A Deep Learning Model for Feature Extraction in Email Spam Classificiation with Machine Learning

Sza Sza Amulya Larasati1, a), Nathan Daud1, b), Diva Kurnianingtyas1, c), Alifi Lazuardi Gunawan1, d), Naufal Fathirachman Mahing1, e), and Irsyad Gema Adzani1, f)

1Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia, 651254

*c) Corresponding author:* [*divaku@ub.ac.id*](mailto:divaku@ub.ac.id)

*a)* [*szaszaal@student.ub.ac.id*](mailto:szaszaal@student.ub.ac.id)

*b)* [*nathandaud@student.ub.ac.id*](mailto:nathandaud@student.ub.ac.id)

*d)* [*alifilazuardi@student.ub.ac.id*](mailto:alifilazuardi@student.ub.ac.id)

*e)* [*naufalmahing@student.ub.ac.id*](mailto:naufalmahing@student.ub.ac.id)

*f)* [*rsyadsp10@student.ub.ac.id*](mailto:rsyadsp10@student.ub.ac.id)

**Abstract.** Spam is the delivery of content that is of no interest to the user and thus damages the user experience, usually accomplished via email. Spam is sent from one source to multiple targets to deceive, persuade, promote, and other negative purposes. This study aims to implement Deep Learning (DL) methods, namely Bidirectional Encoder Representations from Transformers (BERT) and Long Short-Term Memory (LSTM), to capture the meaning of words in whole sentences and detect repetitive or sequential patterns in email content as input for Machine Learning (ML) to help categorize email spam. The classification methods used are K-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), and Naive Bayes (NB). The dataset contains 6046 texts in the email body, divided into spam and non-spam classes. The results are that BERT consistently performs better than LSTM in all types of classifiers tested in the email spam classification task. Additionally, linear models such as LR showed the most optimal performance (0.965). In email spam classification, the BERT-LR approach is practical as it understands the complex nuances of language while precisely mapping important patterns to distinguish spam from ham emails.

# INTRODUCTION

In the modern digital era, email communication has become one of the main ways to exchange information, especially in the work environment. Email allows real-time communication between team members located in different locations [1], so that the decision-making process becomes faster and task completion can be done more efficiently [2]. Along with increasing reliance on email, the volume of email spam has also increased [3]. Email spam refers to electronic messages sent in bulk to many unsolicited recipients that often contain commercial advertisements, malicious links, phishing, and other intrusive content [4], [5]. These threats disrupt user productivity and pose serious cybersecurity risks, such as the theft of personal data and the spread of malware [6].

Email spam have taken on various forms and writing styles that are increasingly complex and adaptive. Traditional systems that rely solely on keyword-based filtering of spam messages have limitations in understanding the context and hidden meaning in the email text [3]. Natural Language Processing (NLP) approaches offer the advantage of understanding the human language patterns and context of email messages to identify the distinctive characteristics of spam messages. NLP techniques such as BERT and LSTM are widely applied to interpret textual content and effectively help identify whether an email is spam or legitimate. DL approaches offer superior context understanding compared to traditional methods like Term Frequency-Inverse Document Frequency (TF-IDF). BERT and LSTM stand out in extracting and understanding the semantic features of the text. These feature extraction results can be combined with ML algorithms as classifiers to improve the effectiveness of spam detection. This approach combines the context understanding power of DL models with the classification accuracy of ML models.

Numerous previous studies have explored different approaches for email spam detection, which can be categorized into three main methodological groups.

**Classical ML Approaches:** Several studies have demonstrated the effectiveness of traditional ML algorithms for spam detection. The study [7] developed a LR-based model emphasizing interpretability and computational efficiency, achieving an accuracy of 0.9935. Similarly, [8] proposed an email spam detection method using LR with TF-IDF feature extraction combined with bigrams, reaching an accuracy of 0.96. Other studies have also undertaken this comparison by evaluating several classifiers and yielding LR accuracies of up to 0.93 [9]. These studies highlight LR's relevance and power in addressing modern spam challenges while maintaining computational efficiency.

**Transformer-based Approaches:** Modern transformer models have shown superior performance in spam detection tasks. The study [10] conducted a comprehensive comparative analysis of various ML models (LR, SVM, RF, KNN, NB) alongside transformer models (BERT, DistilBERT, RoBERTa, XLNet). Their results showed LR achieving 0.984 accuracy, while BERT reached 0.988 accuracy. Another study [11] further evaluated the combination of traditional ML algorithms with BERT tokenization as feature extraction, where LR achieved up to 0.98 accuracy. These findings demonstrate the effectiveness of combining transformer-based feature extraction with classical classifiers.

**DL Sequential Models:** The study [12] investigated LSTM-based approaches for email classification, demonstrating that LSTM models can effectively understand sequential context in email text. Their experiments showed that LSTM achieved a validation accuracy of 0.9677 after data balancing, emphasizing LSTM's capability in capturing temporal dependencies in email content.

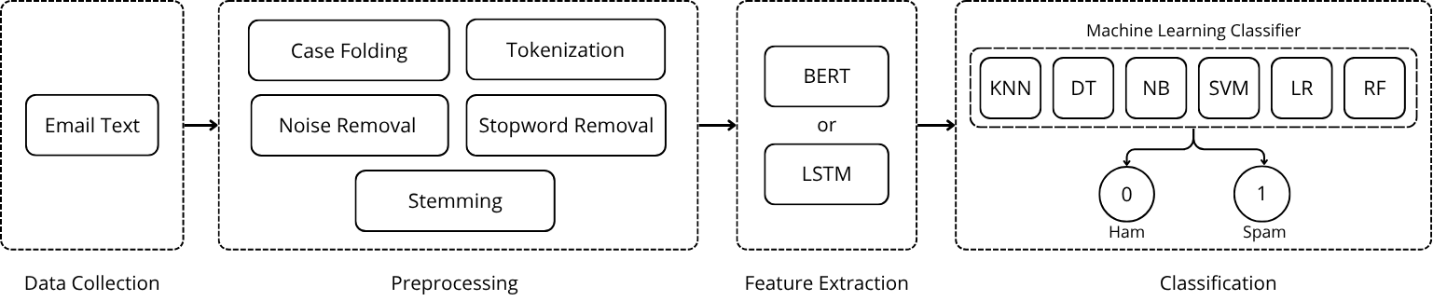
Although various previous studies have demonstrated the effectiveness of classical ML models such as KNN, DT, NB, LR, SVM, and RF in detecting email spam, most of these approaches still rely on traditional feature extraction techniques such as TF-IDF, n-grams, or metadata which have limitations in understanding the context and deep meaning of email text. On the other hand, DL models such as BERT are proven to excel in capturing context and word sequential relationships, but they are usually used in an end-to-end manner that requires large computational resources. In the meantime, no study has explicitly evaluated the effectiveness of BERT and LSTM as feature extractors whose results are used as inputs for classical ML models such as LR. To address these issues, this research proposes a combined approach that integrates LSTM and BERT models as feature extractors on email texts, which are subsequently used as inputs using ML algorithms. The selection of LSTM and BERT is based on empirical evidence from various studies showing that these two models are effective in handling text. BERT has the ability to understand semantic context in a bidirectional manner, while LSTM is reliable in capturing long-term dependencies in word sequences. This combination is expected to provide a more accurate representation of text to determine whether an email is spam or not.

# RELATED WORKS

Numerous previous studies have explored the use of classical ML algorithms for email spam detection. [7] developed a LR-based email spam detection model that emphasizes interpretability, computational efficiency, and high accuracy in email classification. The dataset used is processed through the stages of data cleaning, extraction of text features and metadata (including n-grams and word frequency), and regularization to avoid overfitting. The resulting accuracy reached 0.9935. In addition to predictive performance, this study also emphasizes the importance of model interpretation as features in keywords and email header attributes have a significant contribution to classification. This study shows that LR remains a relevant and powerful method in the face of modern spam challenges. [10] comparatively analyzed various ML models for email spam detection using LR, SVM, RF, KNN, and NB, as well as modern transformer models such as BERT, DistilBERT, RoBERTa, and XLNet. LR gives an accuracy of 0.984, while classification using BERT gives an accuracy of 0.988. Although LR has a slightly lower accuracy, it is much less computationally demanding due to its much smaller number of parameters than transformer-based DL models such as BERT. The LR model does not require multi-layered architecture, self-attention, or context embedding, making its training and inference process much faster and resource-efficient. Another study by [11] undertook a comprehensive evaluation of various ML algorithms using LR, SVM, NB, DT, and RF with BERT tokenization to perform feature extraction. The experimental results show that LR has an accuracy of 0.98. This study highlights the importance of combining textual features and metadata to improve the effectiveness of spam detection systems and provides an overview of the performance comparison of LR with other algorithms. [8] proposed an email spam detection method using the LR algorithm and the TF-IDF feature extraction technique equipped with bigrams to capture broader word context. Evaluation results show that LR achieves a training accuracy of 0.98 with a testing accuracy of 0.96. This approach offers a combination of computational efficiency and model interpretability, which makes it suitable for use as a real-world email filter system. In spite of BERT, [12] research proposed an email classification approach using LSTM. Experiments show that the LSTM model trained and tested on imbalanced data, without data balancing, still produces a validation accuracy of 0.8211, while after balancing, it increases greatly to 0.9677. This study emphasizes that LSTM can understand the sequential context of email text properly.

# MATERIAL AND METHOD

This section describes the stages and approach of a DL method designed to identify patterns in email content to automatically distinguish between spam and non-spam messages. This research consists of three main stages, namely data collection, data preprocessing, and classification. The detailed flow of the proposed method is shown in Figure 1.



**Figure 1.** Proposed Email Spam Detection Methodology Framework Showing the Complete Workflow from Data Collection Through Preprocessing to Classification Using BERT/LSTM Feature Extraction

## Data Collection

The dataset used in this study was retrieved from Kaggle, specifically from the "Email Spam Dataset” published by [13]. The dataset contains a total of 6,046 entries. The labels in the data set are binary: spam (1) represents unwanted, unsolicited emails, and ham (0) refers to legitimate emails.

## Preprocessing

Several text normalization techniques are used sequentially to clean up and ensure the consistency of the input data. Three examples of input data are presented in Table 1. First, case folding is performed by converting all characters in the email body to lowercase to ensure uniformity and reduce redundancy caused by letter sensitivity. Next, tokenization is applied to split the text into words. The email bodies in this dataset have some elements that are irrelevant in text semantics, such as punctuation marks, square brackets, etc. hyperlinks, HTML tags, newline characters (\n), and words containing numeric characters. These elements are removed in the noise removal stage, followed by removing stopwords (e.g. “the”, ‘is’, and “and”) that do not contribute to the semantic meaning using the NLTK library. In the last stage, stemming is implemented to reduce words to their basic form so that semantically similar words are treated as a single term. This normalization technique results in the preprocessing of the data shown in Table 2.

**TABLE 1.** Example of Input Data

|  |  |
| --- | --- |
| **Document ID** | **Text** |
| 1 | I just had to jump in here as Carbonara is one of my favourites |
| 2 | Can someone explain what type of operating system Solaris |
| 3 | My FREE Media Sofware linkenjoy |

**TABLE 2.** Result of Preprocessing

|  |  |  |
| --- | --- | --- |
| **Document ID** | **Text** | **Class** |
| 1 | jump carbonara one favourit | 0 |
| 2 | someon explain type | 0 |
| 3 | free media sofwar | 1 |

## Classification

The dataset was divided into 80% of the data as a training set and the remaining 20% data set aside for testing. The architecture consists of two main DL models: LSTM [14] and BERT [15]. Each architecture is equipped with six additional ML-based classifiers: KNN [16], DT [17], NB [18], SVM [19], LR [20], and RF [21]. This process is undertaken using a hybrid approach, where the feature representation of the email text is first extracted through LSTM and BERT, then passed on to the classifiers for the final classification stage. Detailed parameters for each classifier are shown in Table 3. The selection of this combination aims to explore the performance of each classifier in processing feature representations from DL models that have different characteristics: LSTM is advantageous in handling sequential data, while BERT is effective in understanding the global context between words.

**TABLE 3.** Example of Input Data

|  |  |
| --- | --- |
| **Classifier** | **Hyperparameter Value** |
| KNN | n\_neighbors = 5 |
| DT | criterion = gini |
| NB | priors = None |
| SVM | kernel = rbf; probability = True |
| LR | max\_iter = 1000 |
| RF | n\_estimators = 100 |

LSTM and BERT run over the same dataset with the same parameters for a fair performance comparison. BERT performs feature extraction by understanding the bidirectional context of words in a sentence using a transformer architecture. Initially, tokenization is executed using WordPiece, then embedding is added with positional information. Each token in the sentence is processed by multiple layers of transformers with a self-attention mechanism to capture the relation between words. The final output of BERT is a vector representation of each token or of the whole sentence that represents the contextual features of the text. Unlike BERT, LSTM is designed to handle data sequences. LSTM has a short-term and long-term memory structure through cell state and gating mechanism (input, forget, and output gate). Once text is input as word sequences, the LSTM processes the data sequentially and stores important information through the input gate, discarding irrelevant information at the forget gate. The output of LSTM is a vector representation that captures the temporal dependencies between words in a sentence.

Once the feature representation is obtained from BERT or LSTM, the vector is sent to various traditional classifiers: KNN (based on the label majority of the k nearest neighbors in the feature space), DT (based on the most informative features), NB (assumes independence between features and calculates class probabilities), SVM (find the best hyperplane with maximum margin), LR (the probability of a class with a logistic function on a one-vs-rest basis), and RF (ensemble model of many DT through voting of the entire tree for final prediction).

## Evaluation Metric

Model performance is evaluated using a confusion matrix that assesses the extent to which a model can correctly classify data based on its original label [22]. The prediction results in the confusion matrix are defined through four matrices: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). The classification report provides four evaluation metrics: accuracy, F1-score, precision, and recall. These four metrics are defined by the equation (1) – (4).

(1)

(2)

(3)

(4)

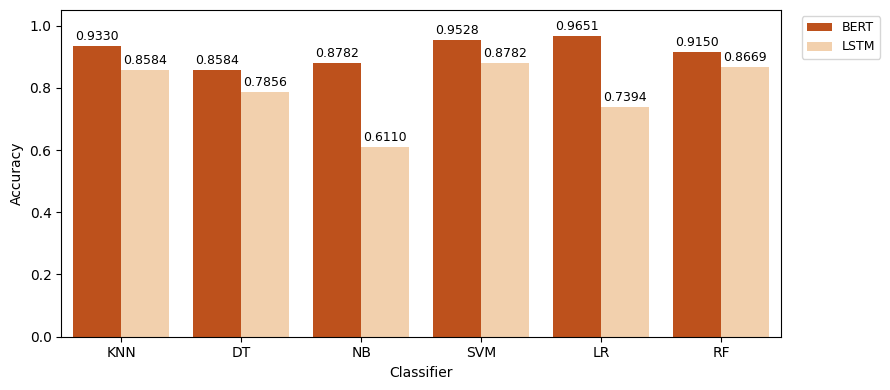
# RESULT AND DISCUSSION

Frequently occurring words in both spam and ham emails are visualized through the Word Cloud in Figure 2. In the spam Word Cloud (a), words such as click, remove, email, free, and credit have a high frequency of occurrence. These words indicate the common characteristics of email spam that often contain calls to click on links, free offers, or requests for removal from certain lists. Meanwhile, in Word Cloud ham (b), words such as date, url, linux, use, and file are more dominant. Additionally, these words indicate that the content of emails is informative, technical, or related to daily professional activities. These vocabulary differences support the analysis that there are quite different linguistic patterns between spam and ham.

|  |  |
| --- | --- |
|  |  |
| (a) | (b) |

**Figure 2.** Word Cloud Visualization Showing Frequently Occurring Words in (a) Spam Emails Featuring Terms like 'click', 'free', 'remove' and (b) Legitimate Emails with Technical Terms like 'date', 'linux', 'file'. Word Size Indicates Frequency of Occurrence.

Precise and contextual feature representation is paramount as email spam are often designed to disguise their original intent and avoid detection by automated systems. Based on the experimental results shown in Figure 3 BERT consistently provides higher accuracy compared to LSTM on all types of classifiers tested. The highest performance of BERT is exhibited in the LR classifier with an accuracy of 0.9651, while the lowest accuracy is still quite good, with an accuracy of 0.8584 in the DT. In contrast, the LSTM features resulted in a lower overall performance. The highest accuracy was recorded in RF at 0.8669, and the lowest accuracy in NB at 0.6110. These results show that the text representation generated by BERT improves the effectiveness of the classification process compared to LSTM. The Transformer-based BERT architecture has a bidirectional attention mechanism to understand the relationship between the entire word in a document and consider the context before and after each word at a time. This ability is very important in understanding email spam because the meaning of words can change depending on the context. For example, the word “free” in an informative sentence will have a different meaning if used in the context of a suspicious promotion. Meanwhile, LSTMs are designed to overcome the vanishing gradient problem of conventional RNNs and can store information in the long term; they still process text in a linear order (from left to right or vice versa). As a result, LSTMs often struggle to capture far-flung word relations or understand complex nuances in long sentences that are common in emails. This becomes a significant drawback for spam detection tasks, as certain patterns can be scattered in different parts of the text and do not always appear in sequence.



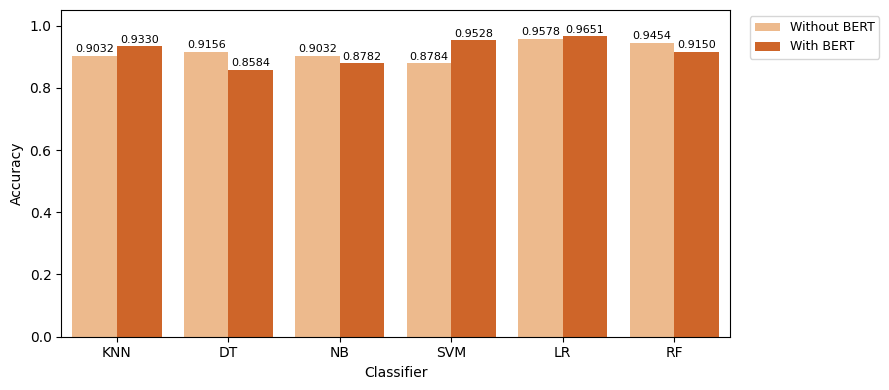
**Figure 3.** Performance Comparison: BERT vs LSTM Feature Extraction Across Different Classifiers

The variation in BERT accuracy shows that classifier selection still impacts classification effectiveness even when useful features are used. Details values of the accuracy, precision, recall, and F1-score for each BERT classifier are shown in Table 4. The best performing classifier is LR, with the highest accuracy of 0.965, followed by SVM with an accuracy of 0.953, and KNN, with an accuracy of 0.933. These results show that linear models such as LR are suitable for BERT's feature representation, which is structurally excellent and close to linearly separable conditions. The advantage of LR lies in its simplicity [8], which is an added value when used with BERT embedding because this model works optimally with informative input features and minimal noise. Despite its high accuracy, LR also showed excellent performance in precision, recall, and F1 score (0.9651 each). These results demonstrate LR's consistent ability to correctly classify spam and non-spam classes without any significant trade-off between precision and recall. Meanwhile, SVM also performs very well in accuracy, precision, recall, and F1-score values of 0.9528, 0.9532, and 0.9528. These results highlight SVM's strength in creating clear class boundaries while maintaining balanced performance across all evaluation metrics. KNN also showed that the BERT embedding can be effectively clustered in vector space and is compatible with distance-based classification methods. On the other hand, RF recorded an accuracy of 0.915, showing that this ensemble method is still quite reliable when used with the BERT representation. With a precision, recall, and F1-score of 0.9324, 0.9330, 0.9316, respectively, KNN maintains good classification consistency, although it is slightly behind SVM and LR. Although not as high as the previous models, RF still provides stable results due to its ability to handle data complexity through voting from many DT. RF achieved fairly balanced metrics with accuracy of 0.9150, precision of 0.919, recall of 0.915, and F1-score of 0.910. Naive Bayes with an accuracy of 0.8782 performs moderately because its basic assumption of independence between features is less suitable for interdependent text features, even though BERT has partially reduced the dependency through contextual mapping. Nevertheless, Naive Bayes yields respectable precision of 0.8835, recall of 0.8782, and F1-score of 0.8801, proving that this model can still be a lightweight and practical baseline. Meanwhile, the DT with the lowest accuracy of 0.858 also exhibits the lowest precision, recall, and F1-score (all around 0.8584–0.8594). This result further highlights its tendency to overfit and become unstable, particularly when dealing with complex and high-dimensional features like those produced by BERT. Overall, these results confirm that although BERT has provided strong feature representation, classifier selection still plays an important role in optimizing classification model performance.

**TABLE 4.** Performance Metrics of BERT with Different Classifiers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classifier** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| KNN | 0.9330 | 0.9324 | 0.9330 | 0.9316 |
| DT | 0.8584 | 0.8594 | 0.8584 | 0.8588 |
| RF | 0.9150 | 0.9194 | 0.9150 | 0.9102 |
| SVM | 0.9528 | 0.9532 | 0.9528 | 0.9518 |
| LR | 0.9651 | 0.9651 | 0.9651 | 0.9651 |
| NB | 0.8782 | 0.8835 | 0.8782 | 0.8801 |

The accuracy comparison results with BERT and pure ML algorithms are shown in Figure 4. The evaluation results of six ML algorithms show that using BERT as a feature extractor has a varying impact on classification performance. Linear-based models such as LR and SVM showed a consistent increase in accuracy when using the feature representation from BERT, reaching 0.9651 and 0.9528 respectively which outperformed the approach without BERT. In contrast, for algorithms such as DT and NB, performance stagnated even decreased when combined with BERT. This is most likely due to the inability of these models to process the high-dimensional dense embedding produced by BERT, rendering them unable to form effective decision rules.



**Figure 4.** Model Performance with and Without BERT

In comparison, the LR model using TF-IDF features produces an accuracy of 0.96 [8], while the BERT model application in an end-to-end manner can achieve an accuracy of up to 0.988 [10]. Another study combining BERT with LR also showed competitive performance, with accuracy reaching 0.98 [11]. Compared with the model using same dataset [9], improvements were seen in some classifiers. The KNN model which only achieved an accuracy of 0.39 [9], the BERT implementation in this study increased the accuracy to 0.93. Similarly, the LR model which initially only 0.93 [9] increased to 0.97. Meanwhile, the SVM model remained stable at 0.95 both with and without BERT demonstrating that although SVM is quite strong in both approaches, the BERT is still able to maintain that performance. In general, this finding confirms that BERT integration is most optimal when combined with models that are able to utilize feature complexity such as SVM and LR, while tree-based models tend to be better suited using simpler feature representations. Notably, the highest performance in this study was achieved by the BERT-LR combination, reaching an accuracy of 0.9651 with balanced precision, recall, and F1-score values. This result highlighting not only its strong predictive power, but also its reliability across all evaluation metrics. This balanced performance underscores the practical advantage of this approach for real-world spam detection tasks, where consistency across metrics is critical.

# CONCLUSION

BERT consistently performed better than LSTM across all classifier types tested in the email spam classification task. The accuracy of each model using BERT's feature representation was consistently higher than that of the LSTMs. This difference in performance suggests that BERT is more adept at capturing the semantic context and relationships between words in the text than LSTM. However, classifier selection in BERT still significantly influences the final performance. Linear models such as LR showed the most optimal performance (0.965), implying that the BERT representation is sufficient to make the data almost linearly separable in the binary case, thus eliminating the need for complex classification structures. These findings not only reaffirm the effectiveness of BERT in text representation, but also introduce a novel perspective by demonstrating that its combination with lightweight, classical classifiers such as LR can achieve competitive performance with lower computational cost. This hybrid strategy leverages rich semantic representations while avoiding the computational overhead of end-to-end deep learning, offering a practical and scalable solution that maintains high accuracy and is well-suited for real-world spam detection tasks, particularly in resource-constrained environments.

# ACKNOWLEDGMENT

The authors would like to thank Universitas Brawijaya and Faculty of Computer Science Universitas Brawijaya for funding this research.

# REFERENCES

1. I. Sulistiani, A. Amin, and M. F. Nastiar, “Professional Communication in the Digital Age: Benefits and Challenges of Using Instant Messaging Applications in the Workplace,” in 2025 *International Journal of Society Reviews (INJOSER)* **3**(2), 346.
2. E. I. Akpabio, *African Communication Systems and the Digital Age*. (Routledge, London, 2021).
3. E. H. Tusher, M. A. Ismail, M. A. Rahman, A. H. Alenezi, and M. Uddin, “Email Spam: A Comprehensive Review of Optimize Detection Methods, Challenges, and Open Research Problems,” in 2024 IEEE Access 12, 143627.
4. OECD, *Report of the OECD Task Force on Spam: Anti-Spam Toolkit of Recommended Policies and Measures*, (OECD Publishing, Paris, 2006).
5. R. Agarwal, A. Dhoot, S. Kant, V. S. Bisht, H. Malik, M. F. Ansari, A. Afthanorhan, and M. A. Hossaini, in 2024 IEEE Access **12**, 124298.
6. J. Jang-Jaccard, and S. Nepal, “A survey of emerging threats in cybersecurity,” in 2024 *Journal of Computer and System Sciences* **80**(5), 973–993.
7. L.S.S. A, Y.S. V, and N. Jayapandian, “Machine Learning Based Spam E-Mail Detection Using Logistic Regression Algorithm,” in *2023 Institute of Electrical and Electronics Engineers International Conference on Information and Communication Technology in Business Industry & Government (ICTBIG)*, (Institute of Electrical and Electronics Engineers, Indore, India, 2023), pp. 1–6.
8. Y. S. Navya, K. Pranathi, G. Srija, S. H. Naaz, “Spam Detection Using Machine Learning: A Logistic Regression Approach,” Advacementof Computer Technology and Its Applications, **8**, 2584–1262 (2025).
9. I. Diyasa, “Implementation of Natural Language Processing for Spam Email Detection in Outcome Based Education (OBE) Application,” in 2023 *International Journal of Entrepreneurship and Business Development (IJEBD)* **6** (6), 1166–1171.
10. N. Raihen, S. Rana, S. Akter, and M. A. Kadir, “Efficient Email Spam Detection Using Machine Learning Techniques: A Comparative Analysis of Classification Models,” in 2024 *International Journal of Intelligent Computing and Information Sciences (IJICIS)* **24**(4), pp. 1–15.
11. S. Shrestha, A.M.O. Khan, P. Jain, V. Ravi, and H.L. Gururaj, “Analysis of Machine Learning Approach for Spamming Electronic Mail Detection,” in IoT Sensors, ML, AI and XAI: Empowering A Smarter World, edited by B. Pradhan and S. Mukhopadhyay, (Springer Nature Switzerland, Cham, 2024), pp. 415–438.
12. P. Bhatti, Z. Jalil, and A. Majeed, “Email Classification using LSTM: A Deep Learning Technique,” in 2021 *International Conference on Cyber Warfare and Security (ICCWS)*, (Institute of Electrical and Electronics Engineers, Islamabad, Pakistan, 2021), pp. 100–105.
13. A. Nitisha, *Email Spam Dataset*, WWW document, (<https://www.kaggle.com/datasets/nitishabharathi/email-spam-dataset>).
14. R. DiPietro and G. D. Hager, “Deep learning: RNNs and LSTM,” in Handbook of Medical Image Computing and Computer Assisted Intervention (Elsevier, 2020), pp. 503–519.
15. I. Tenney, D. Das, and E. Pavlick, “BERT Rediscovers the Classical NLP Pipeline,” (2019).
16. M. Steinbach and P.-N. Tan, “kNN: k-nearest neighbors,” in 2009 *The Top Ten Algorithms in Data Mining*, (Chapman and Hall/CRC, 2009), pp. 165–176.
17. B. De Ville, “Decision trees,” in 2013 *Wiley Interdisciplinary Reviews: Computational Statistics* **5**(6), pp.448–455.
18. F.-J. Yang, “An Implementation of Naive Bayes Classifier,” in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, (Institute of Electrical and Electronics Engineers, Las Vegas, NV, USA, 2018), pp. 301–306.
19. D. A. Pisner and D. M. Schnyer, “Support Vector Machine,” in *Machine Learning*, edited by A. Mechelli and S. Vieira, (Academic Press, 2020), pp. 101–121.
20. G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, “Linear Regression,” in *An Introduction to Statistical Learning*, (Springer International Publishing, Cham, 2023), pp. 69–134.
21. S. J. Rigatti, “Random Forest,” in 2017 *Journal of insurance medicine*, **47**, 31–39.
22. S. Raschka, and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python,* *Scikit-Learn, and TensorFlow 2*, 3rd ed (Packt Publishing, Birmingham, 2020).