A Machine Learning-Based Framework for Accurate and Interpretable SMS Spam Detection

Aman Pal
Department of Computer Scinece Engineering (Internet of Things)
Noida Institute of Engineering And Technology(AKTU Affiliation).
Greater Noida , India
0211csio098@niet.co.in

Kushagra
Department of Computer Scinece Engineering (Internet of Things)
Noida Institute of Engineering And Technology(AKTU Affiliation).
Greater Noida , India
0211csio073@niet.co.in

Abhijeet Krishna
Department of Computer Scinece Engineering (Internet of Things)
Noida Institute of Engineering And Technology(AKTU Affiliation).
Greater Noida , India
0211csio048@niet.co.in

Priya Ranjan Pritesh
Department of Computer Scinece Engineering (Internet of Things)
Noida Institute of Engineering And Technology(AKTU Affiliation).
Greater Noida , India
0211csio020@niet.co.in

Under the Guidance of

Mr. Jaya Nidhi Vashishtha(Assistant Professor)

Department of Computer Science Engineering (Internet of Things)

Noida Institute of Engineering and Technology(AKTU Affiliation)

Abstract*— Amidst mobile communication in daily life, the spread of unwanted and damaging Short Message Service (SMS) spam poses real privacy and security issues. For this research, a comprehensive machine learning-based approach that can classify SMS messages effectively as spam or ham (legitimate) was suggested. With the use of the well-known SMS Spam Collection dataset, some preprocessing techniques were utilized to normalize and clean the text-based data. TF-IDF-based feature extraction and application and testing of various classification algorithms, such as Naive Bayes, Support Vector Machine (SVM), and Logistic Regression, were adopted. The results of the experiment show that machine learning models, particularly those with natural language processing-specific design, can efficiently yield high accuracy rates in spam message detection. The results highlight the need for the presence of efficient spam filtering software for the prevention of the risk of fraud, phishing, and data abuse via SMS. Furthermore, the research points out the issues regarding class imbalance and predicts future improvements with the use of powerful deep learning algorithms or methods like Generative Adversarial Networks (GANs) for data augmentation. This research is a significant contribution towards the creation of more intelligent and secure messaging systems.

Keywords—SMS spam detection, text classification, machine learning, natural language processing (NLF), tf-idf, naive bayes, support vector machine (svm), logistic regression, dataset imbalance, cybersecurity.

I. INTRODUCTION

In modern society, mobile phones have become one of the most widely used means of communication all over the world. As their usage increases, Short Message Service (SMS) has emerged as an ordinary and efficient means of personal communication, economic transactions, and advertisement messages. Nevertheless, this extensive use of SMS has created a terrible issue: the emergence of unwanted spam messages. These messages, extensively employed for advertisement, phishing, or fraud, have serious implications on user privacy and security.

Spam messages do not only cause a lot of annoyance but also lead to severe issues like identity theft, financial scams, or the spreading of malicious URLs. Contrary to email spam, which is supported by strong filtering mechanisms, it is harder to identify SMS spam due to the short and colloquial nature of text messages. Most spam messages are intentionally designed with misspellings, colloquialisms, abbreviations, or obfuscated URLs to avoid common detection methods, thus making rule-based filters useless.

To solve this problem, machine learning has been found to be extremely useful as a tool. Through the identification of patterns in data and learning from labelled examples, such programs are capable of well differentiating spam from non-spam (ham) messages with great accuracy. This work considers the development and testing of a range of different SMS spam models using a public dataset of thousands of labelled SMS.

Four machine learning algorithms were run: Naive Bayes, Decision Tree, Support Vector Machine with a linear kernel, and Random Forest. The text data were pre-processed and transformed using techniques like text cleaning and TF-IDF vectorization. Out of the models run, Random Forest performed

the best at 99%, then SVM and Decision Tree at 98%, with Naive Bayes keeping the performance at 94.3%. This study explains the results of the experiments conducted, assesses the pragmatic value of each model, and presents the limitations along with future avenues for further studies. The primary aim is to demonstrate the capability of machine learning to significantly minimize SMS spam, thus the security and reliability of messaging systems.

A. Background and Motivation for SMS Spam Filtering

In today's digital age, text messaging plays a vital role in everyday communication, with billions of SMS messages exchanged globally each day. Its ease of use, affordability, and broad accessibility make it a popular medium for both personal conversations and business interactions. However, the widespread use of SMS has also led to its exploitation—particularly through spam messages. These unsolicited texts often promote fake offers, deceptive prize claims, suspicious links, or harmful websites. In more serious cases, they serve as phishing tools, attempting to extract sensitive financial or personal information from unsuspecting users.

The increasing volume of SMS spam is more than just a nuisance—it poses a serious risk to user privacy, data security, and overall trust in mobile communications. As spammers constantly refine their techniques to evade detection, traditional filtering methods that rely on static keywords or blacklists have become insufficient. These outdated approaches struggle to keep pace with the evolving nature of spam messages, especially those that use informal language, abbreviations, or creative formatting to bypass filters.

This growing concern has fuelled interest in leveraging machine learning to combat SMS spam more effectively. Unlike rule-based systems, machine learning models have the ability to learn from data and identify complex patterns that might not be immediately visible. By training on real-world, labelled datasets, these models can intelligently differentiate between legitimate texts and spam—even when the spam is subtly disguised.

The focus of this research is to design and evaluate various machine learning models for SMS spam classification. We employed algorithms such as Naive Bayes, Decision Tree, Support Vector Machine (SVM), and Random Forest to compare their performance in accurately filtering out spam messages. The ultimate goal is to enhance user safety, reduce message clutter, and contribute to a more secure mobile communication experience.

B. Problem Statement, Mitigation Approaches and Paper Organization

The growing volume of SMS spam messages has become a major issue for mobile users and service providers. These messages can be disguised as a promotion, a notification that you've won a prize, or an urgent issue. They are designed to fool recipients and can subsequently lead to scams, breaches of personal data, or unwanted advertisements. Existing filtering systems are based on static rules, or keyword matching, and often cannot keep up with the ever-changing language and structure used by spammers.

The primary research problem we will address is the issue related to the complexity of identifying genuine and spam SMS messages. Unlike emails, SMS messages are short, less formal, and typically full of abbreviated forms and symbols, which make it more challenging to detect spam. Additionally, as the techniques employed by spammers develop further, it is even more imperative to develop models that learn from data to recognize the subtle differences between spam and content that should be considered legitimate.

We applied four different models: Naive Bayes, Decision Tree, Support Vector Machine (SVM with linear kernel), and Random Forest to a labelled SMS dataset. We pre-processed the raw text and converted it to numerical features using TF-IDF, so that the models could learn and make predictions based on word importance and frequency.

In terms of results, all four models obtained acceptable results: Random Forest was the highest, with accuracy of 99%, then SVM and Decision Tree with 98%, and Naive Bayes with 94.3%. These results highlight that machine learning techniques can provide the basis for effective intelligent spam filter solutions.

The rest of the document is structured to allow the reader to understand the research process step by step.

Section 2 presents previous work about spam detection with a summary of methods used in past studies.

Section 3 describes the dataset and text processing steps that were performed in preparation for machine learning.

Section 4 presents the specifications of the algorithms chosen and the details of training and testing.

Section 5 shows the results from our experiments, interprets the relevance of the results, and explores possible uses and limitations we observed.

Section 6 provides a summary of the major findings, and comments on future improvements and expansion on this work.

C. Research Aim and Study Objectives

The purpose of this research is to develop an SMS classification system that can effectively identify spam SMS messages and confirm if SMS messages are legitimate (or ham) messages using machine learning. SMS spam has serious implications that encompass an annoyance, fraud, and phishing. Static spam filter systems are limited to their static rules and keyword matching components, which makes it more difficult to detect newer types of spam messages as spam senders develop tactics to work around static spam filters.

The focus of the study is dynamic machine learning as a more scalable solution to these issues. The study implements four supervised learning models (Naive Bayes, Decision Tree,

Support Vector Machine (SVM, linear kernel), and Random Forest), trained on labeled SMS messages, for comparison on our spam classification system effectiveness.

To ensure a meaningful learning experience, the raw SMS data is first thoroughly processed and transformed through initial text cleaning before converting the data into numerical features through TF-IDF (Term Frequency–Inverse Document Frequency). This defining step allows the models to detect trends and ultimately classify messages. The models were not evaluated on the text data; therefore, several metrics were calculated, (accuracy, precision, recall, F1-score) to determine which models performed best on spam identification.

To uncover strengths, weaknesses, and practical applications for each model, particularly with respect to their applicability in real world contexts such as, but not limited to, mobile devices, filters on network level or enterprise messaging systems.

To provide a framework for future improvements by looking at areas of potential performance improvements such as balancing datasets, multi-language, and/or deeper learning interactions.

These objectives mean that, although this work provides a comparative study of algorithm performance, it can also evaluate the potential for real-world machine learning solutions to mitigate SMS spam to end-users and service providers.

II. Literature Review And Problem Formulation

This section explores the historical background, ongoing advancements, and research challenges in the field of SMS spam detection. A review of existing literature highlights how detection strategies have evolved from basic rule-based filters to intelligent models powered by machine learning and deep learning. It also discusses recent efforts to make model predictions more transparent and interpretable for real-world deployment.

A. Evolution of SMS Spam Detection Techniques

During the initial days of SMS spam detection, largely, detection systems revolved on rules-based type filters, which were able to flag message contents by being able to assess the presence of keywords, originating phone numbers, and/or formatting styles. These rules-based systems had the potential to be quick and straight-forward but lacked the capacity to change as spam messages began to evolve to include other elements or obfuscations, making such historical techniques less useful.

The onset of machine learning represented a radical change in spat detection. Rather than solely relying on rules, systems could start to learn which types of SMS messages were spam, by classifying messages using models available in machine learning. Naïve Bayes and Decision tree classifiers were too important in making machine learning for text processing common, as these classifiers were able to learn from labelled SMS dataset examples. Machine learning classifiers could learn richer complexity and layering of messages, with contextual meaning offered messages, as opposed to simply ruling out keywords. This was a profound change in the spam detection space, and more importantly led to improved detection and elimination rate with less false positives.

B. Advanced Techniques in Phishing SMS Detection and Interpretability

As mobile phishing attacks—also known as smishing—grew in scale and sophistication, detection models needed to go beyond basic spam recognition. Phishing-related SMS messages are commonly crafted to appear urgent, include suspicious links, or impersonate trusted sources to trick the recipient into taking harmful actions. Identifying such attacks requires models that can understand context and intent, not just keywords.

Recent approaches have focused on content analysis combined with behavioural features, such as link structure, sender identity, and message timing. Moreover, as these systems are used in sensitive environments like banking or telecom, there's a growing demand for interpretability—the ability to explain why a particular message was flagged. This has led to the adoption of interpretable models and explanation frameworks like LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (Shapley Additive explanations).

C. Model Explainability in SMS Spam Detection

As mobile phishing attacks, or smishing, have continued to grow in scale and sophistication, detection models have had to evolve beyond basic spam detection. Phishing related SMS messages usually incorporate time sensitive language, dubious links or impersonate a trusted party to trick the user into performing harmful actions. We need models that go beyond consideration of keywords and can account for context and intent.

Recent approaches have incorporated content analysis and behavioural features, such as links, identity of who sent the message and sender timing. Moreover, now that these systems operate in sensitive environments like banking or telecommunications, there is significant interest in the ability to provide explanation for why a given message has been categorised in a certain way or flagged as suspicious, which is considered interpretability. It has become paramount/necessary to use easy to understand, interpretable models, and the use of explanation frameworks such as LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (Shapely Additive explanations).

D. Deep Learning-Based Spam Detection and Insights from Existing Research

The newest version of SMS spam filtering utilizes deep learning methods. These models employ architecture-based methods including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to remove the need to rely upon extractable features in the raw text and the

preprocessing steps previously utilized during feature extraction.

The results of studies employing models based upon these deep architectures such as Bi-LSTM, GRU, Transformer (447 extreme transformers i.e. BERT, including multilingual or unstructured text) have demonstrated notable performance in SMS spam filtering relative to other machine learning (ML) techniques. The robustness of deep learning models against common techniques utilized in modern spam SMS, such as misspellings, slang, and hidden messages, is also known to vary between methods of classification.

With the advantages associated with performance, deep learning methods also present higher computational and lower interpretability costs, which normalizes the barrier to adoption for some variations from many real time mobile applications. Some hybrid methods, which employ the best of multiple paradigms (deep and classical), are being explored to gain the performance from the deep models and speed and interpretability from classical ML algorithms.

Our performance findings utilizing classical ML models considering Naïve Bayes (94.3%), Decision Tree (98%), SVM (98%), Random Forest (99%); indicate that even on classical methods, excellent performance can still be achieved without deep learning, if appropriate data is prepared and superior features are extracted using the TF-IDF measure.

III. Proposed Methodology And Dataset Overview

In this section of the write-up, I use a stepwise approach to detail how one would go about creating a machine learning system that can take an SMS text message and classify it as either spam or not spam. This involved identifying appropriate datasets, cleaning and transforming the original text data, applying several machine learning models, and assessing each against benchmark metrics.

A. Overview of the SMS Spam Collection Dataset

For this project, I used the SMS Spam Collection dataset. There are just over 5,000 real text messages from actual mobile communications. Each message is labeled either as spam (usually meaning unwanted or shady) or ham, which just means a normal safe message. Since the data is labeled, it is a great fit for training a machine learning model where it can learn from examples.

One of the first things I realized is that the dataset is not completely balanced. Most of the messages are ham, which makes sense, as people generally get more regular messages than spam messages. However, this can be problematic when it comes to modelling, a model could just predict ham for everything because that is the predominant type of message for this dataset. Part of the modelling challenges was to train the system to still find spam, without it being biased based on the majority class.

B. Class Preprocessing Techniques

To better account for the class imbalance in the dataset, we did look at distributions of the messages. Since no synthetic balancing methods were used in this version of the project, the knowledge of the skewed data helped to inform the evaluation strategy, e.g., focusing on precision and recall rather than just accuracy in evaluating how many spam messages were identified.

In future iterations, oversampling, under sampling, or even synthetic data (e.g., SMOTE or GAN-based techniques) could be applied to have a fairer model across both classes.

C. Data Preprocessing Techniques

Before training any model, raw text data must be cleaned and transformed into a format that machines understand. This section describes the specific text processing pipeline that was carried out to create SMS messages ready for classification.

- **Text Cleaning**

In initial cleaning steps, text messages were cleaned to remove unwanted characters and where available, inconsistencies. As part of this process text messages were:

Converted to lowercase to create a uniform representation of the text, Removed all special characters, numbers, extra whitespace, and punctuation that did not contribute to the meaning of the message, Removed unnecessary symbols and other formatting problems typical with SMS messages, The cleaning steps in this first part of the pipeline removed a large amount of noise from the data and allowed for a more uniform text to analyse later.

- **Tokenization and Lemmatization**

After cleaning the messages, the next step was to tokenize each message into separable words. Tokenization is the process of breaking down sentences into smaller parts so that the parts can be understood or processed by machine learning algorithms.

Once we completed our tokenization of the messages, we performed lemmatization of the words. Lemmatization normally reduces each word down to its simplest definition in the dictionary. For example, all forms of the "eat" i.e. "eats"; "eating"; "eaten" would all be reduced to "eat." This approach decreases the number of variations for a single term and allows the model to treat all forms of a word as one generating more consistency of features in the dataset.

- **Feature Extraction using TF-IDF**

After cleaning and pre-processing the text, the next step was to transform the messages into numerical values using TF-IDF (Term Frequency–Inverse Document Frequency). This will help assign weight to words based on their importance in the individual messages and the dataset.

The term frequency part means how often that word occurs in each message, which indicates the relevance of that word to that instance. The inverse document frequency reduces the weight for words that are more commonly saw in lots of messages, because the less meaningful those common words are going to be for classification.

D. Model Selection

In this work there are four, supervised machine learning algorithms selected to classify SMS messages, either as spam or ham. Each of the four models was chosen because of their demonstrated performance on text classification tasks, along with their other learning methods, which provide a different perspective on model performance. Each model is discussed in the following subsections and under subsequent accuracy results.

- **Naïve Bayes (94.3% Accuracy)**

Naive Bayes is a straightforward but effective model using Bayes Theorem for classification. Naive Bayes assumes all features are independent; this allows for simpler calculations, resulting in the speed of training a naive bayes model. Although this assumption of independence doesn't hold true for all-natural language, it however, performs consistently well for text-based tasks. In our experiment of 94.3% accuracy, it performed respectably well, which made it a strong contender for lightweight applications that require predicting data quickly while using low resources.

- **Decision Tree (98% Accuracy)**

The Decision Tree model creates a list of simple decision rules based on the features in the dataset. Each question reduces the dataset into smaller and smaller groups (yes or no question) to classify a specific message as spam or not spam. The biggest benefit is its interpretability - it is easy to see why a tree did what it did in classifying a dataset. The accuracy of the Decision Tree was 98% for this project which demonstrated it picked up the relevant keywords and/or patterns that separated spam and not-spam messages.

- **Support Vector Machine – Linear (98% Accuracy)**

The SVM with a linear kernel also turned out to be a very competitive model. The SVM linear kernel model works extremely well on text classification problems because of its performance on high-dimensional data like TF-IDF based data. The SVM finds the best boundary/or hyperplane to separate spam and not-spam datasets. By adequately pre-processing the data the model was able to achieve 98% accuracy - which is an incredible result. The results are clear - SVMs can be very powerful when used to distinguish spam from genuine messages.

- **Random Forest (99% Accuracy)**

Random forests are an ensemble technique that accounts for the predictions of multiple decision trees to yield a more accurate prediction. Each tree acts as a model and yields predictions, averaging the predictions across the trees yields a much more accurate prediction from a single model. Further, because it contributes to the averaging of models, random forest also aids the activity of overfitting. Each of the models selected performed well but Random Forest stood out because its accuracy was 99%, which made it the most stable and most accurate model upon comparison.

E. Model Training and Validation

All the selected models were trained on part of the dataset, while they were tested with the other part of the dataset. The dataset was divided in two, 80 percent of the dataset was training the models and 20 percent of the dataset was put aside to test each model. This allows for the evaluation of generalization among each of the models. Prior to training the message was pre-processed and converted into numerical vectors using TF-IDF. The default hyperparameters were used to train the models with an initial evaluation, and results were collected using common performance metrics. This process allowed for each of the models to be trained at the same conditions for a comparable analysis.

F. Performance Metrics

To properly assess how models performed at classifying SMS messages, we utilized four standard performance metrics. These four metrics can help provide a better perspective on the model's ability, especially when the dataset is imbalanced, and spam messages were the less frequent type of message.

- **Accuracy**

This metric is the total percentage of correctly classified messages. The accuracy metric gives a general idea of how a model performed, but it doesn't always provide the full story, particularly in the case of a class (ham) that nominated the spam message. Therefore, the accuracy metric should be evaluated with other metrics.

- **Precision**

Precision gives an indication of how accurate the model is when predicting a message is a spam message. In more common terms, precision means how many of the messages that flagged as spam were spam. A higher precision means the model is better at avoiding false positives and, therefore is less likely to incorrectly flag a legitimate message as spam.

- **Recall**

Recall measures a model’s ability to identify spam messages from the total number of actual spam messages in the dataset. Or, simply put, it represents how many spam messages the

system is identifying. Higher recall rates represent the model is identifying a large percentage of the spam and is useful when the cost of missing spam is high, but it can also reflect the fact the model is misclassifying legitimate messages. A model may have a high recall rate, but if it labels a high number of legitimate messages as spam, then they are making serious mistakes.

- **F1 Score**

The F1 score provides a single measure of the trade-off between precision (how many of the predicted spam messages were correct) and recall (how many of the actual spam messages were caught) as the harmonic mean of both. This is useful especially in imbalanced situations as natural to have a single measure, that indicates how well the model manages both types of classification errors; missing spam and mislabelling legitimate messages.

IV . EXPERIMENTAL RESULTS

This section provides an overview of the result for training and testing the selected machine learning models for SMS spam classification. All models were evaluated, visualized, and compared to determine the most effective approach for accurately detecting the spam classification task.

A. Comparison of Model Performance

Four machine learning algorithms were assessed on the SMS dataset: Naïve Bayes, Decision Tree, Support Vector Machine (Linear), and Random Forest. All models were trained and tested on the same training and test splits to maintain consistency when evaluating and comparing performance. The results of the model accuracy is summarized below:

Random Forest: 99%

Decision Tree: 98%

Support Vector Machine (Linear): 98%

Naïve Bayes: 94.3%

In this evaluation, Random Forest was the most accurate of the models. Random Forest was able to leverage multiple decision trees, which provided the benefits of overfitting reduction and prediction stability. SVM and Decision Tree were very close in performance, and Naïve Bayes was less accurate than both but performed competently; especially given Naïve Bayes is simplistic and computationally efficient

B. Confusion Matrix Analysis

To determine how accurately the models classified the messages, confusion matrices were outputted. Confusion matrices show each model's number of correct classifications, and its number of incorrect classifications for the spam and ham categories. The confusion matrix for the Random Forest algorithm demonstrated a very strong balance between correctly determining spam versus incorrectly identifying spam.

True Positives (Spam correctly identified): Very high

True Negatives (Ham correctly identified): Almost all

False Positives (Ham incorrectly classified as spam): Very few

False Negatives (Spam missed as ham): Very few

The SVM and Decision Tree model had similar trends, but the accuracy included somewhat more classification errors. For example, and number of false positives and false negatives compared to Random Forest led to a slightly lower precision and recall measure.

Overall, it’s clear Random Forest demonstrated not only high accuracy, but low error classifications in determining spam and legitimate messages.

C. Visualizations: ROC Curves, Loss Graphs

To more thoroughly understand model performance, visualizations were incorporated as well, namely, ROC curves and training-validation loss plots.

ROC Curve: In this graph, false positive (FPR) and true positive (TPR) rates are demonstrated for each model computed. The AUC was very close to 1.0 for Random Forest, so that model performance could be emphasized as excellent. SVM and Decision Tree models existed at the high level, while Naïve Bayes had a lower value, but still acceptable.

Loss Graph: With this plot several models' error through time while training can be demonstrated as they decreased over time. Overall, the plots can demonstrate why each model had a sharp decrease in training and validation loss wondered in the beginning of each of their epochs and then levelled off. This evasiveness or timidity of the models' errors shows models were able to learn at very good levels without any instance of overfitting.

While there were conclusions drawn based on the numerical processes offered, the added component of visualizations helped in cross analysing the numerical results as well as explore in more dimensions how each algorithm was learning over time.

D. Discussion of Results

The experimental results suggest that machine learning models can effectively separate spam from ham in SMS messages. Random Forest was the always the most successful of the four algorithms employed, producing nearly perfect classification error.

Random Forest's success can be explained by its ensemble method "averaging" to reduce variance and reliably dealing with fluctuations in noisy data. SVM, Decision Tree, and Naïve Bayes did not produce statistically as reliable results as Random Forest; but were all useful in providing, at least to some degree, reliable classifications.

The consistent distinguishing feature in all the models used was the appropriate data pre-processing, especially applicable for feature extraction with TF-IDF. This was particularly important

to transform the informal wording and short messages in SMS data into machine learning compliant applications.

Although the models provided strong performance, future work can continue to improve overall accuracy, such as using additional alternative deep learning techniques or balancing the dataset composition to an even proportion of spam and ham. Regardless, this outcome will rationally empirically support the promise of using machine learning for informative real-life applications in SMS spam filtering and classification tasks.

V. DISCUSSION

The study illustrates the power of machine learning algorithms to filter spam from SMS messages. The features were taken from a real-world dataset, and we trained four classifiers - Naive Bayes, Decision Tree, Support Vector Machine (SVM) and Random Forest - the three of which had very high scores and were strong classifiers. However, the three classifiers differed in structure and learning method, which lead to different scores.

The Random Forest model scored the highest with 99%; Random Forest is well suited to handling complicated and noisy data structures. As a model that aggregates over several decision trees, Random Forest mitigates the possibility of overfitting data, but also, it reduces variability in predictions. Such factors made Random Forest a particularly good candidate for this domain of text spam detection, where text features may have inherently subtle patterns even though classified as spam.

The Decision Tree and Linear SVM models were almost identical, on average realizing an accuracy score of 98%. The Decision Tree model is explanatory in nature and easy to understand; its straightforwardness supports explaining to users the reasoning for messages when labeled as spam. At the same time, the SVM model is known for its ability to provide consistent classification performance. In binary classification problems, it is often the case that people use a linear kernel with SVM, and that the problem is overly simplified. Given that we have a text feature with 327 unique tags (TF-IDF) and that the SVM model classifies problems in high-dimensional spaces, this specified use case seems justified.

The Naive Bayes model achieved a relatively lower accuracy of 94.3%, but it also provided good efficiency and speed. Naive Bayes uses the principle of probability and works well for short text messages where it is fairer to assume that the words are independent of each other. Although this assumption does not hold true in natural language, Naive Bayes performed very well given the limitation and can be useful in situations that value speed and simplicity over accuracy.

While accuracy was high across all models, one such challenge is the dataset's imbalanced nature where there are significantly more legitimate messages, otherwise known as Ham messages, than there are spam ones; sometimes the classification models favoured the majority class such as Ham and totally missed classifying spam messages. This imbalance is common and as we did standard pre-processing steps and evaluation steps, future work might explore some advanced techniques such as data resampling or synthetic data generation as a solution.

Another point to consider is that SMS messages are often short, informal and reference many abbreviations or links which statistically speaking, can be challenging for traditional models that do not always depict the intent of each message. Our models and the evaluation only used word frequencies as an input which misses the intention or emotional aspect of a message which could mean that a model could classify a message incorrectly.

The recent on spam detection suggests that deep learning models, specifically the models that utilize sequences of contextualized word embeddings (e.g., LSTM and Transformer-based architectures), offer promise by using the way that words relate to each other in a message since they have adversaries that craft spam messages to avoid the basic filters with clever idiosyncratic phrasing or unorthodox manipulations.

That said, exactly like our study has demonstrated, traditional machine learning algorithms seem to still be applicable and clearly important to the domain of SMS spam detection. If the pre-processing of the features and reliable features extraction methods are undertaken (e.g., TF-IDF) Random Forest can provide consistently good performance. Random Forest was the best model in our study, and it was highly accurate and credible. The excellent performance of the other models also supports the value of machine learning for this domain.

The future is overwhelmingly positive. Modern natural language processing (NLP) capabilities, improving upon our class imbalance approach, and further accommodating for different languages/states could take our strictly supervised spam detector to a more robust and flexible spam detection system.

A. Correlation Analysis

Analyzing the forms and meaning of words to understand their relationship to emotion and sentiment classification is a key part of text data analysis. In the context of classifying SMS spam, correlation analysis assists in understanding which words or features are more firmly correlated with spam messages and which words and features are more likely to be in normal (ham) messages.

As part of our preprocessing, we created a numerical representation of the text data with TF-IDF (Term Frequency–Inverse Document Frequency) which provided information about which terms ranked higher in importance across the documents and influenced the resulting class label. For example, terms such as "free," "win," "claim," "urgent," and "congratulations" appeared with stronger correlation in spam messages; conversely, common terms used in day-to-day conversations such as greetings (e.g., "hi" or "hello") or names appeared more strongly correlated in ham messages. Furthermore, during the correlation analysis, we found that spam messages often followed a pattern where spam messages used promotional language, numbers, often in terms of offers or contact information, or had hyperlinks. These high-correlation words demonstrably added value to our models, most notably in the Decision Tree and Random Forest, where the models can split all data based on either the presence or

absence of these very keywords or high-correlation words.

Additionally, correlation analysis showed us that some features contributed little to no value to the classification process. For example, the analysis showed that features consisted of very common stop words or overly short terms. These were all filtered out during the preprocess stage. This helped improve the model by reducing noise and allowed the classifiers to be more focused on the extracts from the underlying data without unnecessary noise. It is important to keep in mind that correlation tells us the significance of the features but not the justification of the relationship. For example, a specific word may appear most frequently in spam messages because a promotional word in it is included, however, just because a message contains that specific word does not mean that the message is junk. Thus, correlation is most valuable when used with strong classification models, which can understand context.
In conclusion, the correlation showed us which features to choose wisely, and improved how we refine our model inputs, improving the models classification performance in the end especially with models such as Random Forest, and SVM were minimal noise between relevant and irrelevant features is critical.

B. Limitations

While the models we created had high accuracy, there were several limiting factors that reduce the efficacy of an SMS spam detection system in practice.

One key issue is the size of the dataset we used for training and testing. Spam messages were a much smaller proportion of the messages compared to legitimate messages. The imbalance in the dataset may have led to models that preferred the easier choice of predicting messages as legitimate. Though overall accuracy measures were good, the imbalance led to less able models to detect rare or new types of spam.

Another limitation stems from the format of the communication itself. SMS messages are often more casual, short, and laden with abbreviations, emojis, special characters, or shortened links. This lack of formality still provides a fashion to identify the intent of the message, but not as well with the trained models. TF-IDF as a technique helps to emphasize the informative terms in the messages but does not consider tone or context, creating additional limits in detecting well-crafted spam messages.

There is a fact we cannot overlook: most messages we focused on are in English. This entails that the usability of the model is reduced in countries where people use other languages. Supporting multiple languages would require training data and separate preprocessing setups for each language-a treat we did not give to the study.

New tricks are invented all the time by spammers to bypass filters. Since our models were trained with a fixed dataset, they might show reduced performance when faced with the newest patterns of spam, that is, if they have not been trained with fresh data recently.

And, though the best accuracy was given by Random Forest, the highest computational costs might have disqualified it for use in any mobile application. Lightweight models-take Naive Bayes for instance-might be more practical for such environments, even though their performance may be somewhat lower.

As a short story, the models may have yielded acceptable results, but improvements in the future must focus more on the imbalance of classes, always changing spam attacks, multiple languages, and resource-constrained optimization.

C. Practical Implication

This study produces valuable real-world implications for environments where mobile messaging is commonplace. As spam messages proliferate and become increasingly complex by design, detection systems are the last line of defense against fraud, phishing, and unwelcome advertising.

Our results indicate that machine learning models, from a Random Forest perspective at 99% accuracy, are efficacious in developing highly reliable spam detection applications. SVMs, and Decision Trees follow with an 98% accuracy rate with Naive Bayes following with a 94.3% accuracy rate when computational resources are a consideration.

When spam detection models are deployed to proactively block volume and harmful messages through messaging applications, mobile operating systems, or whatever appropriate service providers, a better end-user experience is achieved, which is an enhanced level of happy exit from the telecom world – a positive thing as it ends complaints and support tickets concerning spam.

Companies that utilize SMS for customer communications will certainly gain from having a spam filter. A proper spam filter will allow legitimate messages to go through without being flagged as spam and if the spam filter works well, it shouldn't cause any interruptions to your business. Simultaneously the spam detection system can prevent possible spam messages to employees or customers. For low-end devices (like some smartphones) or with low processing power, the low-end Data source filters, such as the Naive Bayes model would be a good fit. For higher-end models with greater computing power, like Random Forests, would be able to be run on cloud infrastructure to easily pass through an enormous amounts-of-messages to evaluate. In conclusion, this project has successfully demonstrated that machine-learning-based SMS spam detection is not merely a concept that relies on machine learning, it can be implemented in practice to protect all forms of communication, improve the legitimacy of SMS-messages, and promote digital trust.

VI . Conclusion And Future Work

The exponential growth of deceptive and unsolicited SMS messages has made it more important than ever to create reliable spam detection systems. This work was focused on using machine learning to automatically classify SMS messages as spam or legitimate (ham). We used a real-world,

labelled dataset and performed multiple text preprocessing steps including cleaning, tokenizing, and transforming the text to numerical form with TF-IDF. Once we preprocessed the text, we trained four different supervised learning models: Naïve Bayes, Decision Tree, Support Vector Machine (with a linear kernel), and Random Forest.

Our results demonstrated that the Random Forest model had the best applicability with a high reliability for detecting spam at a 99% accuracy. The Support Vector Machine and Decision Tree did quite well too at 98% accuracy. The Naïve Bayes achieved good performance at 94.3% accuracy, which is a good option when processing capacity is minimal.

While examining the models, we did not consider accuracy statistics in isolation. We considered other metrics such as precision, recall, and F1-score to observe the performance of these models concerning the imbalance issues in these datasets. Visual approaches such as confusion matrices and ROC curves were better tools for imparting insights about the consistencies and efficiencies of the models.

Overall, the study further unveiled the complexities of the problem. The challenges associated with imbalances between the two classes of messages continue, further compounded by the simplified and short lexical style characteristic of SMS, not to mention the ever-changing methodologies regularly appearing and being used by these spammers. Besides, since all messages in the dataset are English, any attempt of adapting the model to other languages or regional text styles would entail additional tuning and data.

Future Work

There are several methods to improve SMS Spam detection systems as follows:

Diversity and Balancing of Datasets: A good way to fix the model's bias is to have more examples of messages in other languages and regions. Once the dataset is expanded to represent a more diverse user group, we can also add more spam examples to eliminate the class imbalances and bias towards legitimate messages.

Deep Learning Models: Different architectures like, Bi-LSTM, GRU or transformer architectures with BERT can model more complex character for longer SMS preventing much higher detection accuracies.

Lightweight and Resource-Constrained Models in Real-Time: An innovative solution for deploying spam filters as SMS and therefore model training must happen on mobile device while being constrained to light-weight implementation without compromising on resources.

Use of Explainable AI: Using explainable AI (XAI) tools like SHAP and LIME selectively can lead user understanding about AI classification and help build trust especially as it is used by enterprises and organizations which need to trust that the model is basing its decisions based on the AI in the model being unable to explain why it is giving that classification.

Adaptive Learning: Making machine learning systems that automatically learn from incoming data is worth investigating to innovate spam detection methods and stay ahead of spammers adapting to the model's performance.

REFERENCES

[1] [1] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2005.

[2] [2] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with Naive Bayes – Which Naive Bayes?" in *Proc. CEAS*, vol. 17, no. 1, 2006, pp. 28–69.

[3] [3] UCI Machine Learning Repository, "SMS Spam Collection Dataset," [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

[4] [4] M. R. Al Saidat, S. Y. Yerima, and K. Shaalan, "Advancements of SMS spam detection: A comprehensive survey of NLP and ML techniques," *Procedia Computer Science*, vol. 244, pp. 248–259, 2024. [Online]. Available: https://doi.org/10.1016/j.procs.2024.10.198

[5] [5] R. M. Rahman, F. Islam, and S. S. Farid, "Smishing detection using deep learning techniques," *Journal of Information Security and Applications*, vol. 63, pp. 103003, 2022.

[6] [6] S. Al-Gaashani, M. Al-Rousan, and A. Jarrah, "SM-Detector: A hybrid model for detecting smishing SMS using NLP and deep learning," in *Proc. 2023 Int. Conf. on Cybersecurity Trends*, pp. 101–108.

[7] [7] T. S. Lim, W. Y. Loh, and Y. S. Shih, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Machine Learning*, vol. 40, no. 3, pp. 203–228, 2000.

[8] [8] H. Zhang, "The optimality of Naive Bayes," in *Proc. 17th Int. Florida Artif. Intell. Res. Soc. Conf.*, 2004, pp. 562–567.

[9] [9] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity-based approaches," *Security and Privacy*, vol. 3, no. 3, pp. e89, 2020.

[10] [10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[11] [11] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf.*, 2016, pp. 1135–1144.

[12] [12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.