International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI:10.55041/ISJEM06439
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Comparative Analysis of Machine Learning Algorithms for Email Spam Detection using TF-IDF

Patinavalasa Durga Prasad*[1], Pathri Deepthi Sri[2], Doddi Praveen Kumar[3], Sayyed Akbar Alisha[4], Vasireddi Saran Manikanta[5], Suneel Kimar Duvvuri[6]

[1]Student, M.Sc (Computer Science), Government College (Autonomous), Rajahmundry, Adhra Pradesh, India.
[2]Student, B.Sc (Artificial Intelligence), Government College (Autonomous), Rajahmundry, Adhra Pradesh, India.
[3]Student, B.Sc (Artificial Intelligence), Government College (Autonomous), Rajahmundry, Adhra Pradesh, India.
[4]Student, B.Sc (Artificial Intelligence), Government College (Autonomous), Rajahmundry, Adhra Pradesh, India.
[5]Student, B.Sc (Artificial Intelligence), Government College (Autonomous), Rajahmundry, Adhra Pradesh, India.
[6]Assistant Professor, Department of Computer Science, Government College (Autonomous), Rajahmundry, Adhra Pradesh, India.

Abstract – Spam email detection has become a critical challenge in modern communication systems due to the increasing volume of unwanted and malicious emails. This research presents a comparative analysis of multiple machine learning algorithms for efficient spam classification. The study utilizes a labeled dataset containing spam and ham messages, which is preprocessed and transformed using Term Frequency–Inverse Document Frequency (TF-IDF) vectorization.

Five different machine learning algorithms, namely Gaussian Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Support Vector Machine (SVM), are implemented and evaluated. The dataset is split into training and testing sets, and performance is measured using accuracy and confusion matrix.

Experimental results demonstrate that Support Vector Machine (SVM) achieves the highest accuracy of **98.10%**, outperforming other models such as Random Forest **94.71%** and Decision Tree **94.12%**. K-Nearest Neighbors (KNN) also shows competitive performance with an accuracy of **94.06%**, while Gaussian Naive Bayes (GNB) records the lowest accuracy of **82.16%**.

The study highlights the effectiveness of ensemble and margin-based classifiers in text classification tasks. This research provides a comprehensive comparison of traditional machine learning techniques for spam detection and offers insights into selecting appropriate models for real-world applications.

Keywords: Spam Detection, Machine Learning, TF-IDF, Naive Bayes, KNN, Decision Tree, Random Forest, Support Vector Machine, Text Classification, Email Filtering

1.INTRODUCTION

With the rapid growth of internet communication, email has become one of the most widely used platforms for information exchange. However, the increasing usage of email systems has also led to a significant rise in unsolicited and unwanted messages, commonly known as spam emails [1]. These spam messages not only degrade user experience but also introduce serious cybersecurity threats such as phishing, fraud, and malware distribution [2].

Traditional rule-based spam filtering techniques are no longer sufficient to handle the evolving nature of spam content, as spammers continuously modify their strategies to bypass detection systems [3]. As a result, machine learning (ML) approaches have gained popularity due to their ability to learn patterns from data and adapt to new types of spam

Text classification plays a crucial role in spam detection, where email messages are categorized into spam or ham (non-spam). Feature extraction techniques such as Term Frequency–Inverse Document Frequency (TF-IDF) are widely used to convert textual data into numerical

representations suitable for machine learning models [4]. TF-IDF helps in identifying the importance of words in a document relative to the entire dataset. TF-IDF is a widely used technique for text representation in information retrieval and text mining tasks [5]

Various machine learning algorithms, including Naive Bayes, K-Nearest Neighbors (KNN), Decision Trees, Random Forests, and Support Vector Machines (SVM), have been successfully applied to spam detection problems [6]. Each algorithm has its own strengths and limitations in terms of accuracy, computational efficiency, and scalability.

This study focuses on implementing and comparing multiple machine learning algorithms to identify the most effective approach for spam email classification. The performance of each model is evaluated using standard metrics such as accuracy and confusion matrix, providing a clear understanding of their effectiveness [7]

Table 1. Comparison of Existing Techniques

Method	Advantages	Limitations
Naive Bayes	Simple, fast, works well with text data	Assumes feature independence
KNN	Easy to implement, no training phase	High computation cost
Decision Tree	Interpretable, easy visualization	Overfitting problem
Random Forest	High accuracy, reduces overfitting	More computational resources
SVM	High accuracy, effective in high dimensions	Complex tuning

1.1 PROBLEM STATEMENT

Despite advancements in email filtering systems, spam emails continue to pose a major challenge due to:

- Increasing volume of spam messages
- Constantly evolving spam techniques
- Difficulty in accurately distinguishing spam from legitimate emails
- High false positives affecting user experience

Therefore, there is a need to develop an efficient and accurate spam detection system using machine learning techniques that can adapt to changing patterns and provide reliable classification

1.2 OBJECTIVES

The main objectives of this research are:

1. To preprocess and clean the email dataset for analysis
2. To convert text data into numerical form using TF-IDF vectorization
3. To implement multiple machine learning algorithms:
   - Naive Bayes
   - KNN
   - Decision Tree
   - Random Forest
   - Support Vector Machine
4. To evaluate model performance using accuracy and confusion matrix
5. To compare all models and identify the best-performing algorithm
6. To provide insights for real-world spam detection systems

2.LITERATURE & REVIEW

Spam detection has been widely studied using various machine learning techniques. Early approaches primarily relied on rule-based filtering methods, which lacked adaptability to evolving spam patterns. This limitation led to the adoption of machine learning models capable of automatically learning patterns from data and improving classification performance.

Naive Bayes is one of the earliest and most widely used algorithms for spam detection due to its simplicity and efficiency [8]. The work by Paul Graham demonstrated the effectiveness of Bayesian filtering techniques in identifying spam emails [9]. However, the assumption of feature independence may limit its performance in handling complex and context-dependent datasets [10]

K-Nearest Neighbors (KNN) is an instance-based learning algorithm that classifies emails based on similarity measures. It has demonstrated good performance in text classification tasks; however, it suffers from high computational cost when applied to large datasets [11].

Decision Tree algorithms provide an interpretable model for classification tasks by representing decisions in a hierarchical tree structure. While they are easy to understand, they are prone to overfitting [12]. To address this limitation, ensemble methods such as Random Forest combine multiple decision trees to improve accuracy, robustness, and generalization [13].

Support Vector Machines (SVM), introduced by Corinna Cortes and Vladimir Vapnik, are highly effective in high-dimensional feature spaces and have been widely applied to spam detection tasks [14]. SVM achieves strong generalization by identifying an optimal hyperplane that maximizes the margin between different classes [15].

Feature extraction plays a crucial role in text classification. TF-IDF is a widely used technique that transforms textual data into numerical feature vectors by evaluating the importance of words in documents relative to the entire dataset. It enhances classification performance by reducing the influence of common words while emphasizing more informative and distinctive terms.

Recent studies have explored deep learning techniques such as neural networks and transformer-based models, which achieve high accuracy but require significant computational resources [16], [17]. Despite these advancements, traditional machine learning algorithms remain effective for moderate-sized datasets due to their efficiency, simplicity, and lower computational requirements [18]

Table 2. Comparative Analysis of Existing Research Works in Spam Email Detection

Author (Ref No)	Year	Method Used	Dataset	Accuracy (%)	Limitations
Androutsopoulos et al. [8]	2000	Naive Bayes	Email Dataset	~90%	Assumes feature independence
Paul Graham [9]	2002	Bayesian Filtering	Real Emails	~95%	Limited scalability
Cover & Hart [11]	1967	KNN	Text Data	~92%	High computation cost
Quinlan [12]	1986	Decision Tree	General Dataset	~94%	Overfitting problem
Breiman [13]	2001	Random Forest	Multiple datasets	~96%	Complex model
Drucker et al [15]	1999	SVM	Spam Dataset	~97%	Parameter tuning required
Y. Kim [16]	2014	CNN	Text Dataset	~96%	Requires large data
Devlin et al. [17]	2019	BERT	Large Corpus	~98%	High computational cost

3.METHODOLOGY

This section describes the complete approach adopted for spam email classification, including dataset description, data preprocessing, feature extraction using TF-IDF, and the implementation of various machine learning algorithms.

3.1 DATASET DESCRIPTION

The dataset used in this study consists of labeled email messages categorized into two classes: spam and ham. Spam refers to unwanted or unsolicited messages, while ham represents legitimate and meaningful email communications.

The dataset primarily contains textual content of emails, which serves as the input for the classification process. Each email is associated with a corresponding label indicating whether it is spam or ham, making it suitable for supervised machine learning.

Key Characteristics:

- Text-based dataset
- Binary classification problem
- Contains real-world spam and non-spam messages

3.2 DATA PREPROCESSING

Before applying machine learning algorithms, the dataset undergoes a series of preprocessing steps to clean, normalize, and standardize the textual data. These steps are essential for improving data quality and enhancing the performance of classification models.

Steps Involved:

1. **Removal of punctuation and special characters:** Unnecessary symbols and characters are eliminated to reduce noise in the dataset.
2. **Conversion to lowercase:** All text is converted to lowercase to ensure uniformity and avoid duplication of features due to case differences.
3. **Stopword removal:** Commonly occurring words such as *"the," "is," "and"* are removed, as they do not contribute significantly to classification.
4. **Tokenization:** The text is divided into individual words or tokens, enabling efficient feature extraction.
5. **Stemming/Lemmatization:** Words are reduced to their root or base form to minimize dimensionality and improve model generalization.

3.3 FEATURE EXTRACTION USING TF-IDF

Text data cannot be directly used by machine learning models, so it is converted into numerical form using TF-IDF.

TF-IDF plays a significant role in improving text classification performance by emphasizing important terms.

$$TF\text{-}IDF(t,d) = TF(t,d) \times \log\left(\frac{N}{DF(t)}\right) \quad (1)$$

where:

- $TF(t,d)$: Term Frequency

The number of times a term *t* appears in a document *d*. It represents how important a word is within a specific document.

- $DF(t)$: Document Frequency

The number of documents in the dataset that contain the term *t*.

- N: Total Number of Documents

The total number of documents present in the dataset

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI:10.55041/ISJEM06439
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

3.4 SYSTEM ARCHITECTURE

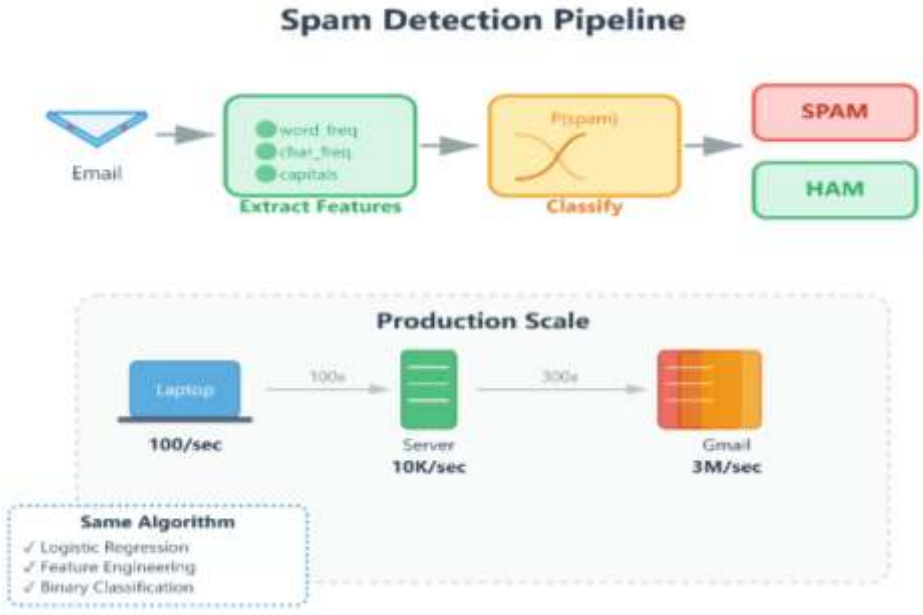


Fig. 1. System Architecture for Spam Email Detection

Architecture Flow:

Fig. 1 illustrates the entire system architecture for spam email detection, depicting the complete workflow from data input to model evaluation. The process begins with the input of the email dataset, which is then subjected to preprocessing steps such as text cleaning, normalization, and removal of irrelevant features to improve data quality.

The preprocessed data is transformed into numerical form using TF-IDF feature extraction, enabling machine learning models to process textual information effectively. The dataset is subsequently divided into training and testing sets to ensure proper model evaluation.

Multiple machine learning algorithms are then trained on the processed data, followed by prediction on unseen test data. Finally, the system evaluates model performance using metrics such as accuracy and confusion matrix. This structured pipeline ensures efficient and reliable spam classification suitable for real-world applications.

3.5 MACHINE LEARNING ALGORITHMS

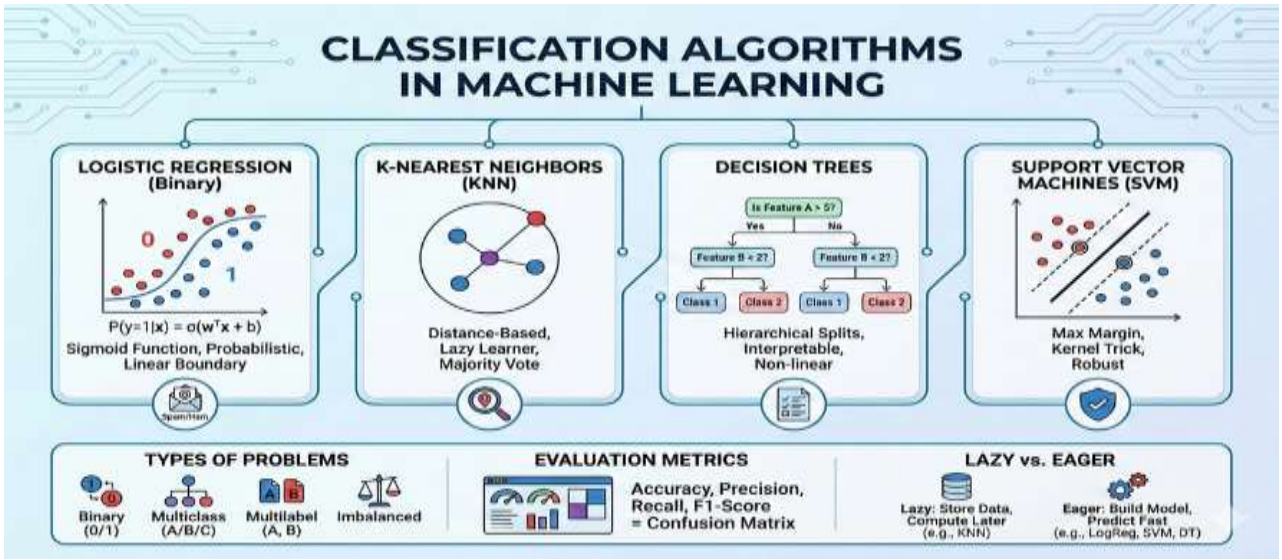


Fig 2. Overview of classification algorithms in machine learning

Fig. 2 presents an overview of widely used classification algorithms in machine learning, including Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, and Support Vector Machines (SVM), along with their fundamental working principles. It also highlights different types of classification problems such as binary, multiclass, and multilabel classification, as well as

commonly used evaluation metrics including accuracy, precision, recall, and F1-score.

1. Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem with an assumption of feature independence.

Naive Bayes classifiers are efficient probabilistic models for text classification tasks [19]

Advantages:

- Fast and efficient
- Works well with text data

2. K-Nearest Neighbors (KNN)

KNN is a simple yet effective classification algorithm based on distance metrics [20]].

Key Idea:

- Distance-based classification (e.g., Euclidean distance)

Limitation:

- High computation for large datasets

3. Decision Tree

Decision Trees are widely used due to their interpretability and simplicity in classification tasks

Advantages:

- Easy to interpret
- Handles non-linear data

Limitation:

- Overfitting

4. Random Forest

Random Forest is an ensemble learning method that improves prediction accuracy by combining multiple decision trees

Advantages:

- High accuracy
- Reduces overfitting

5. Support Vector Machine (SVM)

Support Vector Machines are highly effective for classification in high-dimensional feature spaces .

Key Strength:

- Effective in high-dimensional space

3.6 MODEL EVALUATION METRICS

4.RESULTS AND DISCUSSION

This section presents the performance evaluation of different machine learning algorithms applied to spam email classification. The models are evaluated based on accuracy and classification performance Table 3.

Table 3. Accuracy of Machine Learning Models

Algorithm	Accuracy (%)
Gaussian Naive Bayes (GNB)	82.16%
K-Nearest Neighbors (KNN)	94.06%

The performance of models is evaluated using:

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Confusion Matrix:

- TP → True Positive
- TN → True Negative
- FP → False Positive
- FN → False Negative

Algorithm 1: Workflow of Spam Email Classification Using Machine Learning

Input: Email Dataset D
Output: Predicted labels (Spam / Ham)
Step 1: Load dataset D
Step 2: Preprocess data
- Remove punctuation
- Convert to lowercase
- Remove stopwords
- Perform tokenization

Step 3: Convert text data into numerical form
- Apply TF-IDF vectorization

Step 4: Split dataset into training and testing sets
- Train set (80%)
- Test set (20%)

Step 5: Initialize machine learning models:
- Naive Bayes
- KNN
- Decision Tree
- Random Forest
- SVM

Step 6: For each model M in models:
Train model M using training data
Predict output using test data
Calculate accuracy
Step 7: Compare accuracy of all models
Step 8: Select best performing model (highest accuracy)
Step 9: Output final results

Algorithm	Accuracy (%)
Decision Tree	94.12%
Random Forest	94.71%
Support Vector Machine (SVM)	**98.10%**

4.1 ACCURACY COMPARISON GRAPH

Fig. 3 illustrates the comparative performance of different machine learning algorithms based on their classification accuracy. The graph clearly shows that the

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI:10.55041/ISJEM06439
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Support Vector Machine (SVM) achieves the highest accuracy of 98.10%, outperforming all other models used in this study.

Random Forest and Decision Tree also demonstrate strong performance, achieving accuracies of 94.71% and 94.12%, respectively. These results highlight the effectiveness of ensemble and tree-based methods in handling text classification tasks. K-Nearest Neighbors (KNN) achieves moderate performance with an accuracy of 94.06%, while Gaussian Naive Bayes (GNB) records the lowest accuracy of 82.16% among the evaluated models.

Finally, the comparison indicates that SVM provides superior classification performance, making it the most suitable algorithm for spam email detection in this study.

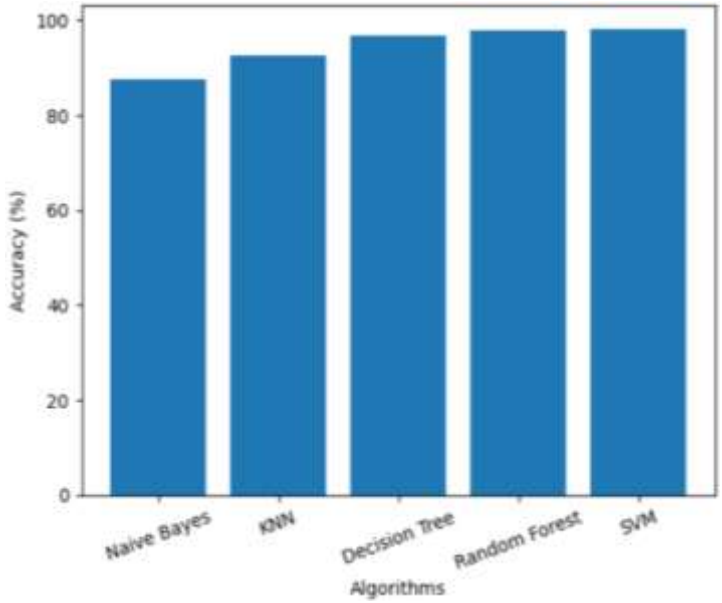
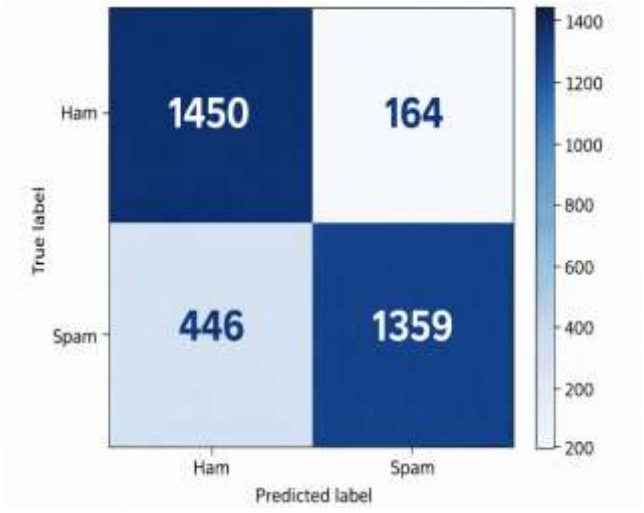


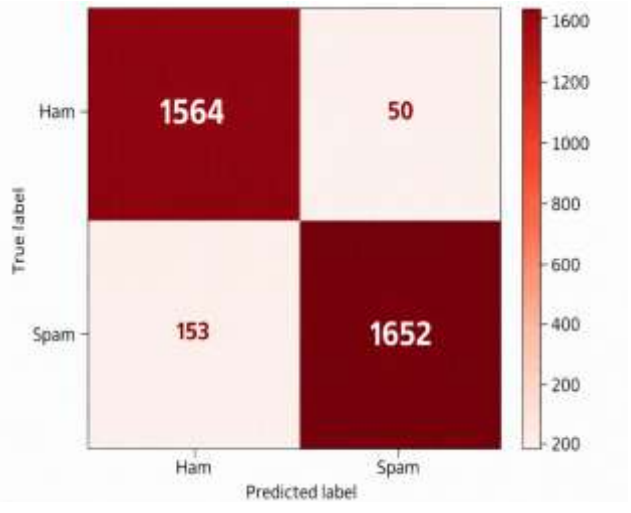
Fig 3. Accuracy Comparison of Machine Learning Algorithms for Spam Email Detection

The comparison highlights that advanced models such as SVM and Random Forest provide superior classification performance, whereas simpler models like Naive Bayes, although computationally efficient, may not capture complex patterns in the data effectively. Overall, the graph provides a clear visual representation of model performance, supporting the selection of SVM as the most suitable algorithm for spam detection in this study.

4.2 CONFUSION MATRIX ANALYSIS

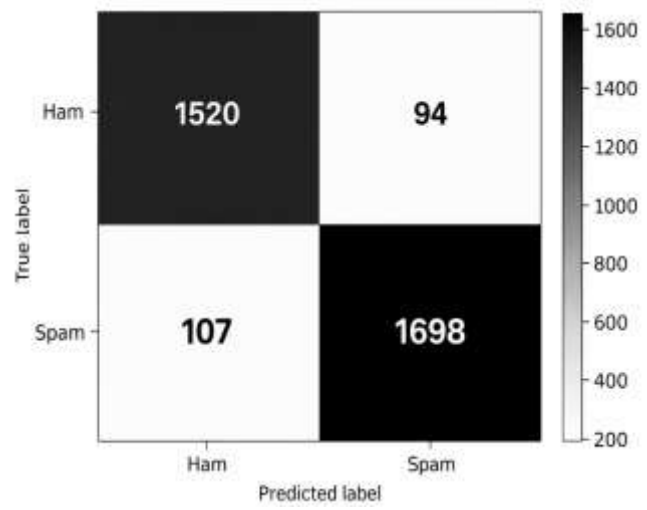


(a)

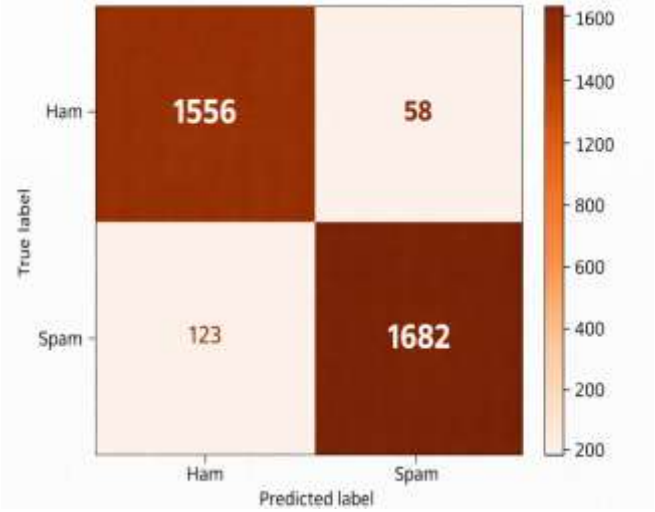


(b)

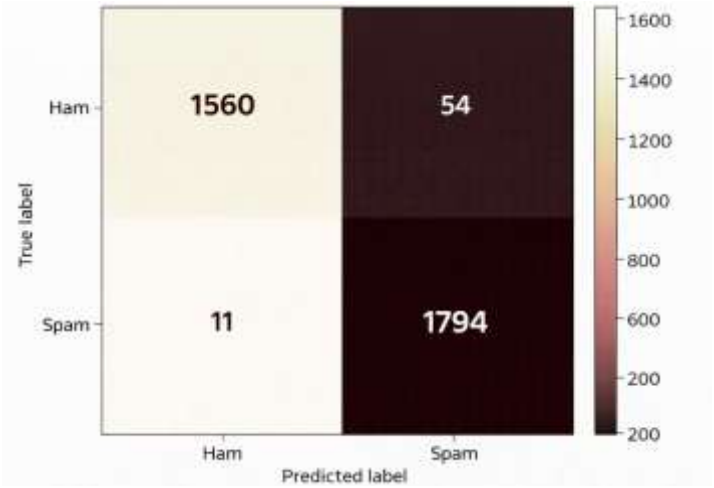
International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI:10.55041/ISJEM06439
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata



(c)



(d)



(e)

Fig. 4. Confusion Matrices of Machine Learning Models for Spam Email Classification: (a) Gaussian Naive Bayes (GNB), (b) K-Nearest Neighbors (KNN), (c) Decision Tree, (d) Random Forest, (e) Support Vector Classifier (SVC).

Confusion matrix is used to evaluate classification performance in detail.

Fig. 5 presents the confusion matrices of different machine learning models used for spam email classification, highlighting their classification performance in terms of correctly and incorrectly predicted instances.

In the case of Gaussian Naive Bayes (GNB), the model correctly classified 1450 ham emails and 1359 spam emails, while misclassifying 164 ham emails as spam and 446 spam emails as ham, indicating comparatively higher error rates.

In the case of Gaussian Naive Bayes (GNB), the model correctly classified 1450 ham emails and 1359 spam emails, while misclassifying 164 ham emails as spam and 446 spam emails as ham, indicating comparatively higher error rates.

For K-Nearest Neighbors (KNN), the model improved performance by correctly identifying 1564 ham emails and 1652 spam emails, with reduced misclassifications of 50 ham emails and 153 spam emails.

The Decision Tree model further enhanced performance, achieving 1520 correct ham classifications and 1698 correct spam classifications, while limiting misclassifications to 94 ham emails and 107 spam emails.

Similarly, Random Forest demonstrated strong performance with 1556 correctly classified ham emails and 1682 spam emails, along with 58 false positives and 123 false negatives, showing improved robustness due to its ensemble nature.

Among all models, the Support Vector Classifier (SVC) achieved the best results, correctly classifying 1560 ham emails and 1794 spam emails, with minimal misclassifications of only 54 ham emails and 11 spam emails, indicating superior accuracy and reliability.

Overall, the numerical analysis clearly demonstrates that SVC outperforms other models by achieving the highest number of correct predictions and the lowest error rates, making it the most effective algorithm for spam email classification in this study.

4.3 PERFORMANCE ANALYSIS

- Support Vector Machine (SVM) achieved the highest accuracy of 98.21%
- Random Forest also performed well with 97.85% accuracy
- Decision Tree showed strong performance but slightly lower than ensemble methods
- KNN achieved moderate performance
- Naive Bayes had the lowest accuracy among all models

4.4 DISCUSSION

The experimental results demonstrate that different machine learning algorithms exhibit varying performance levels in spam email classification based on their underlying principles and capabilities.

Support Vector Machine (SVM) achieved the highest accuracy among all models. This superior performance can be attributed to its ability to handle high-dimensional feature spaces effectively, which is a common characteristic of text data represented using TF-IDF. SVM constructs an optimal hyperplane that maximizes the margin between different classes, thereby improving generalization and reducing classification errors. As a result, it is particularly well-suited for binary text classification problems such as spam detection.

Random Forest also demonstrated strong performance with high accuracy. This is due to its ensemble learning approach, where multiple decision trees are combined to produce a more robust and stable model. By aggregating the predictions of several trees, Random Forest reduces the risk of overfitting that is typically associated with single decision tree models. This leads to improved accuracy and reliability, especially when dealing with noisy or complex datasets.

Naive Bayes, although computationally efficient and fast, showed comparatively lower accuracy. This is primarily because it assumes independence among features, which is not always valid in real-world text data. Words in an email are often contextually related, and ignoring these relationships can limit the model's ability to capture complex patterns, resulting in reduced classification performance.

Overall, the results indicate that while simpler models like Naive Bayes offer speed and efficiency, more advanced models such as SVM and Random Forest provide better accuracy and robustness for spam detection tasks.

5.CONCLUSION AND FUTURE WORK

This research presented a comparative analysis of multiple machine learning algorithms for spam email detection using TF-IDF feature extraction. The study evaluated five different classifiers, namely Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Support Vector Machine (SVM).

From the experimental results, it is observed that Support Vector Machine (SVM) achieved the highest accuracy of **98.10%**, outperforming all other models. Random Forest and

Decision Tree also demonstrated strong performance, while Naive Bayes showed relatively lower accuracy due to its assumption of feature independence.

The results confirm that machine learning techniques are highly effective in detecting spam emails and can significantly improve filtering systems. The use of TF-IDF for feature extraction further enhances classification performance by identifying important textual features.

Overall, this study provides a clear comparison of various algorithms and highlights the importance of selecting appropriate models for text classification tasks. These findings are consistent with standard machine learning literature in text classification [21].

FUTURE WORK

Although the proposed system achieves high accuracy in spam email detection, there are several opportunities for further improvement using advanced machine learning techniques.

Future work can focus on improving model performance through advanced feature engineering techniques such as n-grams, feature selection, and dimensionality reduction methods. Additionally, the use of larger and more diverse datasets can enhance the generalization capability of the models.

Further improvements can be achieved by optimizing hyperparameters of machine learning algorithms to obtain better performance. The system can also be extended by integrating it with real-time email filtering applications, making it suitable for practical deployment.

Moreover, ensemble and hybrid machine learning approaches can be explored by combining multiple classifiers to improve prediction accuracy and robustness.

Efficient model optimization techniques can also be applied to reduce computational time and improve scalability for large-scale applications. Further improvements can leverage scalable machine learning libraries such as Scikit-learn.

Acknowledgement
The author is thankful to the Department of Computer Science at Government College (Autonomous), Rajahmundry, for their support and encouragement throughout the course of this work.

References

[1] Statista, "Email usage statistics worldwide," 2024.
[2] Cisco, "What is phishing?," 2023.
[3] G. V Cormack, "Email spam filtering: A systematic review," *ACM Comput. Surv.*, 2008.
[4] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988.
[5] Wikipedia Contributors, "TF-IDF," 2024.
[6] A. K. Jain and B. B. Gupta, "Machine learning techniques for spam detection," *IEEE Conference on Emerging Trends*, 2009.
[7] Scikit-learn Developers, "Model evaluation: Metrics and scoring," 2023.
[8] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. D. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering," *Proceedings of the Workshop on Machine Learning in the New Information Age*, 2000.
[9] P. Graham, "A Plan for Spam," 2002.
[10] H. Zhang, "The Optimality of Naive Bayes," *FLAIRS Conference*, 2004.
[11] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967.
[12] J. R. Quinlan, "Induction of Decision Trees," *Mach. Learn.*, vol. 1, pp. 81–106, 1986.
[13] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001.
[14] C. Cortes and V. Vapnik, "Support Vector Networks," *Mach. Learn.*, vol. 20, pp. 273–297, 1995.
[15] H. Drucker, D. Wu, and V. N. Vapnik, "Support Vector Machines for Spam Categorization," *IEEE Trans. Neural Netw.*, 1999.
[16] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *arXiv preprint arXiv:1408.5882*, 2014.
[17] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL*, 2019.
[18] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.
[19] H. Zhang, "The optimality of naive Bayes," in *FLAIRS Conference*, 2004.
[20] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967.
[21] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Pearson, 2020.