SMS Spam Detection and Classification

Using Machine Learning and Natural Language Processing

Abishek J 24MDT1032

Master of Science, Data Science

School of Advance Sciences,

VIT Chennai Campus, Vandalur- Kelambakam Road,

Chennai-600127, Tamil Nadu, India

[Abishek.j.2024a@vitstudent.ac.in](mailto:Abishek.j.2024a@vitstudent.ac.in)  
  
Under the Guidance of

Dr. A. Felix

Associate Professor,

School of Advance Sciences,

VIT Chennai Campus, Vandalur-Kelambakam Road,

Chennai-600127, Tamil Nadu, India

[Felix.a@vit.ac.in](mailto:Felix.a@vit.ac.in)

**Abstract** – The widespread use of mobile communication has led to an increase in unsolicited SMS messages, commonly referred to as spam. Detecting and filtering such messages is crucial to ensuring secure and efficient communication. This project aims to develop a robust spam detection model using a dataset of 5,574 SMS messages labeled as "ham" (legitimate) or "spam." By leveraging advanced Natural Language Processing (NLP) techniques and machine learning algorithms, the system seeks to accurately classify SMS messages while minimizing errors. The process involves extensive data preprocessing, including normalization, removal of special characters, stop word elimination, stemming, and lemmatization. Feature extraction is performed using TF-IDF vectorization to transform the text into numerical representations suitable for machine learning models. Both traditional machine learning and deep learning approaches are employed to achieve high classification accuracy. This work offers a reliable solution to improve mobile communication experiences by effectively filtering unwanted messages.,

Keywords:SMS, Artificial Intelligence, NLP, spam detection, vectorization.

1. **Introduction**

The rapid expansion of mobile communication has revolutionized personal and professional interactions worldwide. However, this growth has also been accompanied by a surge in spam SMS messages, which can disrupt communication and pose security risks through phishing and other malicious activities. Addressing this challenge, spam detection systems play a vital role in safeguarding users' communication channels.

This project focuses on building a reliable SMS spam detection system using a dataset of 5,574 messages labeled as either "spam" or "ham." The development process includes cleaning and standardizing text data through techniques like lowercasing, removal of special characters, and stop word elimination. Further, the text data undergoes transformation using methods such as stemming, lemmatization, and TF-IDF vectorization to prepare it for machine learning algorithms. Both machine learning and deep learning models are explored to classify messages accurately, with an emphasis on minimizing false positives and negatives.

The objective of this project is to create a scalable and efficient system that enhances user experiences by filtering out unwanted messages and ensuring secure communication. The findings and methods can be extended to similar applications in detecting spam across various communication platforms.

1. **Literature Review**

The increasing reliance on machine learning techniques for spam detection has been a focal point of research, driven by the limitations of traditional rule-based approaches in handling evolving spam tactics. Studies like those by Mallampati et al. (2019) and Torabi et al. (2015) emphasize the significance of Support Vector Machines (SVMs) in spam classification due to their ability to effectively handle non-linear data distributions through kernel functions, achieving superior accuracy over other algorithms. Similarly, Bayesian filtering, introduced in the 1990s, continues to serve as a foundational technique, estimating spam probabilities based on word frequencies. Recent works have also highlighted the role of feature extraction and selection, with methods such as Term Frequency-Inverse Document Frequency (TF-IDF), N-grams, and document frequency significantly enhancing spam classification accuracy. Additionally, novel approaches like the SIGMM model proposed by Qiu et al. (2018) address challenges such as data imbalance and reliance on user relationships, further advancing spam detection capabilities.

Recent surveys, such as those by Abayomi-Alli et al. (2019) and Świtalski and Kopówka (2019), underscore the growing threat of spam beyond emails, extending to SMS and social media platforms, necessitating adaptive and scalable filtering techniques. These studies also reveal a shift towards ensemble methods and deep learning models, which provide robust solutions by leveraging complex data patterns. However, challenges remain, including the computational demands of neural networks and the reliance on labeled datasets for supervised learning approaches. Notably, Bhowmick and Hazarika (2016) and Agarwal et al. (2024) highlight research gaps in handling large-scale multimedia data and propose innovative strategies such as gradient descent-based models and alternative learning methods. These findings collectively emphasize the importance of continuous innovation in spam detection, focusing on enhancing classification accuracy, minimizing false positives and negatives, and addressing emerging spam threats.tactics of attackers.

1. **Research Gap**

Despite considerable advancements in spam detection, several critical research gaps remain, particularly concerning adaptability, computational efficiency, and handling of new spam formats. While Support Vector Machines (SVMs) and Bayesian filters have proven effective in classifying text-based spam, they encounter limitations with large-scale datasets and multimedia content, which are increasingly common in modern spam. Additionally, traditional machine learning methods often rely on extensive labeled data, which is time-consuming and costly to curate, thus limiting scalability. Existing methods also struggle with real-time detection in dynamic environments, where spammers continuously evolve their techniques to bypass conventional filters. Thus, a pressing need exists for more flexible and efficient approaches that can handle diverse and large datasets, operate in real-time, and reduce dependency on labeled data.

Furthermore, while some research has explored ensemble methods and deep learning models, these approaches typically require high computational power and extensive training, which may not be feasible for all applications, particularly on mobile and resource-constrained devices. Another research gap lies in the limited focus on user feedback mechanisms that could help adapt spam filters based on real-time user input, enhancing overall accuracy and responsiveness. There is also a lack of comprehensive strategies to handle multilingual and multimedia spam content, which complicates detection and classification. Addressing these gaps by developing adaptive, resource-efficient, and user-interactive spam filtering systems could lead to significantly improved spam detection and a more secure communication environment.

1. **Methodology**

This project aims to create a comprehensive spam detection system, focusing on two core components: an SMS spam detection model and an email spam detection chatbot integrated with a user’s inbox.

For SMS spam detection, the approach begins with data preprocessing, including transforming text to lowercase, removing special characters and stop words, and applying lemmatization and stemming to standardize language features. After this, feature extraction techniques, such as TF-IDF vectorization, are employed to convert text data into numerical representations, making it suitable for machine learning models. Traditional machine learning algorithms and deep learning techniques will then be evaluated to find the most accurate model, with a focus on minimizing false positives and negatives.

In parallel, the project also aims to develop an email spam detection model integrated with a chatbot. This chatbot, designed using Google’s Dialogflow, will provide real-time alerts to users when potential spam emails are detected in their inboxes. Dialogflow is selected due to its flexibility and ability to connect with external APIs, which will enable seamless integration with the spam detection model. This dual approach—robust spam detection and proactive alert system—intends to enhance user safety and communication efficiency.

**4.1 Dataset Description**

The dataset, available in CSV format, comprises 5,572 SMS messages labeled as either "spam" or "ham" (legitimate), intended for classification in spam detection research. It contains five columns: the first column, v1, provides the label for each message, specifying whether it is "spam" or "ham," with no missing values across its 5,572 entries. The second column, v2, contains the SMS message text itself, also fully populated across all entries. The remaining three columns, labeled as Unnamed: 2, Unnamed: 3, and Unnamed: 4, appear to hold minimal additional information, with only 50, 12, and 6 non-null entries respectively, suggesting they may contain sparse or irrelevant data. This dataset's structure and format offer a foundational setup for building and evaluating machine learning models for SMS spam classification..

**4.2 Data Preprocessing**

Data preprocessing is an essential step in preparing the dataset for training the model. The following steps were taken to clean and balance the data:

1. To effectively handle missing values and remove unwanted data during preprocessing, several key steps are implemented. First, missing values are addressed by examining the dataset for null entries. If columns like "Unnamed: 2", "Unnamed: 3", and "Unnamed: 4" have excessive missing data, they are dropped since they likely do not contribute meaningful information. For any missing rows in the primary columns ("v1" or "v2"), they are removed to maintain the dataset’s quality.
2. Next, unwanted data in the text is cleaned. HTML tags are removed using regular expressions (re.sub(cleanr, '', sentence)), ensuring that any embedded HTML is eliminated. URLs are stripped from the messages with re.sub(r'http\S+', '', cleantext) to prevent irrelevant content from affecting the analysis. Numbers are also removed through re.sub('[0-9]+', '', rem\_url), as they are often not useful for text classification tasks. Special characters and punctuation are cleaned using a tokenizer and regex methods.
3. The text is then normalized by converting all characters to lowercase, ensuring consistency across the dataset. Tokenization splits the text into words, ignoring punctuation, while stopwords (common words like "the", "is") are filtered out to focus on meaningful content. Finally, stemming and lemmatization are applied to reduce words to their base forms, enhancing model performance by ensuring similar words are treated as the same..

**Data Splitting**: The pre-processed dataset was split into training and testing sets using an 80-20 ratio to evaluate the model on unseen data.

**Feature Engineering**

To process the email text for classification, the data were transformed from raw email text data into a format that the model could understand. TF-IDF (Term Frequency-Inverse Document Frequency) is a popular text representation technique that captures the importance of words in documents, helping the classifier focus on significant terms while ignoring common, less informative ones.

**Term Frequency (TF)**: Measures how often a word appears in a document.

A higher TF indicates that the word is more significant within the specific document.

**Inverse Document Frequency (IDF):** Measures how important a word is across all documents in the corpus.

**Combined TF-IDF score:**

Words that are more indicative of phishing behaviour (e.g., "account suspended," "click here") receive higher weights in phishing emails, enabling the model to make an informed decision. The scores can be used with many types of machine learning models, from simple linear classifiers to more complex ensemble methods. In this project, it is used along with Random Forest Classifier.

**Model Selection Strategy:**

1. **Primary Model: Random Forest Classifier**
   * Given its superior accuracy and flexibility, the Random Forest Classifier will be the main model used for SMS spam classification. It is robust, scalable, and handles complex feature interactions well.
2. **Secondary Model: Multinomial Naive Bayes**
   * This model will be retained as a secondary option, especially for fast, efficient classification when needed, although it may not outperform Random Forest in terms of accuracy.
3. **Evaluation Metrics**:
   * **Accuracy**: The primary metric to evaluate the performance of the models.
   * **Precision, Recall, F1-Score**: Since spam detection is a class imbalance problem, these metrics will help evaluate how well the model detects spam messages without incorrectly labeling legitimate ones as spam (false positives).
   * **Confusion Matrix**: This will help visualize the true positives, true negatives, false positives, and false negatives, providing a more comprehensive understanding of model performance.

By prioritizing Random Forest for its accuracy and versatility, and using Multinomial Naive Bayes for efficiency, you can optimize your model selection process for SMS spam detection.

1. **Results**
   1. **Model Performance**

A screenshot of a computer

Description automatically generated

Fig 1. Model accuracy score

The model performed well across both classes with balanced precision, recall, and F1-scores of 0.99. This indicates that it has a good balance between detecting true positives and minimizing false positives. With an accuracy of 98%, the model is quite reliable for identifying phishing emails as well as safe emails.

* 1. **Confusion Matrix**

**A screenshot of a computer

Description automatically generated**

Fig 2. Confusion Matrix for the Model which describes the TP, TN, FP, FN distribution.

**True Positives (TP):** The bottom-right cell (110) represents the number of times class ‘1’ was correctly predicted.

**True Negatives (TN)**: The top-left cell (982) represents the number of times class '0' was correctly predicted.

**False Positives (FP):** The top-right cell (0) represents the number of times class '1' was incorrectly predicted when it was class '0’.

**False Negatives (FN):** The bottom-left cell (23) represents the number of times class '0' was incorrectly predicted when it was class '1’.

**Retrieving Emails and Parsing Content**

The system fetches a specified number of emails from the inbox using the Gmail API. The query can be customized based on specific date ranges or keywords (for example, "after:2024/01/01"). This allows the system to target relevant emails that are likely to be encountered in real-world phishing scenarios. Each email retrieved is then processed in detail. The system extracts the subject and body of the email from the payload section of the Gmail message object and get its ready for verification.

**SMS Preprocessing**

Text preprocessing is a crucial step in natural language processing (NLP) for tasks like SMS spam detection. Here are the key steps:

1. **Lowercasing**: Convert all text to lowercase to maintain uniformity.
2. **Remove Special Characters & Punctuation**: Eliminate unnecessary symbols or punctuation.
3. **Remove Numbers**: Numbers are often irrelevant in text classification tasks.
4. **Remove URLs**: Remove web links to avoid irrelevant data.
5. **Remove HTML Tags**: Strip any HTML tags from the text.
6. **Tokenization**: Split the text into individual words (tokens).
7. **Remove Stopwords**: Eliminate common words that don’t add meaningful value (e.g., "and", "the").
8. **Stemming**: Reduce words to their root form (e.g., "running" -> "run").
9. **Lemmatization**: Convert words to their valid base form using linguistic rules (e.g., "better" -> "good").
10. **Vectorization**: Convert text into numerical format using methods like TF-IDF or Bag of Words for model input.

These steps help clean and standardize text, improving model performance for classification tasks

**Verification**

After preprocessing the email text, the processed data is input into the machine learning model. In this research, we used a **Random Forest Classifier**, which is effective in handling complex datasets with many features. The classifier predicts whether an email is "Safe" or "Phishing."

The model outputs a prediction label of 1 for phishing and 0 for safe emails. This prediction is then presented to the user, along with a recommendation. If the email is classified as phishing, the chatbot advises the user to report it as suspicious, ensuring users are alerted to potential threats. The system will also show the summary of the inbox check.

A screenshot of a computer

Description automatically generated

Fig 4. Sample output for spam

A screenshot of a computer

Description automatically generated

Fig 5. Sample output for Ham

1. **Future Direction**

Future directions for this SMS spam detection project include exploring deep learning models such as RNN, LSTM, or BERT, which could significantly improve context understanding and accuracy. Another key direction is enabling real-time spam detection on mobile devices or messaging platforms, making the system more practical for everyday use. The project could also benefit from enhanced feature engineering by incorporating additional data points such as sender behavior and message metadata. Expanding the model’s capabilities to support multiple languages would help broaden its applicability globally. To stay ahead of spammers' evolving tactics, adversarial learning could be used to make the model more robust. Additionally, integrating a user feedback loop would allow for continuous improvement and refinement of the model. The project could further extend by incorporating spam detection in chatbots and voice assistants, providing protection across multiple communication channels. Finally, expanding the system for cross-platform integration, including emails and social media, would make it a comprehensive solution for spam detection across various platforms.

**7. References**

1. Yu, Bo, and Zong-ben Xu. "A comparative study for content-based dynamic spam classification using four machine learning algorithms." Knowledge-Based Systems 21.4 (2008): 355-362.

<https://www.sciencedirect.com/science/article/pii/S0950705108000026>

1. Guzella, Thiago S., and Walmir M. Caminhas. "A review of machine learning approaches to spam filtering." Expert Systems with Applications 36.7 (2009): 10206-10222..

<https://www.sciencedirect.com/science/article/pii/S095741740900181X>

1. Lee, Sang Min, et al. "Spam detection using feature selection and parameters optimization." 2010 International conference on complex, intelligent and software intensive systems. IEEE, 2010.

<https://ieeexplore.ieee.org/abstract/document/5447486/>

1. Awad, Wael Abou, and S. M. ELseuofi. "Machine learning methods for spam e-mail classification." International Journal of Computer Science & Information Technology (IJCSIT) 3.1 (2011): 173-184.

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e0113ba1c6a737f685f84639645ce9b58b8ca41e>

1. Torabi, Zahra S., Mohammad H. Nadimi-Shahraki, and Akbar Nabiollahi. "Efficient support vector machines for spam detection: a survey." International Journal of Computer Science and Information Security 13.1 (2015): 11.

<https://www.academia.edu/download/37127910/03_Paper_31121412_IJCSIS_Camera_Ready_pp._11-28.pdf>

1. Trivedi, Shrawan Kumar. "A study of machine learning classifiers for spam detection." 2016 4th international symposium on computational and business intelligence (ISCBI). IEEE, 2016. <https://ieeexplore.ieee.org/abstract/document/7743279/>
2. Kumar, Santosh, et al. "A machine learning based web spam filtering approach." 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA). IEEE, 2016. <https://ieeexplore.ieee.org/abstract/document/7474194/>
3. Bhowmick, Alexy, and Shyamanta M. Hazarika. "Machine learning for e-mail spam filtering: review, techniques and trends." arXiv preprint arXiv:1606.01042 (2016).

<https://arxiv.org/abs/1606.01042>

1. Ahsan, MN Istiaq, et al. "An ensemble approach to detect review spam using hybrid machine learning technique." 2016 19th International Conference on Computer and Information Technology (ICCIT). IEEE, 2016. <https://ieeexplore.ieee.org/abstract/document/7860229/>
2. Iqbal, Muhammad, et al. "Study on the effectiveness of spam detection technologies." International Journal of Information Technology and Computer Science (IJITCS) 8.1 (2016): 11-21. <https://www.researchgate.net/profile/Malik-Muneeb-Abid/publication/292190337_Study_on_the_Effectiveness_of_Spam_Detection_Technologies/links/56cf084a08aeb52500c8a262/Study-on-the-Effectiveness-of-Spam-Detection-Technologies.pdf>
3. Sajedi, Hedieh, Golazin Zarghami Parast, and Fatemeh Akbari. "Sms spam filtering using machine learning techniques: A survey." Machine Learning Research 1.1 (2016): 1-14. <https://www.academia.edu/download/107709765/10.11648.j.mlr.20160101.11.pdf>
4. Qiu, Tie, et al. "SIGMM: A novel machine learning algorithm for spammer identification in industrial mobile cloud computing." IEEE Transactions on Industrial Informatics 15.4 (2018): 2349-2359. <https://ieeexplore.ieee.org/abstract/document/8274937>
5. Gupta, Mehul, et al. "A comparative study of spam SMS detection using machine learning classifiers." 2018 eleventh international conference on contemporary computing (IC3). IEEE, 2018. <https://ieeexplore.ieee.org/abstract/document/8530469/>
6. Ali, Syed Sarmad, and Junaid Maqsood. ". Net library for SMS spam detection using machine learning: A cross platform solution." 2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST). IEEE, 2018. <https://ieeexplore.ieee.org/abstract/document/8312266/>
7. Mallampati, Deepika, K. Chandra Shekar, and K. Ravikanth. "Supervised machine learning classifier for Email spam filtering." Innovations in Computer Science and Engineering: Proceedings of the Sixth ICICSE 2018. Springer Singapore, 2019.

<https://link.springer.com/chapter/10.1007/978-981-13-7082-3_41>

1. Abayomi-Alli, Olusola, et al. "A review of soft techniques for SMS spam classification: Methods, approaches and applications." Engineering Applications of Artificial Intelligence 86 (2019): 197-212. <https://www.sciencedirect.com/science/article/abs/pii/S0952197619302155>
2. Świtalski, Piotr, and Mateusz Kopówka. "Machine learning methods in e-mail spam classification." Studia Informatica: systems and information technology 1 (2019).

<https://bibliotekanauki.pl/articles/1819261.pdf>

1. Agarwal, R., et al. "A novel approach for spam detection using natural language processing with AMALS models." IEEE Access (2024).

<https://ieeexplore.ieee.org/abstract/document/10504798>

1. Barushka, Aliaksandr. "Machine Learning Techniques in Spam Filtering." (2020). <https://dk.upce.cz/handle/10195/75560>
2. Julis, M. Rubin, and S. Alagesan. "Spam detection in SMS using machine learning through textmining." Internatıonal Journal Of Scıentıfıc & Technology Research 9.02 (2020).

<https://www.academia.edu/download/77424969/Spam-Detection-In-Sms-Using-Machine-Learning-Through-Text-Mining.pdf>

1. Sun, Nan, et al. "Near real-time twitter spam detection with machine learning techniques." International Journal of Computers and Applications 44.4 (2022): 338-348. <https://www.tandfonline.com/doi/abs/10.1080/1206212X.2020.1751387>
2. Makkar, Aaisha, et al. "An efficient spam detection technique for IoT devices using machine learning." IEEE Transactions on Industrial Informatics 17.2 (2020): 903-912. <https://ieeexplore.ieee.org/abstract/document/8967128/>
3. Kumar, Nikhil, and Sanket Sonowal. "Email spam detection using machine learning algorithms." 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2020. <https://ieeexplore.ieee.org/abstract/document/9183098/>