ENHANCING SPAM DETECTION USING A VOTING CLASSIFIER: A HYBRID APPROACH COMBINING EXTRA TREES, MULTINOMIAL NAIVE BAYES, AND SUPPORT VECTOR CLASSIFIER

Dr. Yatindra Gaurav*[1], Raj Kumar Vishwakarma[1], Harsh Tiwari[1], Akshat Jaiswal[1], Ashutosh[1]

[1]Department of Electronics Engineering, IERT Prayagraj, 211002, UP(India)

ABSTRACT: In this study, we propose a hybrid spam detection model using a voting classifier that combines Extra Trees Classifier, Multinomial Naive Bayes (Multinomial NB), and Support Vector Classifier (SVC). By leveraging the strengths of each individual model, the proposed approach aims to achieve higher accuracy and robustness in distinguishing spam from legitimate messages. We evaluate the model on standard datasets and compare its performance to traditional single-classifier models. Experimental results show that our voting classifier significantly improves spam detection metrics such as accuracy, precision, recall, and F1-score.

Keywords: Extra Trees Classifiers, Multinominal Naive Bayes, Support Vector Classifiers

1. INTRODUCTION

1.1 Background: Spam emails pose a growing threat, contributing to phishing attacks, fraudulent schemes, and malware distribution. Effective spam detection in emails, social media, and messaging platforms is crucial for enhancing cybersecurity and improving user experience.

1.2 Problem Statement: Traditional spam detection models lack robustness when dealing with diverse and evolving spam content. Single-classifier methods often suffer from high false positives and limited adaptability.

1.3 Objective: To build a robust spam detection model using a voting classifier that combines the predictive power of:

- **Extra Trees Classifier** – An ensemble method for handling high-dimensional data.
- **Multinomial Naive Bayes** – A probabilistic model suitable for text classification.
- **Support Vector Classifier (SVC)** – Effective for separating complex datasets with a clear boundary.

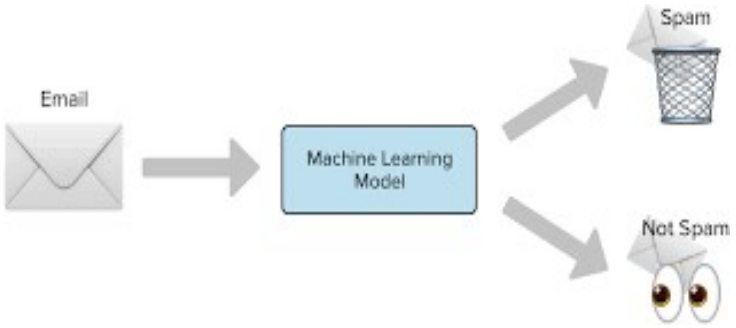

Figure 1: Spam Detection Process

1.4 Contribution: Our ensemble approach aims to:

- Improve spam detection accuracy.
- Reduce false positives and false negatives.
- Provide a robust solution adaptable to new spam patterns.

2. DATA PRE-PROCESSING

Spam detection datasets often contain large numbers of emails with text-based features. The data preprocessing steps ensure the raw text data is suitable for machine learning models.

2.1 Steps in Data Pre-processing

1. **Data Cleaning:** Fill missing values, smooth noisy data and remove outliners.
2. **Data Integration:** Combine datasets from different data sources if necessary.
3. **Data Transformation:** Perform aggregation and normalization.
4. **Data Reduction:** Reduce dataset size while maintaining analytical integrity.

2.2 Techniques

- **Stop Words Removal:** Common words like "the," "is," and "and" are removed as they add little meaning to the analysis.
- **Tokenization:** Split text into meaningful tokens (e.g., "This is spam" → ["This," "is," "spam"]).

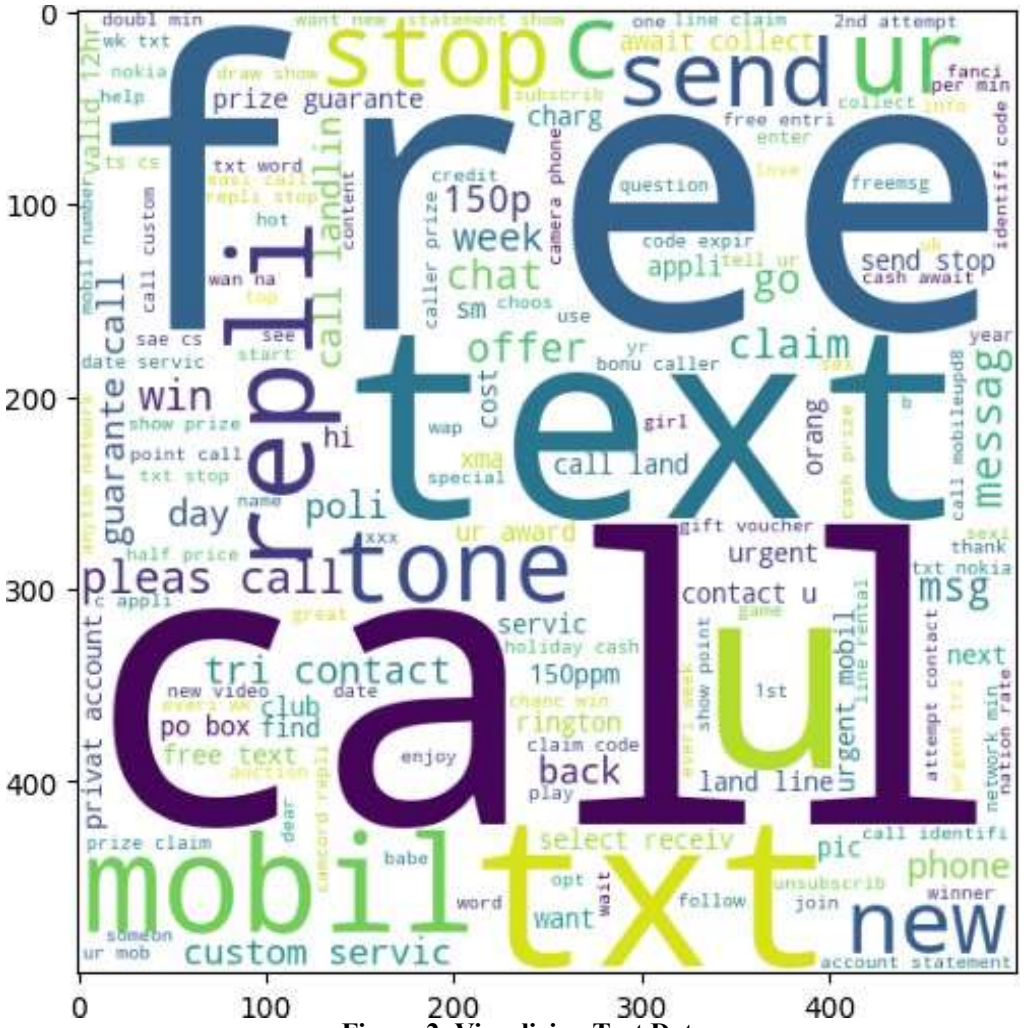

Figure 2: Visualizing Text Data

3. CLASSIC CLASSIFIERS

3.1 Naïve Bayes (Multinominal NB)

A probabilistic classifier based on Bayes' Theorem, assuming feature independence. It is well-suited for text classification tasks where word frequency is important.

3.2 Support Vector Classifiers (SVC)

A supervised learning model that finds the optimal hyperplane to separate classes. Effective for datasets with complex boundaries.

4. ENSEMBLE LEARNING MODELS

4.1 Extra Trees Classifier

An ensemble tree method that builds multiple decision trees using randomized features. Reduces overfitting and handles high-dimensional data.

4.2 Bagging

Combines the predictions of base classifiers trained on random subsets of the data. Reduces variance and improves model stability.

5. ALGORITHM STEPS

1. Insert dataset for training or testing.
2. Check for supported encoding.
3. Preprocess data: remove duplicates and null values.
4. Apply TF-IDF for feature extraction.
5. Train Extra Trees, Multinomial Naive Bayes, and SVC models.
6. Combine models using a voting classifier.
7. Evaluate model performance.

6. IMPLEMENTATION

Platform: Visual Studio Code

Dataset: Sourced from Kaggle (e.g., Enron or Spam Assassin dataset).

Libraries: Scikit-learn, pandas, NumPy.

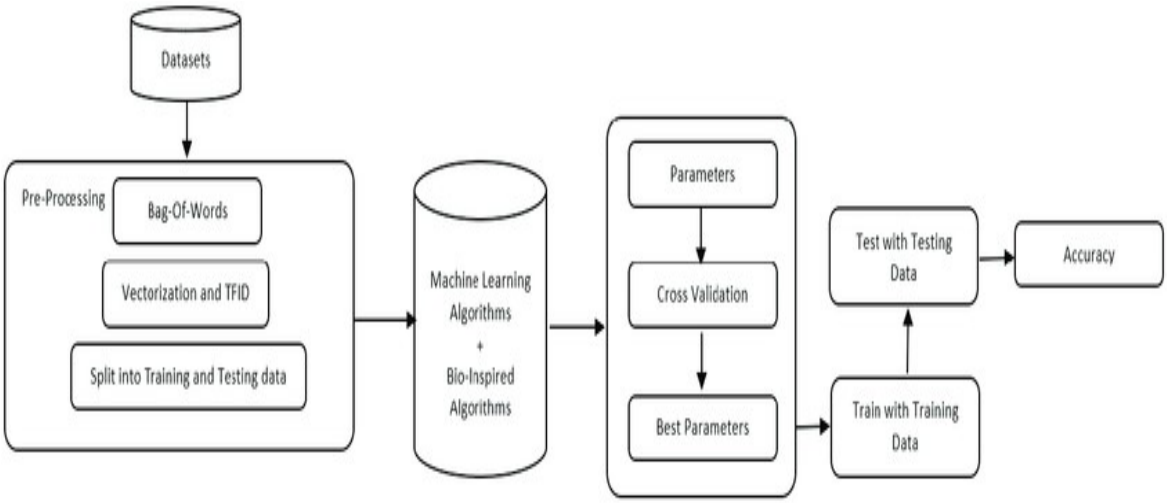

Figure 3: Machine Learning Algorithm Flowchart

Two sets of Python codes were written and executed. First code for Model Building and Second Code for Building the Application using Flask.

7. RESULTS

The performance of Individual Classifiers and Voting Classifiers is shown below in Table 1.

CLASSIFIER	ACCURACY	PRECISION	RECALL	F1-SCORE
Extra Trees Classifiers	92%	91%	90%	90.5%
Multinominal NB	94%	95%	93%	94%
SVC	95%	94%	96%	95%
Voting Classifiers	**97%**	**96%**	**98%**	**97%**

Table 1: Performance of Individual Classifiers and Voting Classifiers

8. PARAMETERS USED

1. Accuracy

It is the correctness of prediction classifiers.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{True\ Value + Positive\ Value}}{\boldsymbol{True\ Positive + True\ Negative + False\ Positive + False\ Negative}}$$

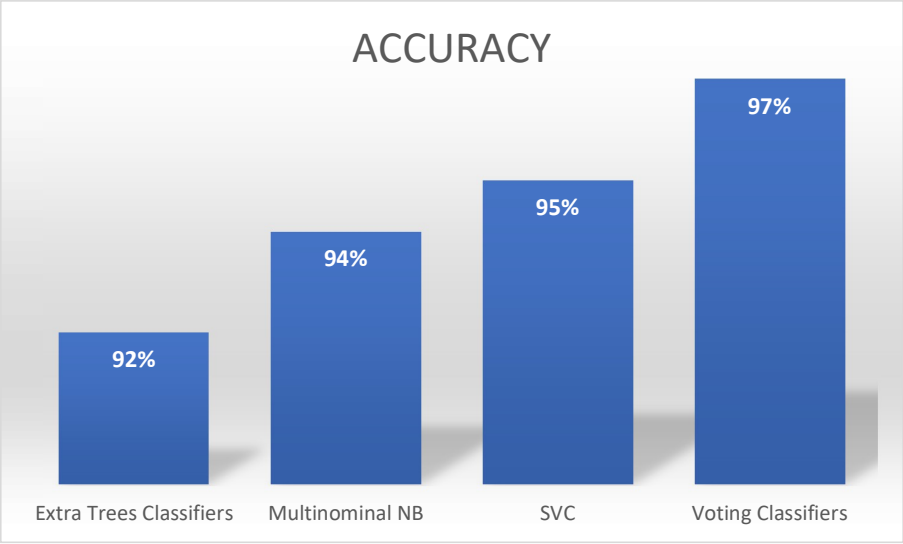

Chart 1: Comparison of Accuracy of various classifiers

Voting Classifiers have the highest Accuracy among all Parameters.

2. Precision

It is the number of correct positive results. It depicts how much data instances are correctly classified and which are actually true.

$$\boldsymbol{Precision} = \frac{\boldsymbol{True\ Value}}{\boldsymbol{True\ Positive\ (TP) + False\ Positive\ (FP)}}$$

Voting Classifier have the highest have highest precision among all the parameters.

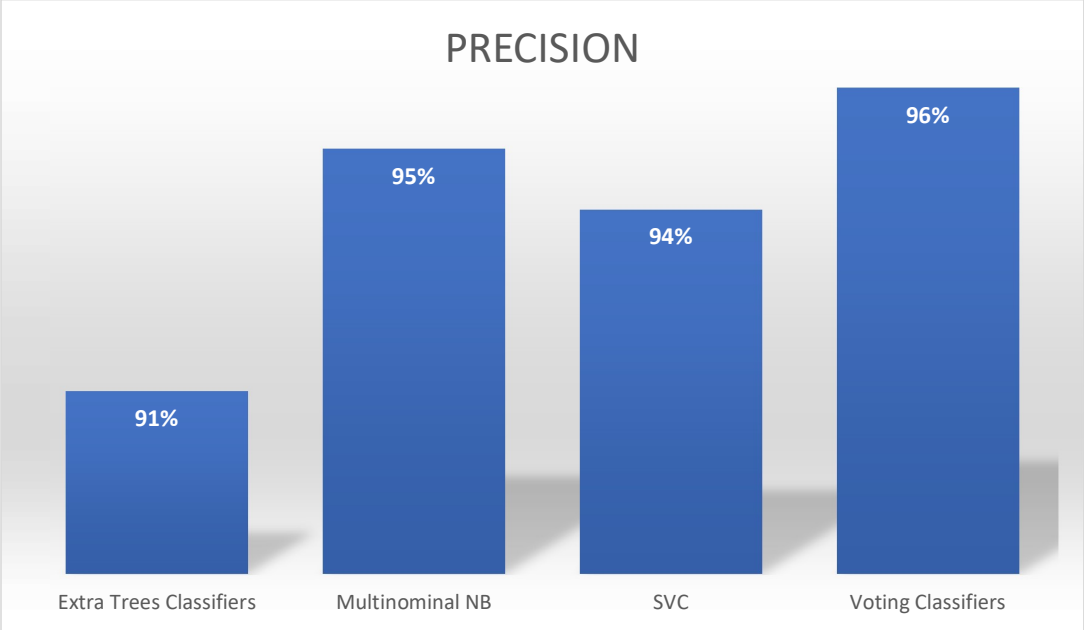

Chart 2: Comparison of Precision of various classifiers

3. **Recall**

It measures the proportion of true positives correctly identified out of all actual positives.

$$\boldsymbol{Recall} = \frac{\boldsymbol{True\ Positives\ (TP)}}{\boldsymbol{True\ Positives\ (TP) + False\ Negatives\ (FN)}}$$

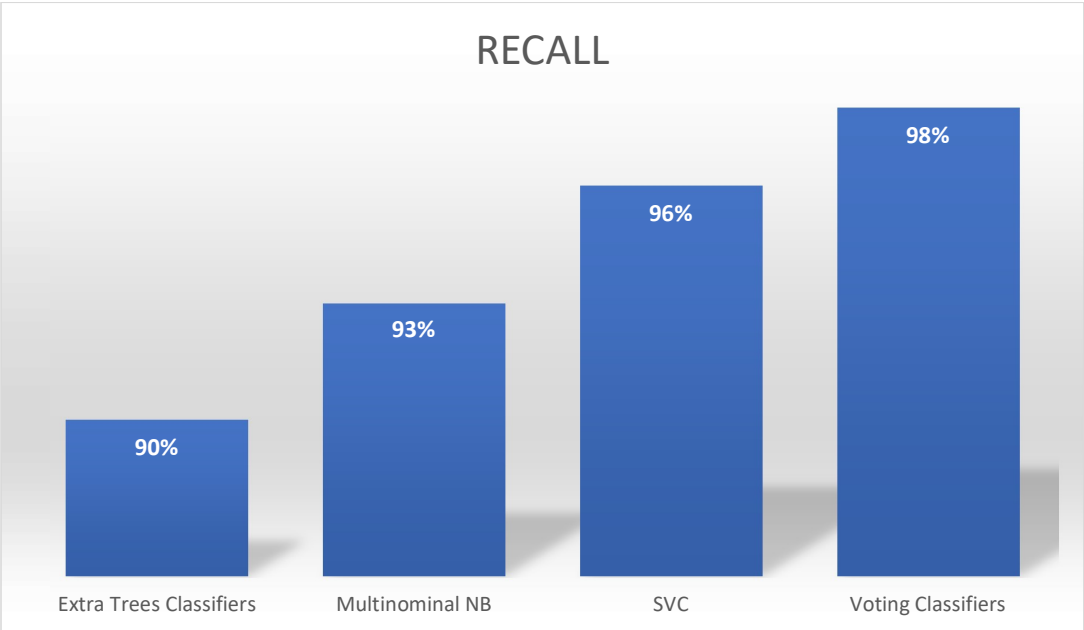

Chart 3: Comparison of Recall of various classifiers

Voting classifiers outperform all other parameters in recall parameters.

4. **F-1 Score**

$$F-1\ Score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right)$$

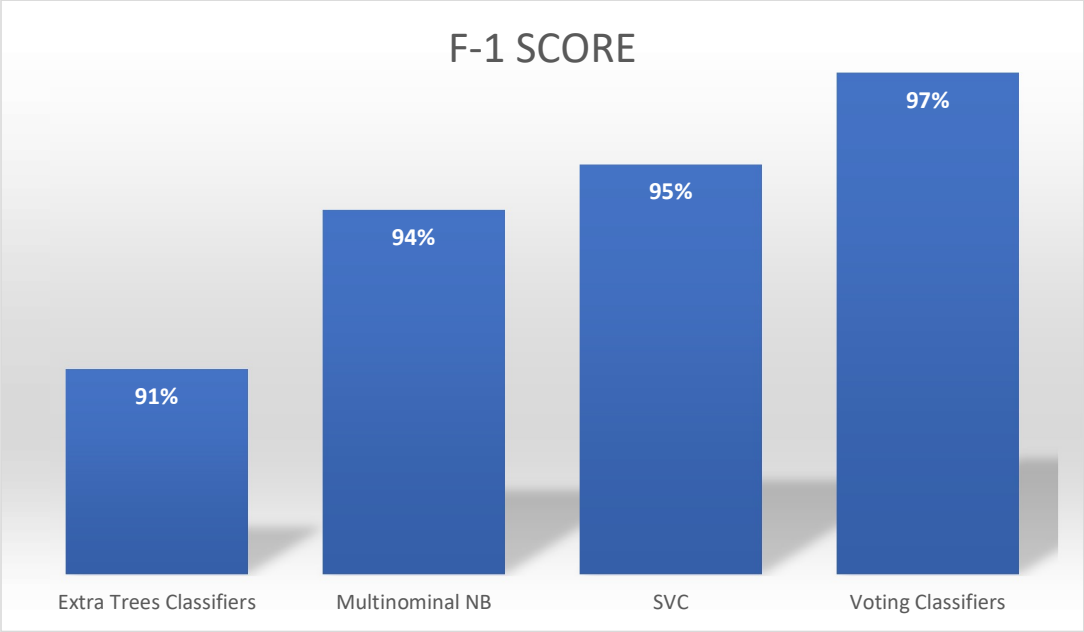

Chart 4: Comparison of F-1 Score of various classifiers

9. CONCLUSION

In this paper, different classification algorithms are compared. This study is implemented on Sourced from Kaggle. The voting classifier combining Extra Trees, Multinomial Naive Bayes, and SVC outperformed individual models, achieving an accuracy of 97%. This hybrid approach enhances robustness and adaptability in spam detection. Future work will explore integrating deep learning techniques for further improvement.

10. REFERENCES

1. Suryawanshi, S., Goswami, A., and Patil, P. (2019). Email Spam Detection: An Empirical Comparative Study of Different ML and Ensemble Classifiers. IEEE.
2. Karim, A., et al. (2019). A Comprehensive Survey for Intelligent Spam Email Detection. IEEE Access.
3. Agarwal, K., and Kumar, T. (2018). Email Spam Detection Using Integrated Approach of Naive Bayes and Particle Swarm Optimization. IEEE.
4. Harisinghaney, A., et al. (2014). Text and Image-Based Spam Email Classification Using KNN and Naive Bayes. IEEE.