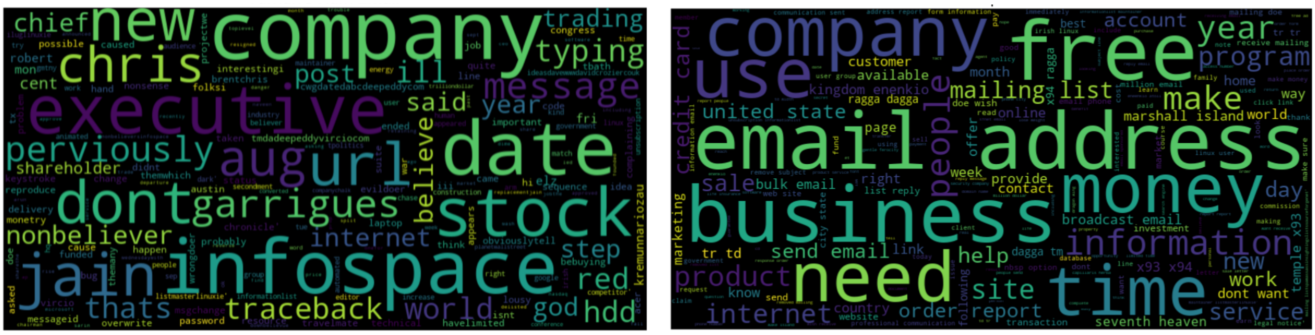
5.1. ***Test Case-I:***

Spam filtering test results with pre-processing email messages using natural language text processing techniques.

1) **Scenario 1a:** This section presents the results for standard Gaussian Naïve Bayes ML algorithm. Word-cloud is the tool that comes in handy for text visualization, as is evident from Figure 3a,b for ham and spam emails, respectively.

   The accuracy, precision, recall, and F1-score are computed using Gaussian Naïve Bayes are shown in Table 3. The Naive Bayes model achieved the best accuracy of 88.02%, with 89.50% precision score, 78.09% recall score, and 83.01% F1-score.

   Figure 4a shows the confusion matrix without normalization whereas Figure 4b illustrates normalized confusion matrix. The diagonal values of confusion metrics represent those truly classified by



(a) Visualizing HAM email

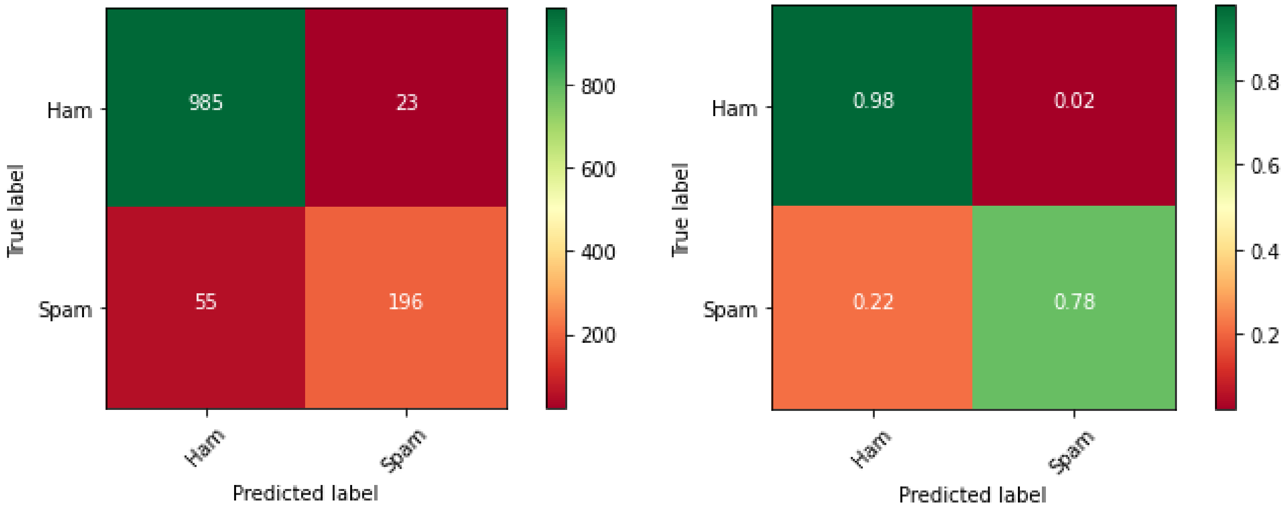
(b) Visualizing SPAM email

Figure 3. Comparison of SPAM and HAM emails.

Table 3. All experimental results with text pre-processing.

Naïve Bayes		LSTM	
Accuracy	88.02%	Accuracy	94.05%
Precision	89.50%	Precision	93.46%
Recall	78.09%	Recall	96.81%
F1-score	83.01%	F1-score	95.03%

the model, while the non-diagonal values are miss-classified by the naive Bayes model.



(a) Confusion Matrix-Non-Normalized

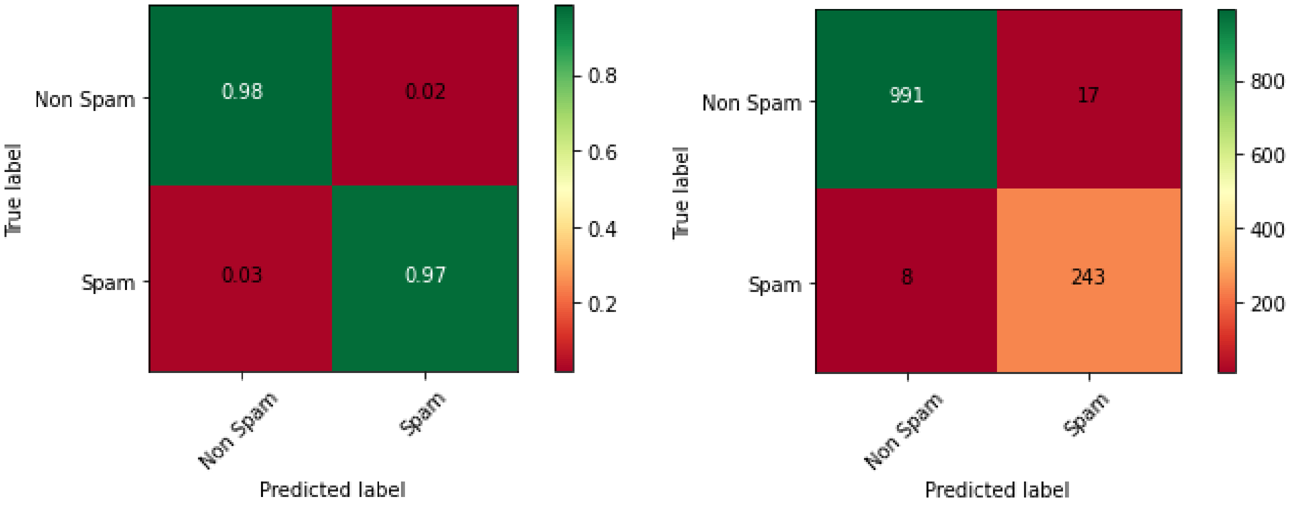
(b) Confusion Matrix-Normalized

Figure 4. Confusion Matrix of Naive Bayes model with text pre-processing.

2) **Scenario 1b:** In this section, we presented the results for LSTM algorithm as can be seen in Table 3 and the confusion matrix of the LSTM model is presented in Figure 5a–c. Figure 6a,b shows the accuracy graph and loss over train and test dataset when operated using the LSTM algorithm. The experimental results of the LSTM model with text pre-processing techniques are higher than the

Naive Bayes model, as shown in Table 3. The LSTM model outperforms the naive Bayes model in accuracy, precision, recall, and F1-score value.

The Figure 5a shows the confusion matrix with normalization whereas Figure 5b illustrates non-normalized confusion matrix. 991 values are classified as Non-spam, and 243 belong to the spam category. However, 17 and 8 non-diagonal values are miss-classify by the LSTM model.

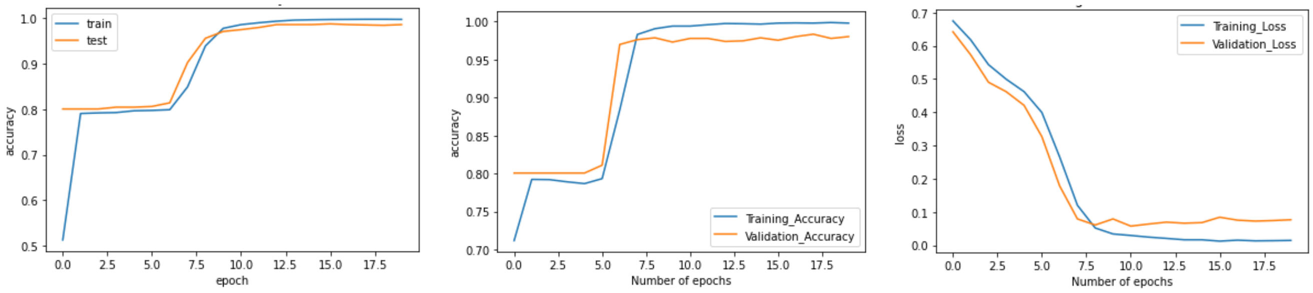


(a) Confusion Matrix-Normalized

(b) Confusion Matrix-Non-Normalized

Figure 5. Confusion Matrix of LSTM model with text pre-processing.

The loss function takes the predicted and real output from the training set. The loss is calculated during training and validation, and its meaning is determined by how well the model performs in these two sets. The loss graph is illustrated in Figure 6c. In the end, the Figure 6a depicts the model accuracy, which shows that the LSTM model achieved better testing and training accuracy on this data while Figure 6b shows training and validation accuracy graph, where the LSTM model training score is 97.15%, and the validation score is 96.23%.



(a) LSTM Model Accuracy

(b) Training and Validation Accuracy

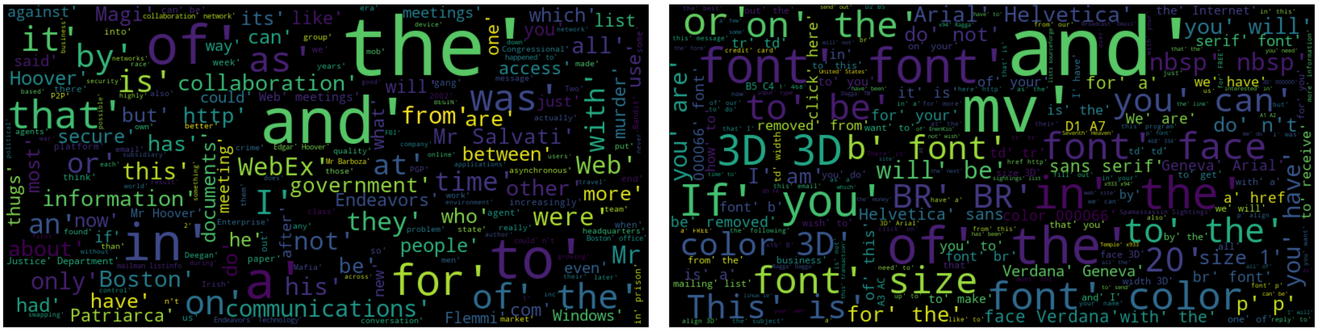
(c) Loss Graph

Figure 6. Comparison of LSTM model results with pre-processing.

5.2. *Test Case-II:*

Spam filtering test results without pre-processing email messages using Natural Language Text Processing Techniques.

1) **Scenario 1a:** In scenario 1a, results for Standard Gaussian Naïve Bayes ML algorithm are presented. First of all, we generate a word cloud of HAM and SPAM emails. A word cloud is a tool, which is useful for text visualization, as is evident from Figure 7a,b for ham and spam emails, respectively.



(a) Visualizing HAM email

(b) Visualizing SPAM email

Figure 7. Word cloud visualization of SPAM and HAM emails.

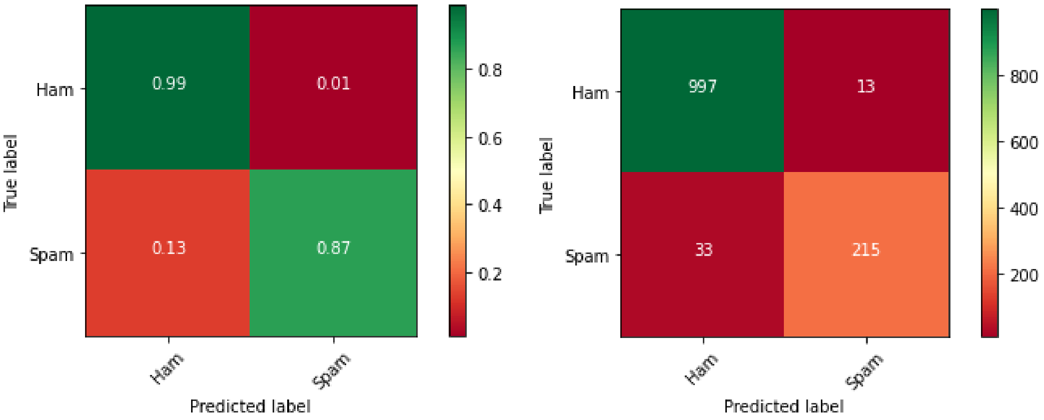
In the next step, we evaluate the model performance. The accuracy precision, recall, and F1-score values computed using Gaussian Naïve Bayes are shown in Table 4. The experimental results of the Naive Bayes algorithm without pre-processing data exhibit the best accuracy score of 93.07% along with 94.30, 86.69, 90.02% precision, recall, and F1-score values.

Table 4. All experimental results without text pre-processing.

Naïve Bayes		LSTM	
Accuracy	93.07%	Accuracy	97.18%
Precision	94.30%	Precision	95.26%
Recall	86.69%	Recall	97.18%
F1-score	90.02%	F1-score	96.01%

In the end, we plot a confusion matrix to find the true and miss-classify values. Figure 8a shows the confusion matrix with normalization whereas Figure 8b illustrates non-normalized confusion matrix.

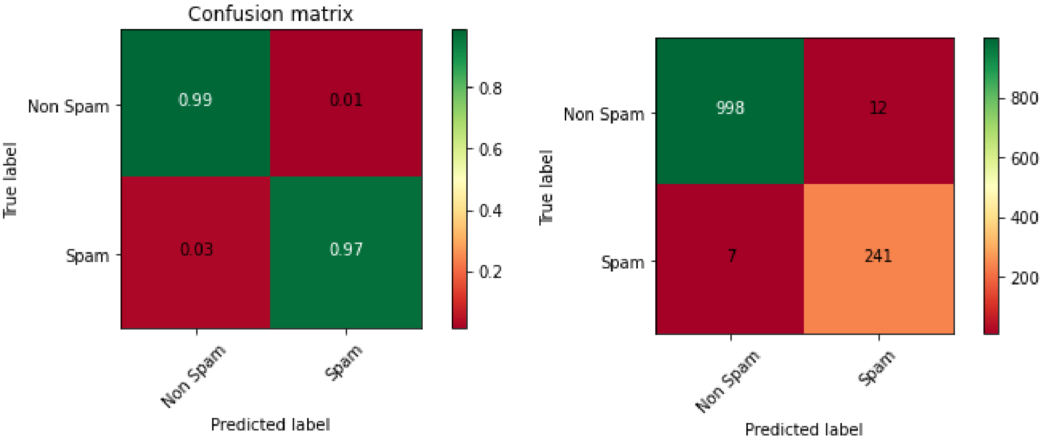
2) **Scenario 1b:** This section shows the results for LSTM DL algorithm. The following shows the performance metrics in Table 4, confusion matrix, the accuracy graph, and loss over train and test dataset when operated using the LSTM algorithm. Without pre-processing techniques, the lstm model achieves accuracy, precision, recall, and F1-score values of 97.18, 95.26, 97.18, and 96.01%, respectively.



(a) Confusion Matrix-Non-Normalized **(b)** Confusion Matrix-Normalized

Figure 8. Confusion Matrix of Naive Bayes model without text pre-processing.

Figure 9a shows the confusion matrix with normalization whereas Figure 9b illustrates non-normalized confusion matrix. Figure 10a depicts the model accuracy while Figure 10b shows training and validation accuracy graph. The loss graph is illustrated in Figure 10c.

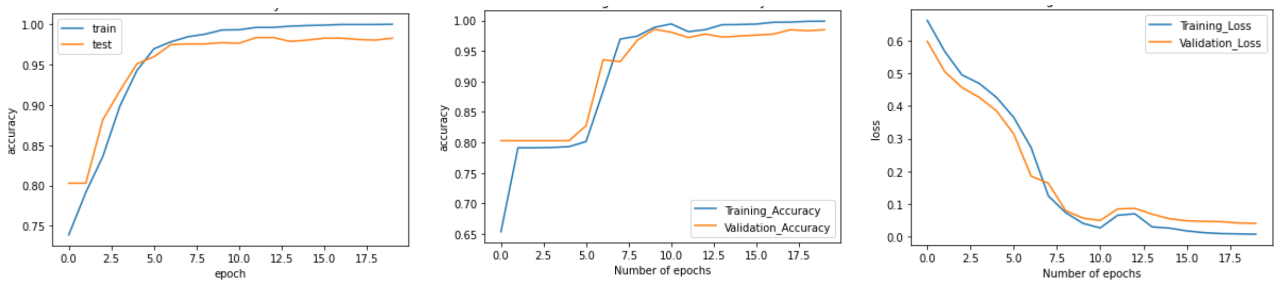


(a) Confusion Matrix-Normalized **(b)** Confusion Matrix-non-Normalized

Figure 9. Confusion Matrix of LSTM model without text pre-processing.

Confusion Matrix [43] determines the performance of a classifier with information about original and predicted taxonomies. At times accuracy may not correctly expand on the performance of data-logy models when evaluated. Suppose what if we have an imbalanced class? Then relying wholly on accuracy might not give us enough assurance on an algorithm's performance. For example, spam constitutes roughly 20% of total data addressing our email spamming issue. A prediction of all the emails as non-spam shall yield an 80% accuracy. Whereas for only 1% of Ham, 99% accuracy in predicting all the samples as unfavorable is evident. However, this kind of prediction model is impractical in real scenarios for which precision and recall come in handy to evaluate the class-imbalanced classification model.

The low 'recall' values above indicate that the Model might not be performing well in determining



(a) Model Accuracy

(b) Training and Validation Accuracy

(c) Loss Graph

Figure 10. Comparison of LSTM model results without pre-processing.

the spam email or feature extraction worked well in the case of non-pre-processed emails. That is, Gaussian Naïve Bayes operated well on a non-preprocessed dataset. A low F1 (measure of Model's accuracy) value in itself speaks that the algorithm does not perform well on pre-processed data.

The LSTM (DL Model) not only performed better on non-pre-processed dataset but outperformed Gaussian (Standard ML Model) from a ML perspective, which is contrary to [44,45,48] in terms of pre-processing. Vector representation of words through Glove has contributed well to the other classical email representation methods. It endows Glove embedding with DL methods in spam email filtering in the actual world situation. Higher F1 values affix the credibility of DL over the standard ML algorithm. One of the main snags in any ML approach is making the model generalized to predict probable results with the new dataset accurately. Visualizing the training accuracy vs. validation accuracy or training loss vs. validation loss over several epochs seems ideal for determining if the model has sufficient training. That is, the model must not be under and over train because memorizing the training data, in turn, shall lessen its capacity to perform an accurate prediction.

Hyperparameters such as the number of nodes per layer and the number of layers in the Neural Network can significantly affect the Model's performance. Envisaging the appropriateness of the training and validation dataset helps augment these values in building an improved model. The accuracy graph is increasing over epochs. Compared to the validation set, the model is doing well for the training set. In case of overfitting, it would have been otherwise. The loss over the number of epochs for training and validation datasets (Figures 6c and 10c) shows that the loss declines as the number of epochs upsurges. The validation loss is higher than training loss, and the difference is more noticeable as the number of epochs increases. Table 5 illustrates the Execution time taken concerning scenario Test Cases I and II, respectively.

Table 5. Execution time.

	Test Case-I	Test Case-II
Execution Time	0:16:58	0:14:05

This is evident because of the text pre-processing steps. However, the fact remains that the exclusion of pre-processing steps positively impacted the spam filtering process. The LSTM models do far better than the Gaussian Naïve Bayes Algorithm because the TFIDF vectorizer is ignorant of ordering the words in the sentence and thus loses essential information. It is also amongst the better algorithms in sequential data processing such as speech, text, and time-series data of the recent time.

It expounds that research focusing on text pre-processing and standard ML algorithms does not contribute satisfactorily to spam filtering.

In continuation of our assumption that words ignored during text pre-processing constitute our stealth message, it is apparent that such messages pass undetected for Ham messages regardless of using ML or DL algorithms. Hence, further research to be DL-focused. Strictly speaking, in the context of covert communication, none of the ML and DL algorithms contributes towards blocking veiled messaging because these algorithms may only prevent stealth messages that get excluded in the purview of being spam. Hence, it is a priori to refine and extend the spam filtering algorithms by additional pre-processing of email messages in stealth communication such as counting and placing excluded words, detecting patterns amongst those, and collectively as a message similar to other processing. Following that shall be a ranking or weight assignment to each observation. Finally, based on aggregated value of the orders, a decision be made for those emails as Ham or spam. That is, an approach similar to the ensemble learning technique.

Lastly, we have not come across any recent research addressing the proposed concern and scenario to the best of our knowledge. Hence, a cross-comparison of our results with similar results is impossible. The fact also endorses the novelty of our research contribution.

6. Conclusions and future works

Detecting spam at a place closer to the email dispatching server contributes much to network security, and ML techniques can drastically reduce the detection time. However, standard ML algorithms fail to contribute to detecting veiled communication. This paper reviewed some ML and DL techniques to detect and classify spam and examine algorithms for detecting stealth communication. Empirical evaluation based on accuracy, precision, recall, and F1 metrics of the two algorithms, Gaussian Naïve Bayes and Long-Short-Term-Memory, were also given. The assessment was based on e-mail messages collected from the public domain Spamassassin website. All the learning classifiers showed aptness to learning, but LSTM based DL algorithm outperformed the Gaussian Naïve Bayes with a maximum accuracy of 97.18%. The concept presented and the proposed spam filtering solution have made this research a novel contribution to cyber security. The dynamism that spam structuring offers and the counter-strategy of spammers to combat have made spam filtering an active research area, leaving a broader scope for exploration and evolution of new spam filters encompassing the present-day cyber security requirements.

Recent research focuses on detecting fake usernames, malicious email addresses, and content filtering [46, 49]. However, in general, text pre-processing remains the foremost step in the NLP pipeline [47, 48], potentially impacting its final performance primarily when used in conjunction with machine learning and neural networks algorithms for spam filtering. There are some suggestions for spam filtering efficacy; 1) the rapid infiltration of malicious network trafficking calls for examining the email contents at a broader level rather than sticking to the traditional spam detection approaches of refining the textual contents meriting lingual context alone, 2) the spam spread can also be minimized by deploying spam filters near main servers to restrict their further traversal across the network 3) gradually drifting and extending the scope of email spam filtering from textual context to a more generalized and broader concept covering detection of obscure messaging.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. C. M. Habito, A. Morgan, C. Vaughan, 'direct'and 'instant': the role of digital technology and social media in young filipinos' intimate relationships, *Cult., Health & Sexual.*, 1–19. doi: 10.1080/13691058.2021.1877825.
2. M. U. Khan, A. R. Javed, M. Ihsan, U. Tariq, A novel category detection of social media reviews in the restaurant industry, *Multimedia Syst.*, 1–14. doi: 10.1007/s00530-020-00704-2.
3. M. Hina, M. Ali, A. R. Javed, F. Ghabban, L. A. Khan, Z. Jalil, Sefaced: Semantic-based forensic analysis and classification of e-mail data using deep learning, *IEEE Access*, **9** (2021), 98398–98411. doi: 10.1109/ACCESS.2021.3095730.
4. R. Kong, H. Zhu, J. A. Konstan, Learning to ignore: A case study of organization-wide bulk email effectiveness, in *Proceedings of the ACM on Human-Computer Interaction*, **5** (2021), 1–23. doi: 10.1145/3479861.
5. E. Kiselev, Trends and features of russian business email: Contrastive analysis based on materials from business communication textbooks, *Jpn. Sl. East Eur. Stud.*, **41** (2021), 18–41.
6. M. Hina, M. Ali, A. R. Javed, G. Srivastava, T. R. Gadekallu, Z. Jalil, Email classification and forensics analysis using ML, in *2021 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*, 2021, 630–635. doi: 10.1109/SWC50871.2021.00093.
7. W. Ahmed, A. Rasool, A. R. Javed, N. Kumar, T. R. Gadekallu, Z. Jalil, et al., Security in next generation mobile payment systems: A comprehensive survey, *IEEE Access*, **9** (2021), 115932–115950. doi: 10.1109/ACCESS.2021.3105450.
8. A. R. Javed, S. U. Rehman, M. U. Khan, M. Alazab, H. U. Khan, Betalogger: Smartphone sensor-based side-channel attack detection and text inference using language modeling and dense multilayer neural network, *Trans. Asian Low-Res. Lang. Inf. Process.*, **20** (2021), 1–17. doi: 10.1145/3460392.
9. A. R. Javed, M. O. Beg, M. Asim, T. Baker, A. H. Al-Bayatti, Alphalogger: Detecting motion-based side-channel attack using smartphone keystrokes, *J. Ambient Intell. Human. Comput.*, 1–14. doi: 10.1007/s12652-020-01770-0.
10. A. Basit, M. Zafar, A. R. Javed, Z. Jalil, A novel ensemble machine learning method to detect phishing attack, in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, IEEE, 2020, 1–5. doi: 10.1109/INMIC50486.2020.9318210.
11. A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, K. Kifayat, A comprehensive survey of ai-enabled phishing attacks detection techniques, *Telecommun. Syst.*, **76** (2021), 139–154. doi: 10.1007/s11235-020-00733-2.

12. S. ur Rehman, M. Khaliq, S. I. Imtiaz, A. Rasool, M. Shafiq, A. R. Javed, et al., Diddos: An approach for detection and identification of distributed denial of service (ddos) cyberattacks using gated recurrent units (gru), *Future Gener. Comput. Syst.*, **118** (2021), 453–466. doi: 10.1016/j.future.2021.01.022.

13. S. I. Imtiaz, S. ur Rehman, A. R. Javed, Z. Jalil, X. Liu, W. S. Alnumay, Deepamd: Detection and identification of android malware using high-efficient deep artificial neural network, *Future Gener. Comput. Syst.*, **115** (2021), 844–856. doi: 10.1016/j.future.2020.10.008.

14. T. Conley, J. Kalita, Language model metrics and procrustes analysis for improved vector transformation of nlp embeddings, preprint, arXiv:2106.02490.

15. L. Kumar, A secure communication with one-time pad encryption and steganography method in cloud, *Turk. J. Comput. Math. Educ. (TURCOMAT)*, **12** (2021), 2567–2576. doi: 10.1007/s00779-021-01607-3.

16. R. Abid, C. Iwendi, A. R. Javed, M. Rizwan, Z. Jalil, J. H. Anajemba, et al., An optimised homomorphic crt-rsa algorithm for secure and efficient communication, *Pers. Ubiquitous Comput.*, 1–14. doi: 10.1007/s00779-021-01607-3.

17. B. Ahuja, R. Doriya, Visual chaos steganography with fractional transform, in *Soft Computing and Signal Processing*, Springer, 2021, 295–304.

18. Q. Li, X. Wang, B. Ma, X. Wang, C. Wang, Z. Xia, Y. Shi, Image steganography based on style transfer and quaternion exponent moments, *Appl. Soft Comput.*, 107618. doi: 10.1016/j.asoc.2021.107618.

19. L. Serpa-Andrade, R. Garcia-Velez, E. Pinos-Velez, C. Flores-Urgilez, Analysis of the application of steganography applied in the field of cybersecurity, in *International Conference on Applied Human Factors and Ergonomics*, Springer, 2021, 366–371.

20. C. Iwendi, Z. Jalil, A. R. Javed, T. Reddy, R. Kaluri, G. Srivastava, et al., Keysplitwatermark: Zero watermarking algorithm for software protection against cyber-attacks, *IEEE Access*, **8** (2020), 72650–72660. doi: 10.1109/ACCESS.2020.2988160.

21. D. A. Putri, D. A. Kristiyanti, E. Indrayuni, A. Nurhadi and D. R. Hadinata, Comparison of naive bayes algorithm and support vector machine using pso feature selection for sentiment analysis on e-wallet review, in *Journal of Physics: Conference Series*, **1641** (2020), 012085. doi: 10.1088/1742-6596/1641/1/012085.

22. A. Mishra, J. A. Latorre, J. Pool, D. Stosic, D. Stosic, G. Venkatesh, et al., Accelerating sparse deep neural networks, preprint, arXiv:2104.08378.

23. M. Ramprasad, N. H. Chowdary, K. J. Reddy, V. Gaurav, Email spam detection using python & machine learning, *Turk. J. Phys. Rehabil.*, **32** (2019), 3.

24. M. Eriksson, G. Heuguet, Genealogies of online content identification-an introduction, *Int. Hist.*, **5** (2021), 1–7. doi: 10.1080/24701475.2021.1878649.

25. M. Neha, M. S. Nair, A novel twitter spam detection technique by integrating inception network with attention based lstm, in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, 2021, 1009–1014. doi: 10.1109/ICOEI51242.2021.9452825.

26. F. Iqbal, R. Batool, B. C. Fung, S. Aleem, A. Abbasi, A. R. Javed, Toward tweet-mining framework for extracting terrorist attack-related information and reporting, *IEEE Access*, **9** (2021), 115535–115547. doi: 10.1109/ACCESS.2021.3102040.

27. S. E. Rahman, S. Ullah, Email spam detection using bidirectional long short term memory with convolutional neural network, in *2020 IEEE Region 10 Symposium (TENSYMP)*, IEEE, 2020, 1307–1311. doi: 10.1109/TENSYMP50017.2020.9230769.

28. N. Garba, S. Rakshit, C. D. Maa, N. R. Vajjhala, An email content-based insider threat detection model using anomaly detection algorithms, in *Proceedings of the International Conference on Innovative Computing Communication (ICICC) 2021*, 2021. doi: 10.2139/ssrn.3833744.

29. T. Sharma, P. Ferronato, M. Bashir, Phishing email detection method: Leveraging data across different organizations, 2020.

30. S. Afzal, M. Asim, A. R. Javed, M. O. Beg, T. Baker, Urldeepdetect: A deep learning approach for detecting malicious urls using semantic vector models, *J. Network Syst. Manage.*, **29** (2021), 1–27. doi: 10.1007/s10922-021-09587-8.

31. R. Chiramdasu, G. Srivastava, S. Bhattacharya, P. K. Reddy, T. R. Gadekallu, Malicious url detection using logistic regression, in *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, IEEE, 2021, 1–6. doi: 10.1109/COINS51742.2021.9524269.

32. C. Rupa, G. Srivastava, S. Bhattacharya, P. Reddy, T. R. Gadekallu, A machine learning driven threat intelligence system for malicious url detection, in *The 16th International Conference on Availability, Reliability and Security*, 2021, 1–7. doi: 10.1145/3465481.3470029.

33. B. Aguirre, *Steganography in Contemporary Cyberattacks and the Link to Child Pornography*, PhD thesis, Utica College, 2020.

34. R. Singh, Analysis of spam email filtering through naive bayes algorithm across different datasets.

35. S. Srinivasan, V. Ravi, M. Alazab, S. Ketha, A. Z. Ala'M, S. K. Padannayil, Spam emails detection based on distributed word embedding with deep learning, in *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, Springer, 2021, 161–189. doi: 10.1002/9781119701859.ch6.

36. A. N. Soni, Spam-e-mail-detection-using-advanced-deep-convolution-neuralnetwork-algorithms, *J. Innovative Dev. Pharm. Tech. Sci.*, **2** (2019), 74–80. doi: 10.1007/s35146-018-0155-y.

37. J. Rastenis, S. Ramanauskaitė, I. Suzdalev, K. Tunaitytė, J. Janulevičius, A. Čenys, Multi-language spam/phishing classification by email body text: Toward automated security incident investigation, *Electronics*, **10** (2021), 668. doi: 10.3390/electronics10060668.

38. S. Manjula, M. Shivamurthaiah, Identification of languages from the text document using natural language processing system, *Turk. J. Comput. Math. Educ. (TURCOMAT)*, **12** (2021), 2465–2472.

39. M. Mukhanova, Text normalization and spelling correction in kazakh language.

40. A. M. Alhassan, W. M. N. W. Zainon, Review of feature selection, dimensionality reduction and classification for chronic disease diagnosis, *IEEE Access*. **9** (2021), 87310–87317. doi: 10.1109/ACCESS.2021.3088613.

41. M. Ontivero-Ortega, A. Lage-Castellanos, G. Valente, R. Goebel, M. Valdes-Sosa, Fast gaussian naïve bayes for searchlight classification analysis, *Neuroimage*, **163** (2017), 471–479. doi: 10.1016/j.neuroimage.2017.09.001.

42. A. R. Javed, Z. Jalil, W. Zehra, T. R. Gadekallu, D. Y. Suh, M. J. Piran, A comprehensive survey on digital video forensics: Taxonomy, challenges, and future directions, *Eng. Appl. Artif. Intell.*, **106** (2021), 104456. doi: 10.1016/j.engappai.2021.104456.

43. S. Visa, B. Ramsay, A. L. Ralescu, E. Van Der Knaap, Confusion matrix-based feature selection, *MAICS*, **710** (2011), 120–127. doi: 10.3917/trans.120.0127.

44. A. Mann, O. Höft, Categorization of swedish e-mails using supervised machine learning, 2021.

45. V. Karunakaran, V. Rajasekar, S. I. T. Joseph, Exploring a filter and wrapper feature selection techniques in machine learning, in *Computational Vision and Bio-Inspired Computing*, Springer, 2021, 497–506.

46. N. P. Wosah, T. Win, Phishing mitigation techniques: A literature survey, preprint, arXiv:2104.06989. doi: 10.5121/ijnsa.2021.13205.

47. A. El Kah, I. Zeroual, The effects of pre-processing techniques on arabic text classification, *Int. J.*, **10**.

48. T. Mehrotra, G. K. Rajput, M. Verma, B. Lakhani, N. Singh, Email spam filtering technique from various perspectives using machine learning algorithms, in *Data Driven Approach Towards Disruptive Technologies: Proceedings of MIDAS 2020*, Springer Singapore, 2021, 423–432. doi: 10.1007/978-981-15-9873-9-33.

49. S. P. Shyry, Y. B. Jinila, Detection and prevention of spam mail with semantics-based text classification of collaborative and content filtering, in *Journal of Physics: Conference Series*, **1770** (2021), 012031. doi: 10.1088/1742-6596/1770/1/012031.

AIMS Press