



Received June 13, 2021, accepted June 29, 2021, date of publication July 8, 2021, date of current version July 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095730

# SeFACED: Semantic-Based Forensic Analysis and Classification of E-Mail Data Using Deep Learning

MARYAM HINA<sup>1</sup>, MOHSIN ALI<sup>1</sup>,  
ABDUL REHMAN JAVED<sup>1</sup>, (Graduate Student Member, IEEE),  
FAHAD GHABBAN<sup>3</sup>, LIAQAT ALI KHAN<sup>2</sup>, AND ZUNERA JALIL<sup>1</sup>, (Member, IEEE)

<sup>1</sup>Department of Computer Science, Air University, Islamabad 442200, Pakistan

<sup>2</sup>Department of Cyber Security, Air University, Islamabad 442200, Pakistan

<sup>3</sup>Information System Department, College of Computer Science and Engineering, Taibah University, Medina 42353, Saudi Arabia


Corresponding author: Abdul Rehman Javed (abdulrehman.cs@au.edu.pk)

**ABSTRACT** Artificial Intelligence (AI), in combination with the Internet of Things (IoT), called (AIoT), an emerging trend in industrial applications, is capable of intelligent decision-making with self-driven analytics. With its extensive usage in diverse scenarios, IoT devices generate bulk data contrived by attackers to disrupt normal operations and services. Hence, there is a need for proactive data analysis to prevent cyber-attacks and crimes. To investigate crimes involving Electronic Mail (e-mail), analysis of both the header and the email body is required since the semantics of communication helps to identify the source of potential evidence. With the continued growth of data shared via emails, investigators now face the daunting challenge of extracting the required semantic information from the bulks of emails, thereby causing a delay in the investigation process. This gives an edge to the criminal in erasing their footprints of malicious acts. The existing keyword-based search techniques and filtration often result in extraneous, short sequence emails, which skips meaningful information. To overcome the above limitation, we propose a novel efficient approach named *SeFACED* that uses Long Short-Term Memory (LSTM) based Gated Recurrent Neural Network (GRU) for multiclass email classification. *SeFACED* not only works on short sequences but with long dependencies of 1000+ characters as well. *SeFACED* focuses on tuning LSTM based GRU parameters to attain the best performance and with assessment by comparing it with traditional machine learning, deep learning models, and state-of-the-art studies on the subject. Experimental results on self-extended benchmark datasets exhibit that *SeFACED* effectively outperforms existing methods while keeping the classification process robust and reliable.

**INDEX TERMS** Artificial intelligence, cybercrimes, multiclass e-mail classification, deep learning, cyber-security.

## I. INTRODUCTION

AIoT is capable of decision-making intelligently and self-driven analytic. IoT devices' data can be analyzed using AI to prevent cybercrime, improve security, and privacy, particularly in industrial applications, [1]–[5]. As the Internet was popularized in the early 90s of the last century, electronic mail became an essential communication source everywhere. E-mail storage has grown exponentially over the years, and a typical user stores half of his/her critical data in e-mail storage [6]–[8].

The associate editor coordinating the review of this manuscript and approving it for publication was Kok-Lim Alvin Yau .

E-mail is an essential application for carrying out transactions and efficiency in business processes to improve productivity. Many organizations share their necessary information utilizing E-mail like delivering a document, sharing messages, collaborations, essential updates, and notifications. According to a recent study, in 2019, global e-mail users amount to 3.9 billion, which is likely to grow up to 4.3 billion in 2023, half of the total world population, and 108.7 billion e-mails exchange every day.<sup>1</sup> E-mail is frequently used as a vital medium of communication and is also being used by cybercriminals to commit crimes [9]. Current and emerging threat agents are increasingly targeting

<sup>1</sup><https://blog.logix.in/Types-of-Email-Threats/>

complex, extensive data networks in modern organizations [10]. With the growing trend of cybercrime and accidents resulting from vulnerabilities, proactive monitoring and post-incident analysis of email data is crucial for organizations [11]. Cybercrimes like hacking, spoofing, phishing, E-mail bombing, whaling, and spamming are being performed through E-mails [7]. According to a study, E-mail is the second most used application on the Internet and the third most common form of cyberbullying.<sup>2</sup> Cybercriminals use it in numerous ways, like sending harassing and threatening messages, attaching viruses to E-mail, including a victim's private information, and sending it to hundreds of people. Spam messages accounted for 53.95% of e-mail traffic in March 2020.

In this study, we consider three different categories of illicit E-mails. The first one is fraudulent E-mails, which are intended for deceptive purposes to get crucial information. The second one is harassing E-mails, which are used in cyberbullying and are designed to threaten people. The third category is suspicious E-mails, which contain some text regarding unlawful activities. In the past, some researchers have contributed in this regard. As per the researcher's opinion, there is some evidence of the exchange of suspicious E-mails before the events of 9/11 took place [12].

To date, only one research piece has been done on private, text, and image-based E-mail classification, terrorist E-mail classification, and VIP E-mail classification [13]. There is only one dataset uploaded after these studies. Different techniques are still implemented on these publicly available datasets, and some researchers collected data on their own to implement different methodologies. In the literature, blocklisting systems, ML algorithms, and the use of forensic tools have all been listed as methods for E-mail detection. The blocklisting process focuses on identifying and documenting individuals, which takes a lot of manpower and time. ML algorithms also need manual feature engineering for the representation of features that are not very conducive. Forensic tools often lead to irrelevant E-mails as they use keyword search-based methods.

The existing email classification approaches lead towards irrelevant E-mails and/or loss of valuable information. Keeping in sight these limitations, this paper makes the following contributions:

- We design a novel efficient approach named *SeFACED* for E-mail classification into four different classes: Normal, Fraudulent, Threatening, and Suspicious E-mails by using LSTM based GRU that not only deals with short sequences as well long dependencies of 1000+ characters. *SeFACED* focuses on tuning LSTM based GRU parameters to attain the best performance.
- The LSTM based GRU efficiently captures meaningful information from E-mails that can be used for forensic analysis as evidence. E-mail content analysis helps spoof

TABLE 1. Acronyms used in *SeFACED*.

Acronym	Full Form
AI	Artificial Intelligence
IOT	Internet of Things
SMTP	Simple Mail Transfer Protocol
NLP	Natural Language Processing
ML	Machine Learning
DL	Deep Learning
SVM	Support Vector Machine
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
TF-IDF	Term frequency-document inverse frequency
ANN	Artificial Neural Network
LSTM	Long short term memory
RCNN	Recurrent Convolutional Network
RELU	Rectified Linear Unit
RF	Random Forest
LR	Logistic Regression
NB	Naïve Bayes
GRU	Gated Recurrent Unit
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ADAM	Adaptive Learning

identification since it is more efficient to analyze the headers of specific E-mails than all E-mails.

- Evaluate the performance of *SeFACED* compared to traditional ML as well as DL models and existing studies on E-mail content analysis and classification of E-mails.
- The results demonstrate that the *SeFACED* effectively classify E-mail content with the accuracy of 95.0%, the precision of 95.0%, recall of 95.1%, and f-score of 95.1% keeping the classification process robust and reliable.

This paper is organized as follows: Section II briefly describes the previous work related to this study. The datasets used for this study are discussed in Section III. Section IV detailed the proposed approach, while Section V presented the experiments' results and their analysis. Section VI gives some discussion surrounding our results, while the last section, namely Section VII, presents the conclusion and future work. TABLE 1 represents the abbreviations used in this study.

## II. LITERATURE REVIEW

Information security organizations have developed several computer forensic products. These products focus on essential functions like E-mail data collection, E-mail decoding, and simple relationship graph drawing. Existing work related to E-mail classification, spoofing, and phishing is divided into ML and DL. The following two subsections contain details about existing techniques applied to E-mail data in each category:

### A. MACHINE LEARNING TECHNIQUES

The authors in [7] presented an intelligent spam E-mail detection survey that concentrated on AI and ML approaches

<sup>2</sup><https://www.statista.com/statistics/420391/spam-email-traffic-share>

for spam detection. They investigated various methods and E-mail structures to analyze data for intelligent analysis like routing information gathered from the header, source and destination, content information, and attachments in the Simple Mail Transfer Protocol (SMTP) envelope. They addressed everything from hashing to DNS blocklisting and content-based methods such as regular content filtering and regular expressions filtering. They concluded that while machine learning algorithms are in high demand for improving cybersecurity, none of these are sufficient for dealing with multiple spam E-mails.

The authors in [11] presented two-phased anomaly detection models to boost the IIoT network's reliability. To predict class labels, they used Support Vector Machine (SVM) and the Naive Bayes (NB), Artificial Neural Network (ANN) with Random Forest (RF) ensemble technique, and to achieve accuracy, they fed the results into a classification unit. They experimented on standard IIoT attack datasets such as WUSTL\_IOT, N\_BaIoT, and Bot IoT. They concluded that the ensemble method outperformed these datasets.

The authors in [14] presented a content-based phishing detection approach. They used RF for the classification of Phishing E-mails. They classified phishing and ham E-mails. They extracted features and improved phishing E-mail classifiers with better prediction accuracy. The authors in [15] presented different efficient ML algorithms to filter spam. They measured 10 classes of E-mails and run the multilayer perceptron (MLP) algorithm on test data. They considered E-mails as spam by setting a threshold. The authors in [16] presented E-mail visualization correlation analysis by creating a visual Foxmail forensics system. This investigated by hash verification, document parsing, mail inquiry, mail relations visual demonstration, the message body, and attachment full-text retrieval. It showed a relationship by a histogram, but it needs improvement in supporting multiple document formats.

The authors in [17] performed a manual method of E-mail analysis. They spotted spoofed messages sent by SMTP, decoded these, analyzed IP addresses, traced their locations, and made a timeline of all the events. They also checked server logs to ensure the timetable's activities, but it was a long and tiring procedure and needed too many E-mails to analyze. The authors in [18] presented a spam classification framework using S-Cuckoo and a hybrid kernel-based SVM. The TF algorithm and images extracted textual features by the S-Cuckoo search algorithm and classified algorithms using a hybrid kernel-based SVM.

The authors in [19] presented a methodology and tool for discovery and information in large E-mail datasets relevant to the investigation. They tried to reduce unnecessary E-mails, performed a context-based visual search, and defined a visual analytical pipeline that supports user interactions with E-mail search results and filters and expands interactions. The authors in [20] performed E-mail forensics analysis on the header and considered storage format availability of backup and protocols used to transport E-mails.

They concluded it a slow process and recommended E-mail forensics tools eMailTrackerPro and Add4Mail, which help in E-mail investigation with limitations that software cannot find a spammer blocklisted in the database and find keywords with user search.

The authors in [21] presented E-mail classification as spam using the Fuzzy C-means algorithm. They implemented a membership threshold value to detect spam. The authors in [22] proposed a model to label unlabeled data and performed sentimental analysis on the Enron data set. They classified data into positive, negative, and neutral classes. They used the unsupervised ML approach k-means clustering to group data and applied supervised ML algorithms SVM and NB to classify them. The authors in [23] presented a method to classify an E-mail as fraudulent and ham. They used Fraudulent and Normal E-mail Dataset [16] for E-mail classification. They used ML techniques to classify E-mails. They improved accuracy and proposed ensemble techniques to improve classification accuracy and reduce misclassification errors. The authors in [24] implemented word embedding or vectorization and presented a neural network-based model for detecting and classifying a phishing E-mail. Their model has six components E-mails, E-mail Classifier, E-mail Parser, E-mail Sanitizer, E-mail Vectorizer, and Neural Network Model, and uses six features and ten-fold cross-validation for training, validation, and testing. They used two datasets for classification: SpamAssassin dataset and real Phish corpus.

## B. DEEP LEARNING TECHNIQUES

The authors in [16] presented E-mail Spam Filtering using a backpropagation neural network (BPNN) Classification Algorithm. They used a backpropagation multilayer feed-forward artificial neural network for the detection of spam and phishing E-mails. They used k-mean clustering in preprocessing and detected spam and phishing E-mails by ANN. The GRU is similar to LSTM, but it has fewer gates than LSTM, which improves the training process's speed. LSTM has three gates, while GRU has two gates, an update gate, and a rest gate. We use LSTM followed by the GRU layer with 60 memory cells and *tanh* activation, and a dense layer with a softmax activation function.

Spam detection in the mail, SMS texts, and reviews of the customer is a hot topic in literature [7]. Several algorithms exist in the state of the artwork. Spam detection with DL algorithms [25] such as CNN, RNN, GRU, and MLP are exciting to work on. The spam detection accuracy using modern DL techniques with *word2vec* word Embedding is relatively better than the traditional spam detection methods in the literature. The LSTM is better than the CNN and other machine learning algorithms because of long-term dependencies in the textual data. The LSTM has 3 gates to control the training process's information flow stated that LSTM is accurate. The yelp dataset is used to compare KNN, SVM, and NB algorithms with DL models [26].

A very effective way of handling the embedding vectors using GRU states is a plus for many Natural language

processing (NLP) problems such as text classification. The effectiveness of GRU is proved with the help of an experimental study conducted by the [27] researchers. They used TREC with 6 distinct classes and Google snippet with 8 distinct classes dataset for proving the strength of GRU with the NLP process for problem-solving. The authors compared GRU, LSTM, RNN, and MV-RNN(Multi-view RNN) and proved that GRU is much better than other models, and LSTM is second in performance. The performance of RNN is at the third number, and multi-view performance comes after RNN. Parameter tuning based on batch size, learning rate, padded sequence, embedding vector dimensions, and unique words in the datasets [28]. The authors proved with experiments that RNN is best for sequence learning, and deep neural networks are better for capturing the position invariant from the data.

GRU uses the point-wise multiplication function and logistic sigmoid activation to control the information flow. The GRU does not have separate memory cells/units for state control, and it has hidden states of storage memory. Weights, gates, and biases are essential variables that must be calculated during the GRU model development and represented by  $W$ ,  $U$ , and  $b$  vectors, respectively. The pre-trained word-embedding is used for training purposes named Glove vector. They stated explicitly that GRU is the better model when having extensive training data of textual categories with word embedding availability and considerable computation support such as GPU [2], [27], [29].

Many researchers have stated that CNN models as hierarchical representation learning models and RNN models as sequence learning models [30]–[32]. Here a substantial question of how we will choose a model for language processing? If we want to classify some tweets, E-mails, or text, we will use hierarchical models such as CNN. Moreover, if we have a sequence problem in textual data, such as sequence modeling, we will follow the RNN models' path. The application of the RNN is text summarization, text generation, and modeling of text. There is no consensus on the selection of DNN for the NLP tasks because, in many tasks, RNN performs better than the CNN for language processing [28].

The document-level and sentence-level representations in RNN also affect the performance of the RNN models. The document-level performance of RNN is better than the sentence-level representations. The LSTM performs best in sentence-level representations for abstractive and extractive summary generations in the Natural language processing field. The sentence vectors for understanding the sentence semantics help the LSTM capture the sequence of meaning. The document representation is used to encode the sentence relationships. The clarity of the dataset is also essential to capture the importance of the data. The noise in the dataset traps the model performance and leads to an overfitting problem. The preprocessing steps must be applied successfully on the dataset to reduce the vector density and to save the storage of the data points in the actual working memory [33].

### III. DATASET DESCRIPTION

The dataset used in this study is an amalgamation of four different datasets. The dataset contains Normal e-mails from Enron Corpora [34], Fraudulent e-mails provided by Phished e-mails corpora [35] which contain misleading information, Harassment messages selected from Hate Speech, Offensive dataset.<sup>3</sup> We enhance the dataset of Email Forensics by adding the suspicious emails data from our email sources, and twitter source. The suspicious dataset contains some terrorism-related messages collected from Twitter by API. These different datasets are merged into a structural file to make the multiclass E-mail classification possible. We extract features from the E-mail body using TF-IDF, Word2vector, and Word Embedding to classify them. TABLE 2 shows the composition of different E-mail corpus used for this study. All the header information such as sender, subject, CC, and BCC are removed; only the E-mail body's content is used for analysis. The dataset after composition contains about 32,427 messages. This newly created dataset is publicly available.<sup>4</sup>

TABLE 2. Composition of dataset.

	Normal E-mails	Fraudulent E-mails	Harassment E-mails	Suspicious E-mails
Number	9001	9001	9138	5287
Percentage	27.8%	27.8%	28.2%	16.3%

### IV. E-MAIL DETECTION AND CLASSIFICATION (SEFACED)

In this paper, E-mails are divided into normal, harassing, suspicious, and fraudulent classes. The architecture of current research work to classify E-mails in multiple classes is shown below in FIGURE 1. The proposed approach comprises data collection, pre-processing, feature extraction, parameter tuning, and classification using the LSTM-GRU model. The E-mail is divided into word levels of the E-mail body, and the embedding layer is applied to train and obtain the sequence of vectors. We input a part of the training validation set into the model. Finally, a testing set is used to verify the model's performance. We use Python language in the Google Colab environment for implementation and experiments.

E-mail data detection and classification. Algorithm 1 presents the steps of the proposed approach for E-mail classification. LSTM comprises the classification of E-mails as Normal, Harassing, Suspicious, and Fraudulent. The LSTM and GRU are both based on the gated network architecture, due to which we combined the GRU and LSTM to utilize the gated architecture of both of them. The gated network helps in tracking the long-term dependencies that exist in the textual data. This research aims to detect any harmful or unfavorable e-mails received at the e-mail server end based on the deep learning-based architecture.

<sup>3</sup>[https://www.kaggle.com/mrmorj/hate-speech-and-offensive-language-dataset?select=labeled\\_data.csv](https://www.kaggle.com/mrmorj/hate-speech-and-offensive-language-dataset?select=labeled_data.csv)

<sup>4</sup><https://github.com/Abdul-Rehman-J/SeFACED>



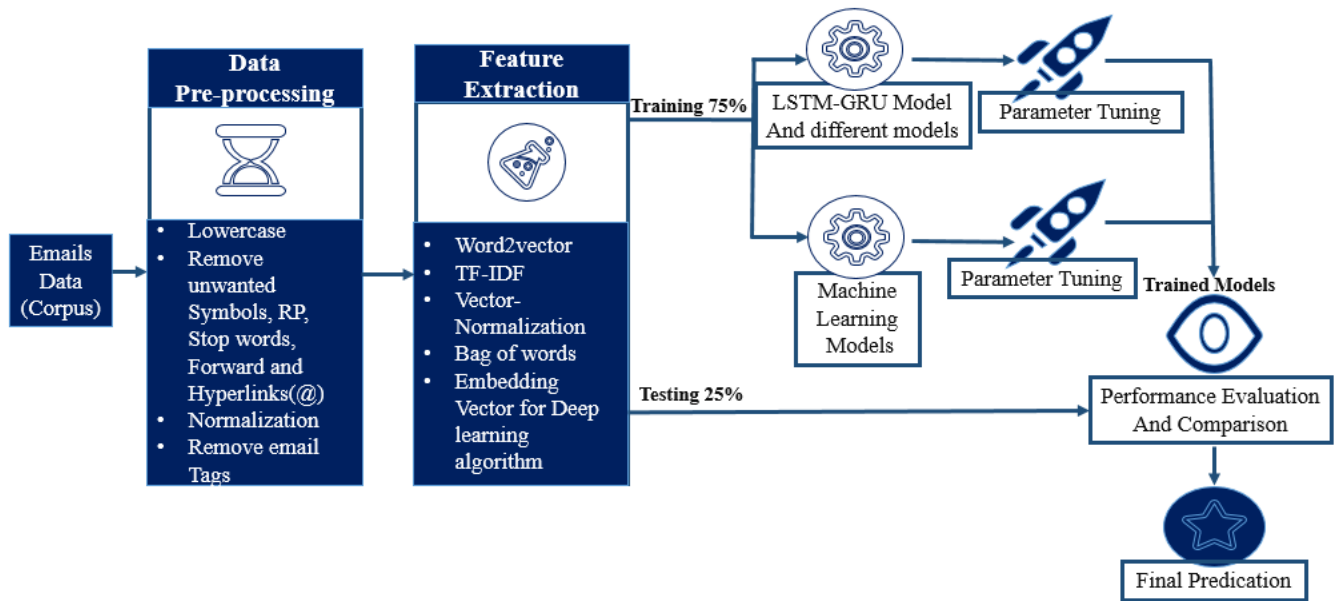


FIGURE 1. Proposed model for E-mail detection and classification.

**Algorithm 1** Algorithm: Multi-Class E-Mail Classification Using LSTM and GRU

```

1: INPUT: Data ← E-mail Messages
2: OUTPUT: Normal, Harassing, Suspicious, Fraudulent
3: For each E-mail message E {Data Preprocessing}
4: Undesired ← {Array of characters to remove}
5: Get document length (l)
6: for char ∈ undesired do
7:   Replace char with whitespaces till (l)
8: end for
9: Remove (tabs, punctuation, stopword, numbers, whitespaces from E)
10: V ← Embedding Layer(Data) {Vector Conversion}
11: LS ← LSTM(V) {LSTM}
12: GR ← GRU(V) {GRU component}
13: Training epochs M
14: for for (l,M) do
15:   Calculate the gradient of Weights W
16:   Backpropagate and Update W
17:   for epoch in range (20) do
18:     Evaluate Loss, Validation Loss
19:     Evaluate Accuracy
20:     Evaluate Precision, Recall, F-score and Confusion Matrix
21:   end for
22: end for
23: return Output
    
```

**A. DATA PREPROCESSING**

The data preprocessing phase consists of natural language-based steps that standardize the text and prepare it for analysis. It comprises different stages, as explained below.

1) TOKENIZATION

Breaking up the original text into component pieces is the tokenization step in natural language processing. There are predefined rules for tokenization of the documents into words. The tokenization step is performed in Python by using the *SpaCy* library.

2) STOP WORDS REMOVAL

Words like “a” and “the” that appear so frequently are not relevant to the context of the E-mail and create noise in the text data. These words are called stop words, and they can be filtered from the text to be processed. We utilized the “NLTK” Python library to remove stop words from the text.

3) PUNCTUATION REMOVAL

Punctuation includes (e.g., full stop (.), comma (,), brackets) to separate sentences and clarify meaning. For punctuation removal, we utilize the “NLTK” library.

**B. FEATURE EXTRACTION**

After eliminating irrelevant information, the elaborated list of words is converted into numbers. The TF-IDF method is applied to accomplish this task. Term Frequency is several occurrences of a word in a document, and IDF is the ratio of a total number of documents and the number of documents containing the term. A popular and straightforward method of feature extraction with text data is called the bag-of-words model of text. A bag-of-words model, or BoW for short, is a way of extracting features from the text for use in modeling, such as machine learning algorithms. A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things (1) A vocabulary of known words, (2) A measure of the presence of known words.

We extract features on the basis of Equations 1,2,3,4,5, and 6. Here  $tf$  represents term frequency and  $df$  represents document frequency.

$$TFIDF = tf * \left(\frac{1}{df}\right) \quad (1)$$

$$TFIDF = tf * Inverse(df) \quad (2)$$

$$TFIDF(t, d, D) = TF(t, d).IDF(t, D) \quad (3)$$

$$TFIDF(t, d) = \log \frac{N}{|d \in D|t \in D|} \quad (4)$$

Feature extraction in DL with the context of words is also essential. The technique used for this purpose is `word2vec` neural network-based algorithm. Equation 5 given below shows how `word2vec` manages the word-context with the help of probability measures. The  $D$  represents the pair-wise illustration of a set of words, and  $(w, c)$  is the word-context pair drawn from the large set  $D$ .

$$P(D = 1 | w, c_{1:k}) = \frac{1}{1 + e^{-(w \cdot c_1 + w \cdot c_2 + \dots + w \cdot c_k)}} \quad (5)$$

The multi-word context is also a variant of `word2vec`, as shown in Equation 6. The variable-length context is also controlled by the given below mathematics.

$$P(D = 1 | w, c) = \frac{1}{1 + e^{-s(w,c)}} \quad (6)$$

### C. WORD EMBEDDING LAYER

Embedding is the representation of words into real numbers. Many machine learning and DL Algorithms cannot process data in raw form (text form) and can only process numerical values as input for learning. Word embedding organizes texts which are converted into numbers. It extracts relevant features from the textual data and structures them up in the form of real values. Word embedding uses a word mapping dictionary to convert the terms (words) to a real value vector. There are two main problems with machine learning feature engineering techniques, one problem is the sparse vectors for data representation, and the second issue is that; it does not take into account the meaning of words to some extent. In embedding vectors, similar words will be represented by the almost near real-valued numbers. For example, the terms love and affection will be near to each other in the embedding vector.

The embedding vector as a data structure in the DL algorithm is used to accomplish the learning. In the experimental setup, the word embedding layer contains information about the sequence length of E-mails. We consider the sequence length of the E-mail 600 characters each. The embedding dimensions used in *SeFACED* is 800. The vocabulary size is set to 70,000 at the start because we set this value after generating the unique tokens of our dataset. The embedding layer takes three arguments such as input dimensions, output dimensions, and input length. In our proposed study, the input dimensions are 800, vocabulary size is 70,000, and input length is 600. We need to be curious when setting the embedding layer dimensions because sometimes we skip the

essential features when dealing with the large size of textual input. The embedding layer output will be used for the LSTM and GRU layers in adjacent layers.

### D. MACHINE LEARNING MODELS

In this study, we use the following machine learning algorithms Logistic Regression (LR), Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), Naive Bayes (NB), and Random Forest (RF) to evaluate and compare the effectiveness of our proposed LSTM approach. We trained machine learning models for comparison purposes and to select the best model for E-mail forensic tools.

### E. HYPER-TUNED LSTM BASED GRU MODEL

The DL models' layered structure helps in learning without intervention in ML model implementation. Several libraries provide an in-depth learning implementation structure. We split the data into three training, validation, and testing sets with a 65 : 10 : 25 ratio. We extracted the features from textual data of E-mail using the word Embedding technique. We encode the target values using the one-hot encoding technique into 4-distinct classes. We pass all preprocessed data to the novel architecture of LSTM layers variants for the perfect classification of E-mails. We use the LSTM layers with different GRU and Conv1D layer variants to transform the input textual data into an efficient E-mail classification system.

Textual data needs special attention when feature extraction comes in the proposed methodology. Different feature extraction methods need to be implemented when solving the Natural language processing problem using DL. The main point is to convert the textual data into real-valued vectors. There is a unique name for the vector in natural language processing, "embedding vector". There are multiple ways to generate the embedding vector from the textual data, but famous methods are GLOVE and Word2Vec techniques. Embedding vector dimensions are essential to get all the features extracted from the data. Let us suppose if we have 8 samples of textual data. The data have two distinct classes. Each sample has a maximum of five tokens in it. The vocabulary size will be the unique words in 8 samples, and the vocabulary size needs to be higher than the available unique tokens in the dataset to avoid collisions with a hash function. In this case, the dimensions of the embedding vector will be  $4 \times 8$ . In the case of the classification problem of NLP, we need to encode the target values using the one-hot encoding method.

After getting the vectors from the words, the similarity between the words is measured using the similarity measure between the corresponding vectors using Equation 7.

$$\text{sim}_{\cos}(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2} = \frac{\sum_i u_{[i]} \cdot v_{[i]}}{\sqrt{\sum_i (u_{[i]})^2} \sqrt{\sum_i (v_{[i]})^2}} \quad (7)$$

There are many other ways to measure the similarity between the word vectors, one of them is Jaccard similarity, which defines Equation 8.

$$\text{sim}_{\text{Jaccard}}(u, v) = \frac{\sum_i \min(u_{[i]}, v_{[i]})}{\sum_i \max(u_{[i]}, v_{[i]})} \quad (8)$$

DL for NLP uses dense vector representation to reduce the memory requirement for large models. Dense vector categorical data encoding is also a famous method, but most literature is based on one-hot encoding techniques. After feature extraction, the language modeling phase comes up. The evaluation criteria for language modeling is the perplexity method based on the probability theory. CNN can also classify textual data combing with pooling and fully connected layers, but we need to flatten the input vectors after CNN. [24]. The example given below illustrates the concepts of 1D Convolution with max-pooling over a sample of corpora.

### 1) ORIGINAL SENTENCE

Semantic-Based Forensic Analysis and Classification of E-Mail Data

### 2) WINDOW SIZE 3

Semantic-Based Forensic Analysis and Classification of E-Mail Data.

A 1D convolution network with max-pooling is applied over the sentence ‘‘Semantic-Based Forensic Analysis and Classification of E-Mail Data.’’ Just a simple explanation is given to illustrate the working of the convolutional network for textual data. The window size is 3 to divide the original corpus into vectors. The embedding dimensions are not shown in this illustration. Sentence after window size 3 is passed through a convolution layer through a filter size  $6 \times 3$ . In the end, max pooling is applied to get the 3-dimensional pooled vector.

The gated architecture of neural networks such as RNN and LSTM performs better than the CNN for language processing. RNN variants such as bidirectional RNN and multilayer (stacked) RNN work better than traditional networks. The advanced version of RNN is LSTM and GRU, which use the gated architecture to enhance the sequence capabilities of the RNN model.

### 3) MATHEMATICAL CONCEPT OF LSTM STORAGE

Equations 9 have logical gates to control the LSTM model’s memory [36]. There are four memory cells(logical) in LSTM: input, output, gate, and forget represented by  $i$ ,  $o$ ,  $z$ , and  $f$  in below Equation 9. The previous state information is controlled by the dot operation of forget gate and memory component  $f \odot c_{j-1}$ . The dot operation between  $i$  and  $z$  ( $i \odot z$ ) decides how much of the proposed update to keep for the next use.  $h_j$  decides the output of the gate over the  $\tanh$  nonlinear function in Equation 9.

$$\begin{aligned} s_j &= R_{\text{LSTM}}(s_{j-1}, x_j) = [c_j; h_j] \\ c_j &= f \odot c_{j-1} + i \odot z \end{aligned}$$

$$\begin{aligned} h_j &= o \odot \tanh(c_j) \\ i &= \sigma(x_j W^{xi} + h_{j-1} W^{hi}) \\ f &= \sigma(x_j W^{xf} + h_{j-1} W^{hf}) \\ o &= \sigma(x_j W^{xo} + h_{j-1} W^{ho}) \\ z &= \tanh(x_j W^{xz} + h_{j-1} W^{hz}) \end{aligned} \quad (9)$$

The idea behind using the LSTM for E-mail classification is that each E-mail may have different lengths. So, the LSTM memory structure deals with large sequences with the help of memory cells (see Equation 10).

$$\begin{aligned} y_j &= O_{\text{LSTM}}(s_j) = h_j s_j \in \mathbb{R}^{2 \cdot d_h}, \quad x_i \in \mathbb{R}^{d_x}, c_j, h_j, i, f, o, z \\ &\in \mathbb{R}^{d_h}, \quad W^{x^o} \in \mathbb{R}^{d_x \times d_h}, \quad W^{h^o} \in \mathbb{R}^{d_h \times d_h} \end{aligned} \quad (10)$$

The vanishing gradient and exploding problem are solved through the gated architecture of LSTM and GRU networks. To manage more long sequences, a combination of LSMT and GRU is a good choice. GRU works well on non-textual datasets. The working environment of the GRU is explained through Equation 11.

$$\begin{aligned} s_j &= R_{\text{GRU}}(s_{j-1}, x_j) = (1 - z) \odot s_{j-1} + z \odot \tilde{s}_j \\ z &= \sigma(x_j W^{xz} + s_{j-1} W^{sz}) \\ r &= \sigma(x_j W^{xr} + s_{j-1} W^{sr}) \\ \tilde{s}_j &= \tanh(x_j W^{xs} + (r \odot s_{j-1}) W^{sg}) \\ y_j &= O_{\text{GRU}}(s_j) = s_j \end{aligned} \quad (11)$$

Gate ( $r$ ) is used to access the previous state used to  $s(j)$ , an updated state.  $s(j)$  is the GRU network’s output state equal to  $y(j)$ . the relation between the  $s$  and  $\tilde{s}$  is controlled via the  $z$  value [37], [38].

LSTM, RNN, and GRU are used to overcome the loss of the long and short-term dependencies within the document. In our case, we initialized the LSTM embedding layer with embedding dimension, maximum sequence length, and vocabulary size in three variants of the LSTM model. We reshaped the input vector to make it easy to use for the Conv1D layer. The LSTM layer returns the sequences of previous states. When the next state is not based on the gated architecture, we must set the LSTM layer ‘‘return sequences’’ equal to False. The number of learning parameters is essential. We set 200 units of LSTM layer and tried different variants of LSTM units. More significantly, the number of units chosen in the LSTM model will require more time to train the model.

The loss function is used to measure the model’s performance in the training phase. The weights will be updated accordingly in the next iteration by checking the output in the training phase, and loss score called the backpropagation technique. The weight update step is called the optimization step. The dense layer tensor helps in getting the probability of occurrence of a label concerning the text data. By the end of the training, we evaluated the model using the testing data’s

unknown samples. TABLE 3 shows the training setting of hyperparameters for the deep neural network LSTM.

**TABLE 3. Details of optimal hyper-parameters deep neural network LSTM.**

Parameter	Value
Initial Bias	0
Internal Layers	10
Dropout	Random-Initialization
Activation Function at all layers	Relu-Rectified Linear Unit
Activation Function at the output layer	Softmax
Batch Size	64 bytes
LSTM layers	2 layers With 250 hidden units
LSTM layers	With return sequences true
GRU layer	1 layer with 250 hidden units
GRU layer	With return sequences false
Learning Optimizer	Adam
Error Function	Categorical Cross-Entropy

Cross-validation is a sampling procedure used on partial data samples by splitting the corpus into a training set to train the model and a test set to evaluate it. In cross-validation, the corpus is randomly divided into subsamples. A single subsample is used to test the model, and the remaining are used as training data. In our case, we applied all cross-validation to reduce the biases of input data instances. We also used the random shuffling method to distribute the data classes equally in all the data subsets, such as training, testing, and validation.

## V. EVALUATION AND RESULTS

Several measurements are used for performance evaluation of classifiers like accuracy, precision, recall, and f-score. These measurements are computed by a confusion matrix, which is composed of four terms.

- True positive (TP): are the positive values correctly classified as positive.
- True Negative (TN): are the negative values correctly classified as negative.
- False Positive (FP): are the negative values incorrectly classified as positive.
- False Negative (FN): are the positive values incorrectly classified as negative.

For the performance evaluation of our proposed model, we use the following metrics.

### A. ACCURACY

Is the fraction of the total number of applications correctly classified. The Accuracy of a detection mechanism can be calculated using Equation 12.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

### B. PRECISION

Is the fraction of the predicted correctly classified applications to the total of all applications that are correctly real positive. It can be calculated using Equation 13.

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

### C. RECALL

The recall is a fraction of the predicted correctly classified applications to the total number of applications classified correctly or incorrectly. Recall can be calculated using Equation 14.

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

### D. F-SCORE

F-score is the harmonic mean of precision and recall. It symbolizes the capability of the model for making fine distinctions. f-score of a detection model can be computed using Equation 15.

$$F - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (15)$$

TABLE 4 shows the accuracy of ML models with different features setup. The normalization of L2 with N-gram features helped in getting the highest results in the form of accuracy. LR produced the best accuracy score among the other classifiers with a bit of parameter tuning technique. A grid search mechanism is used to tune the parameter. C and gamma are predicted based on the grid search algorithm. According to the results, the best value for c is 0.1 in LR parameter tuning. LR model with unigram, bigram, trigram, and normalizations produce almost the same highest results as other algorithms.

**TABLE 4. Comparison of tuned models with different N-gram features and TF-IDF values.**

N-gram with TF-IDF Norm	Accuracy%				
	LR	SVM	SGD	NB	RF
Unigram with L2 norm	0.9191	0.9001	0.8715	0.9045	0.9054
Unigram+bigram with L2 norm	0.9191	0.8990	0.8763	0.9050	0.9050
Unigram+bigram+trigram with L2 norm	0.9179	0.9000	0.8746	0.8950	0.9035

TABLE 5 shows the precision, recall, and f-score of the best machine-learning model, which is LR. The class-wise precision, recall, and f-score helped us get individual labels' intelligence during the prediction. Almost all the classes are precise and distinguished from the other classes in the dataset. Precision, recall, and f-score of the fraudulent class are high in comparison with the other classes.

**TABLE 5. Tuned Logistic Regression (LR) model scores on test data.**

Class	Precision%	Recall%	F-score%
Normal	0.87	0.99	0.93
Harassment	0.90	0.96	0.93
Fraudulent	0.98	0.95	0.96
Suspicious	0.96	0.68	0.80

TABLE 6 illustrate the parameters selected for the ML algorithm for better training. The parameter tuning increases the model's accuracy. We focus on obtaining better accuracy from this tuned parameter using the grid search technique. The value of C in the LR has a high impact on the accuracy of the model.



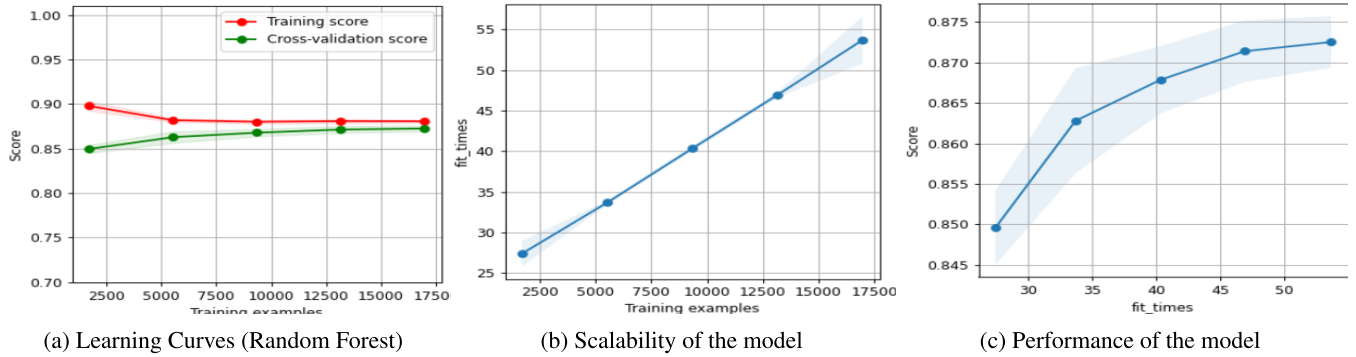


FIGURE 2. Learning curves random forest.

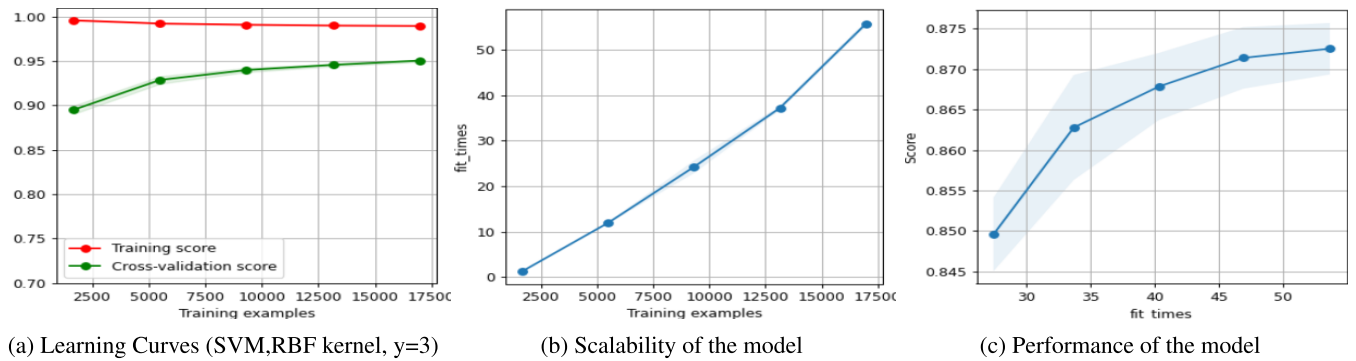


FIGURE 3. SVM learning curves.

TABLE 6. Parameter tuning of the machine learning models.

Model	Parameters
LR	$C=[0.1, 0.001, 1]$ , maxiter= 100, njobs=-1
SVM	$C=[0.1, 0.001, 1, 100, 1000]$
SGD	gamma=0, learning_rate=0.1
NB	alpha=[0.01, 0.001, 0.00000001]
RF	min_samples_leaf=[5,4], n_estimators=[100,150]

TABLE 7 shows each best model’s accuracy from ML and DL. The best accuracy is achieved with the LR model in ML algorithms, and in the case of DL, LSTM + GRU performs well and returns with an accuracy score of 95%. The precision, recall, and f-score of the LSTM + GRU model are well generalized to identify individual classes in the testing dataset. The weighted average helps us combine the precision, recall, and f-score into one value when computing the classification report.

TABLE 7. Multiclass classification performance of algorithms (ML and DL).

Model	Accuracy%	Precision%	Recall%	F1- Score%
LR	0.9191	0.96	0.68	0.80
SVM	0.9001	0.91	0.90	0.89
SGD	0.8763	0.89	0.85	0.86
NB	0.9045	0.91	0.90	0.90
RF	0.9054	0.92	0.91	0.90
LSTM+ Conv1D	0.9316	0.93	0.93	0.93
Stack of LSTM	0.9391	0.94	0.94	0.94
LSTM+ GRU	<b>0.9500</b>	0.95	0.95	0.95

The appropriate time is also an important parameter that needs to be considered when training the model. The exponential growth in a reasonable time is also a drawback of a complex learning classifier. FIGURES 2a, 2b, and 2c curves not just helped in getting the training score it also depicts the validation score for random forest classifiers. These curves depict how accuracy improves concerning training examples and how much time the model consumes to fit the training examples and fit times to achieve the accuracy/score. Similarly FIGURES 3a, 3b, and 3c depicts the learning curves for SVM model. Learning curves depicts the performance of the algorithm over experience and time. The training algorithm was applied several times on the dataset with the hold-out validation method and then plotted the learning curves based on each update. By looking at the curves, diagnosing the model performance should be easy; either the model is overfitted or underfitting. These curves are similar to the DL training and validation curves, but there is a difference in fold validation iterations in ML.

FIGURES 4a, 4b, and 4c depicts the logistic regression model’s learning pattern, scalability, and performance, respectively. These three curves are better than any other learning curves in this research, and we come to the point that the Logistic regression model learning style, scalability, and performance are better than SVM, RF, and SGD. Similarly, FIGURES 5a, 5b, and 5c illustrate the stochastic gradient’s learning process, scalability, and performance for the E-mail

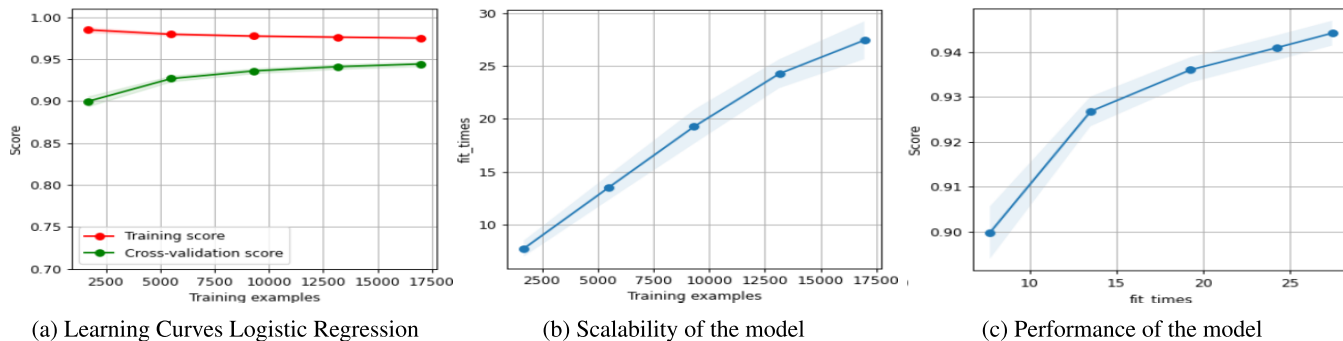


FIGURE 4. Logistic regression learning curves.

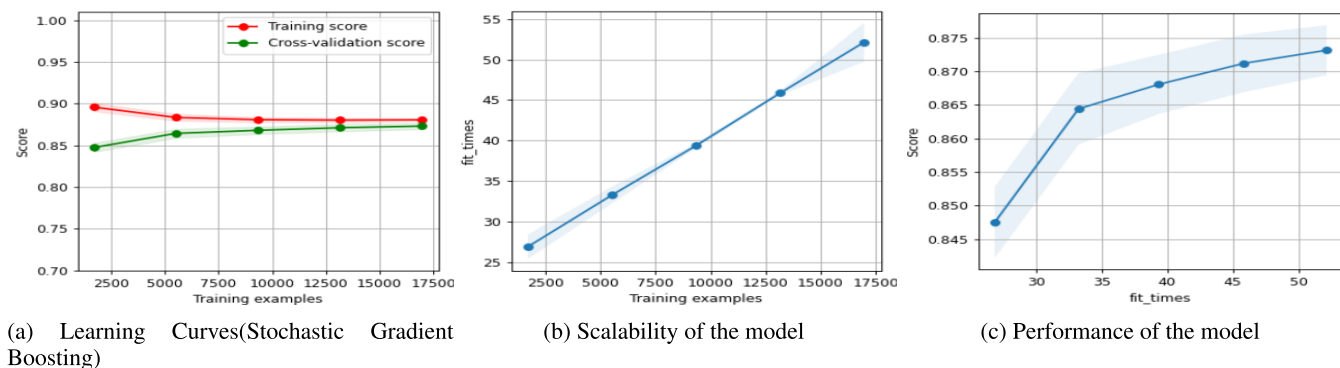


FIGURE 5. Stochastic gradient learning curves.

classification dataset used in this research. The performance given by the SGD is the lowest in our study because this model did not perform well for the long sequence of the textual data.

The proposed LSTM architecture variants comprised five layers; Embedding, LSTM, Conv1d, Dense layer, and an output layer with 4 output values probability using the softmax activation function. A few hidden layers are used to reduce time and space complexity for better and efficient results because this system environment will be used in real-time applications. The input layer contains the input vector dimensions and unique corpus in the dataset, and the embedding layer contains dimensions according to the unique features and input vector length. The LSTM layer dimensions are essential in long-term dependencies and used 600 real-time dimension vectors for the LSTM layer with a dropout rate of 0.2. The fully connected layer is accommodating after the LSTM layer operation to generalize the parameters of training. The Conv1D layer with 100 filters and kernel size 3. also tried many filters and kernel sizes to increase the accuracy. The input of each layer is the output of the last layer. The last dense layer input is (? , 64), and output is a vector with probability values for E-mail classification (? , 4). Our forensic E-mail module’s total trainable parameters are 50,080,804, and there are no non-trainable parameters in the model. The model performance, such as accuracy and robustness, is tested concerning the unseen dataset named the

testing dataset. The model performance is unbiased because we tested the model performance on the unseen testing data, which was not part of the training phase data. The maximum accuracy of 0.9316 is achieved by LSTM + Conv1D as shown in TABLE 7.

According to FIGURE 6a, 95% accuracy is obtained by LSTM + GRU DL algorithm. Epochs are set to 20, and early stopping criteria are set. The system terminates the training process after 6 iterations with 100% training accuracy without overfitting the model. FIGURE 6a shows that the training line converges smoothly after the 2nd iteration during training and validation, which is a positive sign of a generalized model. The LSTM + GRU model is based on a complex structure in the sense of several layers and hidden cells in each layer. We also apply parameter tuning, regularization to control the learning process, and features selection for better results. The complex LSTM + GRU model effectively learns the long sequenced emails content in two to three iterations. The validation loss and training loss are two main parameters that explain that either model learning is good or not. From the curves shown in FIGURE 6b, complex LSTM + GRU model with hyperparameter tuning and regularization model validation loss does not fluctuate more than one to two percent during all the epochs. The proposed model consists of a large number of trainable parameters, loss function, and optimizer, due to which model learning completes in just 3 to 4 iterations as shown in FIGURE 6b below with early stopping.

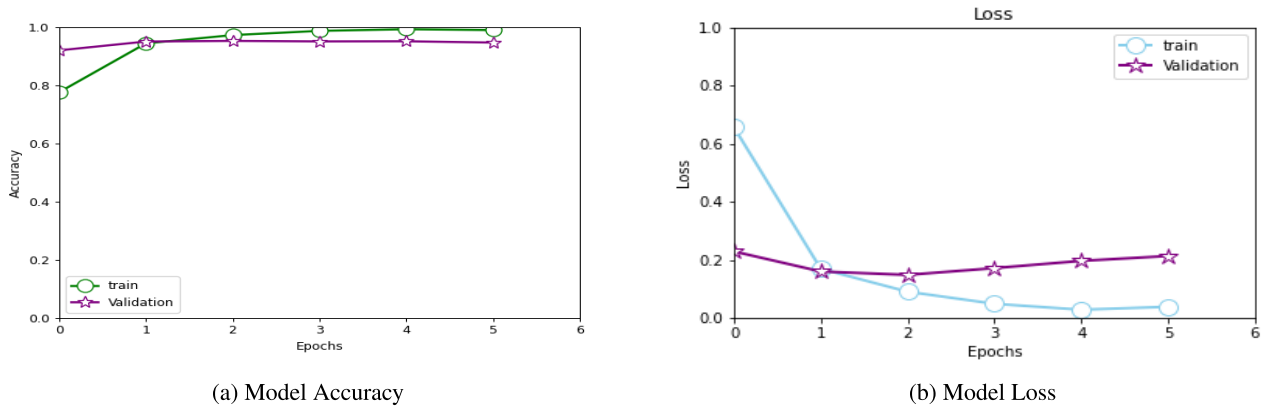


FIGURE 6. Model accuracy and loss of multiclass classification using train and validation datasets with the early stopping of LSTM + GRU model.

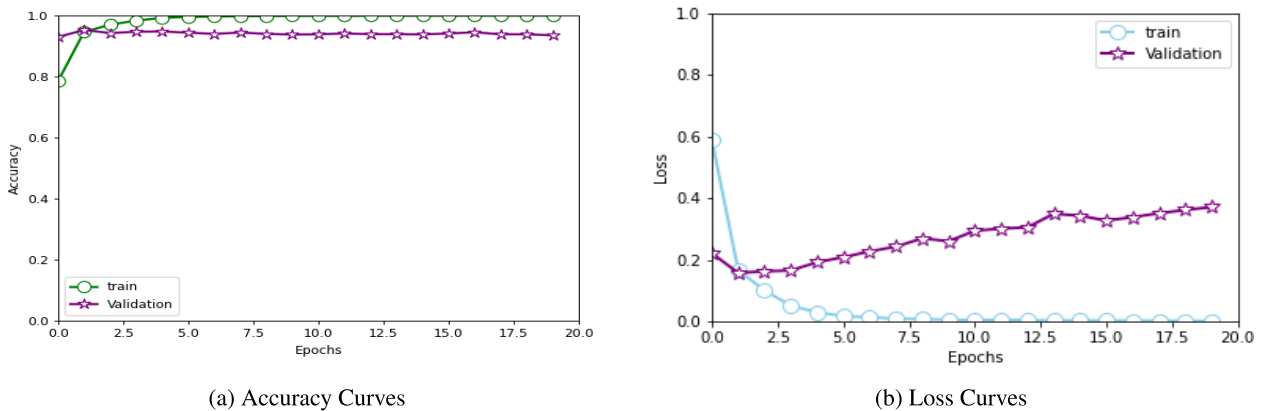


FIGURE 7. Model accuracy and loss of multiclass classification using train and validation datasets without early stopping LSTM + GRU model.

The training loss is almost zero, and we check if validation loss increases again, then the training is stopped. FIGURE 6b explains the parallel loss function visualization of the model training, an apparent reduction in loss after the 2nd iteration, and it goes straight in further training of the model. The impact of the learning is also visible on the validation dataset accuracy curve as shown in the FIGURE 6a. Validation loss and accuracy are aligned with each other during the model learning process. Without early stopping, all 20 epochs are shown in the diagram given below FIGURE 7a. Here x-axis depicts the total number of epochs in both diagrams. One finding is that there is no change in accuracy score without early stopping criteria, and the same score is achieved with 20 epochs as we got with early stopping criteria 95%. The confusion matrix is almost the same as we present above for the 6 epochs, but a slight change in validation loss is observed without early stopping criteria. FIGURE 7b shows the loss curve; the training loss is getting straight after 3 iterations suitable for a good fit model.

After LSTM + Conv1D, a stacked version of LSTM layers enhances the sequence learning capabilities of LSTM with a large number of parameters. We use 250 LSTM units for parameter setting. We notice an improvement in accuracy compared to the LSTM + conv1D model due to many sequence handlers. We use a stack of 3 LSTM layers with sequence return functionality to achieve better results.

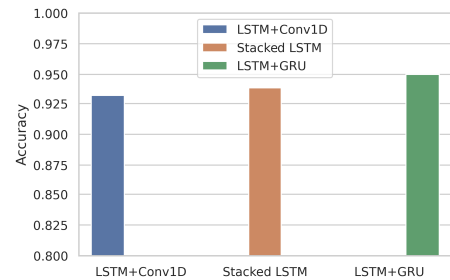
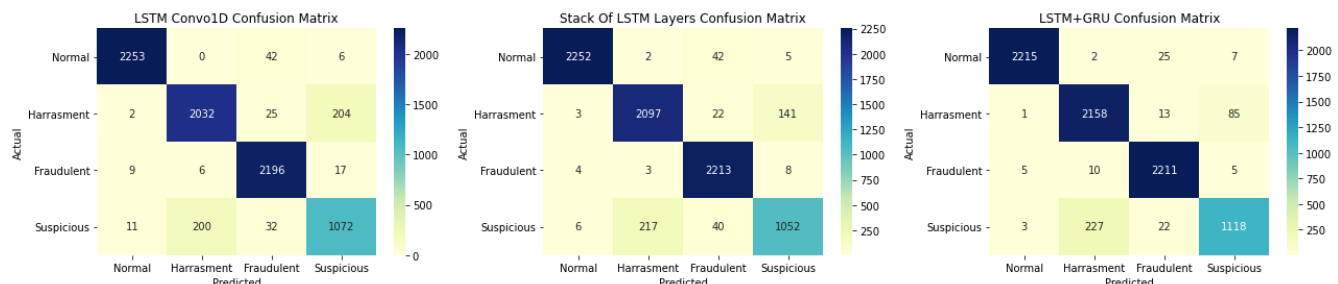


FIGURE 8. Deep learning based algorithm and accuracy comparison.

The results of LSTM + convo and Stacked LSTM improved by using the GRU with the LSTM network. The performance of the model increased by 1 unit. We achieve 95% accuracy, which is higher than other variants of the LSTM model in learning the long-sized E-mail classification. The learning curves of our proposed best model is shown in FIGURES 6a, 6b, 7a and 7b which depicts how well our model is trained and experience with unseen data and well generalized. The Good fit curves are the representation of the best-trained model. FIGURE 8 depicts how the LSTM + GRU novel combination improved the accuracy of the large sequence E-mails dataset forensic E-mail analysis. The maximum E-mail length is more than 1000 words, which needs many sequence learning modules; the LSTM and GRU are prevalent sequence learning algorithms. We proved



(a) LSTM+Conv1D based on Validation Accuracy (b) Stack of LSTM Network based on Validation Accuracy (c) LSTM+GRU based on Validation Accuracy

FIGURE 9. Confusion matrices based on early stopping.

their sequence learning with the E-mail dataset. The stacked LSTM model’s accuracy is at the second number in order, and convo1D + LSTM has minimum accuracy for the E-mail dataset.

The accuracy of the model is described through the predictions that individual DL models, such as Stacked LSTM, Convo1 + LSTM, and LSTM + GRU, in the form of confusion matrices shown in FIGURES 9a, 9b and 9c. The accuracy of the LSTM + GRU is high for the testing dataset in our experimental setup. There are a few misclassifications because of the large size E-mails dataset.

Data availability, fast algorithms, and hardware improvements are the main points for modern learning algorithms to classify e-mails into different categories. The comparison with machine learning is made to get the best-trained model because every mail transfer protocol required efficient and effective e-mail classification with high accuracy. So, our study also proved that modern deep learning is much better in performance than the traditional learning algorithms such as SVM and RF.

The critical consideration while deploying the model is to test the incoming data structure and compatibility with the model’s architecture. If the model architecture data requirement and incoming e-mail data formats are the same, then no modifications must be made for the model to classify E-mails. Otherwise, minor data dimensions and pre-processing will be required for the robust prediction.

VI. DISCUSSION

Existing studies on E-mail classification present various ML approaches to classify E-mails, and some of them focused on the sentiment analysis of E-mail datasets. A significant hindrance in E-mail classification is the non-availability of datasets. Limited E-mail datasets are available publicly, and researchers have to implement their approaches on these datasets or collect data independently. Secondly, data labeling is another limitation, and it is a time-consuming task. For data labeling, some rule-based techniques and tools (VADER) are used, but remarkable results are not obtained, and these techniques affect the performance of models. In this study,

we propose a DL model for multiclass E-mail Classification. We utilize the contents of three original E-mail datasets and gathered data from social media sources (Twitter) as data and vocabulary for criminal activities are the same. In pre-processing, we remove noise, duplicates, and missing values from the data. The first layer in the model is the embedding layer which is used for vectorization. Due to limited data, we perform oversampling techniques for our minority class. Sampling techniques can solve the data imbalance problem but affect the performance of the model. In oversampling, data is repeated randomly, affecting the test data as splitting data may be duplicated. In undersampling, some valuable data may be deleted. So, it is essential to upload datasets on criminal activities to make research on E-mail more effective. Then, LSTM + GRU is applied with different hyperparameters. Categorical cross-entropy is used as a loss function, and the ADAM optimizer optimized the value of weights. The results in TABLE 7 demonstrate that the SeFACED achieved the highest accuracy of 95% in comparison with existing studies.

VII. CONCLUSION AND FUTURE WORK

This paper proposed an LSTM model with an embedding layer for multiclass classification of electronic mails. We evaluated the proposed SeFACED model using evaluation metrics such as precision, recall, accuracy, and f-score. Experimental results revealed that SeFACED performed better than existing ML algorithms and achieved a classification accuracy of 95% using the novel technique of LSTM with recurrent gradient units. As different types of topics are discussed in E-mail content analysis. Many criminal activities are also performed through E-mails, but the E-mail repository is not available for public usage for privacy and security reasons. The non-availability of datasets on negative topics is a big hurdle in this research domain. Many researchers had just mentioned reports about criminal activities performed by E-mails, but they could not experiment due to a lack of datasets. For now, we are considering e-mail classes such as normal, harassment, fraudulent, and suspicious; however, many other classes can be added to this work in the presence



of the massive amount of e-mail data. We intend to produce datasets on these topics and build a generalized model for E-mail classification in the future.

## REFERENCES

- [1] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun. Syst.*, pp. 1–16, Oct. 2020.
- [2] C. Iwendi, Z. Jalil, A. R. Javed, T. Reddy, R. Kaluri, G. Srivastava, and O. Jo, "KeySplitWatermark: Zero watermarking algorithm for software protection against cyber-attacks," *IEEE Access*, vol. 8, pp. 72650–72660, 2020.
- [3] A. Rehman, S. U. Rehman, M. Khan, M. Alazab, and T. Reddy, "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Trans. Netw. Sci. Eng.*, early access, Feb. 19, 2021, doi: [10.1109/TNSE.2021.3059881](https://doi.org/10.1109/TNSE.2021.3059881).
- [4] S. U. Rehman, M. Khaliq, S. I. Imtiaz, A. Rasool, M. Shafiq, A. R. Javed, Z. Jalil, and A. K. Bashir, "DIDDOS: An approach for detection and identification of distributed denial of service (DDoS) cyberattacks using gated recurrent units (GRU)," *Future Gener. Comput. Syst.*, vol. 118, pp. 453–466, May 2021.
- [5] S. I. Imtiaz, S. U. Rehman, A. R. Javed, Z. Jalil, X. Liu, and W. S. Alnumay, "DeepAMD: Detection and identification of Android malware using high-efficient deep artificial neural network," *Future Gener. Comput. Syst.*, vol. 115, pp. 844–856, Feb. 2021.
- [6] Q. Li, M. Cheng, J. Wang, and B. Sun, "LSTM based phishing detection for big email data," *IEEE Trans. Big Data*, early access, Mar. 12, 2020, doi: [10.1109/TBDDATA.2020.2978915](https://doi.org/10.1109/TBDDATA.2020.2978915).
- [7] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," *IEEE Access*, vol. 7, pp. 168261–168295, 2019.
- [8] A. R. Javed, M. O. Beg, M. Asim, T. Baker, and A. H. Al-Bayatti, "AlphaLogger: Detecting motion-based side-channel attack using smartphone keystrokes," *J. Ambient Intell. Humanized Comput.*, pp. 1–14, Feb. 2020.
- [9] A. Yazdinejad, H. Haddadpajouh, A. Dehghantanha, R. M. Parizi, G. Srivastava, and M.-Y. Chen, "Cryptocurrency malware hunting: A deep recurrent neural network approach," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106630.
- [10] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An adaptive multi-layer botnet detection technique using machine learning classifiers," *Appl. Sci.*, vol. 9, no. 11, p. 2375, Jun. 2019.
- [11] V. Priya, I. S. Thaseen, T. R. Gadekallu, M. K. Aboudaif, and E. A. Nasr, "Robust attack detection approach for IIoT using ensemble classifier," 2021, *arXiv:2102.01515*. [Online]. Available: <http://arxiv.org/abs/2102.01515>
- [12] S. Nizamani, N. Memon, M. Glasdam, and D. D. Nguyen, "Detection of fraudulent emails by employing advanced feature abundance," *Egyptian Informat. J.*, vol. 15, no. 3, pp. 169–174, Nov. 2014.
- [13] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, "Email classification research trends: Review and open issues," *IEEE Access*, vol. 5, pp. 9044–9064, 2017.
- [14] A. S. Askari and N. K. Sourati, "Proposed efficient algorithm to filter spam using machine learning techniques," *Pacific Sci. Rev. A, Natural Sci. Eng.*, vol. 18, no. 2, pp. 145–149, Jul. 2016.
- [15] S. K. Tuteja and N. Bogiri, "Email spam filtering using BPNN classification algorithm," in *Proc. Int. Conf. Autom. Control Dyn. Optim. Techn. (ICACDOT)*, Sep. 2016, pp. 915–919.
- [16] Z. Chen, Y. Yang, L. Chen, L. Wen, J. Wang, G. Yang, and M. Guo, "Email visualization correlation analysis forensics research," in *Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Jun. 2017, pp. 339–343.
- [17] M. K. Chae, A. Alsadoon, P. W. C. Prasad, and A. Elchouemi, "Spam filtering email classification (SFECM) using gain and graph mining algorithm," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2017, pp. 217–222.
- [18] N. Moradpoor, B. Clavie, and B. Buchanan, "Employing machine learning techniques for detection and classification of phishing emails," in *Proc. Comput. Conf.*, Jul. 2017, pp. 149–156.
- [19] K. Agarwal and T. Kumar, "Email spam detection using integrated approach of naive Bayes and particle swarm optimization," in *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2018, pp. 685–690.
- [20] J. Koven, E. Bertini, L. Dubois, and N. Memon, "InVEST: Intelligent visual email search and triage," *Digit. Invest.*, vol. 18, pp. S138–S148, Aug. 2016.
- [21] A. K. Singh, S. Bhushan, and S. Vij, "Filtering spam messages and mails using fuzzy C means algorithm," in *Proc. 4th Int. Conf. Internet Things, Smart Innov. Usages (IoT-SIU)*, Apr. 2019, pp. 1–5.
- [22] R. S. H. Ali and N. E. Gayar, "Sentiment analysis using unlabeled email data," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Dec. 2019, pp. 328–333.
- [23] Kaggle. (2020). *Hillary Clinton Email Dataset*. Accessed: Dec. 16, 2020. [Online]. Available: <https://www.kaggle.com/general/16444>
- [24] E. Kiperwasser and Y. Goldberg, "Simple and accurate dependency parsing using bidirectional lstm feature representations," *Trans. Assoc. Comput. Linguistics*, vol. 4, no. 1, pp. 313–327, 2016.
- [25] T. R. Gadekallu, N. Khare, S. Bhattacharya, S. Singh, P. K. R. Maddikunta, and G. Srivastava, "Deep neural networks to predict diabetic retinopathy," *J. Ambient Intell. Humanized Comput.*, pp. 1–14, Apr. 2020.
- [26] G. M. Shahariar, S. Biswas, F. Omar, F. M. Shah, and S. B. Hassan, "Spam review detection using deep learning," in *Proc. IEEE 10th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Oct. 2019, pp. 0027–0033.
- [27] M. Zulqarnain, R. Ghazali, M. G. Ghouse, and M. F. Mushtaq, "Efficient processing of GRU based on word embedding for text classification," *Int. J. Informat. Vis.*, vol. 3, no. 4, pp. 377–383, Nov. 2019.
- [28] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, *arXiv:1702.01923*. [Online]. Available: <http://arxiv.org/abs/1702.01923>
- [29] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, T. R. Gadekallu, and G. Srivastava, "SP2F: A secured privacy-preserving framework for smart agricultural unmanned aerial vehicles," *Comput. Netw.*, vol. 187, Mar. 2021, Art. no. 107819.
- [30] S. M. H. Fard, H. Karimimpour, A. Dehghantanha, A. N. Jahromi, and G. Srivastava, "Ensemble sparse representation-based cyber threat hunting for security of smart cities," *Comput. Electr. Eng.*, vol. 88, Art. no. 106825, Dec. 2020.
- [31] L. Malina, G. Srivastava, P. Dzurenda, J. Hajny, and S. Ricci, "A privacy-enhancing framework for Internet of Things services," in *Proc. Int. Conf. Netw. Syst. Secur. Cham, Switzerland: Springer*, 2019, pp. 77–97.
- [32] G. Sharma, G. Srivastava, and V. Mago, "A framework for automatic categorization of social data into medical domains," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 1, pp. 129–140, Feb. 2020.
- [33] E. Kaiser and I. Sutskever, "Neural GPUs learn algorithms," 2015, *arXiv:1511.08228*. [Online]. Available: <http://arxiv.org/abs/1511.08228>
- [34] Kaggle. (2020). *The Enron Email Dataset*. Accessed: Dec. 16, 2020. [Online]. Available: <https://www.kaggle.com/wcukierski/enron-email-dataset>
- [35] D. Radev, "Clair collection of fraud email, ACL data and code repository," Univ. Michigan, Ann Arbor, MI, USA, Tech. Rep. ADCR2008T001, 2008.
- [36] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2342–2350.
- [37] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," 2016, *arXiv:1602.02410*. [Online]. Available: <http://arxiv.org/abs/1602.02410>
- [38] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," 2016, *arXiv:1610.10099*. [Online]. Available: <http://arxiv.org/abs/1610.10099>



**MARYAM HINA** is currently pursuing the master's degree with Air University, Islamabad, Pakistan. He is currently a Researcher with Air University. He aims to contribute to interdisciplinary research of computer science and human-related disciplines. His current research interests include cybersecurity, artificial intelligence, computer vision, network security, the IoT, smart city, and application development for smart living.



**MOHSIN ALI** is currently pursuing the master's degree with Air University, Islamabad, Pakistan. He is currently a Research Associate with the National Center for Cyber Security, Air University. He aims to contribute to interdisciplinary research of computer science and human-related disciplines. His current research interests include cybersecurity, artificial intelligence, computer vision, network security, the IoT, smart city, and application development for smart living.



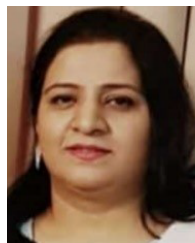
**ABDUL REHMAN JAVED** (Graduate Student Member, IEEE) received the master's degree in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan. He worked with the National Cybercrimes and Forensics Laboratory, Air University, Islamabad. He is currently a Lecturer with the Department of Cyber Security, Air University. He aims to contribute to interdisciplinary research of computer science and human-related disciplines. He has authored more than 20 peer-reviewed articles on cybersecurity, mobile computing, and digital forensics topics. His current research interests include mobile and ubiquitous computing, data analysis, knowledge discovery, data mining, natural language processing, smart homes, and their applications in human activity analysis, human motion analysis, and e-health.



**FAHAD GHABBAN** was born in Medina, Saudi Arabia, in 1982. He received the B.S. degree in computer science from Taibah University, Saudi Arabia, in 2005, the M.S. degree in information system from the University of Wollongong, Australia, in 2010, and the Ph.D. degree in information system from University Technology Malaysia, Malaysia, in 2019. From 2014 to 2018, he was a Lecturer with the College of Computer Science and Engineering, Taibah University. Since 2019, he has been an Assistant Professor with the Information System Department, College of Computer Science and Engineering, Taibah University. He is the author of ten articles. His research interests include key performance indicators, information communication technology, knowledge sharing, software used in research, e-learning, and e-governance. He is the Vice Dean of the College of Computer Science and Engineering for Graduate Studies and Scientific Research, Taibah University.



**LIAQAT ALI KHAN** received the B.E. degree in avionics engineering from the College of Aeronautical Engineering, NED University of Engineering and Technology, Karachi, in 1992, and the Ph.D. degree in information security from the College of Signals, National University of Sciences and Technology (NUST), Pakistan. He was an Associate Professor with the Institute of Avionics and Aeronautics (IAA), Air University, Islamabad, Pakistan, from 1995 to 2017. From 2017 to 2019, he was a Professor and the Head of the Avionics Engineering Department, College of Aeronautical Engineering, NUST. From July 2019 to July 2020, he was the Dean of the College of Aeronautical Engineering, NUST. He is currently serving as the Chair for the Department of Cyber Security, Air University. He has served on various appointments in other research and development organizations, Government of Pakistan.



**ZUNERA JALIL** (Member, IEEE) received the master's degree in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan. He worked with the National Cybercrimes and Forensics Laboratory, Air University, Islamabad. He is currently a Lecturer with the Department of Cyber Security, Air University. He aims to contribute to interdisciplinary research of computer science and human-related disciplines. He has authored more than 20 peer-reviewed articles on cybersecurity, mobile computing, and digital forensics topics. His current research interests include mobile and ubiquitous computing, data analysis, knowledge discovery, data mining, natural language processing, smart homes, and their applications in human activity analysis, human motion analysis, and e-health.

...