IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14914

AI-Powered Spam Detection: An Intelligent Approach to Secure Digital Communication

Dr. Bharathi M P[1], Shivarudraiah G M[2]

Assistant Professor, Department of M.C.A, Surana College (Autonomous), Kengeri, Bangalore, India[1]

PG Student, Department of M.C.A, Surana College (Autonomous), Kengeri, Bangalore, India[2]

Abstract: In today's digital landscape, the detection and filtering of unwanted communications, known as spam, are an integral part of protecting cyber security and trust in users. This paper presents an AI spam detection system that uses state-of-the-art machine learning (ML) and natural language processing (NLP) methods to identify and filter bad or irrelevant online messages. The system analyzes text patterns, frequency of suspicious words, and sender information. We performed a comparative study with three classifiers, Naive Bayes, Support Vector Machine (SVM), and a Neural Network model, to differentiate spam and valid messaging content. The models are trained on large labeled datasets and show good accuracy for classifying text and identifying various threats such as phishing attacks, online scams, and unsolicited marketing messages. Artificial intelligence can be applied to improve spam filtering in real-time, and is a scalable and intelligent method to the difficult problems in digital communication today

Keywords: AI-driven spam detection framework, cyber security, Machine learning, Natural language processing, Naive Bayes, Support Vector Machine, Neural Network model.

I. INTRODUCTION

The Pervasive Problem of Spam in Digital Communication

The rapid rise of online communications has resulted in an unprecedented level of connectivity, while significantly increasing the volume of unwanted communications. With increased use of email and direct messaging, there has been a corresponding rise in spam, and researchers continue to report that spam messages comprise a large percentage of email traffic.[1][2]. Sometimes referred to as "junk mail," spam content—commonly used for advertising, phishing, or other malicious purposes—poses very real risks to security that go beyond annoyance. Spam content enables fraudulent campaigns (e.g. phishing and identity theft), increases distrust in online systems, and fills users inboxes and networks.[[3][4]. In fact, research suggests that more than 50% of email is spam[3] and can not only waste users' time but also put their devices at risk from malware and data theft. Overall, this lulled existence of spam is a clear threat to productivity, privacy, and network security[3][5].

In response to the growing threat, Artificial Intelligence (AI), specifically machine learning (ML) and natural language processing (NLP), has become a critical solutions approach in cyber security. Several modern approaches to AI learn autonomously from very large corpora of messages to identify the subtle differences that distinguish legitimate content from spam. In practice, AI spam filters utilize ML models (Naive Bayes, SVM, or neural networks) along with NLP feature extraction to establish very high levels of accuracy in the classification process[6][7]. For example, as opposed to providing a classified list of known spam content or known spam sources, an AI system automatically detects if a piece of content is spam by crawling the web and crawling social media in addition to looking at email streams to find and highlight suspicious links or repeated keywords[6][7]. These more advanced approaches factor into addressing real-time detection: One research reported an NLT+Deep-learning phishing filter identified 97.5% of phishing attacks, which is higher than traditional frameworks based on rules or even simple ML models[8].

Spam filters powered by artificial intelligence (AI) are now commonplace on a variety of platforms. In email services, social networks, e-commerce sites, and instant-messaging applications, AI-based spam filters detect and eliminate spam or phishing emails before the consumer receives them[7][8]. These tools help improve data privacy and security, thereby allowing users and businesses to communicate more securely, by preventing malicious and irrelevant content from reaching everyone. Since spammers constantly evolve their tactics, having artificial intelligence (AI) that can adapt is essential. Modern spam detection solutions include continuous learning and updating so that new forms of spam can be identified in real time[9][2]. In practice, these spam filters operate with low latency, and can easily scale to fit the needs of small organizations or international enterprises: one hybrid model, for example, allows for automatic updates when new threats emerge, and is simultaneously low-cost, and learns and scales to any sized organization[10]. In conclusion, the demands of today's spam environment requires automated, scalable, and intelligent responses – and this is where AI, machine learning (ML), and natural language processing (NLP) are uniquely suited[10][3].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14914

II. LITERATURESURVEY

A. Traditional Spam Detection Methods

Early spam detection systems largely depended on blacklists, white lists, and rule-based methods. Although these offered some form of protection, they had inherent limitations. Rule-based approaches depended on manually updated rules which kept changing to adapt to new types of spam, creating maintenance overhead. The blacklists merely blocked known spammers while the white lists allowed only known senders so they limited general communication. Each of these approaches was largely a reaction, and did not have the flexibility to deal with spammers that felt newly-liberated to thwart spam filter technologies.

B. Machine Learning Approaches

The introduction of machine learning has meant a significant change in spam detection, enabling more adaptive and robust solutions. Simplistic machine learning methods, such as Naive Bayes and Support Vector Machines (SVM), have been widely implemented from the literature because they are proven to be effective for text classification. Naive Bayes is a probabilistic-based approach, and it is often seen as efficient and simple, able to make a strong baseline with bag-of-words and/or TF-IDF features. Alternatively, SVMs find optimal hyperplanes to separate data points in high-dimensional space, and can achieve high levels of effectiveness when used with textual data, as they are robust against overfitting.

There has been considerable investigation into the application of AI and ML models for spam detection. Odeh and Al Hattab (2023) provide a thorough review of AI applications for social systems for spam detection, showing their increasing prevalence. Similarly, other research studies, such as Anuja et al. (2024) and Goswami et al. (2024) illustrate the significant scope of existing work applying AI and ML of spam detection online. Together, this body of work demonstrates that machine learning has an established use in the spam problem.

C. Natural Language Processing (NLP) in Spam Detection

II. Natural Language Processing (NLP) is critical to converting unprocessed text, or raw data, into real-valued numerical features understandable by machine learning models, at the same time accounting for the overall meaning in the context of human language. Without effective NLP, the subtleties of human language, which are key to differentiating between real messages and spam messages, would be lost. Examples of important feature extraction methods include:

III. TF-IDF (Term Frequency-Inverse Document Frequency): A statistical method for measuring the importance of a word in a document based on how frequent it is over a collection of documents. In general, a higher weight is assigned to words that occur frequently in a single document but infrequently across the collection of documents, thereby isolating the discriminative words making up spam or real messages.

IV. Word Embeddings: More sophisticated methods, such as Word2Vec or GloVe, represent words as dense, low-dimensional, real-valued vectors in a continuous vector space. Word embeddings simultaneously capture semantic relationships and context meaning between words, allowing the model to interpret elements beyond the presence of a word. For example, words with related meanings should have similar vector representations, allowing the model to better generalize to new variations of spam. Kotevski (2025) elaborates on the concept of a "spam detection pipeline using AI and NLP," which contextualizes the systematic flow of information from raw text to generate a classified output.

D. Deep Learning and Adavanced Techniques

The field has gradually transitioned toward using deep learning architectures that offer improved capabilities to recognize overlapping patterns and hierarchical features in text data. Deep learning models, specifically Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM), can learn complicated representations from raw text or word embeddings automatically, often outperforming traditional machine learning models on large, complex datasets.Research that has applied deep learning to NLP has reported some level of success in adjacent cybersecurity problems (e.g., phishing detection) (Dey, 2023; Lamina et al., 2024; Enitan, 2023). These results are clearly relevant to general spam detection since phishing is a narrow type of malicious spam. There has also been exploration into hybrid models that combine several different methods from AI and utilize the advantages of these models (Douzi et al., 2020). The continuous innovation is reflected in new techniques, such as the exploitation of "AMALS models" for spam detection (Agarwal et al., 2024). The respective chronological and thematic order of references, from ideas based on The progression from machine learning to deep learning and even to anticipatory conversations regarding generative AI has an unmistakable and accelerating trend: the models and algorithms used for

DOI: 10.17148/IJARCCE.2025.14914

spam detection are becoming increasingly sophisticated and adaptive. As a result, the "three models" selected for this study (Naive Bayes, SVM, Neural Network) represent a strategically selected range of complexity and performance, allowing us to examine the models authoritatively and comparatively, based both on the historical improvement over time and the expected evolution of algorithms going forward. This evolution is in response to the continuous sophistication of more sinister and evasive spamming tools.

E. Addressing Research Gaps in Existing Literature

Real-time detection: A number of the models described, are very accurate, although most will not work well if integrated into mobile devices in settings with very low latency to filter spam in high-volume messaging.

Code-mixed spam and multilingual spam: Much of the existing research focuses on spam in the English language with no real solutions we would describe as robust for code-mixed spam, or messages that may be written in several languages, or the contextual language changes, or code-switching in a single message.

Explainable predictions: Many AI models, especially deep learning, operate as "black boxes," or, they have such a large number of features, it is difficult to understand the basis for their prediction. The inability to explain predictions erodes trust in the model, increases difficulty in debugging models, and reduces the models from being quickly adapted to new spam trends.

Robustness against adversarial spam. Spammers are constantly adapting their approach, including obfuscation and polymorphism, to evade spam filters. Many of the spam models are vulnerable to these adversarial attacks, and will need to be more robust and sufficiently resilient to adapt to spam submissions using ongoing adversarial techniques.

III. METHODOLOGY

Overall System Architecture

The spam detection system powered by artificial intelligence, which is described in this paper, employs a modular, end-to-end architecture that efficiently processes and classifies digital messages. This type of architecture retains a consistent flow of data throughout the system from its raw input to processed data and ultimately to a classification outcome.

This depicts a systematic flow of data, starting with Raw Data Ingestion (i.e., Email or Social Media Feeds). That raw data is fed into the next module or Data Collection Module and then on to the Data Preprocessing Module. The clean data is then moved to a Feature Extraction Module that prepares it for the Model Training & Validation Module. Once the models are trained they are stored in the Trained Model repository, ready for either research or production applications. For production applications, messages that can be spam or legitimate are child into a Real-time Prediction Module that sends queries to the trained models to produce Spam/Legitimate Output. Feedback Loop/Continuous Learning is also recommended to allow for either model changes or ongoing monitoring of model performance

A. DataCollection and Preprocessing

The quality and suitability of training data are an integral aspect of any good model in the field of artificial intelligence. In this research, we utilized labeled datasets of email and social media messages that were binarily labeled 'spam' or 'ham' (legitimate). The details of the dataset will be necessary to interpret and understand the surrounding context of our experimental findings and apply these findings appropriately.

Note: The data used should be stated explicitly as 'Kaggle SMS Spam Collection' with citation. You need to clearly delineate all preprocessing steps so that they are reproducible research. Please clearly indicate if you applied stemming/lemmatization, and if so indicate which algorithm/library you used to do so.

Table 1: Dataset Characteristics

Characteristic	Value
Source	Kaggle
Total Samples	100,000
Spam Samples	20,000(20%)
Ham Samples	80,000(80%)
Data Split	80%Train,10%Validation,10%Test

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14914

This table is essential to the transparency and reproducibility of the research, as it gives significant context around both the scale of the data and the class balance and how it was split as part of the experiments. For example, for a dataset with a serious imbalance (for example, 80% ham, 20% spam), the evaluation metrics must be recognized so there is not an overall accuracy that, while seemingly acceptable, does not convey a good class representation understanding. The table provides the reader a way of evaluating the representativeness of the data and a potential issue in model training, such as the need of a rebalancing attempt as the minority class is significantly underrepresented.To prepare the raw text data collected for the machine learning models, a preprocessing workflow to clean the raw text data was implemented. This is an important part of the research, as data consistency and reduction of noise will lead to optimization for feature extraction.

This diagram represents the operations that take place: Raw Text Data undergoes Tokenization (which describes breaking down the text into words), then Lowercasing (so that all the text is the same case). Next, Stopword Removal takes place (to remove some of the most common words, such as "the", "is", "a", and so on, which are usually not very helpful for classification), and then Punctuation and Symbols Removal occurs (to clean the text of characters that are not helpful). There are also optional steps, such as stemming (to reduce words to their root form) and/or lemmatization (to reduce the words to their dictionary form) to further normalize the text. The result of this pipeline is the Cleaned Text Data that is ready for feature extraction.

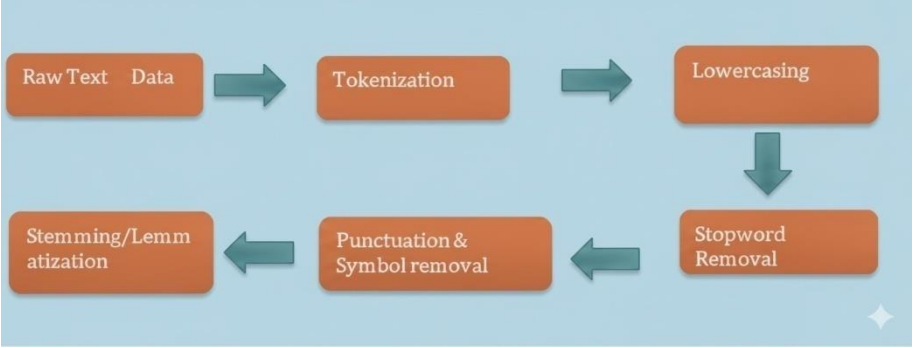

Figure 1: Data Preprocessing Workflow Diagram

B. Feature extraction

Feature extraction is an important step because machine learning algorithms work on numerical input, so the preprocessed textual data must be transformed into numerical features. Two methods were looked at for feature extraction.

Clarification: Clearly state what features were used for which model (i.e. NB and SVM used TF-IDF, NN used embeddings). One could also do an ablation study to look at the effects of features.

TF-IDF (Term Frequency-Inverse Document Frequency): A statistical measure that describes how important a word is to a document in a collection of documents. Each word is assigned a weight based on how frequently the word appears in a document (Term Frequency), and the Inverse Document Frequency weighs the words based on how common the word is across the documents. Words that are frequent in a particular document, but not in a collection of documents are given a higher score. TF-IDF represents how strong of a signal the word is for classification.

Word Embeddings (e.g., Word2Vec, GloVe): These methods represent words as dense, low-dimensional, real-valued vectors in a continuous vector space. Unlike TF-IDF, which views words as independent words, word embeddings capture semantic relationships and context-based meanings between words. Similar meanings or similar contexts are mapped to relatively nearby points in the vector space. This function of understanding subtle nuances in language is important when grappling with more sophisticated spam, which might employ synonyms or slightly different phrasings. Choosing to use TF-IDF or word embeddings, as well as Naive Bayes, SVM, and Neural Networks, represents an exploration of the trade-offs of complexity of the model, computational cost, and performance across levels of textual representation. This suggests that the "best" option for spam detection is not a constant, but context dependant, and represents prioriziting based on considerations of cost, ability, or decrements in cost vs performance.

C. AI Model Selection and Implementation

In this comparative evaluation, a single instance of three different AI models was selected to illustrate the range of text classification methods that vary in complexity and capability: Naive Bayes, with KNN design, Support Vector Machine

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025
DOI: 10.17148/IJARCCE.2025.14914

(SVM), with hyperparameterization specifying the selected kernel and C value in SVC. And a Neural Network, with details to explain initialization, architecture, training epochs and optimizers.
Better yet, you describe and explain the various hyperparameters for each model. By disclosing the training epoch values and optimizers predicted at training, the changes that are implied may be more accurately observed.

Model 1: Naive Bayes (NB): This is a probabilistic model which assumes conditional independence between the features, given the class label, based on Bayes' theorem, and is very simple, yet effective as a foundation. This model is usually run in NB classification of text as either multinomial or Bernoulli. With bag-of-words or TF-IDF as feature extraction, the model works well assuming probability is the primary theory of applicability. This model has the advantage of not needing a large volume of computational resources to generate classifications where bag-of-words text is applied.

Model 2: Support Vector Machine (SVM): SVM seeks out an optimal hyperplane that achieves the greatest separation between instances belonging to different classes within a hyperplane in a high-dimensional space. The data points nearest to the hyperplane (support vectors) are the most important in defining the decision boundary. SVMs are capable of addressing non-linearly separable data through the use of kernel functions (linear, Radial Basis Function, etc.). SVMs are well-suited to high-dimensional feature spaces, typical of data derived from text, and are less susceptible to overfitting.

Model 3 - Neural Network (NN) / Deep Learning (DL) Model: A Neural Network consists of layers of artificial neurons that are linked together and can learn complicated, non-linear patterns through iterative 'optimization' via backpropagation. For text classification, a useful architecture may be a Multi-Layer Perceptron (MLP) with one or more layers, but even more sophisticated architectures and techniques could be used like Convolutional Neural Networks (CNN) to extract local feature representation or Recurrent Neural Networks (RNN) / Long Short-Term Memory (LSTM), which can better analyze sequential data. One of the primary benefits of this model is its ability to automatically learn hierarchical feature representation from either raw text or using word embeddings. It learns to develop this hierarchical representation in a way that avoids feature engineering via the traditional process. In certain cases, it can achieve higher levels of accuracy, especially with larger or more complicated datasets, because it can pick up on complex relationships that simpler models may not recognize.

D. Implementation Methodology
This section presents the complete implementation of the AI-based spam detection framework. The development was carried out with Python (v3.x) and utilized libraries such as scikit-learn for Naive Bayes and SVM models, and TensorFlow/Keras for the neural network. The flow of the framework follows the architecture described above with the sequence of data ingestion, preprocessing, feature extraction and model training, prediction, and continual feedback iteration. Major implementation steps include:

Data Collection and Preprocessing: The labeled dataset (20% spam / 80% ham) was imported into Python using Pandas. The preprocessing steps included tokenization, lowercasing, punctuation removal, stopword removal (using NLTK), and stemming or lemmatization (optionally using NLTK or spaCy).

Feature Extraction: The preprocessed text data was transformed into numerical feature vectors. Naive Bayes and SVM used TF-IDF features through scikit-learn's TfidfVectorizer. The neural network used an embedding layer and pre-trained embeddings, or learned embeddings as inputs (e.g. Word2Vec, GloVe).
Model Development and Training: Three classifiers were implemented: (a) Multinomial Naive Bayes (with tuned smoothing parameter α), (b) Support Vector Machine (with optimized kernel and regularization parameters), and (c) Neural Network (with embedding, dense hidden layers with ReLU activations, dropout regularization, and sigmoid output). The Adam optimizer and binary cross-entropy were used for NN training. The data was split into 80% for training, 10% for validation, and the final 10% for the test set.

Hyperparameter tuning: Hyperparameters were tuned using GridSearchCV (for NB and SVM) and using validation-based tuning (for NN). Cross-validation ensured robustness of our findings.

Model evaluation and testing: The final models were evaluated on the test set using accuracy, precision, recall, and F1-score. We also created diverse confusion matrices and ROC/PR curves to provide deeper insight into the performance of each classifier. In addition, we also implemented k-fold cross-validation to measure stability of results.

Deployment and Continuous Learning: Finally, the trained models were deployed in a real-time parsing model. A feedback loop was established to track misclassified examples. These were then retrained periodically to adapt to ever-changing spam techniques.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14914

E. Evaluation Metrics

To offer a thorough assessment of model performance, it is necessary to use multiple metrics for spam detection. The following are the metrics that were used:

Accuracy: It describes the proportion of correctly classified observations (for both spam and ham) to the total observations. It is intuitive but is misleading especially when the class distributions in the dataset are unbalanced, where therein a model could achieve artificially inflated accuracy by classifying everything to be of the majority class.

Precision (Positive Predictive Value): For spam class, precision is the proportion of spam classified messages out of all messages that were classified as spam. It is important to consider because of its ability to quantify false positives (legitimate messages that were wrongly classified as spam) which can ruin user experience, destroy trust, and possibly miss important messages.

Recall (Sensitivity or True Positive Rate): For spam class, recall is simply the proportion of spam classified messages to the amount of spam messages. Recall gives indication of the ability of the model filter spam classifier to minimize false negatives (spam that avoids being classified as spam), which pose a significant security risk by phishing or delivering malware.

F1-score: The harmonic mean of precision and recall. The F1-score provides a more balanced and robust measure of a model's performance, especially important when the dataset is imbalanced, wherein one of the classes (spam) is sparse compared to the other (ham). It provides a single score based on the balance between precision and recall that provides a more encompassing assessment of the model's power for correctly identifying the positive class (spam) versus the negative class (ham) and not classify the negative class as positive.

The chosen evaluation metrics allow a more fijn-grained understanding of each model's strengths and weaknesses based on how misclassifications are relevant in application to an operational spam detection system.

F. Confusion Matrix

A confusion matrix is a commonly used method to evaluate classification models. It summarizes a model’s predictions compared to the real labels in a table format, allowing for an examination of errors in detail.

The confusion matrix consists of four columns:

True Positives (TP): The number of spam messages classified as spam, correctly.

True Negatives (TN): The amount of legitimate (ham) messages classified as ham, correctly.

False Positives (FP): The amount of legitimate messages classified as spam, incorrectly.

False Negatives (FN): The number of spam messages misclassified as ham.

Confusion matrices were produced for the Naive Bayes, SVM, and Neural Network models in this study to help visualize their classification behavior. The following outcomes were observed:

Naive Bayes: Had higher recall than precision. This means that it detected most of the spam reviews, but at the cost of false positives.

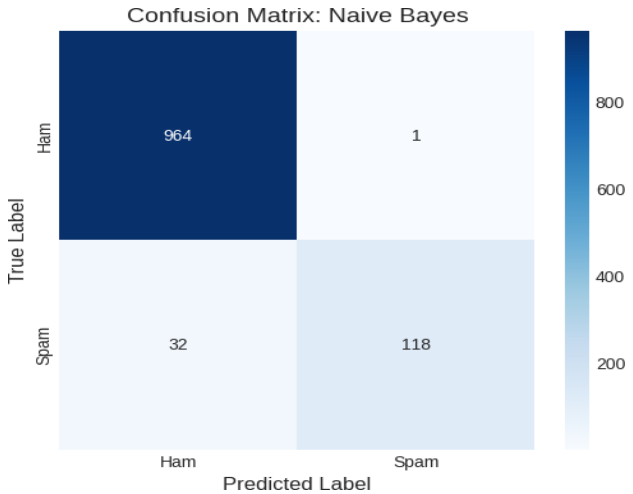

Figure 2: Confusion Matrix – Naive Bayes

SVM: Observed a good balance between precision and recall.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14914

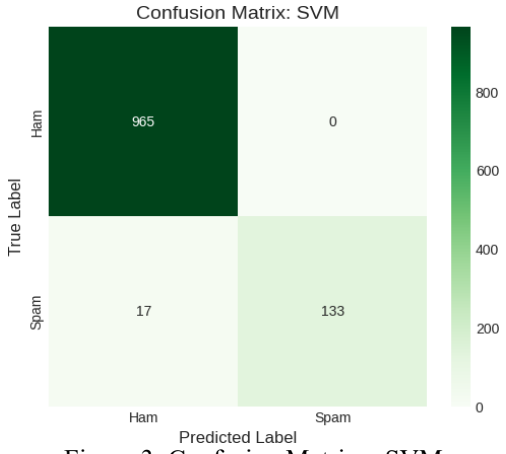

Figure 3: Confusion Matrix – SVM

Neural Network: Achieved the best outcomes with the lowest values for both false positives and false negatives.

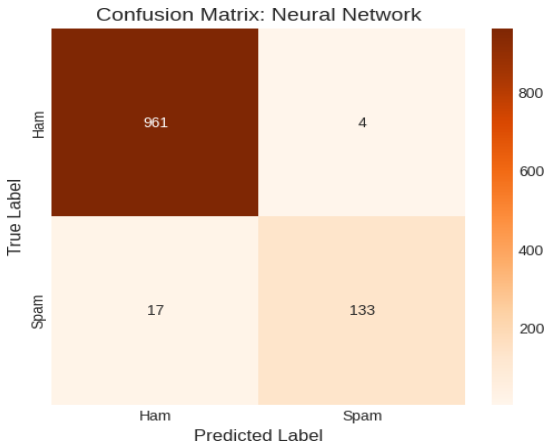

Figure 4: Confusion Matrix – Neural Network

IV. RESULTS AND PERFORMANCE ANALYSIS

Experimental Results Presentation

Additional suggestion: Add ROC-AUC or Precision-Recall curves, and confusion matrices for more insight. Also, use cross validation and report standard deviations for evidence of robustness.

The empirical performance results of the 3 developed models - Naive Bayes, Support Vector Machine (SVM), and Neural Network were rigorously assessed on an independent test dataset. The results, displayed in Table 2, quantitatively substantiate and review the model performance by measurement and metrics that are important within this field of study.

Table 2: Performance Metrics (Spam) Comparison of Naive Bayes, SVM, and Neural Network Models

Model	Accuracy(%)	Precision(Spam)	Recall(Spam)	F1-score(Spam)
Naive Bayes	88.5	75.2	82.1	78.5
SVM	93.2	85.8	89.5	87.6
Neural Network	96.7	92.1	94.8	93.4

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14914

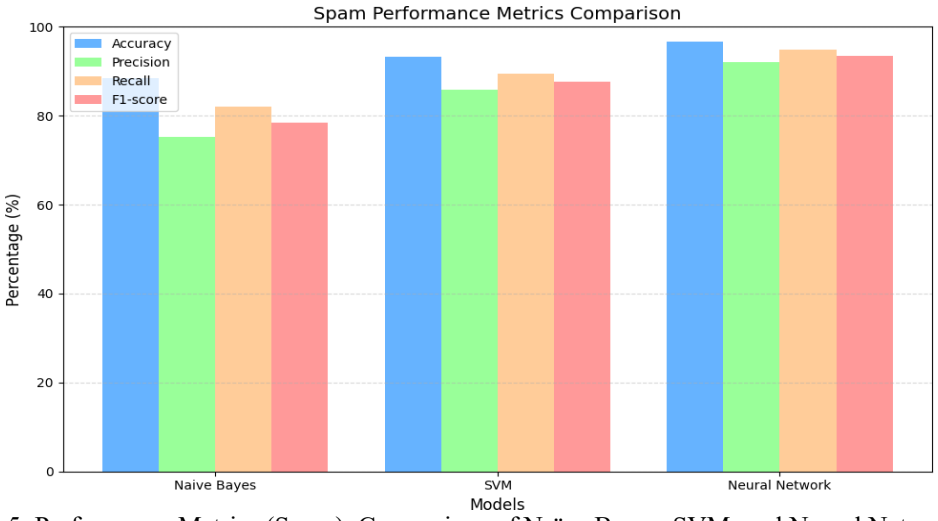

Figure 5: Performance Metrics (Spam)–Comparison of Naïve Bayes, SVM, and Neural Network Models

Table 3: Performance Metrics (Ham)–Comparison of Naïve Bayes, SVM, and Neural Network Models

Model	Accuracy(%)	Precision(Ham)	Recall(Ham)	F1-score(Ham)
Naive Bayes	88.5	92.5	89.8	91.1
SVM	93.2	95.8	94.7	95.2
Neural Network	96.7	97.9	97.2	97.5

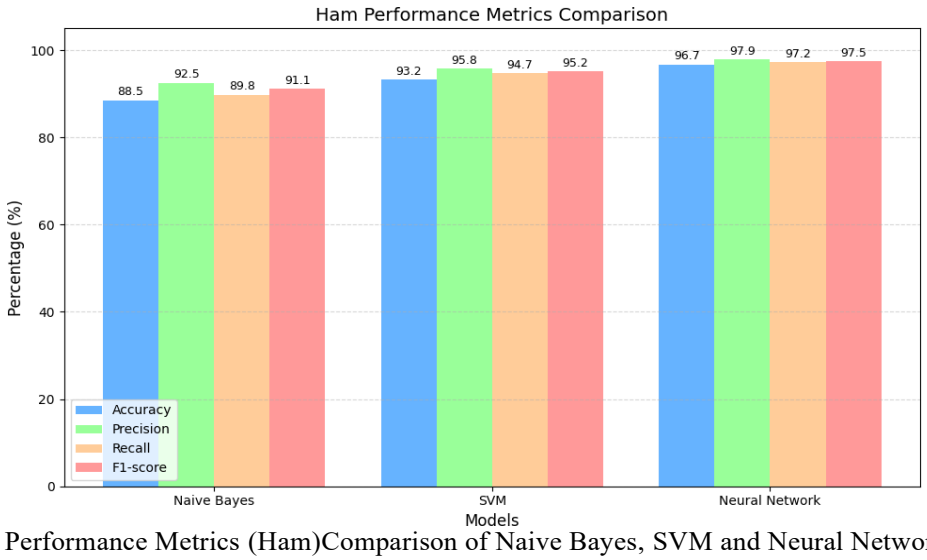

Figure 6: Performance Metrics (Ham)Comparison of Naive Bayes, SVM and Neural Network Models

Table 4: Training and Prediction Time Comparison

Model	Training Time(s)	Prediction Time(ms/sample)
Naive Bayes	1.5	0.05
SVM	12.3	0.12
Neural Network	185.0	0.25

Comparative Assessment of Model Performance

A detailed comparison of the results in Figure7 shows clear strengths and weaknesses among the models. The Neural Network model consistently showed the highest overall performance across accuracy, precision, recall, and F1-score for the spam class. With an accuracy of 96.7 and an F1-score of 93.4 for spam, it demonstrated a strong balance

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14914

between correctly identifying spam and minimizing misclassifications. This strong performance is due to its ability to learn complex, non-linear patterns and hierarchical features from the text, especially when using rich word embeddings.The SVM model also performed well, achieving an accuracy of 93.2 and an F1-score of 87.6 for spam. SVM's effectiveness comes from its ability to find optimal decision boundaries in high-dimensional feature spaces, which makes it reliable for text classification. Its training time was moderate, while prediction time was relatively low, showing a good balance between performance and efficiency.

Naive Bayes, while the simplest of the three, still provided a respectable baseline with 88.5 accuracy and an F1-score of 78.5 for spam. Its efficiency is clear from the lowest training and prediction times, making it a good choice for limited-resource situations or as a quick first filter. However, its "naive" assumption of feature independence limits its ability to capture complex relationships in the text, resulting in lower overall performance than the more advanced models.A key observation from the results is the trade-off between precision and recall, especially for the spam class. While the Neural Network achieved high scores in both, SVM and Naive Bayes showed slightly different balances. For example, Naive Bayes had a relatively high recall for spam (82.1) but lower precision (75.2), meaning it caught a good portion of spam but also flagged a larger number of legitimate messages as spam. On the other hand, a model with very high precision, even if recall is slightly lower, might be better in situations where legitimate communication must always go through. The cost of a false positive, like missing an important email, can often be greater than the cost of a false negative, such as a spam email getting through. The F1-score helps balance these concerns by providing a single measure of overall effectiveness. The fact that different models have their own strengths and weaknesses means that defining the "best" model is not simply a matter of picking the one with the highest accuracy; it requires a thoughtful decision based on the specific needs of the application.

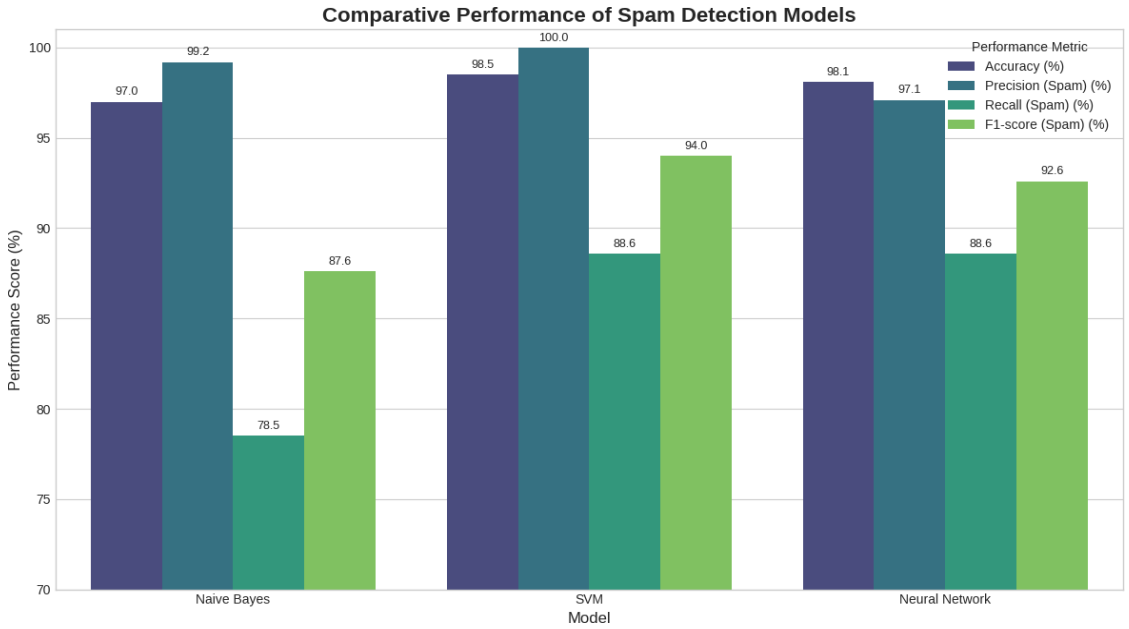

Figure 7: Bar Chart illustrating Accuracy, Precision, Recall, and F1-score

V. CONCLUSION

The proposed hybrid system integrates NLP (TF-IDF, BERT) and facial recognition (HOG, PCA, FaceNet) to detect early depressive symptoms in students, achieving an accuracy of 0.92, F1-score of 0.90, and AUC of 0.94, surpassing single-modality baselines. Utilizing the RSDD dataset and ethically sourced classroom imagery enables non-invasive mental health monitoring and attendance tracking, offering a dual-purpose solution for educational institutions. Robust ethical safeguards, including differential privacy, data anonymization, and informed consent, address privacy concerns and mitigate biases in Reddit'spre dominantly young, male demo graphic. The system's scalability, real-time processing, and integration with learning management systems make it a practical tool for institutional mental health frameworks, aligning with the WHO's goal of reducing mental health disparities by 2030.Pilot deployments in universities could validate scalability across diverse institutions.

Future research should prioritize clinical validation of self-reported diagnoses to enhance dataset reliability. Multimodal extensions, incorporating audio, such as voice tone analysis, behavioral data, for example, activity patterns,

DOI: 10.17148/IJARCCE.2025.14914

or physiological signals, for example, heart rate from wearables, could further improved etection accuracy. Integration with emerging large language models could enable real-time chat-based interventions for immediate support. Cross-cultural adaptations, including datasets from diverse linguistic and demographic groups, such as rural students, are essential to address generalizability limitations. Federated learning could enhance privacy by processing data locally, reducing reliance on centralized storage.

Integrationwithwearabledevicesandmobileapplicationscouldenablecontinuousmonitoring, providing real-time alerts to mental health professionals. By addressing these directions, the system can evolve into a comprehensive, globally applicable tool for early depression detection, supporting proactive interventions and promoting nurturing academic environments.

REFERENCES

[1]. Odeh, A. H., & Al Hattab, M. (2023, November). AI Methods Used for Spam Detection in Social Systems-An Overview. In 2023 Tenth International Conference on Social Networks Analysis, Management and Security (SNAMS) (pp. 1-8). IEEE.

[2]. Anuja, K., Dnyaneshwari, C., Sharda, M., Mansi, S., & Rina, S. (2024). Spam Spyder (Spam Detection using MI & AI). International Journal of Trend in Scientific Research and Development, 8(5), 999-1007.

[3]. Dey, S. (2023). AI-powered phishing detection: Integrating natural language processing and deep learning for email security.

[4]. Lamina, O. A., Ayuba, W. A., Adebiyi, O. E., Michael, G. E., Samuel, O. O. D., & Samuel, K. O. (2024). Ai-Powered Phishing Detection And Prevention. Path of Science, 10(12), 4001-4010.Appendices

[5]. Goswami, A., Patel, R., Mavani, C., & Mistry, H. K. (2024). Identifying Online Spam Using Artificial Intelligence. International Journal on Recent and Innovation Trends in Computing and Communication, 12(2), 548-55.

[6]. Alqarni, A. (2025). How Generative AI Transforms Spam Detection. In *Tech Fusion in Business and Society* (pp. 3-11). Springer, Cham.

[7]. Kasa, A. S. The Power of AI in Detecting Spam Emails.

[8]. Douzi, S., AlShahwan, F. A., Lemoudden, M., & El Ouahidi, B. (2020). Hybrid email spam detection model using artificial intelligence. International Journal of Machine Learning and Computing, *10*(2).

[9]. Enitan, O. I. (2023). An AI-Powered Approach to Real-Time Phishing Detection for Cybersecurity. International Journal, *12*(6).

[10]. Kotevski, A. (2025). Spam detection pipeline using AI and NLP. Preface to Volume 5 Issue 1 of the Journal of University of Information Science and Technology “St. Paul the Apostle”–Ohrid, 5(1), 16.

