Semantic Analysis and Classification of Emails through Informative Selection of Features and Ensemble AI Model

Shivangi Sachan*
Department of CSE
IIIT Lucknow
Lucknow, UP, India
mcs21025@iiitl.ac.in

Khushbu Doulani
Vardhaman College of Engineering
Hyderabad, India
khushidoulani@gmail.com

Mainak Adhikari
Department of CSE
IIIT Lucknow
UP, India
mainak.ism@gmail.com

ABSTRACT

The emergence of novel types of communication, such as email, has been brought on by the development of the internet, which radically concentrated the way in that individuals communicate socially and with one another. It is now establishing itself as a crucial aspect of the communication network which has been adopted by a variety of commercial enterprises such as retail outlets. So in this research paper, we have built a unique spam-detection methodology based on email-body sentiment analysis. The proposed hybrid model is put into practice and preprocessing the data, extracting the properties, and categorizing data are all steps in the process. To examine the emotive and sequential aspects of texts, we use word embedding and a bi-directional LSTM network. this model frequently shortens the training period, then utilizes the Convolution Layer to extract text features at a higher level for the BiLSTM network. Our model performs better than previous versions, with an accuracy rate of 97–98%. In addition, we show that our model beats not just some well-known machine learning classifiers but also cutting-edge methods for identifying spam communications, demonstrating its superiority on its own. Suggested Ensemble model's results are examined in terms of recall, accuracy, and precision

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Dataset, KNN, Gaussian Naive Bayes, LSTM, SVM, Bidirectional LSTM, GRU, Word-Embeddings, CNN

ACM Reference Format:
Shivangi Sachan, Khushbu Doulani, and Mainak Adhikari. 2023. Semantic Analysis and Classification of Emails through Informative Selection of Features and Ensemble AI Model. In *2023 Fifteenth International Conference on Contemporary Computing (IC3-2023) (IC3 2023), August 03–05, 2023, Noida, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3607947.3607979

1 INTRODUCTION

Over the past few years, a clear surge of both the amount of spammers as well as spam emails. This is likely due to a fact that the investment necessary for engaging in the spamming industry is relatively low. As a result of this, we currently have a system that identifies every email as suspicious, which has caused major expenditures in the investment of defense systems [12]. Emails are used for online crimes like fraud, hacking, phishing, E-mail bombing, bullying, and spamming. [16]. Algorithms that are based on machine learning (ML) are now the most effective and often used approach to the recognition of spam. Phishing, which is defined as a fraudulent attempt to acquire private information by masquerading as a trustworthy party in electronic communication, has rapidly advanced past use of simple techniques and the tactic of casting a wide net; instead, spear phishing uses a variety of sophisticated techniques to target a single high-value individual. Other researchers used NB, Decision Trees, and SVM to compare the performance of supervised ML algorithms for spam identification [6]. Spam emails clog up recipients' inboxes with unsolicited communications, which frustrate them and push them into the attacker's planned traps [7]. As a result, spam messages unquestionably pose a risk to both email users and the Internet community. In addition, Users may occasionally read the entire text of an unsolicited message that is delivered to the target users' inboxes without realizing that the message is junk and then choosing to avoid it. Building a framework for email spam detection is the aim of this project. In this approach, we combine the Word-Embedding Network with the CNN layer, Bi-LSTM, and GRU (BiLSTM+GRU). CNN layers are used to speed up training time before the Bi-LSTM network, and more advanced textual characteristics are extracted with the use of this network in comparison to the straight LSTM network, in less time. Gated recurrent neural networks (GRUs) are then added because they train more quickly and perform better for language modeling. To evaluate and investigate various machine learning algorithms for predicting email spam, and develop a hybrid classification algorithm to filter email spam before employing an ensemble classification algorithm to forecast it. To put an innovative technique into practice and compare it to the current method in terms of various metrics. Ensemble learning, a successful machine learning paradigm, combines a group of learners rather than a single learner to forecast unknown target attributes. Bagging, boosting, voting, and stacking are the four main types of ensemble learning techniques. To increase performance, an integrated method and the combining of two or three algorithms are also suggested. Extraction of text-based features takes a long time. Furthermore, it can be challenging to extract all of the crucial information from a short text. Over the span associated with this

[*] Both authors contributed equally to this research.

IC3 2023, August 03–05, 2023, Noida, India

ACM ISBN 979-8-4007-0022-4/23/08…$15.00
https://doi.org/10.1145/3607947.3607979

research, we utilize Bidirectional Large Short-Term Memories (Bi-LSTM) in conjunction with Convolutional Neural Networks (CNN) to come up with an innovative method to the detection of spam. Bagging and boosting approaches were widely preferred in this study. Contribution and paper organization is as follows: section 1.1 describes literature study, section 1.2 describe motivation for this research work, section 2 sketches procedure of details implementation, Section 3 present experimental setup, dataset description and evaluation metrics, and section 4 summarizing outcomes of the experiment.

1.1 Related Work

Email is indeed the second most frequently utilized Internet application as well as the third most common method of cyberbullying, claims one study. Cybercriminals exploit it in a number of ways, including as sending obscene or abusive messages, adding viruses to emails, snatching the private information of victims, and exposing it to a broad audience. Spam letters made up 53.95% of all email traffic in March 2020. We examine three main types of unlawful emails in our study. First are fake emails, which are sent to manipulate recipients to submit sensitive information. The second as being cyberbullying's use of harassing emails to threaten individuals. Suspicious emails that describe illegal activities belong to the third category. Many researchers have earlier contributed massively to this subject. The researcher claims there is some proof that suspicious emails were sent before to the events of 9/11. [14]. When it comes to data labeling, there are also convinced rule-based approaches and technologies (like VADER) that are used, even though their efficiency of the are together is adversely affected. A hidden layer, which itself is essential for vectorization, is the top layer of the model. We use oversampling methods for this minority class because of the absence of data. Sampling techniques can help with multicollinearity, but they have an impact on simulation results. Oversampling causes data to be randomly repeated, which affects test data because dividing data may result in duplicates. Undersampling may result in the loss of some strong information. In order to advance email research, it is crucial to provide datasets on criminal activity. P. Garg et al. (2021) [5], which revealed that spam in an email was detected in 70 percent of business emails, spam was established as an obstacle for email administrators. Recognizing spam and getting rid of it were the primary concerns, as spam can be offensive, may lead to other internet sites being tricked, which can offer harmful data, and can feature those who are not particular with their content using NLP. To select the best-trained model, each mail transmission protocol requires precise and effective email classification, a machine learning comparison is done. Our study has suggested that innovative deep learning outperforms learning algorithms like SVM and RF. Current studies on the classification of emails use a variety of machine learning (ML) techniques, with a few of them focusing on the study of the sentiments consisted of within email databases. The lack of datasets is a significant obstacle to email classification. There are few publicly accessible E-mail datasets, thus researchers must use these datasets to test their hypotheses or gather data on their own. Authors[15] describe supplied two-phased outlier detection models to enhance the IIOT network's dependability. Artificial Neural Network, SVM, Gaussian NB, and RF (random forest) ensemble techniques were performed to forecast class labels, and the outputs were input into a classifying unit to increase accuracy. A method for content-based phishing detection was presented by the authors in [2], to classify phishing emails, they employed RF. They categorize spam and phishing emails. They enhanced phishing email classifiers with more accurate predictions by extracting features. They showed some effective Machine learning spam filtering techniques. When the PCA method is used, it will lower the number of features in the dataset. The collected features go through the PCA algorithm to reduce the number of features. The PCA method is used to make a straightforward representation of the information which illustrates the amount of variability there is in the data. The authors of [20] presented the Fuzzy C-means method for classifying spam email. To stop spam, they implemented a membership threshold value. A methodology to identify unlabeled data was put forth by the authors of [1] and applied motive analysis to the Enron data collection. They divided the data into categories that were favorable, negative, and neutral. They grouped the data using k-means clustering, an unsupervised ML technique and then classified it using the supervised ML techniques SVM and NB. Hina, Maryam, and colleagues (2021) implemented Sefaced: Deep learning-based semantic analysis and categorization of e-mail data using a forensic technique. For multiclass email classification, SeFACED employs a Gated Recurrent Neural Network (GRU) based on Long Short-Term Memory (LSTM). Different random weight initializations affect LSTMs [9]. Zhang, Yan, et al.(2019) Experiments on three-way game-theoretic rough set (GTRS) email spam filtering show that it is feasible to significantly boost coverage without decreasing accuracy [23]. According to Xia et al. [22], SMS spam has been identified using machine learning model such as naive bayes , vector-space modeling, support vector machines (SVM), long selective memory machines (LSTM), and convolutional neural networks including every instance of a method for categorizing data. Elshoush, Huwaida, et al. (2019) Using adaboost and stochastic gradient descent (sgd) algorithms for e-mail filtering with R and orange software spam [3]. Orange software was used to create the classifications, which included Adaboost and SGD. The majority of researchers focused on text-based email spam classification methods because image-based spam can be filtered in the early stages of pre-processing. There are widely used word bag (BoW) model, which believes that documents are merely unordered collections of words, is the foundation for these techniques. Kumaresan [11] explains SVM with a cuckoo search algorithm was used to extract textual features for spam detection. Renuka and Visalakshi made use of svm [17] spam email identification, followed by selecting features using Latent Semantic Indexing (LSI). Here we have used labeled dataset to train the hybrid classifier. We used TF-IDF for feature extraction [20] and Textual features for spam detection were extracted using SVM and a cuckoo search algorithm. [4] for filtering out the spam email. Combining the integrated strategy to the pure SVM and NB methods, overall accuracy is really improved. Moreover, accurate detection for spam email has been proposed using the Negative Selection Algorithm (NSA) and Particle Swarm Optimization's (PSO) algorithm. PSO is used in this instance to improve the effectiveness of the classifier.

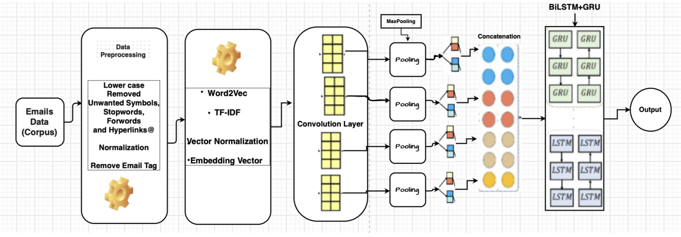
1.2 Motivation and Novelty

Email is most common form of communication between people in this digital age. Many users have been victims of spam emails, and their personal information has been compromised. The email Classification technique is employed to identify and filter junk mail, junk, and virus-infected emails prior to reach a user's inbox. Existing email classification methods result in irrelevant emails and/or the loss of valuable information. Keeping these constraints in mind, the following contributions are made in this paper:

- Text-based feature extraction is a lengthy process. Furthermore, extracting every important feature from text is difficult. In this paper, we show how to employ GRU with Convolutional Neural Networks and Bidirectional-LSTM to find spam.
- Used Word-Embeddings, BiLSTM, and Gated Recurrent Neural Networks to examine the relationships, sentimental content, and sequential way of email contents.
- Applied CNN before the Bi-LSTM network, training time can be sped up. This network can also extract more advanced textual features faster than the Bi-LSTM network alone when combined with the GRU network.
- We use Enorn Corpora datasets and compute precision, recall, and f-score to assess how well the suggested technique performs. Our model outperforms several well-known machine learning techniques as well as more contemporary methods for spam message detection.

2 PROPOSED SYSTEM ARCHITECTURE AND MODEL

E-mail is a valuable tool for communicating with other users. Email allows the sender to efficiently forward millions of advertisements at no cost. Unfortunately, this scheme is now being used in a variety of organizations. As a result, a massive amount of redundant emails is known as spam or junk mail, many people are confused about the emails in their E- Mailboxes. Each learning sequence is given forward as well as backward to two different LSTM networks that are attached to the same outputs layer in order for bidirectional Lstms to function. This indicates that the Bi-LSTM has detailed sequential information about all points before and following each point in a specific sequence. In other words, we concatenate the outputs from both the forward and the backward LSTM at each time step rather than just encoding the sequence in the forward direction. Each word's encoded form now comprehends the words that come before and after it. This is a problem for the Internet community. The diagram depicts various stages that aid in the prediction of email spam:



Because real-world data is messy and contains unnecessary information and duplication, data preprocessing is critical in natural language processing (NLP). The major preprocessing steps are depicted below.

2.1 NLP Tokenization

Tokenization of documents into words follows predefined rules. The tokenization step is carried out in Python with spacy library.

2.2 Stop Words Removal

Stop words appear infrequently or frequently in the document, but they are less significant in terms of importance. As a result, these are removed to improve data processing.

2.3 Text Normalization

A word's lexicon form or order may differ. Thus, they must all be changed to their root word to be correctly analyzed. Lemmatization and stemming are the two methods that can be used for normalization. When a word's final few characters are removed to create a shorter form, even if that form has no meaning, the procedure is known as stemming. lemmatization [21] is a mixture of corpus-based an rule-based methods, and it retains the context of a term while changing it back to its root.

2.4 Feature Extraction

feature extraction which transforms the initial text into its features so that it may be used for modeling after being cleaned up and normalized. Before predicting them, we use a specific way to give weights to specific terms in our document. While it is simple for a computer to process numbers, we choose to represent individual words numerically. In such cases, we choose word embeddings. IDF is the count of documents containing the term divided by the total number of documents, and occurrence is the amount of instances a word appears in a document. We derive characteristics based on equations. 1,2,3,4,5, and 6. We use equations to derive properties.

$$TfIdf = tf * \left(\frac{1}{df}\right) \tag{1}$$

$$TfIdf = tf * \text{Inverse}(df) \tag{2}$$

$$TfIdf(t,d,D) = Tf(t,d).Idf(t,D) \tag{3}$$

$$TIdf(t,d) = \log \frac{N}{|d\epsilon Dt\epsilon D|} \tag{4}$$

A word2vec neural network-based approach is the method that is utilized for this goal as the tool. The following equation, referred to as 5, shows how word2vec handles word context through the use of probability-accurate measurements. Here letter D stands for the paired-wise display of a set of words, while the letters w and c0 or c1 represent paired word context that originated from a larger collection of set D.

$$P\left(D=1 \mid w, c_{11:k}\right) = \frac{1}{1+e^{-\left(w \cdot c_1 1 + w \cdot c_1 2 + \ldots + w \cdot c_1 k\right)}} \tag{5}$$

$$P\left(D=1 \mid w, c_{1:k}\right) = \frac{1}{1+e^{-\left(w \cdot c0\right)}} \tag{6}$$

2.5 Word-Embeddings

Word-Embedding helps to improve on the typical "bag-of-words" worldview, which requires a massive sparse feature vector to score every word individually to represent this same entire vocabulary. This perception is sparse because the vocabulary is large, and each word or document is defined by a massive vector. Using a word map-based dictionary, word embedding needs to be converted terms (words) into real value feature vectors. There are two basic issues with standard feature engineering techniques for deep learning. Data is represented using sparse vectors, and the second is that some of the meanings of words are not taken into consideration. Similar phrases will have values in embedding vectors that are almost real-valued. The Input length in our proposed study is set to 700 for our suggested model. If the texts seemed to be integer encoded with value systems between 10 and 20, the vocabulary distance would be 11. Our data is encoded as integers, and the input and output dimensions are both set to 50,000. The embedding layer outcome will be used in successive layers and for BiLSTM and GRU layers.

2.6 Machine Learning Model

Within the scope of the research, we are using the subsequent machine learning techniques, to examine and compare the overall efficacy of our suggested Bi-LSTM strategy: Support Vector Machine, Gaussian NB, Logistic Regression, K - nearest neighbors, and Random Forest (RF).

2.7 Convolution Network

The popular RNN model generally performs well but takes too long to train the model incorporating the textual sequential data. When a layer is added after the RNN layer, the model's learning duration is considerably decreased. Higher-level feature extraction is another benefit. [19] additionally possible using the convolutional layer. In essence, the convolution layer looks for combinations of the various words or paragraphs in the document that involve the filters. We use features with 128 dimensions and a size 10 for each. For this task, the Relu activation function is utilized. After that, the one-dimensional largest pooling layers with a pooling size of 4 are put on the data in order to obtain higher-level features.

2.8 BiLSTM Network with GRU

Recurrent Neural Network (RNN) technique of text sentiment analysis is particularly well-liked and frequently applied. Recurrent neural networks (RNN) surpass conventional neural networks. because it can remember the information from earlier time steps thanks to its memory. A state vector is combined with an RNN's data to create a new state vector. The resulting state vector uses the present to recollect past knowledge. The RNN is straightforward and is based on the following equations:

$$h_t = \tanh\left(W_{hh} h_{t-1} + W_{\pi h} x_t\right) \tag{7}$$

$$y_t = W_{h_y} h_t \tag{8}$$

The vanilla RNN[18]is not very good at remembering previous sequences. In addition to that, RNN struggles with diminishing gradient descent. A kind of RNN is a long short-term recall network (LSTM), solves a vanishing gradient descent problem and learns long-term dependencies[10]. LSTM was actually created to address the problem of long-term reliance. LSTM has the unique ability to recall. The cell state is the LSTM model's central concept. With only a small amount of linear interaction, the cell state follows the sequence essentially unmodified from beginning to end. gate of an LSTM is also significant. Under the command of these gates, information is safely inserted to or eliminated from the cell stated. The following equations are used by the LSTM model to update each cell:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{9}$$

In this case, Xt denotes input, and ht is the hidden state at the t time step. The following is the revised cell state Ct:

$$\mathrm{i}_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{10}$$

$$C_T = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_{ct}\right) \tag{11}$$

$$C_t = f_t * C_{t-1} + i_t * C_T \tag{12}$$

Here, we may compute the output and hidden state at t time steps using the point-wise multiplication operator *.

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{13}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{14}$$

Due to the reality it only considers all prior contexts from the present one, LSTM does have a few drawbacks. As a result of this, it may accept data from preceding time steps through LSTM as well as RNN. Therefore, in order to avoid this issue, further improvements are carried out with the help of a bidirectional recurrent neural network(Bi-RNN). BiRNN [13] can handle two pieces of information from both the front and the back. Bi-LSTM is created by combining the Bi-RNN and LSTM. As a result, operating LSTM has advantages such as cell state storage so that BiRNN have way to acknowledge from the context before and after. As a consequence of this, it provides the Bi-LSTM with the advantages of an LSTM with feedback for the next layer. Remembering long-term dependencies is a significant new benefit of Bi-LSTM. The output, which is a feature vector, will be based on the call state. Finally, we forecast the probability of email content as Normal, Fraudulent, Harassment, and Suspicious Emails using as an input to the softmax activation function, which is a weighted sum of the dense layer's outputs. To regulate the information flow, GRU employs the point-wise multiplying function and logistic sigmoid activation. The GRU has hidden states of storage memory and does not have distinct memory cells or units for state control. The W, U, and b vectors, which stand for weights, gates, and biases, respectively, are crucial variables that must be calculated during the creation of the GRU model. For training reasons, the pre-trained word embedding known as the Glove vector is used. They made it clear that GRU is the superior model when there is a large amount of training data for textual groups and word embedding is available. BiLSTM, CNN, and GRU is required so as to compensate for the deletion of the document's long-term and short-term connections. In our case, the embedding dimension, maximum sequence length, and lexicon size were used to start the LSTM embedding layer in three separate LSTM models. The input vector was modified to make it appropriate for such a Conv1D layer, prior situations' sequences are returned by LSTM layer. The "return sequences" of the LSTM layer must be set to False when the subsequent state is free of the gated architecture. Quantity of learning parameters must be taken into consideration. A 350-unit LSTM layer was set - up, and different LSTM unit combinations were tested. More importantly, because it has more parts, the model made with BiLSTM will take longer to train. Bidirectional LSTM is the name of a particular kind of recurrent neural network that is primarily used for the processing of natural languages. (BiLSTM). It is able to use data from both sides, and, in contrast to regular LSTM, it enables input flow in both directions. It is an effective instrument for demonstrating the logical relationships between words and phrases, and this involves both the forward and backward directions of the sequence. In conclusion, BiLSTM works by adding one extra layer of LSTM, causing the information flow to travel in the other direction. It only denotes that the input sequence runs in reverse at the next LSTM layer. Multiple operations, including averaging, summation, multiplication, and concatenation, are then applied to the results of the two LSTM layers. The gated design of Bi-LSTM and GRU networks solves the disappearing gradient and exploding problems. A good way to handle more long sequences is to use Bi-LSMT and GRU together. GRU works well with datasets that don't have text. In two to three rounds, the complicated CNN+BiLSTM+GRU model learns the long sequence of email text well. We have used word embedding, cnn, bidirectional lstm and gru networks as our three building blocks to separate email messages based on their sentiment and text's sequential features. Also, we succinctly demonstrate below why these blocks help identify email spam:

- First, We have used the Sequence - to - sequence Lstm as the current block in the networks since it can retrieve both the previous and next sequences from the current. More so than a straightforward LSTM network, it can also recognize and extract text sentiment and sequential properties.
- Second, we extract the more complex and advanced characteristics for Bi-LSTM network using Convolutional Network block, which is the network's second block after the Bi-LSTM block. Bi-LSTM takes a long time to extract text-based features, hence one of the reasons for using this block is to reduce the network's overall training time.

3 EXPERIMENTAL EVALUATION

3.1 Experimental Setup

We divided the information into training and testing groups of 80/20. We divided the remaining 20% of the 80 percent training data into test data for the model. Construct, compute, and evaluate the efficacy of the suggested method using the Pythonic packages Keras, as TensorFlow and Scikit learn.

3.2 Dataset Description

Email spam detection is the foundation of this research project. The dataset includes normal emails from the Enron corpora, deceptive emails from phished email corpora, harassment emails chosen from hate speech, and the offensive dataset. Only the content of the email body is used for analysis; all header information, including sender, topic, CC, and BCC, are eliminated. Word2vector, TF-IDF, and Word Embedding are used to extract characteristics from the email message and classify them. This dataset[8] is publicly available. The presented model is implemented using Python, and several metrics, including accuracy, precision, and recall, are used to examine the outcomes.

3.3 Evaluation Metrics and Results

Classifier performance is assessed Using metrics such as accuracy, precision, and recall. Four terms make up a confusion matrix that is used to calculate these metrics.

- True positives (TP) are positive values that have been accurately assigned the positive label.
- The negative values that are accurately identified as negative are known as True Negatives (TN).
- True Negative values are those that can be accurately identified as being negative (TN).
- Positive readings that have been mistakenly labeled as negative are known as False Negatives (FN).

Assess the efficacy of the suggested model is listed below:

3.3.1 *Accuracy.*

Accuracy reveals how frequently the ML model was overall correct.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

3.3.2 *Precision.*

The accuracy of the model gauges how effectively it can predict a specific category.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

3.3.3 *Recall.*

Recall tells us how often the model was able to recognize a specific category.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

Model	Accuracy	Precision	Recall
Gaussian NB	91.3	90.1	91.8
Random Forest	88.41	90	88
KNN	86.6	89	87
SVM	92.4	91	92
LSTM	95.2	95	95.7
Proposed Ensemble (CNN,BiLSTM+GRU)	97.32	95.6	95.3

Table 1: Differet Model's Score on Test Data

Accuracy, Precision, and Recall metrics are computed. In the given Table 1 where six different classifiers are Gaussian NB, Random Forest, KNN, SVM, LSTM, and Propose Ensemble Hybrid Model (CNN+BiLSTM+GRU) have been used in this work. In the CNN, Bi-LSTM, and GRU architectures which enable sequence prediction, CNN strands for feature extraction on data input which are combined with LSTM. It requires less time training and a higher expandable model. Any bottlenecks are created by predictions and the increasing number of distinct units of information. This model is useful for dealing with issue-related classifications that consist of two or more than two classes. So suggested Ensemble model, out of these six classifiers, produces more accurate findings.

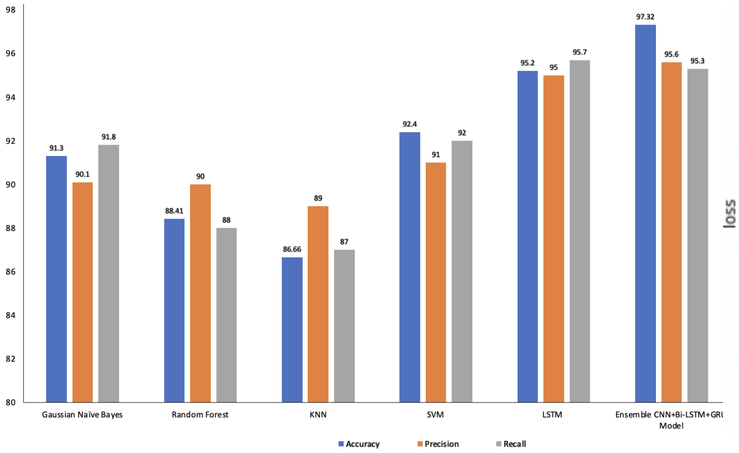


Figure 1: Performance Analysis

3.4 Comparative Analysis

A model's ability to fit new data is measured by the validation loss, whereas its ability to fit training data is determined by the training loss. The two main variables that decide whether in which learning is efficient or not are validation loss and training loss. LSTM and Suggested Ensemble hybrid Models have equivalent loss and accuracy. In this context, we are contrasting the LSTM with the proposed model (CNN, Bilstm, and GRU) in terms of their respective validation accuracies and losses. The model's accuracy was at its highest after 14 epochs of operation when it achieved an accuracy of roughly 97-98% while minimizing model loss.

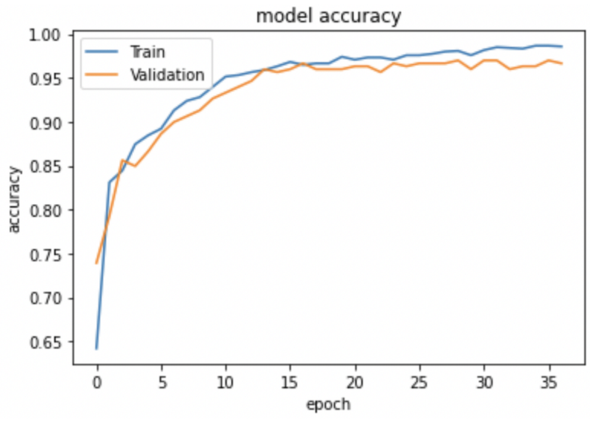


Figure 2: LSTM Model Training and Validation Accuracy

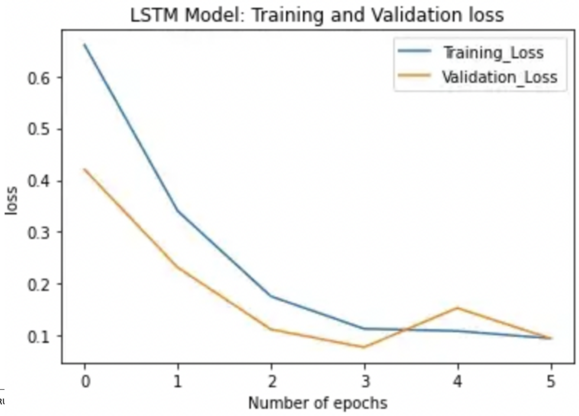


Figure 3: LSTM Model Training and Validation Loss

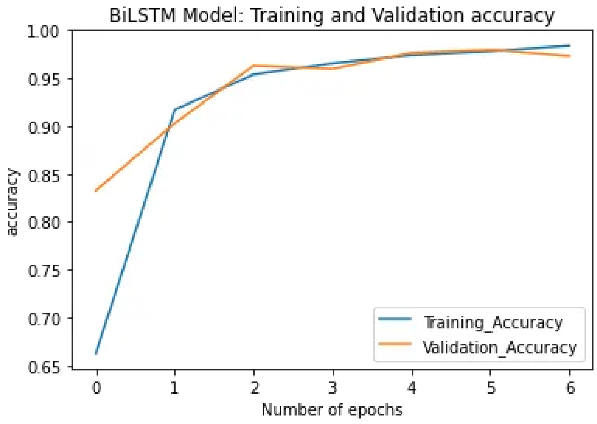


Figure 4: Ensemble Model (CNN,BiLSTM+GRU) Training and Validation Accuracy

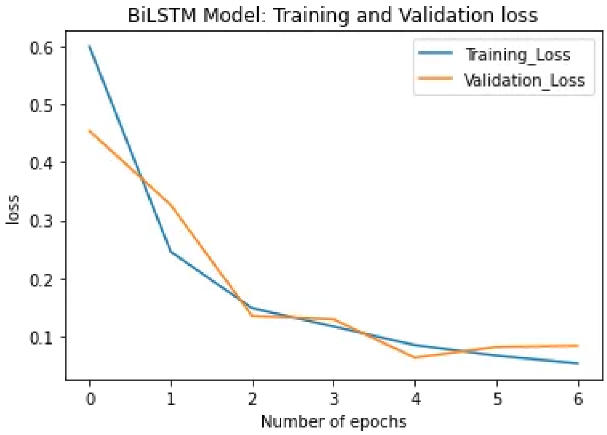


Figure 5: Ensemble Model (CNN,BiLSTM+GRU)Training and Validation Loss

In this Proposed ensemble hybrid model's train accuracy is 98.7% Validation accuracy is 97.32% and LSTM has train accuracy of 97.41% and validation accuracy is 95.2%. So based on figures 3 and 5 indicate the validation loss for LSTM and the proposed ensemble hybrid model to be 0.93 and 0.84, respectively, and figures 2 and 4 show the validation accuracy to be 95.2% and 97.3%, respectively. LSTM and the proposed hybrid model used ensemble artificial intelligence, with the proposed hybrid model outperforming the LSTM. We decide on dense architecture as the final model for identifying the text messages as spam or nonspam based on loss, accuracy, and the aforementioned charts. The loss and accuracy over epochs are more stable than LSTM, and the Proposed classifier has a straightforward structure.

4 CONCLUSION

The model is composed of four networks Word-Embeddings, CNN, Bi-LSTM, and GRU. We may train the model more quickly by using the convolutional layer first, followed by the word-embedding layer, and then the BiLSTM network. The Bidirectional LSTM network also has higher-level properties that we can extract. We have used a bidirectional LSTM(BiLSTM)and GRU network to memorize a sentence's contextual meaning and sequential structure, which improves the model's performance accuracy to roughly 97.32 percent.

REFERENCES

[1] Rayan Salah Hag Ali and Neamat El Gayar. 2019. Sentiment analysis using unlabeled email data. In *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*. IEEE, 328–333.

[2] Ali Shafigh Aski and Navid Khalilzadeh Sourati. 2016. Proposed efficient algorithm to filter spam using machine learning techniques. *Pacific Science Review A: Natural Science and Engineering* 18, 2 (2016), 145–149.

[3] Huwaida T Elshoush and Esraa A Dinar. 2019. Using adaboost and stochastic gradient descent (sgd) algorithms with R and orange software for filtering e-mail spam. In *2019 11th Computer Science and Electronic Engineering (CEEC)*. IEEE, 41–46.

[4] Weimiao Feng, Jianguo Sun, Liguo Zhang, Cuiling Cao, and Qing Yang. 2016. A support vector machine based naive Bayes algorithm for spam filtering. In *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 1–8.

[5] Pranjul Garg and Nancy Girdhar. 2021. A Systematic Review on Spam Filtering Techniques based on Natural Language Processing Framework. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 30–35.

[6] Adam Kavon Ghazi-Tehrani and Henry N Pontell. 2021. Phishing evolves: Analyzing the enduring cybercrime. *Victims & Offenders* 16, 3 (2021), 316–342.

[7] Radicati Group et al. 2015. Email Statistics Report 2015–2019. *Radicati Group. Accessed August* 13 (2015), 2019.

[8] Maryam Hina, Mohsin Ali, and Javed. 2021. Sefaced: Semantic-based forensic analysis and classification of e-mail data using deep learning. *IEEE Access* 9 (2021), 98398–98411.

[9] Maryam Hina, Mohsin Ali, Abdul Rehman Javed, Fahad Ghabban, Liaqat Ali Khan, and Zunera Jalil. 2021. Sefaced: Semantic-based forensic analysis and classification of e-mail data using deep learning. *IEEE Access* 9 (2021), 98398–98411.

[10] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. 2017. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE transactions on smart grid* 10, 1 (2017), 841–851.

[11] T Kumaresan and C Palanisamy. 2017. E-mail spam classification using S-cuckoo search and support vector machine. *International Journal of Bio-Inspired Computation* 9, 3 (2017), 142–156.

[12] Nuha H Marza, Mehdi E Manaa, and Hussein A Lafta. 2021. Classification of spam emails using deep learning. In *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*. IEEE, 63–68.

[13] Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 234–239.

[14] Sarwat Nizamani, Nasrullah Memon, Mathies Glasdam, and Dong Duong Nguyen. 2014. Detection of fraudulent emails by employing advanced feature abundance. *Egyptian Informatics Journal* 15, 3 (2014), 169–174.

[15] V Priya, I Sumaiya Thaseen, Thippa Reddy Gadekallu, Mohamed K Aboudaif, and Emad Abouel Nasr. 2021. Robust attack detection approach for IIoT using ensemble classifier. *arXiv preprint arXiv:2102.01515* (2021).

[16] Justinas Rastenis, Simona Ramanauskaitė, Justinas Janulevičius, Antanas Čenys, Asta Slotkienė, and Kęstutis Pakrijauskas. 2020. E-mail-based phishing attack taxonomy. *Applied Sciences* 10, 7 (2020), 2363.

[17] Karthika D Renuka and P Visalakshi. 2014. Latent semantic indexing based SVM model for email spam classification. (2014).

[18] Shuvendu Roy, Sk Imran Hossain, MAH Akhand, and N Siddique. 2018. Sequence modeling for intelligent typing assistant with Bangla and English keyboard. In *2018 International Conference on Innovation in Engineering and Technology (ICIET)*. IEEE, 1–6.

[19] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. 2015. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. Ieee, 4580–4584.

[20] Anuj Kumar Singh, Shashi Bhushan, and Sonakshi Vij. 2019. Filtering spam messages and mails using fuzzy C means algorithm. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*. IEEE, 1–5.

[21] Kristina Toutanova and Colin Cherry. 2009. A global model for joint lemmatization and part-of-speech prediction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. 486–494.

[22] Tian Xia. 2020. A constant time complexity spam detection algorithm for boosting throughput on rule-based filtering systems. *IEEE Access* 8 (2020), 82653–82661.

[23] Yan Zhang, PengFei Liu, and JingTao Yao. 2019. Three-way email spam filtering with game-theoretic rough sets. In *2019 International conference on computing, networking and communications (ICNC)*. IEEE, 552–556.

Received 15 April 2023