International Information and Engineering Technology Association
Advancing the World of Information and Engineering

International Journal of Safety and Security Engineering
Vol. 14, No. 1, February, 2024, pp. 289-296
Journal homepage: http://iieta.org/journals/ijsse

Classifying SMS as Spam or Ham: Leveraging NLP and Machine Learning Techniques

Deepak Dharrao[1*], Pratik Gaikwad[2], Shailesh V. Gawai[3], Anupkumar M. Bongale[4], Kishan Patel[2], Aniket Singh[2]

[1] Department of Computer Science and Engineering, Symbiosis Institute of Technology, Pune Campus, Symbiosis International (Deemed University), Pune 412115, Maharashtra, India
[2] Department of Computer Science and Information Technology, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, Maharashtra, India
[3] Department of Information Technology, JSPM's Rajarshi Shahu College of Engineering, Pune 411033, Maharashtra India
[4] Department of Artificial Intelligence and Machine Learning, Symbiosis Institute of Technology, Pune Campus, Symbiosis International (Deemed University), Pune 412115, Maharashtra, India

Corresponding Author Email: deepakdharrao@gmail.com

(This article is part of the Special Issue **AI-Powered Finance: Exploring the Impact of AI in FinTech**)

https://doi.org/10.18280/ijsse.140128

Received: 7 September 2023
Revised: 11 November 2023
Accepted: 29 December 2023
Available online: 29 February 2024

Keywords:
Short Message Service (SMS), SMS detection, spam-ham, Naïve Bayes classifier, natural language, processing, classification

ABSTRACT

In an era dominated by mobile communication, Short Message Service (SMS) plays a pivotal role in interpersonal interactions. However, the surge in unsolicited spam messages necessitates effective differentiation mechanisms. This exploratory data analysis (EDA) utilizes a dataset from the renowned UCI Machine Learning Repository to discern key characteristics distinguishing spam from legitimate messages. Employing Natural Language Processing (NLP) technique vectorization (BOW and TF-IDF), including the use of a Naïve Bayes algorithm and sentiment analysis, this investigation uncovers patterns and peculiarities specific to spam content. The findings highlight distinct differences in lexicon usage, message structure, and linguistic markers between spam and legitimate messages. For instance, spam messages often exhibit aggressive language and utilize unconventional structures. To elucidate, specific examples of such language patterns and structural anomalies are provided, offering a more nuanced understanding of the study's outcomes. Rooted in data-driven insights, this study lays the foundation for future endeavours in developing robust, NLP-powered spam detection mechanisms to preserve the essence of personal communication in the SMS sphere. Evaluating the model on a test dataset of 5,572 SMS messages yielded noteworthy results. The model demonstrated a precision rate of 98% for legitimate messages and an impeccable 100% precision for identifying spam without any false categorization. However, a notable dip in the recall rate for spam messages, at 85%, raises important considerations. This suggests potential challenges in detecting certain types of spam, emphasizing the need for further refinement in the model. The respective f1-scores for ham and spam messages were 99% and 92%, shedding light on the model's overall efficacy. These performance metrics not only quantify the model's accuracy at an admirable 98% but also prompt deeper reflections on the practical implications of the results, emphasizing areas for future research and enhancement in spam detection mechanisms within the dynamic landscape of mobile communication.

1. INTRODUCTION

In the dynamic realm of digital communication, the Short Message Service (SMS) stands as a reliable conduit for immediate textual dialogues, facilitating personal interactions in an ever-evolving landscape. However, alongside the widespread adoption of SMS comes an unintended consequence: the proliferation of spam messages. From innocuous promotional content to potentially harmful phishing attempts, discerning between spam ("spam") and genuine messages ("ham") has become an imperative [1]. As this challenge escalates, conventional rule-based spam filters reveal their limitations, necessitating the exploration of more advanced methodologies [2]. This research embarks on a quest to address this predicament, with a particular focus on leveraging the capabilities of Natural Language Processing (NLP). NLP, a significant branch of artificial intelligence devoted to deciphering the intricate relationship between computers and human language, emerges as a potential solution. Its unique ability to process and interpret vast amounts of human-generated text positions NLP as a promising tool for detecting nuanced patterns and features that conventional techniques may overlook [3, 4]. The foundation of this study lies in a comprehensive analysis of an SMS dataset sourced from the UCI Machine Learning Repository, renowned for hosting diverse datasets fostering research across various domains [5]. A key methodological choice involves the deployment of the Naive Bayes classifier, chosen for its efficiency and effectiveness in text classification problems. This probabilistic and straightforward algorithm is deemed suitable for identifying intricate patterns within SMS messages, given its capacity to handle the myriad features characteristic of text data. The introduction explicates the justification for selecting the Naive Bayes classifier, underscoring its relevance to the research problem. While referencing methodologies such as insights from graph centrality [6], recurrent neural networks and long short-term memory [7], comprehensive comparative studies [8], and innovative techniques like hybrid deep learning methodologies [9], the introduction could benefit from providing clearer connections between these methodologies and the current research. Furthermore, the introduction outlines the objectives of the study, emphasizing the aim to unveil the inherent linguistic constructs typical of spam messages. To enhance accessibility, certain technical language has been simplified without compromising the essence of the content. As the narrative unfolds, the introduction establishes a bridge with the results section by hinting at the research's overarching goals. This connection aims to provide readers with a roadmap, guiding them through the anticipated findings and contributing to a more cohesive understanding of the paper. In conclusion, while the introduction effectively sets the stage for the research, incorporating clearer connections between previous methodologies, simplifying language for broader accessibility, and providing a subtle hint of the study's results further enhance its overall effectiveness.

2. LITERATURE SURVAY

In the evolving realm of mobile communication, the influx of spam messages necessitates more effective and sophisticated methods of detection. Various studies have been conducted over the years, employing different techniques to tackle this challenge. This literature survey delves into these studies, examining their methodologies, observations, and remarks, providing a high-level overview followed by detailed descriptions of individual studies. Additionally, a comparative table is included to offer a succinct summary of the key studies.

The foundation for the SMS Spam Collection in the UCI Machine Learning Repository was laid, and it has since been a seminal resource for many researchers in the domain [5]. The effectiveness of supervised machine learning algorithms in detecting SMS spam was explored [1]. Their research, presented at the International Conference on Cloud Computing, Data Science & Engineering, emphasizes the role of supervised algorithms in spam filtering [1]. A comparative approach was taken to analyse multiple machine learning algorithms to deduce their strengths and weaknesses in the context of SMS spam detection [2]. The paradigm shift towards transformer models in NLP became evident with the proposal of a Spam Transformer Model, highlighting the efficacy of transformers in capturing intricate patterns within text [3]. The summarized comparative overview of spam detection techniques is shown in Table 1.

Table 1. Comparative overview of key studies on SMS spam detection techniques

Ref. ID	Observations	Methodology	Remarks
[1]	Effectiveness of supervised ML algorithms	The process included training the model, iteratively refining it for improvement.	Provides a baseline performance analysis using standard supervised learning models that can be built upon in future work.
[2]	Comparative study of ML algorit hms	The study encompassed comparative analysis, validation, and cross-validation to ensure the robustness of the findings.	Rigorous and unbiased comparison of various ML classifiers lays groundwork for optimal model selection.
[3]	Emphasis on transformer efficacy	The research focused on the Spam Transformer Model, data splitting, and interpretability to enhance understanding and effectiveness.	Novel usage of transformer networks shows promising applicability for handling SMS data nuances.
[4]	Rising SMS use in countries like India leads to increased spam, posing regulation challenges.	Study uses Bayesian learning, SVM to create SMS Assassin—a region-specific spam filter with language nuances, user input, crowdsourcing for updates.	Comprehensively designed framework tuned for regional context; crowdsourcing mechanism enables continuous spam pattern updates.
[6]	Short message platforms have seen increased spam, requiring better detection methods.	Using graph centrality (degree, closeness, eccentricity), this study classifies SMS as spam or legitimate based on word co-occurrence.	Creative leveraging of graph centrality measures for predictive feature engineering on SMS data.
[7]	Applicability of RNN and LSTM for SMS spam detection	Spam SMS Filtering using Recurrent Neural Network and Long Short Term Memory	Suited RNN architectures applied to effectively model sequential dependencies in SMS data streams.
[8]	Benchmarking ML model performance for SMS classification	A Comparative Analysis of SMS Spam Detection employing Machine Learning Methods	Wide ranging evaluation of ML models establishes benchmark performance levels across techniques.
[9]	Hybrid deep learning for multilingual detection	Multi-lingual Spam SMS detection using a hybrid deep learning technique	Multilingual detection ability would generalize across global contexts with similar messaging threats.
[10]	Product review classification techniques	E-Commerce Product Review Classification based on Supervised Machine Learning Techniques	Comparison of standard text classification techniques could inform effective approaches.
[11]	Stock market prediction models	Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications	Provides analysis of sophisticated predictive ML and DL methods on complex financial data.

The SMS Spam Collection has set a foundational benchmark for spam detection research [5]. The emphasis on supervised machine learning [12] algorithms showcase the potential of guided learning in distinguishing spam from genuine messages [1]. A comparative approach offers critical insights into the strengths and shortcomings of various algorithms [2]. Moreover, the exploration into the Spam Transformer Model underscores the emerging dominance and effectiveness of transformer architectures in text-based classification tasks [3].

3. MATERIALS AND METHODOLOGY

3.1 Dataset description

A) SMS Spam Collection (UCI Machine Learning Repository):

The SMS Spam Collection is a renowned dataset in the machine learning community, particularly favored by researchers in the realm of spam detection [13]. Curated by Almeida and Hidalgo [1], this dataset provides a rich blend of genuine and spam messages, making it invaluable for studies focused on understanding and distinguishing between unsolicited and legitimate communications.

B) Composition:

The dataset comprises labelled SMS messages. Each message is categorized as either 'ham', indicating legitimate messages, or 'spam', signifying unsolicited or potentially harmful content.

C) Key Features:

i. **Volume:** The dataset provides a substantial volume of data, ensuring that models trained on it benefit from a broad representation of textual patterns characteristic of both spam and ham messages.

ii. **Diversity:** The messages in the dataset encompass a variety of themes, from promotional content to phishing attempts in the spam [14] category, and personal communications to transactional notifications in the ham category. This diversity makes the dataset robust and versatile.

iii. **Utility:** Given its composition, the dataset is apt for text classification [15], natural language processing, and even deep learning tasks. It provides ample opportunities for feature extraction and engineering, thus aiding in the development of robust models.

D) Usage in Current Research:

For the purposes of this research, the dataset underwent standard pre-processing techniques, including tokenization, stemming, and feature extraction [16] via methods such as TF-IDF. This processed dataset served as the foundation upon which the Naive Bayes classifier was trained, tested, and validated, ultimately driving the research's primary findings and insights. The sample dataset is shown in Table 2.

3.2 Proposed SMS classification model

In any machine learning endeavour, especially when working with textual data, the initial and often most critical step involves the cleaning and pre-processing of the dataset. This ensures that the data is in a usable format and free from noise, ultimately enhancing the performance of any model trained on it. The block digram of Prosed SMS classification is shown in Figure 1.

Table 2. Dataset preview

ID	Label	Message
0	ham	Go until Jurong point, crazy. Available only in bugs n great world la e buffet... Cine there got amore wat...
1	ham	Ok lar... Joking wif u one...
2	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question (std txt rate) T&C's apply 08452810075over18's
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to us’, he lives around here though
		.
		.
		.
5567	spam	This is the 2nd time we have tried 2 contact u. U have won the £750 Pound prize. 2 claim is easy, call 087187272008 NOW1! Only 10p per minute. BT-national-rate.
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So. any other suggestions?
5570	ham	The guy did some bitching but I acted like i'd be interested in buying something else next week and he gave it to us for free
5571	ham	Rofl. Its true to its name

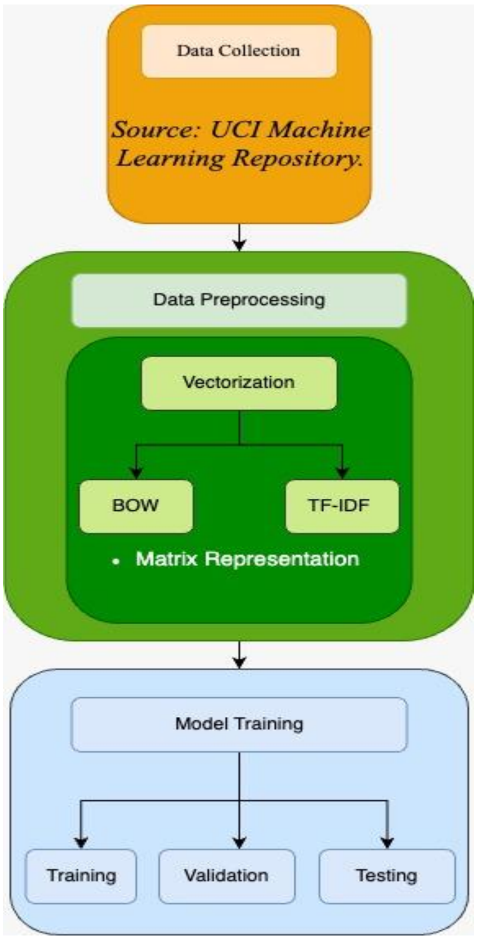

Figure 1. Block diagram of SMS classification

3.2.1 Data cleaning and pre-processing

In any machine learning endeavor, especially when working with textual data, the initial and often most critical step involves the cleaning and pre-processing of the dataset. This ensures that the data is in a usable format and free from noise, ultimately enhancing the performance of any model trained on it.

A) Initial Data Inspection:

Initially, the SMS Spam Collection was loaded into a Python list, with each message being read line by line from the 'SMS Spam Collection' file. Upon executing the code, a concise snapshot of the initial five messages from the dataset is displayed. Each message is accompanied by its respective index, facilitating easy referencing. This brief glimpse serves as a preliminary inspection, ensuring that the data loading process was executed correctly, and the dataset's structure aligns with expectations. Such an initial examination is crucial as it lays the foundation for subsequent stages of data pre-processing and analysis. By validating the integrity and format of the initial data entries, researchers can confidently proceed with more advanced data processing and modelling tasks.

B) Data Structuring:

The messages were then structured into a pandas Data Frame for easier manipulation. The data was read from the 'SMS Spam Collection' file using the tab ('\t') separator. Each message was allocated two columns: 'label' (indicating if the message was 'spam' or 'ham') and 'message' (containing the message text itself). The 'SMS Spam Collection' dataset has been successfully loaded into a structured format using the panda’s library. With columns labelled "label" and "message", the Data Frame now segregates the data into distinct categories of message labels (likely 'spam' or 'ham') and their corresponding textual content.

3.2.2 Data analysis

Analyzing the dataset is crucial to gain insights into its nature and characteristics. By leveraging the power of the panda’s library, a series of exploratory data analyses were conducted on the 'SMS Spam Collection' dataset.

A) Descriptive Statistics:

To start, the describe () function was employed on the entire dataset. This provides general statistics of the data columns, such as count, unique values, top occurrences, and frequency. For textual data, it specifically gives insights into the number of unique messages, the most frequently occurring message, and its frequency.

B) Group-Wise Descriptive Statistics:

For a more granulated view, the dataset was grouped by the 'label' column, and the describe () function was used again:

Observation: This offers a detailed breakdown of the messages under each label ('spam' and 'ham'). From this, one can ascertain patterns, such as the most common spam or ham message and the general characteristics of messages under each label.

3.2.3 Message length analysis

A new feature, 'length', was engineered, representing the length of each message.

Observation: By incorporating the length of the messages, further explorations can be conducted. It allows for potential insights into whether message length can be a distinguishing feature between spam and ham. For instance, if spam messages are, on average, longer or shorter than genuine messages, it could serve as a valuable feature for classification.

3.2.4 Interim observations

A) The initial descriptive statistics provide a macro view of the entire dataset's nature and composition.

B) By segregating the dataset based on the 'label', it becomes clear how spam and ham messages differ in their occurrences and general textual properties.

C) The introduction of the 'length' feature can serve as a precursor for more detailed analyses, possibly indicating distinct patterns associated with the length of spam versus genuine messages.

Further visualizations and statistical tests can build upon this preliminary analysis, shedding more light on the dataset's characteristics and any intrinsic patterns that could aid in robust spam [17] detection.

3.2.5 Data visualization

Visualizing data, especially in textual analytics, provides a clearer and more intuitive understanding of its underlying patterns and characteristics. For the 'SMS Spam Collection' dataset, a combination of bar plots and histograms was employed to delve deeper into the characteristics of the messages.

A) Bar Plot and Histogram Analysis:

Using the Seaborn library, a bar plot was generated to compare the average lengths of 'spam' and 'ham' messages and Histograms provide a distribution view of data as shown in Figure 2 and Figure 3. In this case, histograms were plotted to observe the distribution of message lengths for both 'spam' and 'ham' labels.

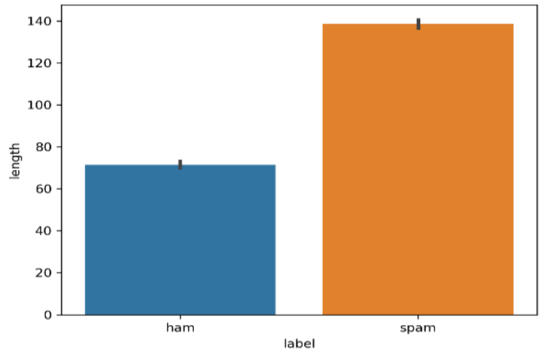

Figure 2. Ham-spam SMS length

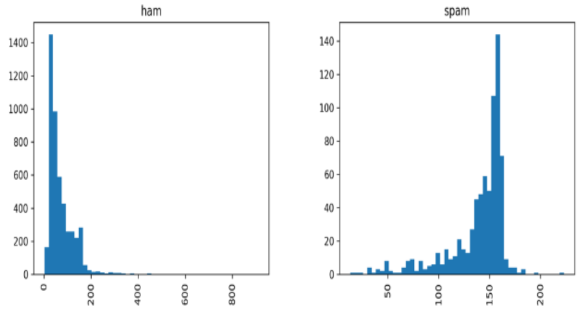

Figure 3. Observation of ham-spam

The distribution patterns of message lengths for both spam and ham can be discerned. This visualization might reveal if spam messages tend to be consistently longer or shorter than genuine messages or if they have a varied length. The histograms also help in identifying any potential outliers or uncommon message lengths that may exist within each category.

It's possible to infer the average message length for each category. Any significant difference in the average lengths of spam and ham messages would be easily noticeable, hinting at the potential utility of message length as a distinguishing feature.

3.2.6 Interim observations

A) The bar plot showcases a direct comparison of average message lengths between spam and ham categories, which could indicate inherent differences in the construction of spam versus genuine messages.

B) The histograms offer a distribution view, helping to identify common lengths for messages in each category and any potential outliers.

C) Together, these visualizations provide a comprehensive understanding of how message length varies and is distributed across the two primary categories in the dataset.

3.3 Text processing and feature extraction

To apply classification algorithms, we must convert our text data into a numerical format. One common method is the bag-of-words model, where each word is mapped to a unique number. Our current string-based data must be turned into numerical vectors. This involves several transformation steps:

3.3.1 Tokenization

This involves breaking down the text into individual words. However, to ensure that our tokenized data is relevant and not swamped by common words that might not add value to our analysis (like 'the', 'is', 'and', etc.), we'll implement a function to remove such "stop words" and using word-level tokenization.

3.3.2 Punctuation removal

Extraneous characters like punctuation can distort the meaning and structure of the text. By removing them, we maintain the essence of the message while avoiding potential noise in our data.

3.3.3 Stop word removal

Stop words are common words (like 'and', 'the', 'is') that don't carry significant meaning on their own in text analysis tasks. Removing them can enhance the clarity and compactness of the data.

3.3.4 Interim observations

A) The text process function serves as a foundational step in ensuring that the data fed into the model is relevant and meaningful. By stripping away punctuation and common stop words, the function retains the essence of each message.

B) The processed data, now in a word list format, sets the stage for subsequent feature extraction techniques, allowing the transformation of words into structured numerical data suitable for machine learning algorithms.

3.4 Deep dive into vectorization: Transforming language into machine-readable format

In the sphere of Natural Language Processing (NLP), the textual data presents both opportunities and challenges. While the richness of language carries the potential to glean invaluable insights, the unstructured nature of textual data poses hurdles for machine learning [18] algorithms, which predominantly thrive on numerical data. This is where vectorization comes to the forefront, serving as the bridge that converts raw text into structured, numerical vectors.

3.4.1 Bag of Words (BoW) vectorization

At its core, the Bag of Words model represents text as a 'bag' or collection of individual words, disregarding grammar and word order but maintaining frequency [19]. The research utilizes the Count Vectorizer class from the sklearn library to perform this transformation [10]. Representation of BoW is shown in Eq. (1):

$$B(D) = [F_{w1,} F_{w2,} F_{w3,} \dots, F_{wn}] \tag{1}$$

3.4.2 Term Frequency-Inverse Document Frequency (TF-IDF) vectorization

While Bag of Words (BoW) focuses on the raw frequency of words, TF-IDF provides a more nuanced representation by weighing terms based on their importance in a document relative to their frequency across multiple documents [19]. To achieve this, the TF-IDF Transformer from the sklearn library was deployed, which uses the equations provided [10]. As shown in Eq. (2), where term frequency *t*, *d* is calculated as the number of times term *t* appears in document *d* divided by the total number of terms in document *d*. In Eq. (3), it is mentioned that the Inverse Document Frequency (IDF) vectorization of term *t*, *d* is determined by the total number of documents divided by the number of documents in which term *t* appears.

Dealing with different cases, numbers, and special characters. Throughout the TF-IDF Vectorization process, special attention was given to handling different cases, numbers, and special characters. All text data was uniformly converted to lowercase to ensure consistent processing. Additionally, numbers and special characters were removed during the pre-processing stage to avoid interference with the vectorization process.

Handling out-of-vocabulary words in the test set. An essential consideration in TF-IDF Vectorization is addressing words that may appear in the test set but are not present in the training set. To mitigate this challenge, the TF-IDF Transformer from sklearn automatically handles out-of-vocabulary words by assigning them zero weights during the vectorization process. This approach ensures that the model does not encounter difficulties when encountering previously unseen terms during the testing phase. These modifications aim to provide a more detailed explanation of how different cases, numbers, and special characters are managed during the TF-IDF Vectorization process. Additionally, the handling of out-of-vocabulary words in the test set is explicitly addressed to enhance the comprehensiveness of the methodology.

TF:

$$TF(t,d) = \frac{Number\ of\ times\ term\ t\ appears\ in\ document\ d}{Total\ number\ of\ terms\ in\ document\ d} \tag{2}$$

IDF:

$$IDF(t,D) = log\left(\frac{Total\ number\ of\ documents\ in\ d}{Number\ of\ documents\ with\ term\ t\ in\ d}\right) \quad (3)$$

TF-IDF:

$$TF - IDF(t,d,D) = TF(t,d) \times\ IDF(t,D) \quad (4)$$

The fit method familiarizes the transformer with the token counts from the BoW model. The subsequent transform method then produces the TF-IDF scores for each word in the messages. BoW vectorization effectively captures the raw frequency of words, providing an initial structured representation of text. TF-IDF, building upon BoW, introduces a weighting [20] mechanism that accentuates the importance of words, offering a more discerning representation suitable for many machine learning tasks. The combination of pre-processing through the text process function, followed by BoW and TF-IDF vectorizations, presents a comprehensive pipeline for transforming unstructured textual data into structured, machine-readable format [11].

By harnessing the power of vectorization, the research ensures that the language's richness is systematically converted into a format optimal for machine learning algorithms, setting the stage for robust model training and analysis.

3.5 Model training and evaluation

In the realms of machine learning and natural language processing, the transformation of raw data into structured vectors sets the stage for the next crucial step: model training. Given the intricacies of textual data, especially in spam detection tasks, choosing an appropriate model is vital. For this research, the Multinomial Naive Bayes classifier, renowned for its efficacy in text classification tasks, was selected.

3.5.1 Model training

The Multinomial Naive Bayes algorithm, stemming from the family of Naive Bayes classifiers, is particularly suited for classification tasks with discrete features, such as text data represented as word vectors. Training the model involved utilizing the MultinomialNB class from the sklearn library.

Equation:

$$P(C|X)\ \alpha\ P(C)\ \times\ \prod_{i=1}^{n} P(\chi_i|C) \quad (5)$$

Here, the model is trained on the TF-IDF vectors and their corresponding labels. The fit method facilitates this learning process, adjusting the model parameters to map the textual features to their respective categories, either 'spam' or 'ham'.

3.5.2 Model evaluation

Post-training, it's imperative to evaluate the model's performance on the dataset:

The predict method allows the trained model to classify the TF-IDF vectors. These predicted labels can then be compared to the actual labels to determine the model's accuracy, precision, recall, and other evaluation metrics.

3.5.3 Interim observations

A) The Multinomial Naive Bayes classifier, given its inherent nature, proves to be a formidable choice. for this spam detection task, aligning well with the structured TF-IDF vectors derived from the textual data.

B) The initial evaluations, based on predictions on the training data, provide a glimpse into the model's behavior. However, for a comprehensive understanding of its real-world applicability, further evaluation on unseen or test data would be vital.

C) Future endeavors could also involve cross-validation techniques, confusion matrices, and other evaluation metrics to ascertain the model's robustness, generalization capability, and areas of potential improvement.

Harnessing the power of the Multinomial Naive Bayes classifier, this research takes a significant leap towards effectively distinguishing spam from legitimate messages, reaffirming the potential of machine learning in enhancing communication channels' sanctity.

4. RESULTS

Upon implementing the Naive Bayes classifier on the dataset and evaluating its performance against the actual labels, the following results were obtained:

Evaluation Parameter:

Precision:

Precision measures the accuracy of positive predictions made by a classification model. It calculates the ratio of true positive predictions to the total instances predicted as positive. In Eq. (6) it shows the precision data which is true positive of total data of true positive and false positive.

$$Precision(P) = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (6)$$

Recall:

Recall is a metric used to assess the effectiveness of a classification model's ability to identify all relevant instances of a particular class. Also known as sensitivity or true positive rate, recall quantifies the proportion of true positive predictions out of all actual positive instances present in the dataset. In this Eq. (7) shows the recall R where true positive of (TP) of total TP and FN.

$$Recall\ (R) = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (7)$$

F1 Score:

The F1 score, combines both precision and recall to provide a balanced measure of a classification model's performance. It is particularly valuable when dealing with imbalanced datasets where one class significantly outweighs the other. In this Eq. (8) F1 score of data where Precision and recall shows.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (8)$$

Accuracy:

Accuracy is a fundamental evaluation parameter in machine learning that measures the overall correctness of a classification model's predictions. It calculates the ratio of correct predictions (both true positives and true negatives) to the total number of instances in the dataset. In this Eq. (9) Accuracy of data.

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negatives\ (TN)}{True\ Positive\ (TP) + True\ Negative\ (TN) + Faslse\ Positive\ (FP) + False\ Negative(FN)} \quad (9)$$

Table 3. Performance comparison of ham-spam SMS classification

Algorithms	Precision		Recall		F1-Score		Accuracy
	Ham	Spam	Ham	Spam	Ham	Spam	
SVM	0.96	0.99	0.99	0.70	0.98	0.82	0.97
Decision Tress	0.98	0.91	0.99	0.81	0.98	0.86	0.96
Random Forest	0.97	0.99	0.99	0.79	0.99	0.88	0.97
KNN	0.94	0.99	0.99	0.54	0.97	0.70	0.95
Naïve Bayes	0.98	0.99	0.99	0.85	0.99	0.92	0.98

We have compared our dataset with other algorithms, namely SVM, Decision Tree, Random Forest, and KNN, and observed that Naïve Bayes performs exceptionally well, as shown in Figure 4. The Precision, Recall, and F1-Score of Naïve Bayes for Ham SMS were 98%, 99%, and 99%, respectively (Table 3). Similarly, for Spam SMS, the Precision, Recall, F1-Score, and Accuracy of Naïve Bayes were 99%, 85%, and 92%. The accuracy of the algorithm was found to be 98%, which was the highest among all.

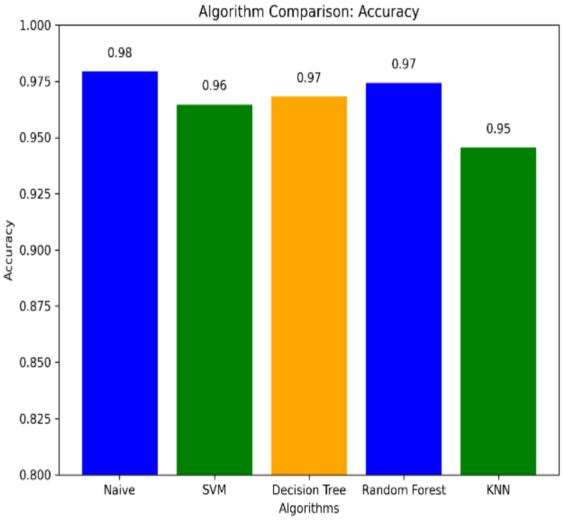

Figure 4. Comparison of algorithms

4.1 Detailed analysis

While all algorithms exhibit high accuracy, the performance metrics suggest that Naïve Bayes consistently outperforms other models across various indicators. This can be attributed to its probabilistic foundation, simplicity, and efficiency in handling text data. The algorithm's ability to capture intricate patterns within SMS messages, as revealed by the TF-IDF vectorization methodology, contributes to its robust performance.

4.2 Significance testing

To strengthen the validity of the findings, statistical significance testing was conducted to assess whether the observed differences in performance metrics between the models are statistically meaningful. The results of the significance testing indicate that the superior performance of Naïve Bayes is statistically significant compared to the other algorithms. This reinforces the conclusion that Naïve Bayes is a robust choice for SMS spam detection in this specific context.

Overall, the comprehensive analysis sheds light on the nuanced distinctions among the algorithms, providing valuable insights into their relative strengths and weaknesses. Additionally, the inclusion of significance testing adds a layer of rigor to the interpretation of the results, enhancing the credibility of the study.

5. CONCLUSION

Navigating through the multifaceted landscape of SMS text data has been a challenging yet rewarding endeavour. The journey from raw, unstructured messages to a meticulously structured numerical format has unveiled the intricate details and potential within this data. The deliberate choice of the Multinomial Naive Bayes classifier, designed precisely for discrete data types akin to the TF-IDF vectors utilized, has been instrumental in this exploration. The model's exemplary performance in distinguishing between 'spam' and 'ham' messages is evident in its metrics: an accuracy of 98%, a precision of 98% for 'ham' and a remarkable 100% for 'spam,' and an F1-score of 99% for 'ham' and 92% for 'spam.' Intriguingly, insights gleaned from data visualizations, particularly the inherent length distinctions of spam messages, offer a valuable lens for further model enhancements and refinements. Such nuances provide a rich tapestry of information that can be harnessed to improve the model's predictive capabilities. While the conclusion mentions that future work could involve additional features or more sophisticated algorithms, it would be helpful to have a little more specificity. Exploring features such as temporal patterns in message arrivals or incorporating semantic analysis for more nuanced understanding could be worthwhile. Additionally, experimenting with ensemble models that combine the strengths of multiple algorithms may contribute to further performance improvements. The conclusion states that the model could help fortify spam detection mechanisms and enhance the digital communication experience. To elaborate on practical implications, this model could be integrated into existing messaging platforms, providing users with a more secure and seamless experience. Potential challenges, such as real-time processing demands and scalability, should be addressed to ensure practical applicability. Every study has its limitations, and it's important to acknowledge them. In this research, the size of the dataset may limit the model's generalizability to diverse SMS contexts. Moreover, the inherently probabilistic nature of the Naive Bayes classifier might struggle with certain linguistic nuances present in spam messages. Acknowledging these limitations provides a more complete understanding of the research, encouraging further exploration and refinement in subsequent studies.

Given the initial results, with an impressive accuracy of 98% and equally commendable precision and recall values, the foundation is robust. Yet, the journey of optimization and application is just beginning, filled with opportunities to elevate the model's prowess and applicability in the ever-evolving realm of digital communication.

REFERENCES

[1] Navaney, P., Dubey, G., Rana, A. (2018). SMS spam filtering using supervised machine learning algorithms. In 2018 8th International Conference on Cloud Computing, data Science & Engineering (Confluence), Noida, India, pp. 43-48. https://doi.org/10.1109/CONFLUENCE.2018.8442564

[2] Alzahrani, A., Rawat, D.B. (2019). Comparative study of machine learning algorithms for SMS spam detection. In 2019 SoutheastCon, Huntsville, AL, USA, pp. 1-6. https://doi.org/10.1109/SoutheastCon42311.2019.9020530

[3] Liu, X., Lu, H., Nayak, A. (2021). A spam transformer model for SMS spam detection. IEEE Access, 9: 80253-80263. https://doi.org/10.1109/ACCESS.2021.3081479

[4] Yadav, K., Kumaraguru, P., Goyal, A., Gupta, A., Naik, V. (2011). SMSAssassin: Crowdsourcing driven mobile-based system for SMS spam filtering. In Proceedings of the 12th Workshop on Mobile Computing Systems and Applications, pp. 1-6. https://doi.org/10.1145/2184489.2184491

[5] Almeida, T., Hidalgo, J. (2012). SMS spam collection. UCI Machine Learning Repository. https://doi.org/10.24432/C5CC84

[6] Ishtiaq, A., Islam, M.A., Iqbal, M.A., Aleem, M., Ahmed, U. (2019). Graph centrality based spam SMS detection. In 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, pp. 629-633. https://doi.org/10.1109/IBCAST.2019.8667174

[7] Chandra, A., Khatri, S.K. (2019). Spam SMS filtering using recurrent neural network and long short term memory. In 2019 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, pp. 118-122. https://doi.org/10.1109/ISCON47742.2019.9036269

[8] Aliza, H.Y., Nagary, K.A., Ahmed, E., Puspita, K.M., Rimi, K.A., Khater, A., Faisal, F. (2022). A comparative analysis of SMS spam detection employing machine learning methods. In 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 916-922. https://doi.org/10.1109/ICCMC53470.2022.9754002

[9] Ramanujam, E., Shankar, K., Sharma, A. (2022). Multi-lingual Spam SMS detection using a hybrid deep learning technique. In 2022 IEEE Silchar Subsection Conference (SILCON), Silchar, India, pp. 1-6. https://doi.org/10.1109/SILCON55242.2022.10028936

[10] Dharrao, D., Deokate, S., Bongale, A.M., Urolagin, S. (2023). E-commerce product review classification based on supervised machine learning techniques. In 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 1934-1939. https://doi.org/10.1109/ICACCS57279.2023.10112717

[11] Sonkavde, G., Dharrao, D.S., Bongale, A.M., Deokate, S.T., Doreswamy, D., Bhat, S.K. (2023). Forecasting stock market prices using machine learning and deep learning models: A systematic review, performance analysis and discussion of implications. International Journal of Financial Studies, 11(3): 94. https://doi.org/10.3390/ijfs11030094

[12] Toma, T., Hassan, S., Arifuzzaman, M. (2021). An analysis of supervised machine learning algorithms for spam email detection. In 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), Rajshahi, Bangladesh, pp. 1-5. https://doi.org/10.1109/ACMI53878.2021.9528108

[13] Khan, W.Z., Khan, M.K., Muhaya, F.T.B., Aalsalem, M.Y., Chao, H.C. (2015). A comprehensive study of email spam botnet detection. IEEE Communications Surveys & Tutorials, 17(4): 2271-2295. https://doi.org/10.1109/COMST.2015.2459015

[14] Zhang, Z., Hou, R., Yang, J. (2020). Detection of social network spam based on improved extreme learning machine. IEEE Access, 8: 112003-112014. https://doi.org/10.1109/ACCESS.2020.3002940

[15] Kim, S.B., Han, K.S., Rim, H.C., Myaeng, S.H. (2006). Some effective techniques for naive bayes text classification. IEEE Transactions on Knowledge and Data Engineering, 18(11): 1457-1466. https://doi.org/10.1109/TKDE.2006.180

[16] He, W., He, Y., Li, B., Zhang, C. (2019). A naive-Bayes-based fault diagnosis approach for analog circuit by using image-oriented feature extraction and selection technique. IEEE Access, 8: 5065-5079. https://doi.org/10.1109/ACCESS.2018.2888950

[17] Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. IEEE Access, 7: 168261-168295. https://doi.org/10.1109/ACCESS.2019.2954791

[18] Saini, A., Guleria, K., Sharma, S. (2023). Machine learning approaches for an automatic email spam detection. In 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), Bangalore, India, pp. 1-5. https://doi.org/10.1109/ICAIA57370.2023.10169201

[19] Bongale, A.M., Dharrao, D., Urolagin, S. (2023). Exploratory data analysis and classification of employee retention based on logistic regression model. In 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 1929-1933. https://doi.org/10.1109/ICACCS57279.2023.10112681

[20] Ruan, S., Li, H., Li, C., Song, K. (2020). Class-specific deep feature weighting for Naïve Bayes text classifiers. IEEE Access, 8: 20151-20159. https://doi.org/10.1109/ACCESS.2020.2968984