Multidisciplinar (Montevideo). 2025; 3:44
doi: 10.62486/agmu202544
ISSN: 3046-4064

ORIGINAL

Enhancing the Optimization of BI-LSTM Classifier with Ensemble Methods, Regularization, and Cross-Validation Techniques for Email Spam Detection

Mejora de la optimización del clasificador BI-LSTM con métodos conjuntos, regularización y técnicas de validación cruzada para la detección de spam de correo electrónico

Arepalli Gopi[1], Sudha L.R[2], Iwin Thanakumar Joseph S[3]

[1]Research Scholar, Department of Computer Science & Engineering, Annamalai University. Chidambaram, Tamilnadu, India.
[2]Associate Professor, Department of Computer Science & Engineering, Annamalai University. Chidambaram, Tamilnadu, India.
[3]Associate Professor, Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation. Vaddeswaram, India.

Cite as: Gopi A, Sudha L, Iwin Thanakumar JS. Enhancing the Optimization of BI-LSTM Classifier with Ensemble Methods, Regularization, and Cross-Validation Techniques for Email Spam Detection. Multidisciplinar (Montevideo). 2025; 3:44. https://doi.org/10.62486/agmu202544

Submitted: 20-03-2024 **Revised:** 22-07-2024 **Accepted:** 01-11-2024 **Published:** 01-01-2025

Editor: Prof. Dr. Javier Gonzalez-Argote

Corresponding author: Arepalli Gopi

ABSTRACT

Email spam, a persistent and escalating issue, continues to disrupt the digital communication landscape, causing inconvenience and time loss for users worldwide. With technological advancements, spammers continually adapt and refine their tactics to infiltrate email inboxes. Staying current with state-of-the-art anti-spam techniques is imperative to secure emails and eliminate unwanted messages. Our research work embarks on an exploration of supercharging email spam detection through the augmentation of a Bidirectional Long Short-Term Memory (BI-LSTM) classifier. Our approach integrates ensemble methods, regularization techniques, and cross-validation into the fabric of the BI-LSTM model, creating a formidable spam detection system. Our paper delves into the intricate technical aspects of these methodologies, elucidating their synergy in fortifying the classifier's performance.

Keywords: Machine Learning; Bi LSTM; LSTM; Sigmoid; Optimization; Regularization; K-Cross Fold; Ensemble.

RESUMEN

El spam por correo electrónico, un problema persistente y creciente, sigue perturbando el panorama de la comunicación digital, causando molestias y pérdidas de tiempo a usuarios de todo el mundo. Con los avances tecnológicos, los spammers adaptan y perfeccionan continuamente sus tácticas para infiltrarse en las bandejas de entrada de los correos electrónicos. Estar al día de las técnicas antispam más avanzadas es imprescindible para proteger los correos electrónicos y eliminar los mensajes no deseados. Nuestro trabajo de investigación explora cómo mejorar la detección de spam mediante el aumento de un clasificador de memoria bidireccional a corto plazo (BI-LSTM). Nuestro enfoque integra métodos de ensemble, técnicas de regularización y validación cruzada en la estructura del modelo BI-LSTM, creando un formidable sistema de detección de spam. Nuestro artículo profundiza en los intrincados aspectos técnicos de estas metodologías, elucidando su sinergia a la hora de fortificar el rendimiento del clasificador.

Palabras clave: Aprendizaje Automático; Bi LSTM; LSTM; Sigmoide; Optimización; Regularización; K-Cross Fold; Ensemble.

INTRODUCTION

In today's digital age, where communication heavily relies on emails, the prevalence of spam emails has become a significant concern for individuals and businesses alike. Spam emails not only clog up inboxes but also pose various security threats, such as phishing attempts, malware distribution, and identity theft. To combat this ever-growing problem, email spam detection systems have been developed to filter out unwanted and potentially harmful emails. These systems employ sophisticated algorithms and machine learning techniques to accurately recognize and group messages as either genuine or spam.

The focus of this research work is to introduce a powerful approach to email spam detection, using a combination of techniques including ensemble methods, regularization, and cross-validation. By enhancing the performance of the BI-LSTM (Bi-directional Long Short-Term Memory) classifier, we aim to supercharge the accuracy and efficiency of spam detection systems mail spam detection is a challenging task due to the constantly evolving nature of spamming techniques. Spammers employ various tactics to bypass traditional filters and deceive users. Therefore, it is crucial to develop advanced algorithms that can adapt and stay ahead of these malicious activities.

Throughout this research work, we will delve into the details of the BI-LSTM classifier and explain how it can effectively analyze the textual content of emails, considering the sequential nature of the data. Additionally, we will explore ensemble methods, which combine the predictions of multiple classifiers to improve overall accuracy. Regularization techniques will also be discussed as a means to prevent overfitting and improve generalization capabilities. Cross-validation, on the other hand, will enable us to assess the performance of our classifier on unseen data, ensuring its reliability and robustness.

By leveraging these techniques and combining them into a unified approach, we can empower email spam detection systems to provide enhanced protection against unwanted and potentially harmful emails. We explore the intricacies of supercharging email spam detection in the digital age, addressing the multifaceted challenges posed by ever-evolving spamming techniques.

Shaik China Moulali et al.(10), done research on machine learning and DL classifiers, such as SVM, Random Forests, Decision Trees, NB, LSTM, & Bi-LSTM. They used the spam.csv dataset to detect spam emails. By implementing all the classifiers, they said that Bi-LSTM is performing more accurately than all other ML classifiers for email spam detection.

According to Lanka Sai Charan et al.(4), they implemented ML procedures like NB, SVM, Decision trees, Random Forest, Boosting, and Bagging to filter spam and ham emails. By using all the proposed ML algorithms, they concluded that Naïve Bayes is best among all-proposed algorithms to detect the spam mails.

According to Adnan Muhammad et al.(2), so they used ML algorithms such as logistic regression, decision tree, AdaBoost, GNB, and KNN classifiers. In their work they used all the ml approaches and later they formed a Stacking Meta Classifier. They conclude that the stacking method is a robust and effective approach for spam email classification. It consistently outperforms individual classifiers, particularly when diverse base classifiers are combined. The choice of base classifier combinations should be made considering both performance and computational efficiency.

As indicated by Agarwal Deepak Kumar et al.(3), they utilized one of the most well-known AI Calculation support vector machines (SVM) with various boundaries utilizing different bit capabilities (direct, polynomial, RBF, sigmoid) are carried out on spambase-dataset. Furthermore, they done Examination of SVM execution for all portions (direct, polynomial, RBF, sigmoid) utilizing various boundaries (C-SVC, NU-SVC) assessed on spambase-dataset to get best precision.

As per Shuaib Marayam et al.(7), the research introduces the Whale Optimization Algorithm (WOA) for feature selection and the Rotation Forest algorithm for spam classification. By applying WOA, the Rotation Forest algorithm achieved an impressive 99,9 % accuracy and a minimal false positive rate of 0,0019, surpassing previous methods significantly.

According to H Bhuiyan et al.(14), they used several algorithms such as SVM, NB, Bayes Additive Regression, Decision Tree, K-Nearest Neighbour. They done research on categorizing, evaluating, and comparing different systems in order to summarize their overall accuracy rates.

As per Salloum Said et al.(9), They discusses the evolving landscape of phishing detection, emphasizing the use of natural language processing (NLP) techniques. They highlights the prevalence of machine learning algorithms, the need for semantic analysis in email body text, and the necessity of adaptable evaluation plans to address the dynamic nature of cyber threats.

In a study by Al-Rawashdeh et al.(6), they aimed to improve feature selection precision by combining the Water Cycle Algorithm (WCA) and Simulated Annealing for spam detection. They evaluated the accuracy and feature selection outcomes using WCA and compared them with other algorithms like PSO, HS, and GA. The results showed that the WCA-based method achieved slightly better accuracy when applied to the SPAM E BASE dataset.

According to Anshumaanmishra, et al.(12) ,they done research on Deep Learning algorithms for email spam detection. They used RNN,LSTM and Bi-LSTM in their work, and finally from the comparison of accuracy and AUC score of all three models they exclaimed that Bi-LSTM is more precise than RNN and LSTM classifiers.

METHOD

In this section, every detail regarding the Optimization techniques that we are going to use in this research work will be described.

Here we are going to optimize the Bi-LSTM for better performance in regards to Email Spam Detection.

The figure 1 shows the Flow chart for building a DL model.

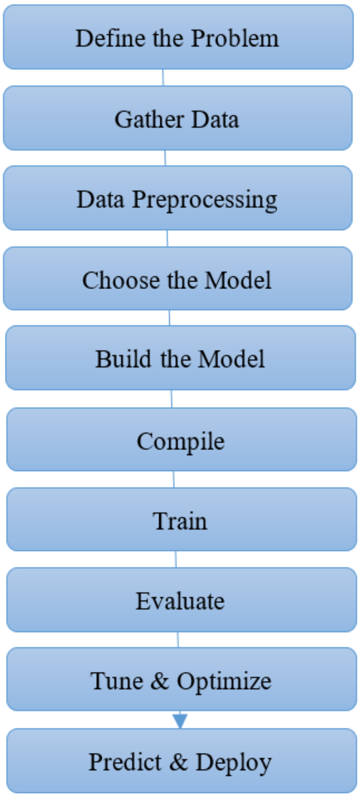

Figure 1. Flow Chart

Data Source

In our study, two datasets "spam_ham_dataset.csv"[17] and "spam.csv"[16] are used .These datasets are taken from Open source Kaggle website. As shown in Figure2 the dataset "spam_ham_dataset.csv" consists of 3672 Ham mails & 1499 Spam mails.

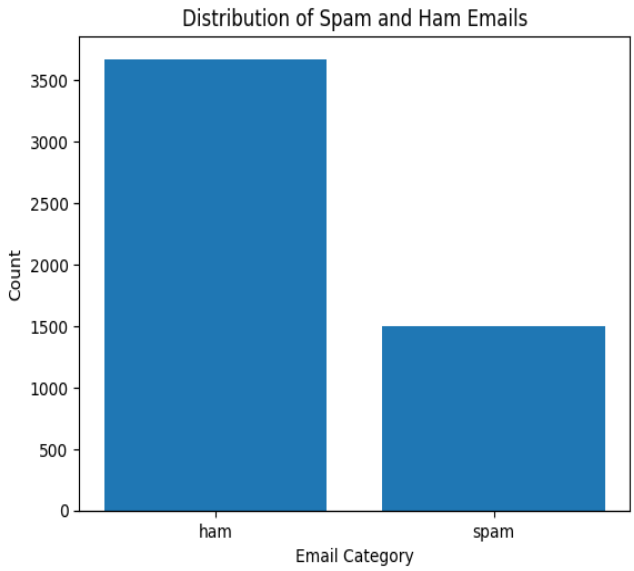

Figure 2. Spam_Ham_dataset count

https://doi.org/10.62486/agmu202544 ISSN: 3046-4064

In figure 3 the percentage of spam and ham mails in the dataset “spam_ham_dataset” is shown.

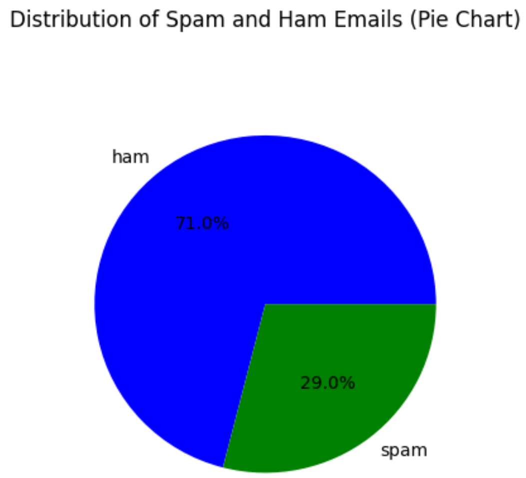

Figure 3. Percentage of Spam & Ham mails in dataset

spam_ham_dataset

As shown in Figure4 the dataset “spam.csv” consists of 4825 - “Spam ” and 747 - “Ham” mails.

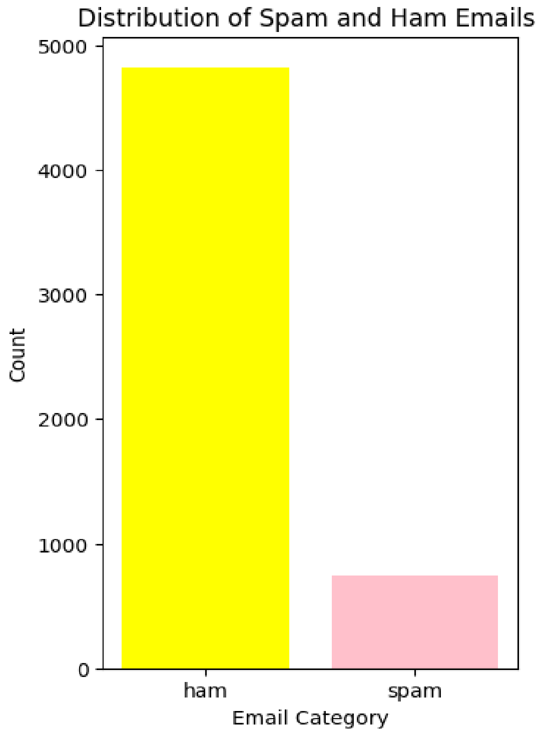

Figure 4. Number of Spam and Ham mails in dataset “Spam.csv”

Data Pre-Processing

The Data Preprocessing done in our research work follows the steps:

a. Data Loading and Character Encoding:

i) The dataset is loaded from a CSV file using pd.read_csv().

ii) The ‘ISO-8859-1’ encoding is specified to handle character encoding issues that may exist in the dataset.

b. Column Removal:

i) The unnamed columns are dropped using the data.drop() method. These columns appear to be extraneous or empty and are therefore removed.

c. Column Renaming:

i) The columns are renamed to provide more descriptive column names.

d. Label Encoding:

i) A new column ‘label_in_num’ is created based on the ‘label’ column. The values ‘ham’ are mapped to 0, and ‘spam’ are mapped to 1. This step encodes the text labels as numerical values, which is often necessary for binary classification tasks.

e. Tokenization

i) We tokenized the text data using the Keras Tokenizer and adjusted the vocabulary size as needed. We then padded the sequences to ensure they have a uniform length using the pad_ sequences function.

Bi-LSTM(Bi-Directional Long Short Term Memory)

The Bi-LSTM (Bi-directional Long Short Term Memory) classifier is a powerful tool used in email spam detection. It is a variant of the LSTM model, which is an sort of Recurrent Neural Network(RNN) that is capable of capturing long-term needs in sequential data. As shown in Figure5. The Bi-LSTM classifier takes advantage of the bidirectional nature of LSTM by processing the input sequence in both forward and backward directions, allowing it to gather information from past and future timesteps simultaneously.

One of the key advantages of the BI-LSTM classifier is its ability to effectively model the context and semantics of email text, making it well-suited for spam detection tasks. It leverages the sequential nature of emails, considering the order in which words and sentences appear to capture the nuances of spam messages. By analyzing the content of emails, including textual features such as subject lines, body text, and sender information, the Bi-LSTM classifier can recognize authentic messages and spam with high exactness.

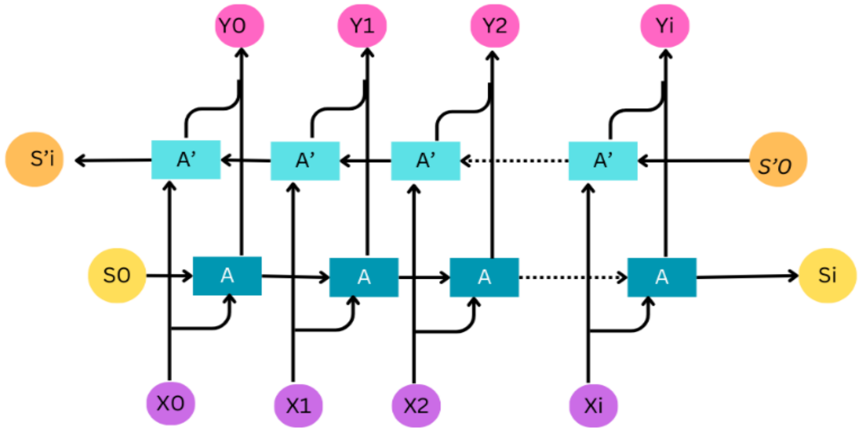

Figure 5. Bi-LSTM classifier

The Bi-LSTM classifier is trained on a large labeled dataset, where each email is labeled like either spam or ham. During training, the classifer learns to recognize patterns and features that are indicative of spam, such as specific keywords, excessive use of capital letters, or suspicious URLs. The model is optimized using backpropagation and gradient descent, adjusting the weights and biases of the neural network to minimize the classification error.

Once trained, the BI-LSTM classifier can be applied to new, unseen emails to arrange them as spam or ham. By utilizing the learned representations and patterns, the model can make predictions with high precision and recall. However, to further enhance the performance and robustness of the classifier, ensemble methods, regularization techniques, and cross-validation are often employed. These approaches help mitigate overfitting, improve generalization, and ensure the classifier performs well on diverse email datasets.

Optimization Techniques

Ensemble Learning

Ensemble methods have gained popularity in the field of machine learning for their ability to improve classification performance. In the context of email spam detection, they can be particularly effective in enhancing the accuracy and reliability of the classification process.

Ensemble techniques encompass the amalgamation of numerous foundational classifiers to create a more potent and resilient classifier.

This can be achieved through different techniques such as bagging, boosting, or stacking. Bagging, for

https://doi.org/10.62486/agmu202544 ISSN: 3046-4064

instance, works by training each base classifier on different subsets of the training data and then combining their predictions through majority voting or averaging. Boosting, on the other hand, focuses on iteratively training weak classifiers and assigning higher weights to misclassified instances in order to improve overall accuracy.

One of the main advantages of ensemble methods is their ability to reduce overfitting. By combining multiple classifiers, the ensemble is able to generalize better and make more accurate predictions on unseen data. And we used Bagging as part of Ensemble method in our research work.

Bagging **:** Bagging is an ensemble strategy that encompasses the training of multiple models on distinct subsets of the training data and then amalgamating their predictions.

For each ensemble member, randomly select a subset of your dataset with replacement. Each subset is called a "bag." And we trained Bi-LSTM on each bag, creating multiple Bi-LSTMs

Regularization

Regularization techniques play a crucial role in preventing overfitting when training machine learning models for email spam detection. Overfitting takes place when a model becomes excessively tailored to the training data, leading to a lack of effective generalization to new, unseen data.

L1 Regularization:

One popular regularization technique is L1 regularization, also known as Lasso regularization. It adds a penalty term to the loss function, encouraging the model to learn sparse weights by shrinking less important features towards zero. This helps in reducing the complexity of the model and prevents it from relying too heavily on specific features that may be present in the training data but not in the test data.

L2 Regularization:

Another regularization technique is L2 regularization, also known as Ridge regularization. Similar to L1 regularization, it adds a penalty term to the loss function. However, instead of promoting sparsity, it encourages the model to distribute the weights more evenly across all features. This can prevent the model from overemphasizing any particular feature and improve its generalization performance.

Cross -Validation

Cross-validation techniques play a crucial role in evaluating the performance of the email spam detection model. In the context of our enhanced BI-LSTM classifier, cross-validation allows us to assess the generalizability and robustness of the model.

K-cross Validation:

A widely employed cross-validation approach is k-fold cross-validation, which involves dividing the dataset into k equitably sized folds like in Figure6. The model undergoes training and assessment k times, with each fold taking a turn as the validation set while the remaining folds are dedicated to training. This method ensures that the model's performance remains consistent and is not influenced by any specific data partition.

As shown in Equation I, we can calculate the overall performance by averaging the performance achieved in each iteration: Iteration 1 Iteration 2 Iteration 3 Iteration 4

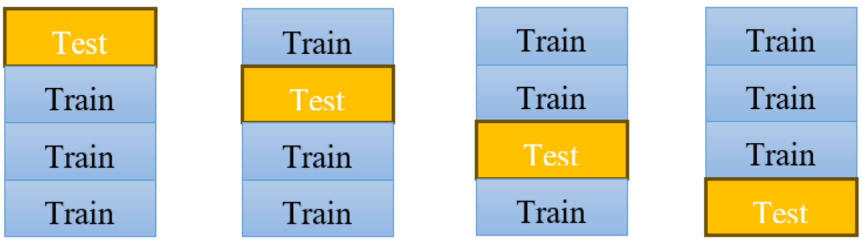

Figure 6. K-Cross Validation Iterations

$$\text{Overall Performance} = \frac{1}{k}\sum_{i=1}^{k} Perfomrmance_i \text{ ------(1)}$$

ALGORITHM 1

Algorithm: Enhancing Email Spam Detection with BI-LSTM, Ensemble, Regularization, and Cross-Validation.

Data Preprocessing

1.1 Load the email spam dataset (e.g., "spam.csv") with proper character encoding.

https://doi.org/10.62486/agmu202544 ISSN: 3046-4064

1.2 Clean the dataset by removing unwanted columns, if any.
1.3 Rename columns to 'label' and 'text' for clarity.
1.4 Encode text labels ('ham' and 'spam') into numerical values (0 and 1).

Split Data

2.1 Divide the dataset into separate training and testing subsets, with a common split ratio, such as allocating 80 % for training and 20 % for testing.

Tokenization and Padding

3.1 Tokenize the text data to convert it into numerical sequences.
3.2 Pad or truncate sequences to ensure uniform length (e.g., max_sequence_length).

BI-LSTM Model

4.1 Initialize a Bi-Directional Long Short-Term Memory (BI-LSTM) classifer.
4.2 Configure model architecture with embedding layers, LSTM layers, and output layer.
4.3 Compile the model with appropriate loss function, optimizer, and metrics.

Ensemble Learning

5.1 Implement ensemble methods (e.g., Bagging) by training multiple BI-LSTM models.
5.2 Train each BI-LSTM model on bootstrap samples of the training data.
5.3 Combine predictions from ensemble models (e.g., majority voting or weighted voting).

Regularization

6.1 Apply L1 and L2 regularization to the BI-LSTM models to mitigate overfitting.
6.2 Integrate regularization terms into the loss function during training.

Cross-Validation

7.1 Implement k-fold cross-validation for robust model evaluation.
7.2 Train and validate the model on different data subsets in each fold.
7.3 Calculate classification report for each fold.

Model Evaluation

1.2 Evaluate the ensemble BI-LSTM model on the testing dataset.
8.2 Assess its performance using appropriate evaluation metrics.

Experimental Setup

The experimental setup used in our study is Google Colab. The study employed two datasets, "spam.csv" and "spm_ham_dataset.csv," sourced from Kaggle. "spam.csv" contains labeled data for email spam classification, while "spm_ham_dataset.csv" is likely a dataset with spam and non-spam (ham) emails. For each dataset, data preprocessing tasks were performed. Hyperparameters, such as learning rates, batch sizes, and optimizer choices, were carefully selected to ensure optimal training performance for the BI-LSTM models. The ensemble methods, such as Bagging, were leveraged to improve the classification accuracy of the models. L1 and L2 regularization techniques were applied to the BI-LSTM models in both datasets. The k-fold cross-validation technique was applied to assess model performance for both datasets. All these optimization techniques are .applied on the Bi-LSTM classifier and they are trained and evaluated on both the datasets .And noted down the performance metrics of each optimization technique.

RESULTS AND DISCUSSIONS

The Bi-LSTM classifier is trained to get optimize using Ensemble learning,L1 Regularization,L2 Regularization and K-Cross validation. And among these optimization techniques K-Cross Validation performed well in optimization of Bi-LSTM classifier. As shown in Figure7.and Figure8. by using K-Cross validation we got best accuracy among remaining optimized techniques on both the datasets.

As shown in figure7. By using the "spam_ham_dataset.csv" the Standard Bi-LSTM classifier predicts an accuracy of 97,58 %, and the optimization techniques L1& L2 Regularization techniques estimated accuracy of 78,45 % & 96,13 % respectively. By using Ensemble Bagging it predicted an value of 97,29 % and the K-Cross validation predicted an accuracy of 98,24 % , which is best among the optimization techniques on Dataset "spam_ham_dataset.csv".

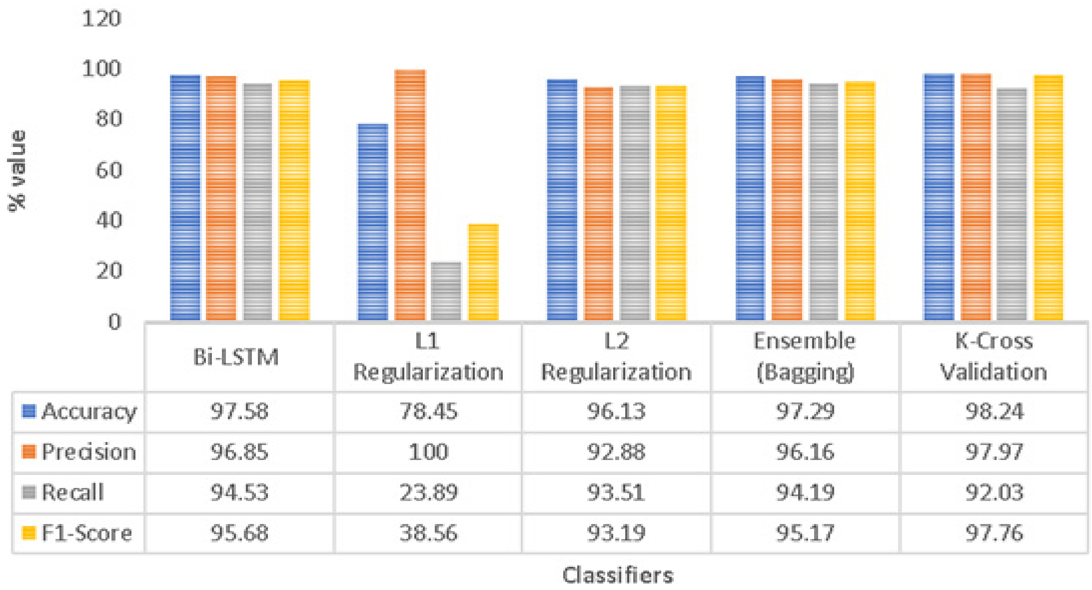

Figure 7. Performance metrics using spam_ham_dataset.csv

As shown in figure 8, by using the "spam.csv" the Standard Bi-LSTM classifier predicts an accuracy of 97,93 %. And the optimization techniques L1& L2 Regularization techniques estimated accuracy of 86,54 % & 98,11 % respectively. By using Ensemble Bagging it predicted an value of 98,29 % and the K-Cross validation predicted an accuracy of 98,63 %, which is best among the optimization techniques on Dataset "spam.csv".

Figure 8. Performance metrics using spam.csv

CONCLUSION

From the study in[10] the Bi-LSTM is outperformed well among ML and DL classifiers due to the Resides in its capacity to proficiently manage sequential data while comprehending dependencies in both the forward and backward directions. So, We have worked on Bi-LSTM to optimize it using optimization techniques. And from the Results[7,8] in both the datasets[16,17] the K-Cross validation technique is performed well to predict best accuracy.It optimized the Bi-LSTM classifier better than L1,L2 Regularization and Ensemble (Bagging) optimization techniques.

And finally from our study,the K-Cross validation Technique optimized the Standard Bi-LSTM classifier well among remaining optimization techniques by providing a robust evaluation.

And further research can be performed on hybrid model of Bi-LSTM and LSTM to gain better results. An hybrid model, formed using two best optimal models can be perform well to create marvelous results.

REFERENCES

1. Gupta, Vashu, Aman Mehta, Akshay Goel, Utkarsh Dixit, and Avinash Chandra Pandey. "Spam detection using ensemble learning." In Harmony Search and Nature Inspired Optimization Algorithms: Theory and Applications,

ICHSA 2018, pp. 661-668. Springer Singapore, 2019.

2. Adnan, Muhammad, Muhammad Osama Imam, Muhammad Furqan Javed, and Iqbal Murtza. “Improving spam email classification accuracy using ensemble techniques: a stacking approach.” International Journal of Information Security (2023): 1-13.

3. Agarwal, Deepak Kumar, and Rahul Kumar. “Spam filtering using SVM with different kernel functions.” International Journal of Computer Applications 136, no. 5 (2016): 16-23.

4. Lanka, Sai Charan, Kommana Akhila, Kodali Pujita, P. Vidya Sagar, Shayan Mondal, and Suneetha Bulla. “Spam based Email Identification and Detection using Machine Learning Techniques.” In 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), pp. 69-74. IEEE, 2023.

5. Bacanin, Nebojsa, Miodrag Zivkovic, Catalin Stoean, Milos Antonijevic, Stefana Janicijevic, Marko Sarac, and Ivana Strumberger. “Application of natural language processing and machine learning boosted with swarm intelligence for spam email filtering.” Mathematics 10, no. 22 (2022): 4173.

6. Al-Rawashdeh, Ghada, Rabiei Mamat, and Noor Hafhizah Binti Abd Rahim. “Hybrid water cycle optimization algorithm with simulated annealing for spam e-mail detection.” IEEE Access 7 (2019): 143721-143734.

7. Shuaib, Maryam, Shafi’I. Muhammad Abdulhamid, Olawale Surajudeen Adebayo, Oluwafemi Osho, Ismaila Idris, John K. Alhassan, and Nadim Rana. “Whale optimization algorithm-based email spam feature selection method using rotation forest algorithm for classification.” SN Applied Sciences 1 (2019): 1-17.

8. Karim, Asif, Sami Azam, Bharanidharan Shanmugam, Krishnan Kannoorpatti, and Mamoun Alazab. “A comprehensive survey for intelligent spam email detection.” IEEE Access 7 (2019): 168261-168295.

9. Salloum, Said, Tarek Gaber, Sunil Vadera, and Khaled Shaalan. “A systematic literature review on phishing email detection using natural language processing techniques.” IEEE Access 10 (2022): 65703-65727.

10. Shaik, China Moulali, Narasimha Murthy Penumaka, Suneel Kumar Abbireddy, Vinod Kumar, and S. S. Aravinth. “Bi-LSTM and Conventional Classifiers for Email Spam Filtering.” In 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), pp. 1350-1355. IEEE, 2023.

11. Abid, Muhammad Adeel, Saleem Ullah, Muhammad Abubakar Siddique, Muhammad Faheem Mushtaq, Wajdi Aljedaani, and Furqan Rustam. “Spam SMS filtering based on text features and supervised machine learning techniques.” Multimedia Tools and Applications 81, no. 28 (2022): 39853-39871.

12. Anshumaanmishra, & VigneshwaranPandi,. (2022). Classifications of E-MAIL SPAM Using Deep Learning Approaches. 10.3233/APC220058.

13. Kanmani, Sujithra, and Surendiran Balasubramanian. “Leveraging Readability and Sentiment in Spam Review Filtering Using Transformer Models.” Computer Systems Science & Engineering 45, no. 2 (2023).

14. Bhuiyan, H., Ashiquzzaman, A., Juthi, T. I., Biswas, S., & Ara, J. (2018). A survey of existing e-mail spam filtering methods considering machine learning techniques. Global Journal of Computer Science and Technology, 18(2), 20-29.

15. Kulkarni, Ritwik, Mercè Vintró, Stelios Kapetanakis, and Michele Sama. “Performance comparison of popular text vectorising models on multi-class email classification.” In Intelligent Systems and Applications: Proceedings of the 2018 Intelligent Systems Conference (IntelliSys) Volume 1, pp. 567-578. Springer International Publishing, 2019

16. https://www.kaggle.com/datasets/venky73/spam-mails-dataset/?select=spam_ham_dataset.csv

17. https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

FINANCING

The authors did not receive funding for the development of this research.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Arepalli Gopi, Sudha L.R, Iwin Thanakumar Joseph S.
Data curation: Arepalli Gopi, Sudha L.R, Iwin Thanakumar Joseph S.
Formal analysis: Arepalli Gopi, Sudha L.R, Iwin Thanakumar Joseph S.
Research: Arepalli Gopi, Sudha L.R, Iwin Thanakumar Joseph S.
Methodology: Arepalli Gopi, Sudha L.R, Iwin Thanakumar Joseph S.
Software: Arepalli Gopi, Sudha L.R, Iwin Thanakumar Joseph S.
Writing - original draft: Arepalli Gopi, Sudha L.R, Iwin Thanakumar Joseph S.
Writing - proofreading and editing: Arepalli Gopi, Sudha L.R, Iwin Thanakumar Joseph S.