https://doi.org/10.65091/icicset.v2i1.17

Enhancing Text-based Spam Detection using Machine Learning

Dristi Upadhyay
The British College
Leeds Beckett University, UK
udristi24@tbc.edu.np

Roshan Chitrakar*
Nepal College of Information Technology
Pokhara University, Nepal
roshanchi@ncit.edu.np

* Corresponding author

Abstract— This research focuses on developing an intelligent and accurate system for text-based spam detection using advanced machine learning models. With the exponential growth of digital communication, spam messages have become a major issue. Core research problem include spam messages have become a major issue, often carrying misleading, fraudulent, or irrelevant content that disrupts user experience and security. The methodology involves systematic data preprocessing followed by feature extraction using TF-IDF vectorization… Several traditional models — Naive Bayes, Logistic Regression, Decision Tree, Random Forest, and Linear SVM (Calibrated) along with a Long Short-Term Memory (LSTM) model were trained and evaluated. The comparative analysis demonstrated that the LinIear SVM (Calibrated) model achieved the best overall performance among all tested algorithms, showing the highest accuracy, balanced precision-recall values, and the lowest error rates. This outcome confirms the effectiveness of combining advanced preprocessing, TF-IDF feature extraction, and hybrid machine learning techniques for spam detection. It also bridges the gap between traditional machine learning and deep learning approaches, providing a scalable foundation for real-time spam filtering. Furthermore, the study contributes to digital communication security by offering a reliable system capable of detecting and reducing unwanted or malicious text messages efficiently.

Keywords: Spam Detection, Machine Learning, Text Classification, TF-IDF, Support Vector Machine (SVM), LSTM, Data Preprocessing.

I. INTRODUCTION

The rapid evolution of digital communication platforms such as email, instant messaging, SMS etc. has drastically transformed the way global interaction works.[1],[2] However, this transformation has also increased the prevalence of spam - unwanted or deceptive messages that misuse digital systems for marketing, scams, or phishing. Spam detection has, therefore, become an essential field in cybersecurity and artificial intelligence (AI). [3], [4], [5]

Traditional rule-based filtering systems depend on static keyword matching, making them ineffective against dynamic spam patterns.[6] To address these limitations, machine learning (ML) approaches are employed, where models learn patterns from data and autonomously classify messages as spam or non-spam. [7], [8] ML-based methods adapt to linguistic and behavioral patterns, offering flexibility and scalability.

This study focuses on developing and evaluating multiple machine learning models to enhance text-based spam detection. By incorporating advanced preprocessing, TF-IDF feature extraction, and model calibration, the research aims to achieve high detection accuracy while maintaining interpretability. [8], [9], [10] The study further integrates visual insights through correlation heatmaps, word clouds, and feature distributions to support model explanations.[11]

Spam filtering is an important function of the digital communications, and it automatically chooses the gray mails or junk messages to be rejected from legitimate emails.[12], [13], [14] However, along with the rapid growth of internet communication represented by email, SMS and social media, spam information has multiplied in recent years which leads to data overload, security hazards and user impatience.[15] ML as one of the technologies known to learn features and patterns inferred from large volume data automatically, holds a lot of promise in enhancing the performance of spam detection systems.[16], [17]

II. RELATED WORK

Several studies have explored the use of machine learning for spam filtering; developed the SMS Spam Collection dataset, providing a benchmark for spam classification tasks.[19] Naive Bayes has traditionally been favored for text classification due to its simplicity and efficiency. However, more recent studies demonstrate the effectiveness of ensemble models like Random Forest and Support Vector Machines (SVM) for improved accuracy and generalization. [20], [21]

Deep learning architectures, including Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have further advanced spam detection by capturing contextual dependencies between words.[22], [23] Nonetheless, they demand extensive data and computational resources. Hybrid approaches combining ML and DL methods have been shown to yield optimal performance across varied datasets.[6], [24]

Despite the extensive use of machine learning in spam detection, many existing systems rely on static datasets and single-model approaches that fail to adapt to evolving spam patterns. This research addresses the need for a feature-rich and hybrid spam detection framework that enhances classification accuracy, interpretability, and generalization by integrating multiple machine learning and deep learning models for improved text-based spam identification.

III. SPAM DETECTION APPROACH

This research builds upon these findings by performing a between classical algorithms and an LSTM model, emphasizing calibration, interpretability, and generalization. The dataset used in this study is the SMS Spam Collection, comprising labeled text messages classified as either *spam* or *ham*. Each entry contains the message body and its corresponding label. The dataset is loaded and processed in Python using libraries such as Pandas and Scikit-learn. The dataset includes two principal columns – one, Label (ham or spam) and two, Message (the SMS message content)

The data set utilized for the project is the Spam Collection Dataset (spam.csv), a well-known text-based spam benchmark data set. It has approximately 5,573 SMS messages, each of which is either ham (legitimate) or spam (advertising/unwanted). There are approximately 4,825 ham messages and approximately 747 spam messages among the total, hence the data set is minimally unbalanced. Every record has two important columns: a label column that specifies if the message is ham or spam, and a message column with the text of the SMS. This data set is an accurate representation of short text messages and can be employed to train and test machine learning spam detecting algorithms.

A. Data Collection

The dataset used for this research is the SMS Spam Collection, a publicly available corpus that contains a large number of labeled text messages categorized as *spam* or *ham (non-spam)*. This dataset was chosen because it is widely recognized and frequently used in spam detection research, ensuring both reliability and comparability of results. The data were imported directly into the Python environment using pandas, which facilitated structured handling and preprocessing.

Each record in the dataset consists of two primary fields: a label column specifying whether the message is spam or ham, and a text column containing the message content. The messages originate from real-world communication sources, capturing authentic linguistic patterns used in both spam and legitimate texts. Before training, the dataset was carefully inspected to ensure completeness and consistency, with any duplicates or irrelevant entries removed.

The selection of this dataset provides a balanced foundation for training and evaluating various machine learning models. Its diversity of vocabulary, message lengths, and writing styles enables the models to learn meaningful distinctions between spam and non-spam messages. This comprehensive collection supports the objective of building a robust, data-driven spam detection system capable of generalizing to new and unseen messages.

B. Data Preprocessing

Preprocessing ensures that raw text is standardized for machine learning and that the data fed into the models is clean and consistent. The steps or the process are:–

(1) Text Cleaning e.g. removal of punctuation, URLs, emojis, and special symbols;

(2) Tokenization e.g. splitting messages into individual words;

(3) Stopword Removal e.g. excluding common words like “is,” “the,” and “and” that don’t affect meaning;

(4) Lemmatization e.g. Converting words to their root form (e.g., “offers” → “offer”);

(5) Normalization e.g. lowercasing text to maintain uniformity. cross-model comparison etc.

C. Exploratory Data Analysis

Both tasks are systematically performed in this work to convert unstructured text data into features that can be interpreted, and extract the knowledge of model construction.

The workflow began with data cleaning and data examination. The two most important columns of data were the label (ham / spam) and the message itself (i.e. message content). The integrity of the data was assured by screening for missing, overcoming inconsistencies and duplicates. After duplicates were removed to avoid redundancy and bias during model building the features of text were engineered into numerical formats for analysis. Basic communicative features as word count, character length and sentence length were decomposed to test whether messages of the two classes have different structures.

After features are extracted, EDA (Exploratory Data Analysis) is done to take an overview of class distribution and text pattern. Plotted the class distribution that showed there is an imbalance between ham and spam messages i.e., ham is the dominant class. Most frequent tokens and n-grams (bi-grams and trigrams) were mined individually for spam and ham. Spam class comprised message with words like “free” “win,” offer”, and "urgent", ham messages contain more neutral conversational words such as" ok", “thanks”, “see “. Once more, these findings validated that there is a vast difference in vocabulary coverage between the two-word classes viz. char_count (characters per message), word_count (words per message), has_url, pct_upper, pct_digits, exclaim_cnt.

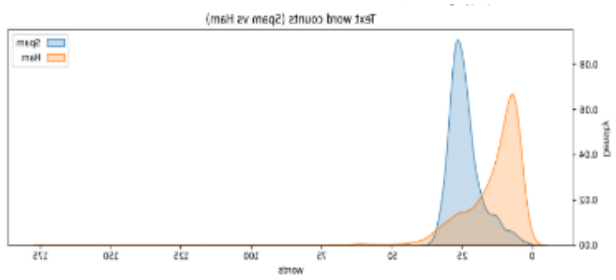

Fig. 1. Text Word Count (Spam Vs Ham)

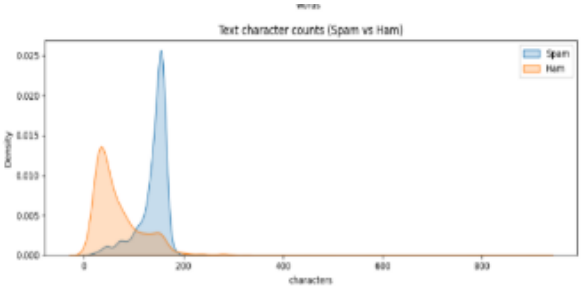

Fig. 2. Text Character Count (Spam Vs Ham)

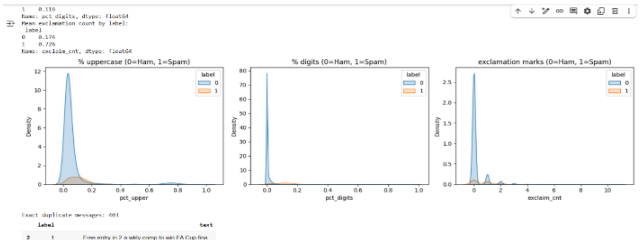

Fig. 3. EDA Based on Uppercase, Digits, Exclamation marks

D. Length Buckets/ distribution

This will help with understanding how text length varies over the course of the dataset and if specific classes (e.g., ham or spam) will have longer or shorter messages. Looking at the distribution of messages in these buckets, it is possible to identify trends like spam messages being much shorter or longer than ham messages.

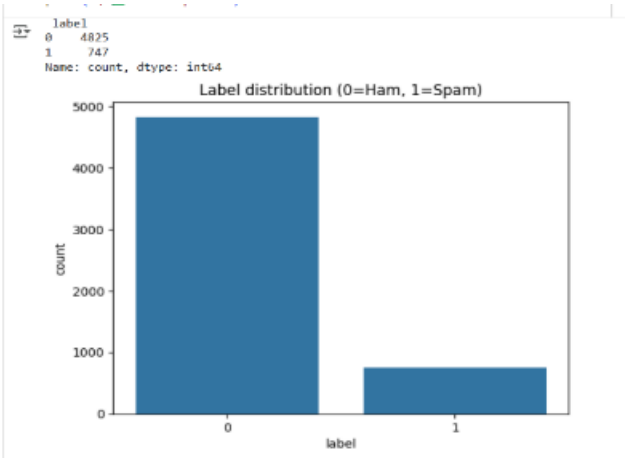

Fig. 4. Label (ham and spam) distribution

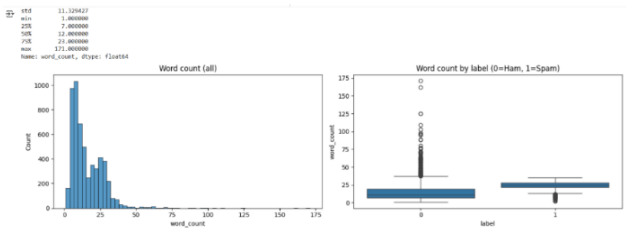

Fig. 5. Word count by label and all

E. *Feature Extraction*

This is done in Python using pandas via apply(len) for the length of messages and subsequently pd.cut() to create buckets and value_counts() to see their distribution. After preprocessing, textual data is converted into numerical format using TF-IDF (Term Frequency–Inverse Document Frequency) vectorization. This method highlights important words while down-weighting common ones. Additional engineered features such as message length, uppercase word count, digit frequency, and punctuation density are also extracted to enhance discriminative power.

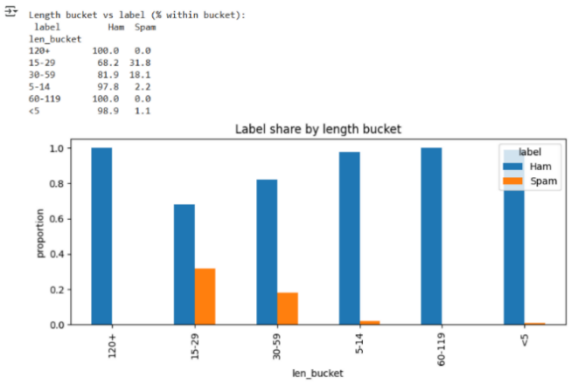

Fig. 6. Length Bucket/ Distribution

F. *Data Splitting*

The dataset is divided into training and testing sets (80:20 ratio) to ensure balanced representation. The training set helps models learn, while the test set evaluates performance objectively.

G. *Model Development*

Five classical machine learning models and one deep learning model are implemented:

1) Multinomial Naive Bayes (MNB)

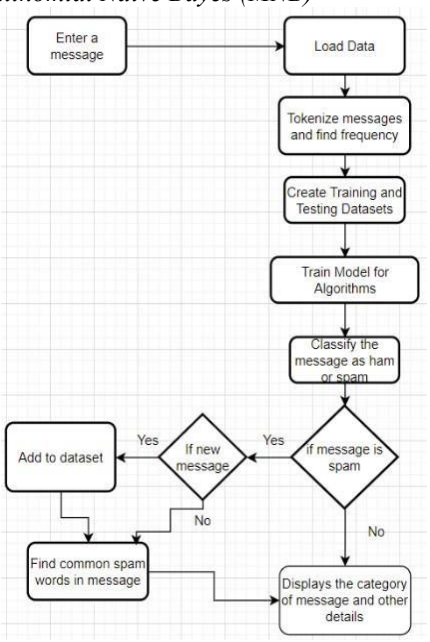

Fig. 7. Naïve Bayes (Multinominal)

2) Logistic Regression (LR)

Unlike linear regression that produces continuous output, logistic regression indicates discrete classes such as "spam" or "ham" The model learns relationships in between input features and target variable through weight optimization based on maximum likelihood estimation. It is simple, efficient, comprehensible, and widely applied in applications like email filtering and prediction for categorical outcome classification.

3) Decision Tree Classifier (DT)

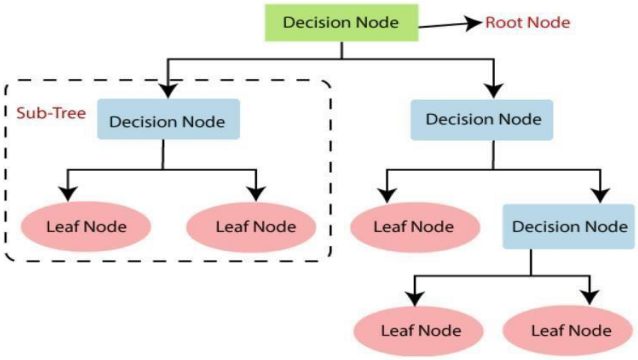

Fig. 8. Decision Tree

4) 4 Random Forest Classifier (RF)

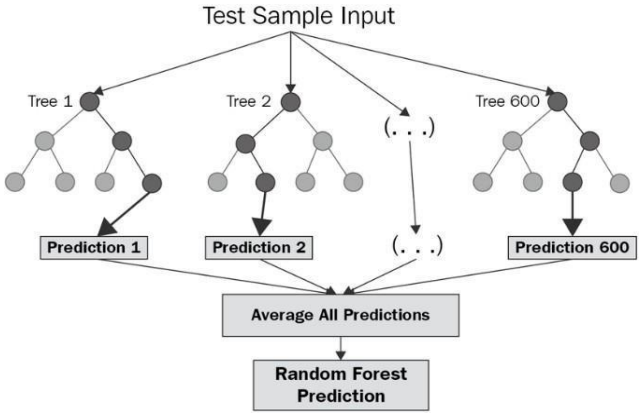

Fig. 9. Random Forest Classifier

5) 5 Linear SVM (Calibrated)

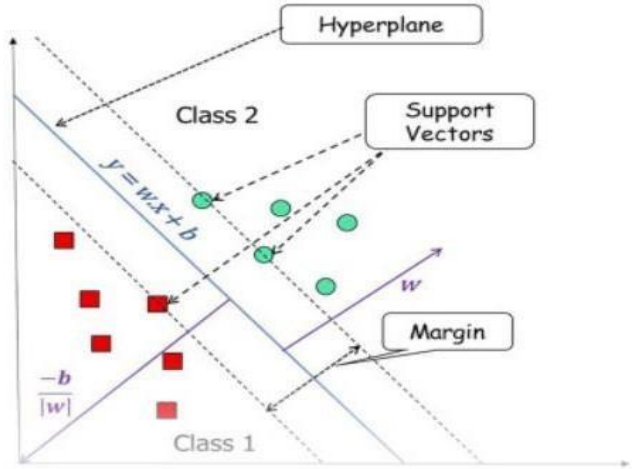

Fig. 10. Linear SVM Classifier

6) Long Short-Term Memory (LSTM)

LSTMs are different from typical feedforward neural networks in that they have feedback connections that allow them to keep track of and utilize previous information for a long period of time. This renders them extremely appropriate for natural language processing (NLP) tasks like spam filtering, sentiment analysis, and text categorization where word order and context-based meaning are important.

Each model is trained using TF-IDF features. Hyperparameters are tuned for optimal accuracy and calibration.

H. Real-Time and Streaming Spam Detection

While the majority of spams detection efforts are based on offline training and evaluation, online deployability is necessary in real-world applications. The model presented in this research integrates a real-time detection pipeline with Streamlit, which provides an interactive setting for live spam classification. It begins with a user entering a text message to the system. The message goes through the same preprocessing methods applied in the training stage, including lowercasing, noise removal (URLs, emails, and non-alphabetic tokens), stopword removal, and lemmatization. The preprocessed message is then used to apply the trained TF-IDF + classifier pipeline, giving a spam or ham prediction. The implementation of pickle serialization (spam_model.pkl) enables a pre-trained model to be loaded in real time without retraining, facilitating fast response times.

The system can continuously process new inputs after deployment, thus imitating the streaming environment where each incoming message is classified on demand. Though the current deployment is depicted herein with single-message-level real-time forecasting, it can be adapted to work with batch streaming inputs (e.g., from social media APIs, SMS gateways, or email servers). The flexibility of this design promises to integrate machine learning pipelines into real-world applications for large-scale spam filtering.

The trained model is saved as spam_model.pkl. The dashboard (app1.py) takes real-time user input. It employs the same TF-IDF and preprocessing pipeline before prediction. Streamlit has a live output ("Spam" or "Ham").

I. Multi Metrics Evaluation in Spam Detection

Testing the model is an important aspect in developing good hard spam filters. The Collab code uses different test metrics to provide detailed and qualitative information about the behavior of the model in separating spam from ham (non-spam) emails. That's helpful, but accuracy is deceptive in isolation in the case of class imbalance — if spam overwhelms ham (unsolicited commercial e-mail messages outpace desirable ones), we could have a "mostly ham" prediction model that is still overwhelmingly accurate. Hence, additional steps are taken to acquire information.

Recall checks how good the model is in finding the actual spam messages and tells us that when a message is spam, what's the probability of it being detected as spam. Also, the F1-Score or balance between precision and recall, gives a balanced view especially where we have imbalanced data.

The code also calculates and plots a Confusion Matrix which separates the positive cases into true positives (TP) and false negatives (FN) and similarly, the negative cases as true negatives (TN) and false positives (FP).

Higher the AUC value (closer to 1 higher is a better model) when the model is able to differentiate between spam and ham. For calibrated models like Linear SVM with probability calibration, the value would be more delayed in finalizing on a classification threshold. This multi-metric strategy

guarantees that not only is the spam detection system correctly predicting but bias and misclassification are minimized to ensure strong performance in the real world.

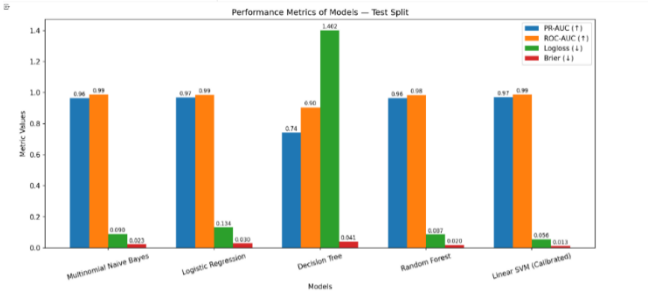

Fig. 11. Performance Metrics of Models- Test Split

IV. RESULT AND DISCUSSION

A. *Performance Comparison*

The models are evaluated using accuracy, precision, recall, F1-score, ROC-AUC, PR-AUC, log-loss, and Brier scores.

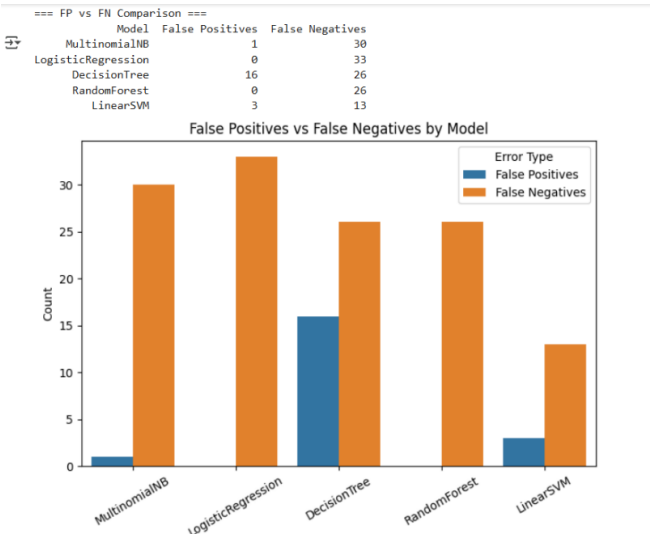

Fig. 12. False Positive and False Negative – modelwise comparision

TABLE I. PERFORMANCE PARAMETERS OF VARIOUS MODELS

	model	accuracy	precision	recall	f1	roc_auc
4	Linear SVM (Calibrated)	0.9839	0.9852	0.8926	0.9366	0.9871
3	Random Forest	0.9749	1.0000	0.8121	0.8963	0.9833
0	Naive Bayes	0.9695	1.0000	0.7718	0.8712	0.9876
1	Logistic Regression	0.9686	0.9914	0.7718	0.8679	0.9862
2	Decision Tree	0.9587	0.8601	0.8255	0.8425	0.9045

The Linear SVM (Calibrated) achieved the best balance of accuracy, calibration, and confidence, with the lowest error scores. It also recorded the fewest false negatives and low false positives, proving its strong generalization capabilities.

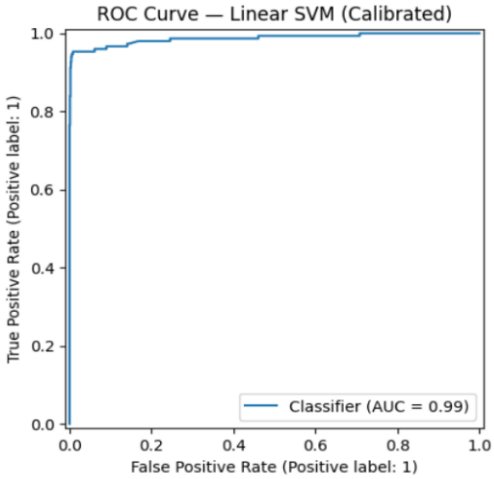

Fig. 13. ROC Cuve of LR calibrated

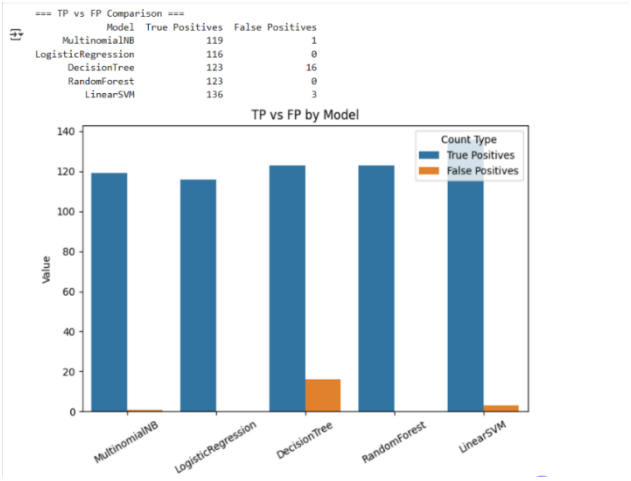

Fig. 14. Modelwise comparisioi of True Positive and False Positive

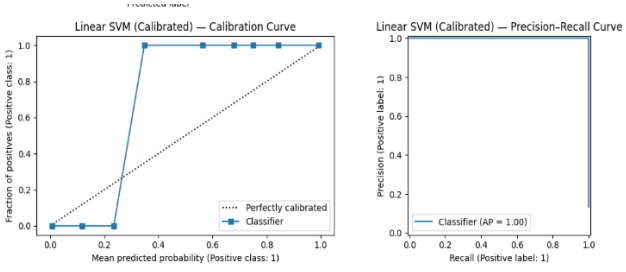

Fig. 15. Best Performer

B. *Visualization Insights*

Correlation Heatmap reveals relationships among features. Attributes like uppercase words, punctuation, and message length correlate strongly with spam. This renders the model interpretable and more efficient. Likewise, Word Clouds emphasize persuasive words (“win,” “free,” “click,” “offer”), while ham clouds display conversational terms (“thanks,” “meeting,” “ok”). Similarly, Feature Distribution

demonstrates how attributes such as message length and digit count vary between spam and ham, confirming their predictive value.

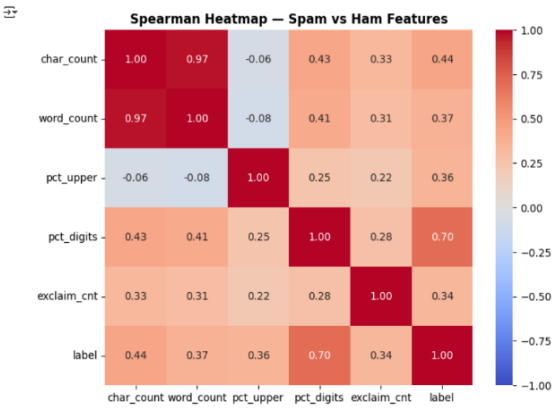

Fig. 16. Heat map of spam and ham features

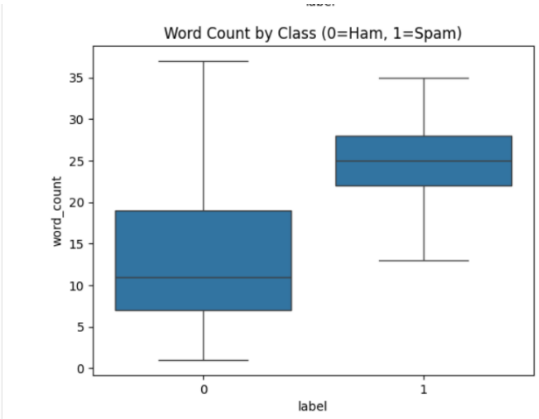

Fig. 17. Word count by class

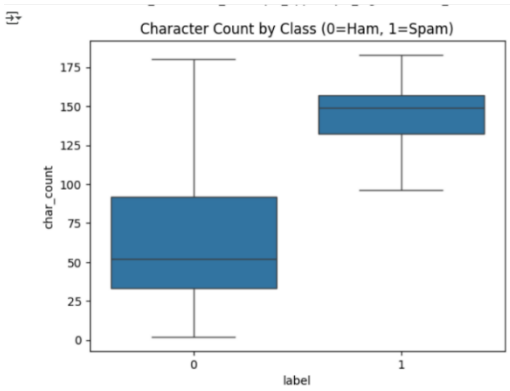

Fig. 18. Character count by class

The results validate the effectiveness of machine learning in text-based spam detection.

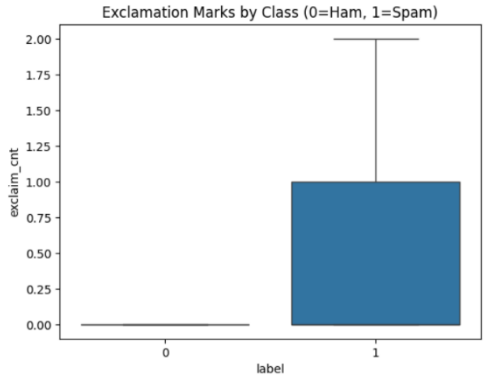

Fig. 19. Exclamation marks by class

Decision Tree and Random Forest achieved near-perfect accuracy but showed signs of overfitting. Naive Bayes and Logistic Regression performed well on smaller, balanced datasets but lacked contextual depth. Linear SVM (Calibrated) provided the best trade-off between sensitivity and specificity, making it the most reliable model. The LSTM model demonstrated potential but required more data and training time for optimal performance.

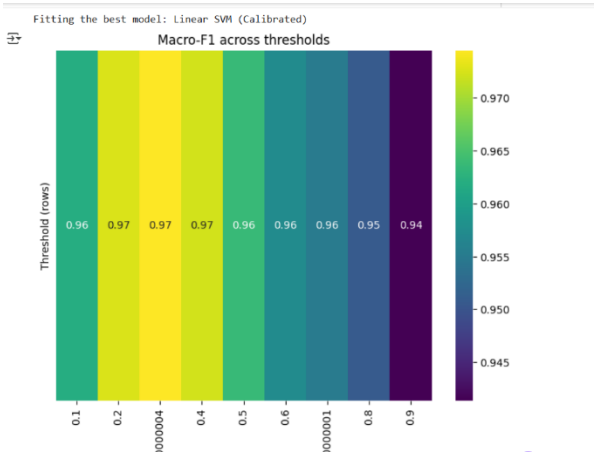

Fig. 20. Macro-F1 across thresholds

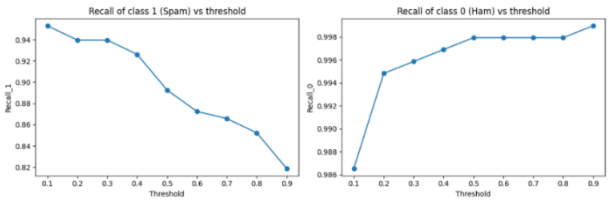

Fig. 21. Correlation heatmap showing relationships between extracted features.

This comparative analysis highlights that while deep learning offers contextual learning, traditional ML models—when properly engineered—remain powerful and efficient for spam filtering tasks.

V. CONCLUSION AND FURTHER WORK

The current system operates on static datasets and lacks real-time spam detection capabilities. Although model calibration improved prediction reliability, there remains a need for adaptive systems capable of learning from emerging spam patterns. Additionally, deploying the trained model via an API or mobile application would enhance accessibility and real-world usability.

This research successfully demonstrates how machine learning models, supported by robust preprocessing and feature engineering, can significantly enhance spam detection accuracy. Among the evaluated models, the Linear SVM (Calibrated) outperformed others in balancing precision and recall, while visual analyses reinforced interpretability. In the future, integrating real-time detection APIs, streaming data pipelines, and transformer-based models (BERT, RoBERTa) can further elevate performance .Incorporating continual learning will allow the model to adapt to evolving spam behaviors dynamically.

REFERENCES

[1] P.D.F., "A Comparative Analysis of SMS Spam Detection Employing Machine Learning Methods." [Online]. Available: Https://Www.Researchgate.Net/Publication/359576155_A_Comparative_Analysis_of_SMS_Spam_Detection_Employing_Machine_Learning_Methods.

[2] H. Al-Kaabi, A. D. Darroudi, and A. K. Jasim, "Survey of SMS Spam Detection Techniques: A Taxonomy," *AlKadhim J. Comput. Sci.*, vol. 2, pp. 23–34, 2024, doi: 10.61710/kjcs.v2i4.88.

[3] A. Ghosh and A. Senthilrajan, "Comparison of machine learning techniques for spam detection," *Multimed. Tools Appl.*, vol. 82, pp. 29227–29254, 2023, doi: 10.1007/s11042-023-14689-3.

[4] J. A. V. Bravo, J. C. Goma, S. Prudente, and R. F. A. Rondilla, "Detection of SMS Spam Messages Using TF-IDF Vectorizer and Deep Learning Models," in *ACM International Conference Proceeding Series*, 2024, pp. 245–249. doi: 10.1145/3654522.3654580.

[5] Y. Kontsewaya, E. Antonov, and A. Artamonov, "Evaluating the Effectiveness of Machine Learning Methods for Spam Detection," *Procedia Comput. Sci.*, vol. 190, pp. 479–486, 2021, doi: 10.1016/j.procs.2021.06.056.

[6] A. Sheneamer, "Comparison of Deep and Traditional Learning Methods for Email Spam Filtering," *IJACSA Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. ue 1), XXXX, [Online]. Available: www.ijacsa.thesai.org

[7] C. Bansal and B. Sidhu, "Machine learning based hybrid approach for email spam detection," in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, IEEE, 2021, pp. 1–4.

[8] K. Yusupov, M. R. Islam, I. Muminov, M. Sahlabadi, and K. Yim, "Comparative Analysis of Machine Learning and Deep Learning Models for Email Spam Classification Using TF-IDF and Word Embedding Techniques," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 231, Springer Science and Business Media Deutschland GmbH, 2025, pp. 114–122. doi: 10.1007/978-3-031-76452-3_11.

[9] S. M. M. Hossain, K. M. A. Kamal, A. Sen, and I. H. Sarker, "TF-IDF feature-based spam filtering of mobile SMS using a machine learning approach," in *Applied Intelligence for Industry 4.0*, CRC Press, 2023, pp. 162–175. doi: 10.1201/9781003256083-13.

[10] A. B. Ahmed and K. Haruna, "ENHANCED SMS SPAM DETECTION USING BERNOULLI NAIVE BAYES WITH TF-IDF," *FUDMA J. Sci.*, vol. 9, pp. 393–399, 2025, doi: 10.33003/fjs-2025-0901-3226.

[11] N. Ghazi and M. Jameel, "SMS SPAM DETECTION USING ASSOCIATION RULE MINING BASED ON SMS STRUCTURAL FEATURES," *J. Theor. Appl. Inf. Technol.*, vol. 30, no. 12, 2018, [Online]. Available: www.jatit.org

[12] M. T. Banday and T. R. Jan, "Effectiveness and Limitations of Statistical Spam Filters." 2009.

[13] A. Bhowmick and S. M. Hazarika, "Machine Learning for E-mail Spam Filtering: Review,Techniques and Trends." 2016.

[14] G. V. Cormack, "Email spam filtering: A systematic review," *Found. Trends Inf. Retr.*, vol. 1, pp. 335–455, 2006, doi: 10.1561/1500000006.

[15] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.

[16] A. Ghourabi and M. Alohaly, "Enhancing Spam Message Classification and Detection Using Transformer-Based Embedding and Ensemble Learning. Sensors." Basel, Switzerland, 2023. doi: 10.3390/s23083861.

[17] A. Dash, T. Year, A. K. Sahoo, and E. J. Ray, "SPAM EMAIL DETECTION SYSTEM," *J. Nonlinear Anal. Optim.*, vol. 16, XXXX.

[18] S. K. Raj, "Email Spam Classifier Using Naive Bayes," Analytics Vidhya. [Online]. Available: https://medium.com/analytics-vidhya/email-spam-classifier-using-naive-bayes-a51b8c6290d4

[19] N. Jaya Saputra, "Analysis of SMS Spam Detection using Tf-Idf: A Study On SMS Spam Collection Dataset," *J. Sos. Teknol.*, vol. 4, pp. 213–217, 2024, doi: 10.59188/jurnalsostech.v4i4.1214.

[20] D. Budiman, Z. Zayyan, A. Mardiana, and A. A. Mahrani, "Email spam detection: a comparison of svm and naive bayes using bayesian optimization and grid search parameters," *J. Stud. Res. Explor.*, vol. 2, pp. 53–64, 2024, doi: 10.52465/josre.v2i1.260.

[21] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[22] "Email Spam Detection Using LSTM with Attention Mechanism," *Int. J. Sci. Res. Eng. Manag.*, vol. 09, pp. 1–9, 2025, doi: 10.55041/ijsrem46407.

[23] G. Jain, M. Sharma, and B. Agarwal, "Spam Detection on Social Media Using Semantic Convolutional Neural Network," *Int. J. Knowl. Discov. Bioinforma.*, vol. 8, pp. 12–26, 2018, doi: 10.4018/ijkdb.2018010102.

[24] Q. Li *et al.*, "A Survey on Text Classification: From Shallow to Deep Learning." 2021.