International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(9s) 2026 pp. 1261-1271
https://doi.org/10.70917/ijcisim-2026-3551
Article

Ensemble-Based Deep Learning Model for Identifying Malicious Text in Natural Language Processing

Snehalatha N[1*], Mohana Kumar S[2]

*[1]JSS Academy of Technical Education, Bangalore
[2] M S Ramaiah Institute of Technology, Bangalore

Abstract: Text classification in Natural Language Processing (NLP) is crucial for sorting text data into classes. Detecting internet spam is a long-standing issue. Solutions exist for identifying spam on social media and in emails. Deep learning models are effective for text classification post-preprocessing. Preprocessing involves steps like converting text to lowercase and removing numbers, eliminating web addresses, punctuations, and stop words, as well as tokenization and stemming. Then, Feature extraction is used to generate functions from the dataset the use of area expertise and those features are extracted via Bag-of-words (BoW), word2vec, and TF-IDF. A feature set is created and it can additionally contain functions that are not meaningful or of very short duration, such capabilities want to be eliminated for better outcomes. Finally, ensemble deep learning algorithms such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Bidirectional Long Short-Term Memory (Bi-LSTM) are used to classify the malicious content within the text classification. This method is used in classifying the text and performs higher whilst compared to the existing strategies. The suggested model is implemented using the PYTHON programming language, and its performance is assessed using metrics such as accuracy, precision, recall, F-measure, balanced accuracy, brier score loss, and MAP. The suggested ensembled deep learning model achieves higher accuracy with a learning rate of 70% is 99% and a learning rate of 80% is 99.61%.
Keywords: Natural Language Processing; Malicious and Non-malicious classification; Text Classification; Spam classification; Deep Learning.

1. INTRODUCTION

NLP is a artificial intelligence region that makes a specialty of supporting computer systems in detecting, deciphering, and generating human language. This subject includes a variety of duties like speech popularity, sentiment analysis, language translation, and text classification [1]. Text type, which is also known as categorization or category, is one of the primary duties in NLP, in which the project is to mechanically label text facts with predetermined classes. The attempt is extraordinarily vital for numerous packages, which include language identity, sentiment evaluation, junk mail detection, and concern rely elegance [2]. Through this manner, deep learning fashions analyze patterns and functionalities from categorised textual content records, permitting them to categorize newly found fabric into applicable lessons [3]. This offers a unique set of features that could be applied to a wide range of tasks, such as sentiment analysis (good, bad, or neutral), unsolicited mail detection (junk mail or not), topic classification (sports, politics, and generation), and motive reputation (customer service inquiries) [4]. As such, it makes it easier to effectively organize and comprehend large amounts of textual material. The majority of core system learning techniques, like Support Vector Machines (SVM), logistic regression, naive Bayes, rely on manually generated features extracted from textual content entries. These models are based on weak models that may be shallow or naive, impacting the domain [5] [6].

There are several difficulties in making use of deep mastering-primarily based text class for NLP including feature identity from unstructured statistics, managing huge datasets efficaciously, ensuring model generalization throughout domain names, and addressing version interpretability [7]. Overcoming those demanding situations

includes improvements in function extraction techniques, scalable version architectures, transfer studying strategies for domain models, and interpretability methods such as attention mechanisms and version visualization strategies [8]. Ensembling multiple fashions and utilising pretrained embeddings consisting of BERT can beautify accuracy and robustness in textual content classification obligations [9]. However, interpreting deep studying fashions remains tough due to their complexity, making it hard to recognize the decision-making system in the back of class effects [10]. To manage those obstacles, advances in feature extraction, version scalability, transfer studying strategies, and interpretability version techniques are required [11]. Text category fashions improve their accuracy, resilience, and applicability throughout a much wider variety of NLP packages and domains, addressing those difficulties [12].

Several deep learning-based textual content types exist for NLP realization on revolutionary architectures and techniques for improving type accuracy and performance [13] [14]. These are using advanced neural network architectures like Transformers and BERT for contextual knowledge, incorporating attention mechanisms to capture critical facts, leveraging pre-skilled word embeddings for feature illustration, and using switch-study techniques for domain variation [15]. In addition, there are efforts on ensemble learning, self-supervised learning, and consideration of model interpretability toward improving the performance and scalability of text class models across a variety of NLP tasks and domains [16].

This paper aims to develop a collective deep learning approach, particularly for NLP-based harmful text classification. The focus is on resolving issues by identifying relevant features, effectively managing huge datasets, ensuring version generalization, and addressing version interpretability. The suggested method makes use of ensemble formats and advanced deep learning techniques such as CNN, RNN, and Bi-LSTM to achieve superior performance in detecting harmful content within text data.

The major contributions of the paper are as follows:

- The research presents an ensemble of deep learning methods (CNN, RNN, and Bi-LSTM) for text classification that specifically aims to identify harmful content using natural language processing. This work broadens the range of approaches available for effectively handling complex textual content category responsibilities.
- Through the use of preprocessing methods such as text conversion to lowercase, and removing numerals, URLs, punctuation, and stop words, the paper exhibits improved overall performance in dangerous content classification.
- Effective use of feature extraction techniques like word2vec, BoW, and TF-IDF allowed for extraction of useful features from dataset, which improved performance and comprehension of model.

The following sections are organized as follows: Section 2 explores relevant research and literature reviews, Section 3 introduces the proposed framework, Section 4 provides a detailed analysis of the observed results and discussions, and Section 5 offers the final assessment of this study.

2. LITERATURE SURVEY

Some of the recent research works related to Text classification were reviewed in this section

An analytical method for a CNN text classification model's interpretability was proposed by Ce and Tie [17] in 2020. The proposed approach may perform multi-angle analysis on the discriminant outcomes of multi-classified text and multi-label classification tasks by applying backtracking analysis to model prediction findings. Finally, depending on their understandability, the model's analytical results can be displayed from numerous angles utilizing visualization tools.

Gong *et al.*, [18] developed a novel Hierarchical Graph Transformer model for large-scale multi-label text classification in 2020. The text was transformed into a graph structure to capture semantics and connections. A multi-layer transformer with a multi-head care mechanism was employed at various stages to extract text features. Label representations were generated using hierarchical label relationships, and a weighted loss function based on semantic distances between labels was constructed.

Zhang *et al.*, [19] highlighted an ensemble-based method for digital communication spam identification, addressing the growing problem of unsolicited messages, or spam in 2024. Effective information filtering systems have to be developed to preserve efficiency and security due to the exponential rise of online platforms. Three key elements comprise the suggested method: decision fusion, classifier selection, and feature extraction. The best classifiers for spam detection are determined by evaluating a number of them, including RNN, LSTM, and GRU.

In 2021, Behera and Kumaravelan proposed a modified CNN framework for feature extraction and text document classification [20]. Next, using both TF-IDF and modified CNN-based feature extraction algorithms, FRS and FRS-RNN were constructed for textual document classification on benchmark datasets such as 20 Newsgroup and Reuter-21578.

In 2021, Gan *et al.*, [21] suggested a structure-extended MNB (SEMNB). Acclimate the well-known hidden NB (HNB) to propose a single model named hidden MNB (HMNB). For every feature, HMNB generates a hidden parent that combines the influences of all the other eligible features. Suggest a straightforward yet efficient learning technique for HMNB that avoids a high computational complexity structure-learning procedure. The refined concept can also be applied to enhance the one-versus-all-but-one model (OVA) and complement NB (CNB).

In 2022, Zheng et al., [22] presented a deep learning model integrating Bi-LSTM and an attention mechanism. Bi-LSTM was used to extract verbal features and keywords, and the attention mechanism was used to weigh the generated keywords. Twelve thousand sentences generated during online collaborative discussion exercises have been categorized into six categories: statement, management, question, emotion, negotiation, and others.

Pawar et al., [23] suggested a system employing the c4.5 Decision tree classifier and the Ensemble Learning Technique of the Random Forest approach to categorize and document text in 2023. The system's processes involve constructing decision tree text classifiers, reducing dimensions, employing TF/IDF indexing, training models, clustering terms with brown clustering, and categorizing documents by testing the dataset through classifiers.

A hybrid Deep Learning-based model by Ahmad *et al.,* [24] in 2021 showed the effectiveness for eight personality traits using Convolutional Neural Network and Long Short-Term Memory. The personality trait categorization task was completed by applying our experimental evaluations to the benchmark dataset. Compared to state-of-the-art methods, the suggested model may successfully categorize the user's personality features based on evaluations of the datasets that produced better results. Lastly, use statistical analysis to assess the efficacy of the strategy.

In 2020, Zhang et al., [25] supplied a unique technique that efficiently confuses human observers at the same time as producing legible antagonistic writings with a few modifications. The suggested method, based at the continuous bag-of-words (CBOW) model, manipulates the perturbation path vectors to discover the proper perturbations with the intention to produce the adverse texts.

3. PROPOSED METHODOLOGY

Text classification is one of the most crucial NLP functions to automatically put text in predefined classes. The problem of spam detection has seen successful solutions, particularly concerning identifying spam comments on social media and fraudulent emails. Such solutions have already been achieved by using many different techniques of text analysis, including machine learning algorithms and rule-based systems. The critical role of text classification is the difference between legitimate and malicious content for the enhancement of online security, content moderation, and user experience. The use extends to several other purposes like sentiment analysis, topic categorization, and intent recognition and influences various aspects of digital communication and decision-making processes.

The text preprocessing step involves creating the dataset by processing the text, converting it into lowercase, removing numbers, and website addresses, and removing punctuations, stop words, tokenization, and stemming. Afterward, feature engineering is done using domain knowledge through BoW, Word2Vec, and TF-IDF. Finally, using deep learning models like CNN, RNN, and Bi-LSTM, harmful material in text categorization is classified. Figure 1 shows the overall architecture of the proposed methodology.

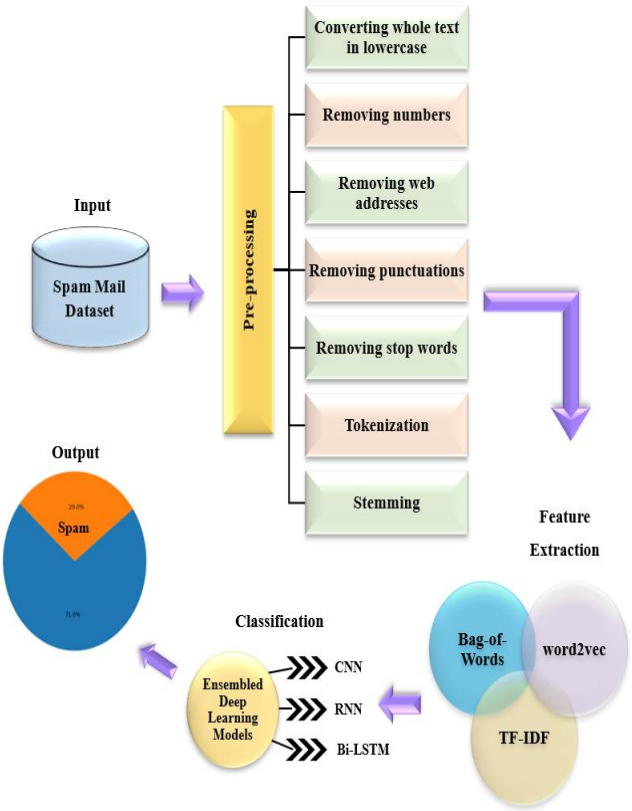

Figure 1: Overall architecture of the proposed methodology

3.1 Dataset Description

The data is composed from https://www.kaggle.com/datasets/venky73/spam-mails-dataset. The enron1 folder is the source of the spam mail dataset, which includes categorized emails (spam and ham). Various email files are stored in each folder. The dataset is made available for analysis and machine learning activities by iteratively going through these files, extracting pertinent information such as sender and message content, and arranging it into a Data Frame.

3.2 Preprocessing

Preprocessing is the first stage in the text class domain; it refines raw textual content facts to enhance their appropriateness for analysis. This crucial step entails several actions, including changing all text to lowercase for consistency and focusing just on the content while removing punctuation, numerical values, and web addresses. Tokenization, which divides text into meaningful devices, stemming, which returns sentences to their basic form, and the removal of common prevent words are also essential.

3.2.1 *Converting whole text in lowercase*

The process involves changing entire review text terms to lowercase. To simplify, everything is typically reduced to lowercase. Lowercasing applies to the majority of text mining and natural language processing tasks and considerably improves output consistency. It is essential to observe, however, that some words, such as "US" and "us," might alter meanings when reduced to lowercase.

3.2.2 *Removing Numbers*

It is used to describe a preprocessing step in which text data has its number of digits removed. Depending on the use cases, a number may or may not contain any important information. Therefore, removal of them is preferable to keeping them. This stage is frequently taken to standardize and clean the language in preparation for more modeling or analysis.

3.2.3 *Removing Web Addresses*

Eliminating web addresses (URLs) from textual data, applied for various purposes, is essential due to several reasons. The URLs provide directions to online locations in the sense that often non-semantic elements like alphanumeric sequences, special symbols, and domain-specific terms that are not relevant to NLP tasks are present within them. With URLs, linguistic content is uniform and purified, reducing disturbance and focusing on what matters.

3.2.4 *Removing Punctuations*

NLP is a step in extracting textual information, and it involves removing all punctuation from texts among other preprocessing procedures. This action guarantees that each one texts are processed uniformly, no matter the various punctuation. For example, "This" and "This!" come to be indistinguishable once the punctuation is eliminated, making the text tokenization greater every day. This normalization assures the performance of different NLP tasks like text classification and sentiment analysis to be of increased accuracy and reliability.

3.2.5 *Removing Stop Words*

The removal of stop words is an important part of NLP preprocessing since it eliminates words that aren't necessary, such as "the," "is," "a," and "and," and have no semantic significance. These terms, known as stop words, are widely used in English yet have limited meaning when used alone. Eliminating them simplifies text data by emphasizing words that provide content.

3.2.6 *Tokenization*

All sentences of the text were divided into pieces is called Tokenization. Tokens were used to break down each sentence in the review texts. Tokenization NLP converts text into tokens such as words, phrases, and characters. It helps to organize data for analysis tasks like text classification or sentiment analysis. Word tokenization separates text into words, whereas sentence tokenization divides it into sentences. Character tokenization separates text into distinct characters.

3.2.7 *Stemming*

Stemming is a linguistic normalization technique for NLP and information retrieval, decreases words to their base or root form, known as stem. To normalize words, suffixes and prefixes are removed such as ing, s, es, and so on. Stemming reduces vocabulary size and consolidates related terms, which improves text analysis tasks like indexing and information retrieval. For example, "publishing" would be stemmed to "publish".

3.3 Feature Extraction

Following preprocessing, it is the next step in the text classification process. Features are generated from the processed text using know-how of the sector. Methods which include Word2Vec, Bag-of-Words, and TF-IDF are used to extract applicable features. Together, Word2Vec facts phrase embeddings, Bag-of-Words depicts textual content as phrase frequency vectors, and TF-IDF balances the importance of phrases within the dataset to facilitate the improvement of robust type fashions.

3.3.1 *Bag-of-Words (BoW)*

In NLP, BoW method turns textual content into numerical representations for Computer algorithms. It is a fundamental notion in NLP that simplifies text illustration and evaluation. It converts textual content records to numerical vectors by way of calculating the frequency of phrases in a document. This version ignores phrase order and grammar, focusing solely on the presence and frequency of words. Despite its simplicity, BoW is usually hired for tasks like text categorization, sentiment evaluation, and file clustering.

3.3.2 *Word2vec*

It is a member of the embedding circle of relatives of textual content vectorization algorithms, in which words are projected right into a decrease-dimensional numeric vector space primarily based on their linguistic context the usage of a shallow neural community. Words with similar meanings inside the same context are assumed to be represented further on this allotted vector illustration of words. Word vectors with associated meanings are consequently grouped inside the vector space. This own family of vectorization techniques addresses the problems of excessive dimensionality and loss of phrase context which might be unique to the bag of phrases and its n-gram equivalents. Word2Vec has primary models: Skip-gram and Continuous Bag of Words (CBOW).

Continuous Bag of Words (CBOW): It predicts the target phrase based totally on its surroundings. The CBOW is a vector representation of words. Unlike different algorithms, along with Skip-gram, which predicts context phrases based on a target word, CBOW predicts a target phrase the usage of its context words. This model plays well on smaller datasets and captures the text's worldwide context. It works by using instructing a neural network to understand the institutions between phrases in a positive set of context words. By aggregating context facts, CBOW produces dense vector representations that preserve semantic statistics and may be used for responsibilities which includes sentiment analysis, textual content categorization, and gadget translation. This structure stores the words within the vocabulary as columns in the matrix W, each of that is mapped to a wonderful one-warm encoded vector. Figure 2 explains the architecture of the Continuous Bag of Words.

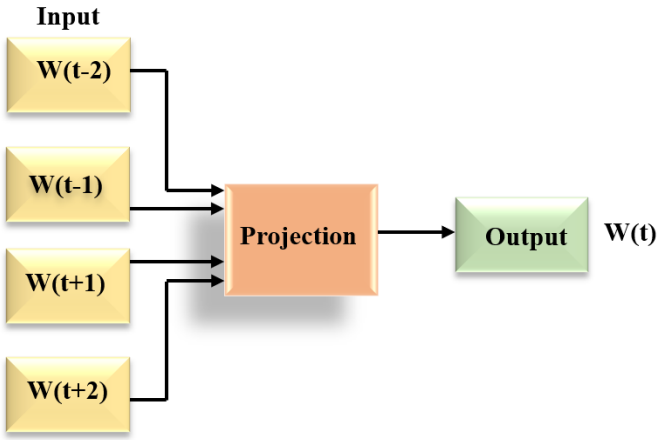

Figure 2: Architecture of Continuous Bag of Words

Skip-gram model: In NLP, this model proposes an efficient word embedding method: forecasting context words from a target word. It may be better at collecting local context than CBOW, which makes it particularly useful for larger datasets. CBOW is generally good at capturing the overall context. By training neural networks on loss reduction concerning expected and actual context words, skip-gram models word representations. Skip-gram is useful for tasks like word analogy completion and similarity computation because it is excellent at capturing semantic associations. Figure 3 demonstrates the architecture of the Skip-gram model.

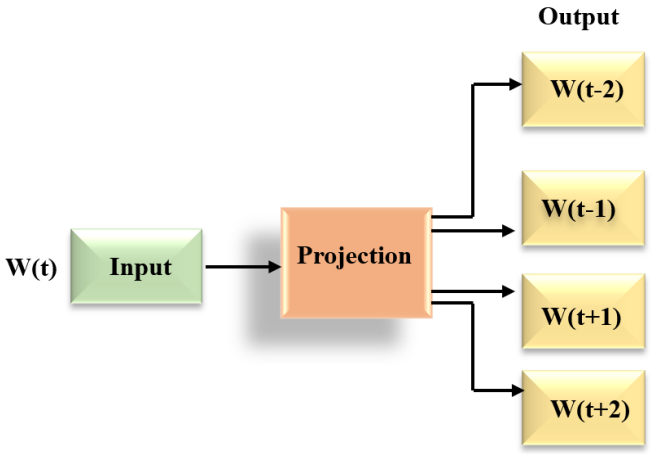

Figure 3: Architecture of Skip-gram model

3.3.3 *TF-IDF*

The frequency with which a word seems in a report is referred to as Term frequency (TF). It is an essential component of the weighting system known as Term Frequency-Inverse Document Frequency (TF-IDF). A term's frequency of occurrence in a text is measured using term frequency (TF) about total number of terms in document. Mathematical expression of TF can be shown in Eq. (1)

$$TF = \frac{\text{number of times a word appears in a document}}{\text{The total word count of a document}} \quad (1)$$

Computation of Inverse Domain Frequency (IDF), which is the frequency of a word throughout the set of documents, is displayed below. IDF highlights the significance of the term. Less common words are typically more informative. The mathematical expression of IDF can be shown in Eq. (2)

$$IDF = \log \frac{\text{the overall count of the documents}}{\text{quantity of documents with the term}} \quad (2)$$

Finally, the TF-IDF score is calculated by Eq. (3)

$$TF - IDF = TF \times IDF \quad (3)$$

3.4 Classification using Ensemble deep learning models

The primary area of this observe is to increase a way for categorizing hazardous content in textual content data through employing CNN, RNN, and Bi-LSTM structure as part of the Ensemble deep getting to know structure. The primary process that exhibits greater performance as compared to current approaches is text categorization. Ensemble frameworks, which combine CNNs, RNNs, and Bi-LSTM networks, offer a comprehensive method for tackling difficult natural language processing (NLP) challenges. CNNs specialize in recognizing particular, localized patterns and features in textual input, making them ideal for tasks like sentiment analysis and text categorization. However, in applications like named things identification, machine translation, and language modeling, RNNs and Bi-LSTMs are specifically intended for modeling sequential data and capturing persistent dependencies.

3.4.1 *Convolutional Neural Network*

Within the field of deep learning architectures, CNNs are a subset designed specifically to handle and analyze visual data, especially images. They may also be utilized for efficient processing of consecutive data arrangements, such as time series, text, and audio spectrograms. This provides them with a broad range of applications in a variety of fields, including but not limited to NLP.

Convolutional Layers: To excerpt features from incoming data, these layers apply convolution processes. When convolutions are applied to images, patterns like textures, edges, and forms are identified by swiping tiny filters, or kernels, across input image. To obtain n-gram features and local patterns from text input, 1D convolutions can be applied.

Pooling Layers: Convolutional layers provide feature maps, which are then utilized to down-sample their spatial dimensions. Common pooling methods like average pooling and max pooling minimize feature map sizes without sacrificing crucial information, increasing computational effectiveness and lowering overfitting.

Fully Connected Layers: These layers are used for classification or regression tasks after convolutional and pooling layers have extracted information. By establishing connections between each neuron in preceding layer and each neuron in subsequent layer, these layers enable network to understand intricate relationships and provide predictions by utilizing the properties that have been retrieved.

3.4.2 *Recurrent Neural Network*

RNN is a form of Neural Network (NN) where output from CNN is suckled as an input to current phase. It plays a critical function in text classification because of its sequential processing functionality, which traditional neural networks lack. In traditional NN, all of inputs and outputs are unbiased of each different. Nonetheless, there may be a necessity to recall words, came before in circumstances where it's essential to forecast sentence's following phrase. Thus, RNN was developed and with the help of a Hidden Layer, it resolved this issue. The maximum critical feature of an RNN is its hidden state, which lets in it to do not forget precise information about a series. This hidden layer acts as a shape of recollection, permitting the network to recall and employ past facts during modern-day computations. It makes it viable for the model to maintain context during the textual content, which makes it nicely-desirable for duties regarding sequences, like sentiment evaluation, device translation, and language modeling. It generates the output by way of the use of the identical parameters for every input to look like a same project on each input or hidden layer. This reduces the complexity of parameters, in evaluation to distinct neural networks. This parameter sharing reduces the model's complexity in comparison to standard feedforward networks, which require separate parameters for every entry.

3.4.3 *Bidirectional Long-Short Term Memory*

BiLSTM is a series model that integrates two LSTM layers, one for forward and backward input processing. One of the key packages of BiLSTM in the area of NLP is the enhancement of collection know-how thinking about each the previous and following contexts. The twin-directional processing allows the model to seize elaborate sequence relationships, important for responsibilities like language modeling or sentiment evaluation. BiLSTM is specifically useful for this form of gaining knowledge of, which entails developing the capacity to study dependencies sequentially. Its use in NLP demonstrates how well it can cope with textual content information by using completely expertise the subtleties of sequential facts interior textual contexts.

CNN with Batch Normalization and MaxPooling layers processes the input features. RNN with a Bi-LSTM layer comes next. Output of these layers is then compressed and routed through several Dense layers that are absolutely connected. Output of ultimate Dense layer is generated the use of sigmoid activation characteristic. Figure 4 demonstrates the architecture of Ensembled DL models (CNN, RNN, and Bi-LSTM)

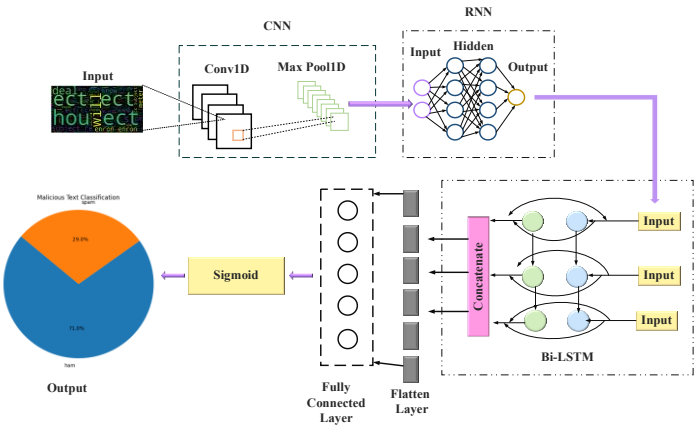

Figure 4: Architecture of Ensembled DL Models (CNN, RNN, and Bi-LSTM)

4. RESULT AND DISCUSSION

An ensemble of DL models (CNN, RNN, and Bi-LSTM) for text classification is presented in this paper. Execution of recommended method is assessed using a variety of metrics, including Balanced Accuracy, Brier Score Loss, Precision, Accuracy, Recall, Mean Average Precision (MAP), and F-measure. The recently constructed framework is evaluated in terms of how much its performance has increased by comparing it with other models, such as Proposed, CNN [17], SVM [19], NB [21], DT [23], and Ensembled RNN [25].

4.1 Evaluation Setup

The recommended framework has been implemented in the PYTHON platform. The Spam Mail Dataset (https://www.kaggle.com/datasets/venky73/spam-mails-dataset) has been used to assess the proposed framework. A learning rate of 70% and 80% was utilized.

4.2 Performance Metrics with existing works

Table 4: Comparative analysis of the performance metrics with existing models-Learning rate 70%

	CNN [17]	**SVM [19]**	**Naive Bayes [21]**	**D-Tree [23]**	**Ensemble RNN [25]**	**PROPOSED**
Precision	0.885947	0.842829	0.832669	0.782692	0.929936	0.986784
Accuracy	0.950387	0.931057	0.921392	0.895619	0.967139	0.990979
F-Measure	0.918691	0.889119	0.872651	0.834016	0.944984	0.984615
Recall	0.953947	0.940789	0.916667	0.892544	0.960526	0.982456

MAP	0.858678	0.810322	0.787765	0.730159	0.904826	0.974627
Balanced Accuracy	0.951426	0.933898	0.920012	0.894721	0.965208	0.988491
Brier Score Loss	0.049613	0.068943	0.078608	0.104381	0.032861	0.009021

A detailed comparison of the suggested model for malicious text categorization with a learning rate of 70% in natural language processing, along with other classification models like CNN, Naive Bayes, SVM, Decision Tree, and Ensemble RNN, is given in Table 4. Important indicators including Accuracy, Precision, Recall, F-Measure, MAP, Balanced Accuracy, and Brier Score Loss are used to assess each model's performance. Comparing the suggested model to the other models assessed, it is noted that it shows greater accuracy (0.990979), F-Measure (0.984615), precision (0.986784), MAP (0.974627), recall (0.982456), Balanced Accuracy (0.988491), and lowest Brier Score Loss (0.009021). These findings demonstrate the ensemble deep learning-based approach's resilience and effectiveness in handling text classification problems, especially when it comes to spotting harmful content. Figure 6 shows the Validation loss and validation accuracy for the learning rate 70%.

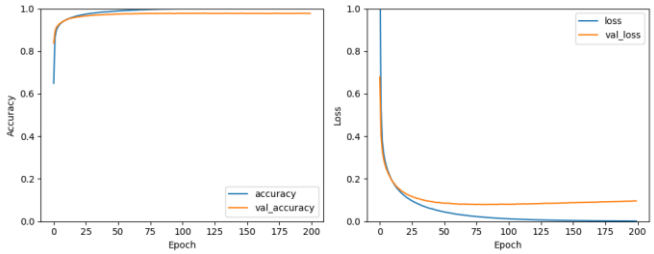

Figure 6: Validation loss and validation accuracy for learning rate 70%

Table 5: Comparative analysis of the performance metrics with existing models-Learning rate 80%

	CNN [17]	**SVM [19]**	**Naive Bayes [21]**	**D-Tree [23]**	**Ensemble RNN [25]**	**PROPOSED**
Precision	0.894895	0.876106	0.862573	0.823708	0.943396	0.990323
Accuracy	0.956522	0.948792	0.942029	0.908213	0.974879	0.996135
F-Measure	0.929797	0.918083	0.907692	0.850863	0.958466	0.993528
Recall	0.967532	0.964286	0.957792	0.87987	0.974026	0.996753
MAP	0.875502	0.855445	0.838726	0.760505	0.926622	0.988073
Balanced Accuracy	0.959695	0.953257	0.946571	0.900045	0.974633	0.996313
Brier Score Loss	0.043478	0.051208	0.057971	0.091787	0.025121	0.003865

Table 5 presents a thorough analysis, based on different evaluation criteria for malicious text categorization in natural language processing, of six classification models with a learning rate of 80%: CNN, SVM, Naive Bayes, Decision Tree (D-Tree), Ensemble RNN, and a new model. Of all the models, it achieves the best results in terms of Precision (0.990323), Accuracy (0.996135), F-Measure (0.993528), Recall (0.996753), Balanced Accuracy (0.996313), and Brier Score Loss (0.003865). These findings demonstrate the robustness, efficacy, and superior prediction powers of the suggested model in correctly categorizing hazardous text, highlighting its potential for useful implementation in real-world natural language processing tasks. Figure 7 shows the Validation loss and validation accuracy for the learning rate 80%.

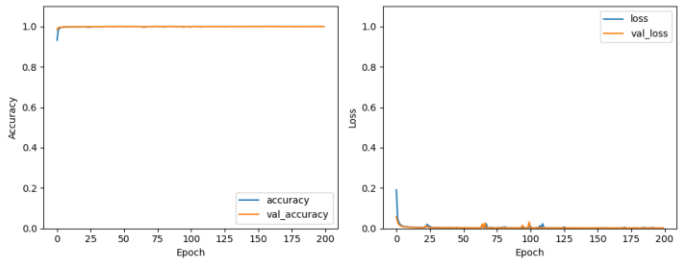

Figure 7: Validation Loss and validation accuracy for Learning rate of 80%

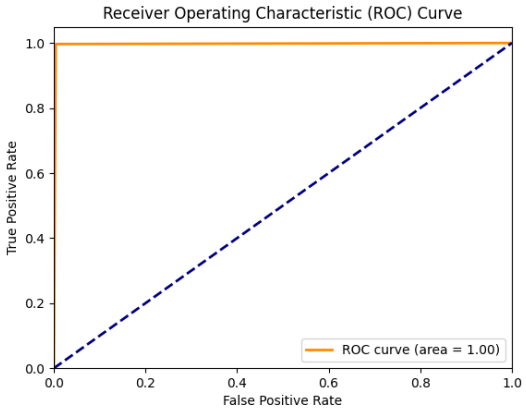

Figure 10: ROC Curve

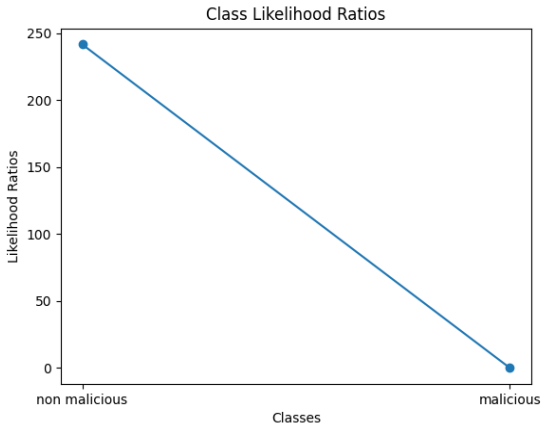

Figure 11: Ratio between Non-malicious and Malicious

The figures above represent the ROC Curve in Figure 10, the ratio between malicious and non-malicious in Figure 11.

5. CONCLUSION

Text classification played a crucial role as a preprocessing step in NLP, especially for tasks like internet spam detection. Text categorization was a preprocessing technique that required several preprocessing stages to fine-tune the textual input. Lowercasing, and eliminating web URLs, digits, punctuation, and stop words were all included in this. Next, to assist in producing significant features from the dataset, the tokenization and stemming methods were applied. Subsequently, the dataset was processed using characteristic feature extraction strategies like BoW, TF-IDF, and word2vec to supply significant features. Ensemble DL methods including Bi-LSTM, RNN, and CNN were integrated to enhance the categorization technique, mainly when it came to detecting harmful content. The

performance was executed by the PYTHON, and Experimental results show accuracy rate proposed with learning rate of 70% was 99% and the learning rate of 80% was 99.61%.

References

1. Chowdhary, K. and Chowdhary, K.R., 2020. Natural language processing. Fundamentals of artificial intelligence, pp.603-649.
2. Umer, M., Imtiaz, Z., Ahmad, M., Nappi, M., Medaglia, C., Choi, G.S. and Mehmood, A., 2023. Impact of convolutional neural network and FastText embedding on text classification. Multimedia Tools and Applications, 82(4), pp.5569-5585.
3. Duan, L., You, Q., Wu, X. and Sun, J., 2022. Multilabel text classification algorithm based on fusion of two-stream transformer. Electronics, 11(14), p.2138.
4. Tan, Z., Chen, J., Kang, Q., Zhou, M., Abusorrah, A. and Sedraoui, K., 2021. Dynamic embedding projection-gated convolutional neural networks for text classification. IEEE Transactions on Neural Networks and Learning Systems, 33(3), pp.973-982.
5. Balyan, R., McCarthy, K.S. and McNamara, D.S., 2020. Applying natural language processing and hierarchical machine learning approaches to text difficulty classification. International Journal of Artificial Intelligence in Education, 30(3), pp.337-370.
6. Jang, J., Kim, Y., Choi, K. and Suh, S., 2021. Sequential targeting: a continual learning approach for data imbalance in text classification. Expert Systems with Applications, 179, p.115067.
7. Mohammed, A. and Kora, R., 2022. An effective ensemble deep learning framework for text classification. Journal of King Saud University-Computer and Information Sciences, 34(10), pp.8825-8837
8. Li, H. and Li, Z., 2022. Text classification based on machine learning and natural language processing algorithms. Wireless Communications and Mobile Computing, 2022.
9. Trappey, A.J., Trappey, C.V., Wu, J.L. and Wang, J.W., 2020. Intelligent compilation of patent summaries using machine learning and natural language processing techniques. Advanced Engineering Informatics, 43, p.101027.
10. Rathnayake, H., Sumanapala, J., Rukshani, R. and Ranathunga, S., 2022. Adapter-based fine-tuning of pre-trained multilingual language models for code-mixed and code-switched text classification. Knowledge and Information Systems, 64(7), pp.1937-1966.
11. Gücükbel, E., 2023. Evaluating The Explanation of Black Box Decision for Text Classification.
12. Luo, X., 2021. Efficient English text classification using selected machine learning techniques. Alexandria Engineering Journal, 60(3), pp.3401-3409.
13. Van Landeghem, J., Blaschko, M., Anckaert, B. and Moens, M.F., 2022. Benchmarking scalable predictive uncertainty in text classification. Ieee Access, 10, pp.43703-43737.
14. Ameer, I., Bölücü, N., Siddiqui, M.H.F., Can, B., Sidorov, G. and Gelbukh, A., 2023. Multi-label emotion classification in texts using transfer learning. Expert Systems with Applications, 213, p.118534.
15. Carnot, M.L., Bernardino, J., Laranjeiro, N. and Gonçalo Oliveira, H., 2020. Applying text analytics for studying research trends in dependability. Entropy, 22(11), p.1303.
16. Alsaleh, D. and Larabi-Marie-Sainte, S., 2021. Arabic text classification using convolutional neural network and genetic algorithms. IEEE Access, 9, pp.91670-91685.
17. Ce, P. and Tie, B., 2020. An analysis method for interpretability of CNN text classification model. Future Internet, 12(12), p.228.
18. Gong, J., Teng, Z., Teng, Q., Zhang, H., Du, L., Chen, S., Bhuiyan, M.Z.A., Li, J., Liu, M. and Ma, H., 2020. Hierarchical graph transformer-based deep learning model for large-scale multi-label text classification. IEEE Access, 8, pp.30885-30896.
19. Zhang, M., 2024. Ensemble-Based Text Classification for Spam Detection. Informatica, 48(6).
20. Behera, B. and Kumaravelan, G., 2021. Text document classification using fuzzy rough set based on robust nearest neighbor (FRS-RNN). Soft Computing, 25(15), pp.9915-9923.
21. Gan, S., Shao, S., Chen, L., Yu, L. and Jiang, L., 2021. Adapting hidden naive Bayes for text classification. Mathematics, 9(19), p.2378.
22. Pawar, S., Rao, M.G. and Pandith, K., 2023. Text document categorisation using random forest and C4. 5 decision tree classifier. International Journal of Computational Systems Engineering, 7(2-4), pp.211-220.
23. Zheng, Y., Gao, Z., Shen, J. and Zhai, X., 2022. Optimizing Automatic Text Classification Approach in Adaptive Online Collaborative Discussion–A Perspective of Attention Mechanism-Based Bi-LSTM. IEEE Transactions on Learning Technologies, 16(5), pp.591-602.
24. Ahmad, H., Asghar, M.U., Asghar, M.Z., Khan, A. and Mosavi, A.H., 2021. A hybrid deep learning technique for personality trait classification from text. IEEE Access, 9, pp.146214-146232
25. Zhang, W., Chen, Q. and Chen, Y., 2020. Deep learning based robust text classification method via virtual adversarial training. IEEE Access, 8, pp.61174-61182.