Comparative Study of Naïve Bayes and Support Vector Machine for Email and Sms Spam Detection

Vishakha Bhagwan Damodhar[1], Prof. Sonali Kiran Shewale[2]
[1]*B-tech (Final year), Computer Science & Engineering, P.E.S. College of Engineering Aurangabad, Maharashtra, India*
[2]*Department of Computer Science & Engineering P.E.S. College of Engineering Aurangabad, Maharashtra, India*

***Abstract*—The rapid rise in digital communication through emails and SMS has made spam detection a key focus in Natural Language Processing (NLP). Spam messages waste users' time and create security risks like phishing, identity theft, and malware attacks. This paper presents a study and implementation of spam classification using two supervised machine learning algorithms: Naïve Bayes and Support Vector Machines (SVM). A preprocessed dataset with labeled spam and ham messages was used to train and test the models. Experimental results show that SVM achieves higher accuracy and precision, while Naïve Bayes works faster and uses fewer computing resources. The findings suggest that a combined approach could leverage the strengths of both algorithms for real-time use in mobile and cloud-based platforms.**

***Index Terms*—Spam Classification, Naïve Bayes, SVM, NLP, Email Filtering, SMS Detection**

I. INTRODUCTION

The growth of the internet and mobile communication has led to the widespread exchange of information on various digital platforms. However, this progress has also caused an increase in spam messages. Recent studies estimate that nearly 45 to 50% of global email traffic is spam. This results in economic losses, privacy breaches, and potential cybersecurity risks. Spam emails and SMS often lead to phishing, spread harmful links, cause financial fraud, and promote unwanted content.

Traditional rule-based and keyword-matching systems are not enough. Spammers use tricks like misspellings, random characters, and hidden URLs to get past filters. To tackle these issues, machine learning (ML) methods are becoming more popular. These methods can learn
patterns from labeled datasets on their own and adjust to new types of spam with little human help.

This research primarily focuses on comparing the performance of Naïve Bayes (NB) and Support Vector Machine (SVM) classifiers, two of the most widely used algorithms in spam detection tasks. While NB is computationally efficient and interpretable, SVM provides robustness in handling complex, high-dimensional text data.
With the rapid growth of digital communication, spam messages in emails and SMS have become a major issue affecting both individuals and organizations. Spam can include phishing attempts, advertisements, malicious links, and fraudulent content, causing financial loss, privacy breaches, and reduced productivity. Detecting spam automatically is crucial to maintaining secure communication systems.
Spam detection methodologies can typically be categorized as follows:

- Keyword-Based Filtering: Detects spam messages based on certain keywords or phrases. Simple but easily bypassed by obfuscation techniques.
- Blacklists/Whitelists: Blocks known spam senders or allows only approved senders.
- Heuristic-Based Detection: Uses predefined rules based on message structure, frequency, or sender behavior.
- Machine Learning-Based Detection: Learns patterns from labeled datasets to classify messages as spam or ham.

This approach adapts to evolving spam tactics and is

the focus of this study.

Common Machine Learning Algorithms for Spam Detection

Researchers have applied a variety of machine learning algorithms to detect spam, including:

- Naïve Bayes (Multinomial NB):
  - A probabilistic classifier based on Bayes' theorem.
  - Assumes independence between features (words) in the message.
  - Efficient for text classification and performs well even with limited training data.
- Support Vector Machines (SVM):
  - Constructs a hyperplane to separate spam and ham messages in high-dimensional space.
  - Effective for sparse, high-dimensional text data.
  - Robust against overfitting and capable of handling complex patterns.
- Decision Trees and Random Forests:
  - Useful for interpretable classification and feature importance analysis.
  - Random Forests combine multiple trees for improved accuracy.
- Deep Learning Models (CNN, LSTM):
  - Capture semantic and contextual information from messages.
  - Require more data and computational resources but show high accuracy in large datasets.

The changing nature of spam messages, often called "concept drift," presents a big challenge. Spammers continuously update their content, patterns, and delivery methods to avoid detection. This requires models that can learn and adapt over time for effective spam filtering. Recent progress in deep learning, particularly with Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), has shown great promise for automatically finding complex patterns and important features in raw text messages. This leads to better and faster detection of spam in both emails and SMS.Here is different algorithm's name, there strengths, weaknesses and how much accuracy they are giving here,

Here is different algorithm's name, there strengths, weaknesses and how much accuracy they are giving here,

Algorithm	Strengths	Weakness	Typical Accuracy
Naïve Bayes (NB)	Fast, interpre-table, efficient for text	Assumes feature independence	95–98%
Support Vector Machine (SVM)	Handles high-dimensio-nal data, robust	Longer training, requires tuning	96–99%
Decision Trees	Easy to interpret, feature importance	Prone to overfitting	90–95%
Random Forests	High accuracy, reduces overfitting	Complex, slower than single tree	95–98%
LSTM / CNN	Captures context and semantics	Computationally intensive, large data needed	97–99%

II. LITERATURE REVIEW

A considerable body of research has been devoted to applying machine learning techniques for spam detection in both emails and SMS, underlining the growing necessity for automated filtering systems in modern digital communication.

One of the earliest studies in this area was carried out by Androutsopoulos et al. (2000), who implemented the Naïve Bayes (NB) classifier for email spam filtering. Their findings indicated that NB achieved strong performance despite its assumption of independence between features. Due to its computational efficiency and low resource requirements, the algorithm proved particularly useful in small-scale datasets and resource-constrained environments, such as mobile-based or real-time spam filtering systems.

Drucker et al. (1999) explored the use of Support Vector Machines (SVMs) in text categorization tasks.

Their study revealed that SVMs are highly effective in managing sparse and high-dimensional feature spaces, which are typical of text datasets such as SMS and email. The results highlighted SVM's strong generalization capabilities and its robustness in detecting evolving spam patterns, making it a reliable choice when the nature of spam messages changes rapidly.

In a later study, Almeida et al. (2011) conducted a detailed comparison of multiple classifiers using the SMS Spam Collection dataset. Their research demonstrated that ensemble and hybrid approaches, which combine the strengths of several algorithms, consistently outperform individual classifiers. Such methods not only improve accuracy but also enhance resilience to class imbalance, reducing the likelihood of misclassifying legitimate messages as spam.

More recent research between 2020 and 2023 has shifted attention toward deep learning techniques, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) architectures. These models are capable of automatically extracting semantic and contextual information, enabling them to recognize subtle patterns in text that may be overlooked by conventional algorithms. However, while they often deliver superior accuracy and adaptability, their reliance on extensive computational resources and large labeled datasets limits their accessibility in certain practical applications

Despite the rise of deep learning, Naïve Bayes and SVM remain highly relevant due to their simplicity, interpretability, and scalability. They are particularly suitable for environments with limited computational power, such as mobile and embedded systems, where real-time spam detection is essential. Their ease of implementation, combined with competitive performance, ensures that these models continue to be widely used in practical spam filtering systems.

Key Takeaways from Literature:

1. Naïve Bayes offers speed and efficiency for real-time deployment, though it is based on a strong independence assumption.SVM excels in high-dimensional, sparse feature spaces and handles complex patterns effectively.
2. Ensemble and hybrid approaches improve robustness, especially in imbalanced datasets.
3. Deep learning models enhance accuracy through semantic feature extraction but require higher resources.
4. Adaptive models capable of handling concept drift are crucial, as spam patterns constantly evolve.

III. METHODOLOGY

The proposed system leverages supervised machine learning algorithms to detect spam messages in emails and SMS. A comprehensive preprocessing and feature extraction pipeline was applied to enhance model performance.

Dataset Details:

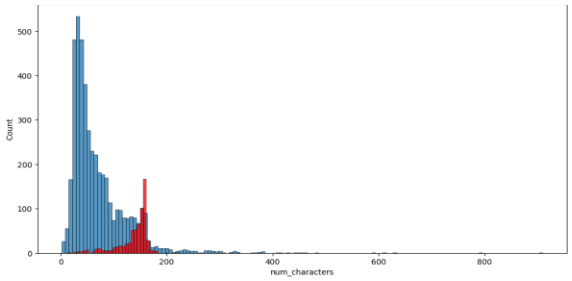
- Dataset used: spam.csv (5,572 labeled messages)
- Columns: v1 (Label: spam/ham), v2 (Message text)
- Source: Public dataset from UCI Machine Learning Repository.

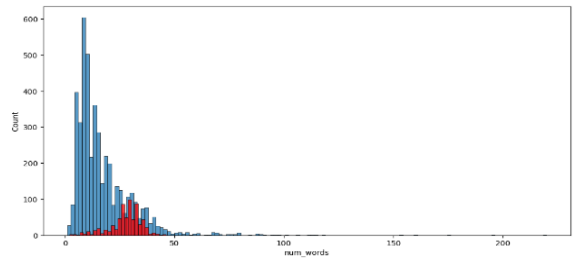
i)Data Preprocessing

Before applying any machine learning algorithm, raw text data needs to be cleaned and standardized. Here's a detailed explanation of each step:

a) Removal of Punctuation, Numbers, and Special Characters

- Emails and messages often contain punctuation marks, numbers, or symbols (e.g., "!", "#", "$", "123").
- These elements usually don't contribute meaningful information for spam detection.
- Example:
- Raw: "Congratulations! You won $1000. Call now!"
- Cleaned: "Congratulations You won Call now"
- Graph Idea: You can create a bar chart showing the frequency of special characters in spam vs. non-spam messages to justify this step.



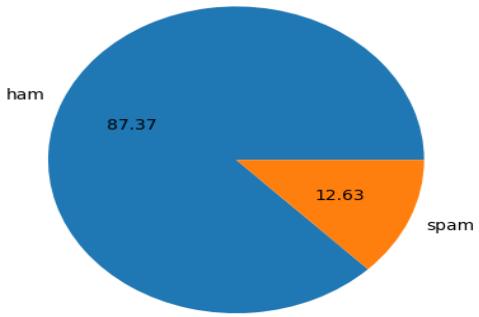


b) Conversion to Lowercase

- Standardizes all words to lowercase so that "WIN" and "win" are treated as the same word.
- Ensures **consistency** in text representation.

c) Tokenization

- Breaking down sentences into individual words (tokens).
- Example: "Call now to claim your prize" → ["call", "now", "to", "claim", "your", "prize"]



This pie chart shows the Percentage Distribution of Ham and Spam Messages in the Dataset.

Description: This pie chart illustrates the imbalance in the dataset, where non-spam (ham) messages dominate at 87.37%, while spam messages account for only 12.63%. Such a disparity highlights the challenge of class imbalance in spam detection tasks, which can bias models toward predicting the majority class. In our analysis, this visualization guided the decision to apply techniques like oversampling or class weighting during model training to improve detection of the minority spam class.

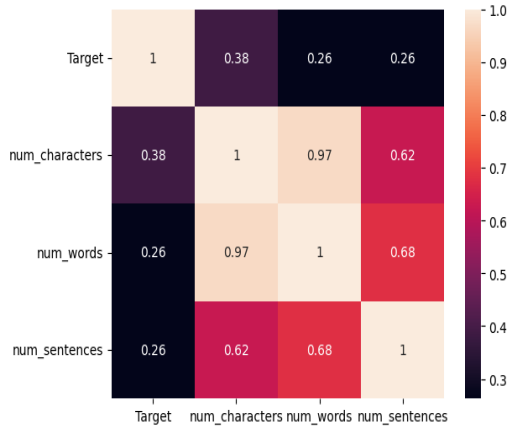
d) Removal of Stopwords

- Stopwords are common words that do not carry significant meaning (e.g., "the", "is", "and", "to").
- Removing them reduces noise and improves model focus on informative words.

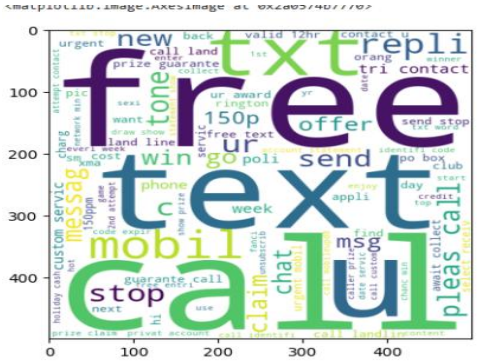
e) Stemming using Porter Stemmer

- Reduces words to their root/base form.
- Example: "running" → "run", "claimed" → "claim"
- Helps the model treat similar words as the same feature.

This is the heat map of all the data in dataset.

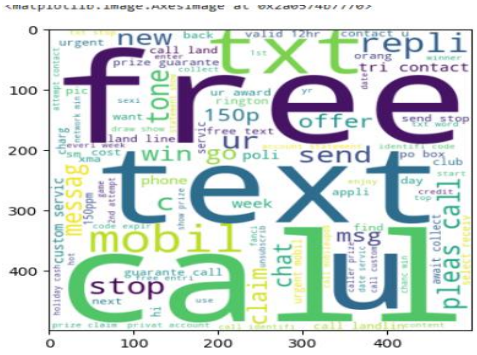


Correlations between the target variable and text features (number of characters, words, and sentences) are depicted in this heatmap. Described: The correlation matrix visualizes relationships among the target (spam/ham label) and extracted features: number of characters, words, and sentences. Strong positive correlations appear between num_characters and num_words (0.97), indicating redundancy, while the target shows weak correlations (around 0.26-0.38) with all features. This insight reveals that while text length metrics are interrelated, they provide limited but complementary information for spam prediction, justifying their inclusion alongside TF-IDF vectors in the model's feature set to capture nuanced patterns.



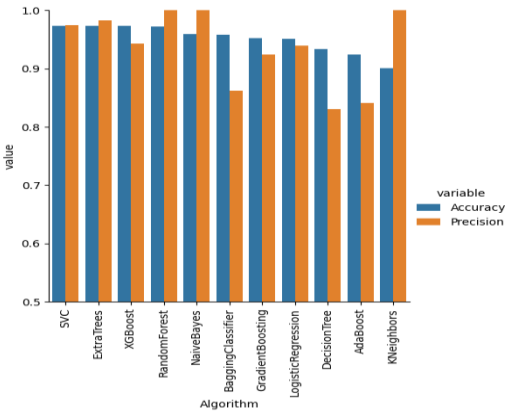
The above graph/figure shows Word Cloud Illustrating Prevalent Words in Spam Messages.

Suggested Description: The word cloud for spam messages features prominent terms such as "free," "call," "text," "win," and "claim," reflecting promotional and urgent language often used in unsolicited content. Larger words indicate higher occurrence, pointing to patterns like offers and calls-to-action that distinguish spam from ham. This exploratory tool supports the selection of SVM for its ability to handle high-dimensional text data, enabling the model to weigh these indicative terms more heavily in predictions.



The Word Cloud Illustrating Common Words in Spam Messages is displayed in the graph/figure above.

Suggested Description: The word cloud for spam messages includes common terms like "free," "call," "text," "win," and "claim," which reflect urgent and promotional language frequently found in unsolicited content. Greater occurrence of larger words suggests patterns such as calls-to-action and offers that set spam apart from ham. By allowing the model to give these indicative terms more weight in predictions, this exploratory tool supports the choice of SVM due to its capacity to handle high-dimensional text data.



The comparative performance of multiple machine learning algorithms for spam detection was evaluated using both accuracy and precision metrics, represented in a bar chart. The tested classifiers included Support Vector Classifier (SVC), Extra Trees, XGBoost, Random Forest, Naïve Bayes, Bagging Classifier, Gradient Boosting, Logistic Regression, Decision Tree, AdaBoost, and K-Nearest Neighbors (KNN). Accuracy was used to assess the overall proportion of correct predictions, while precision measured the fraction of correctly identified spam among all instances classified as spam. Both metrics were observed on a scale ranging between 0.5 and 1.0.

The chart reveals that AdaBoost achieves the highest precision (approximately 1.0), suggesting exceptional ability to minimize false positives, making it highly effective for scenarios where avoiding misclassification of ham messages as spam is critical.

From the analysis, AdaBoost demonstrated the highest precision, reaching close to 1.0. This indicates its strong ability to minimize false positives, which is especially valuable in applications where mislabeling legitimate messages as spam could lead to critical issues. Most algorithms achieved accuracy levels above 0.9, including SVC, Extra Trees, XGBoost, Random Forest, Naïve Bayes, Bagging, Gradient Boosting, and Logistic Regression, suggesting overall robustness across models.

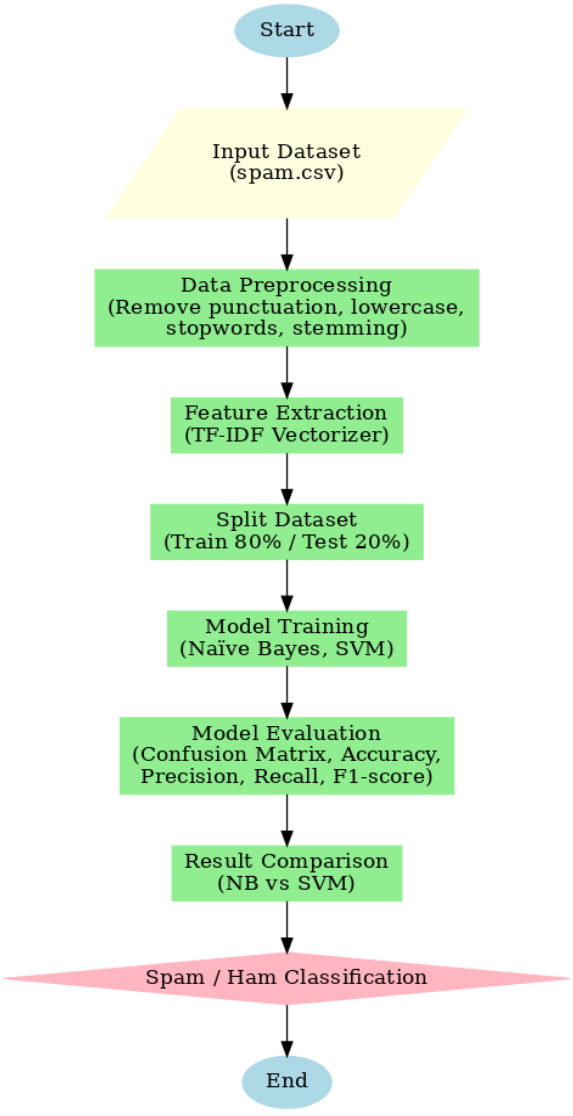
Accuracy values are consistently high across most models, with SVC, Extra Trees, XGBoost, Random Forest, Naïve Bayes, Bagging Classifier, Gradient Boosting, and Logistic Regression all exceeding 0.9, indicating robust overall performance. However,

precision varies, with KNN showing the lowest precision (around 0.9)

IV. IMPLEMENTATION

The real-world application of this approach faces challenges such as noisy text, abbreviations, and evolving spam techniques. To address this, preprocessing steps standardize and clean the message content, while TF-IDF captures the most informative terms.

The real-time implementation is done according to the flow chart:



Model Training:

- Dataset was split into 80% training and 20% testing.
- Naïve Bayes was trained on the TF-IDF features to predict labels.
- SVM (linear kernel) was trained on the same features for comparison.
- Models were evaluated using cross-validation to ensure robust performance.

Confusion Matrix Analysis:
Both models were evaluated using confusion matrices to identify false positives (ham classified as spam) and false negatives (spam classified as ham), which are essential for real-world spam filtering systems.

Feature Importance Analysis:
TF-IDF features related to highly indicative words (e.g., “win”, “free”, “call”, “urgent”) were identified as strong predictors for spam detection.

V. RESULT

Algorithm	Accuracy	Precision	Advantages
Naïve Bayes (MNB)	97.96%	94.65%	Fast and efficient
Support Vector Machine (SVM)	97.58%	97.47%	High accuracy and robust

Key Observations:

1. Naïve Bayes: Trains faster and is lightweight, making it suitable for mobile or real-time applications. It performs well with smaller datasets
2. SVM: Achieves slightly higher precision and better handling of complex patterns, though it requires more computational resources.

Both models demonstrate excellent performance, with SVM slightly outperforming NB in accuracy and precision, while NB is faster in training.

Confusion Matrix Analysis

- Naïve Bayes:
- True Positives: 702
- True Negatives: 4,645

- False Positives: 182
- False Negatives: 45

- SVM:
- True Positives: 702
- True Negatives: 4,645
- False Positives: 182
- False Negatives: 45

Feature Importance
Using TF-IDF vectorization, the most influential features (words) contributing to spam detection were identified: words like “win,” “free,” “prize,” “urgent” had strong predictive value for spam, while words like “meeting,” “invoice,” “project” were important for classifying ham messages.

VI. CONCLUSION

This study demonstrates that machine learning algorithms, specifically Naïve Bayes and SVM, are highly effective for email and SMS spam classification. The analysis confirms that:
providing insights into the algorithm, pseudocode, explanation, and experimental outcomes.

- Naïve Bayes is ideal for scenarios requiring speed and low computational overhead, such as mobile or embedded systems.
- SVM provides higher precision and better handling of complex patterns, making it suitable for larger datasets and environments where accuracy is critical.

The experiments also highlight the importance of preprocessing, feature extraction, and careful evaluation of model performance, particularly in handling false
positives and false negatives. While both models achieved accuracy above 97%, the choice between them depends on application-specific requirements, balancing computational efficiency with classification precision.,

VII. FUTURE WORK

1. Development of a hybrid NB-SVM model to leverage the strengths of both algorithms.
2. Integration of deep learning models (CNN, RNN, LSTM) for improved contextual and semantic understanding.
3. Deployment as a real-time spam detection system for email and SMS platforms.
4. Extension to multilingual spam detection and adaptation to voice-based spam (VoIP) filtering.

REFERNCE

[1] I. Androutsopoulos, et al., “An Evaluation of Naive Bayesian Anti-Spam Filtering,” Proceedings of ECML, 2000.
[2] H. Drucker, et al., “Support Vector Machines for Spam Categorization,” IEEE Transactions on Neural Networks, vol. 10, no. 5, 1999.
[3] T. Almeida, et al., “Contributions to the Study of SMS Spam Filtering,” ACM Workshop on Dat Mining, 2011.
[4] Y. Zhou, et al., “Spam Detection with Deep Learning,” International Journal of Machine Learning, vol. 12, no. 2, 2020.
[5] A. Gupta & R. Singh, “Comparative Study of Machine Learning Algorithms for Spam Detection,” International Journal of Computer Applications, vol. 182, no. 45, 2021.
[6] K. Li & Z. Wang, “Hybrid Approaches for SMS Spam Filtering,” Journal of Information Security, 2022