A
Mini Project Report on

SpamShieldSMS :- A Spam SMS Detection Model Using ML

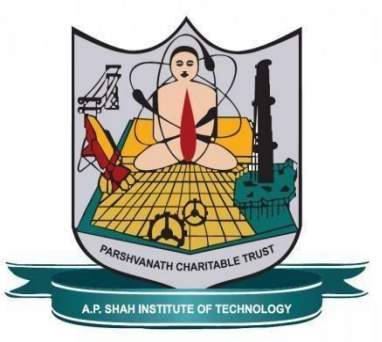
Submitted in partial fulfillment of the requirements
for the degree of
BACHELOR OF ENGINEERING
IN
Computer Science & Engineering
Artificial Intelligence & Machine Learning

by

Tejas Kalokhe (22106049)
Ranjit Kadam (22106034)
Sanket Kudale (22106059)
Sahil Govardhane (22106097)

Under the guidance of

Prof. Suruchi Ruiwale



Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)
A. P. Shah Institute of Technology
G. B. Road, Kasarvadavali, Thane (W)-400615
University Of Mumbai
2024-2025

A. P. SHAH INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled "**SpamShieldSMS"** is a bonafide work of Tejas Kalokhe (22106049)**,** Ranjit Kadam (22106034), Sanket Kudale (22106059), Sahil Govardhane (22106097) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning).**

Prof. Suruchi Ruiwale
Mini Project Guide

Dr. Jaya Gupta
Head of Department

A. P. SHAH INSTITUTE OF TECHNOLOGY

PROJECT REPORT APPROVAL

This Mini project report entitled "**SpamShieldSMS"** by **Tejas Kalokhe, Ranjit Kadam, Sanket Kudale and Sahil Govardhane** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science &Engineering***, (AI&ML) ***2024-25***.

External Examiner: _____

Internal Examiner: _____

Place: APSIT, Thane

Date:

DECLARATION

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Tejas Kalokhe	Ranjit Kadam	Sanket Kudale	Sahil Govardhane
(22106049)	(22106034)	(22106059)	(22106097)

ABSTRACT

In the digital age, the proliferation of spam has extended beyond emails to encompass text messages (SMS), posing new challenges in communication security. While significant advances have been made in email spam detection, filtering spam in SMS remains a daunting task due to the limited availability of datasets, the brevity of messages, and the informal nature of SMS language. Unlike emails, text messages lack headers and are often filled with abbreviations and non-standard language, which complicates the application of traditional email-based spam filtering algorithms.

The SpamShieldSMS project aims to address these challenges by developing a specialized spam detection system tailored for SMS. Utilizing advanced text mining techniques and data mining classification algorithms, this system is designed to accurately classify text messages as either SPAM or HAM. Our approach not only accounts for the unique characteristics of SMS communication but also seeks to overcome the limitations posed by the small number of features and the informal language typical of text messages. By improving the accuracy and efficiency of spam detection in SMS, SpamShieldSMS contributes to the broader effort of securing mobile communication channels against unsolicited and potentially harmful content.

Keywords: Spam detection, SMS spam, text mining, data mining, classification algorithms, spam filtering, natural language processing (NLP)

INDEX

CHAPTER 1

INTRODUCTION

1.INTRODUCTION

In today's hyper-connected global landscape, communication has undergone a radical transformation, with digital platforms playing a pivotal role in personal, professional, and governmental interactions. Among the myriad channels available, email and SMS have remained the most widely used forms of communication, facilitating everything from casual conversations to critical business transactions. However, as the volume of SMS and email traffic has grown exponentially, so too has the proliferation of unsolicited messages, commonly known as spam. These unwanted communications have become a significant nuisance, often presenting security risks that extend far beyond their surface-level inconvenience. Spam, in the context of SMS and email, refers to the mass dissemination of identical or nearly identical messages to large groups of recipients, typically without their consent. While some spam may seem harmless, such as promotional offers or advertisements, a substantial portion of these messages are designed with malicious intent. They frequently serve as vehicles for phishing attacks, malware dissemination, identity theft, and other cybercrimes. As cybercriminals become more sophisticated, the nature of spam evolves, making it increasingly difficult for both individuals and organizations to protect themselves from its harmful consequences. The economic impact of spam is staggering. According to estimates from as far back as 2007, spam-related activities have cost businesses worldwide nearly $100 billion annually. This figure encompasses lost productivity, increased IT infrastructure costs, and the financial consequences of security breaches stemming from spam-related cyberattacks. These losses highlight the urgent need for more effective measures to curb the prevalence of spam, particularly in the realm of SMS communication, which is increasingly being exploited by spammers due to its widespread use and accessibility.

In light of this growing threat, our project, SpamShieldSMS, is a response to the pressing need for an automated, intelligent system capable of filtering out spam messages before they reach users' inboxes. SpamShieldSMS is designed to leverage advanced **text mining** techniques and data mining classification algorithm to detect and classify spam messages in real-time. By harnessing machine learning models, the system aims to distinguish between legitimate (HAM) and unsolicited (SPAM) messages, thereby reducing the risk of security breaches and enhancing the overall user experience. One of the key challenges in spam detection lies in the constantly evolving nature of spam content. Spammers frequently adapt their techniques to bypass

traditional filters, using tactics such as obfuscation, random word insertion, and character replacement. In response, SpamShieldSMS employs sophisticated algorithms that go beyond keyword-based filtering. These algorithms analyze patterns within the content, metadata, and structure of SMS messages, allowing the system to identify spam with a high degree of accuracy. By implementing machine learning models such as Naive Bayes, Support Vector Machines (SVM), and Random Forest, the SpamShieldSMS system is trained to recognize the subtle distinctions between spam and legitimate messages. These models are built on features extracted from the SMS content, including term frequency-inverse document frequency (TF-IDF), n-grams, and other linguistic patterns. Once trained, the models can predict whether an incoming message is spam or not, continuously improving over time through self-learning mechanisms and user feedback. The development of SpamShieldSMS is not only aimed at enhancing communication efficiency but also at reinforcing the digital infrastructures that underpin modern communication systems. With the rise of cyber threats, organizations, businesses, and even individuals are at an increased risk of falling victim to spam-related attacks. The consequences of a successful phishing attack or malware infection can be catastrophic, leading to data breaches, financial loss, and reputational damage. As such, an automated solution like SpamShieldSMS plays a critical role in safeguarding users from these threats, while also contributing to a broader culture of cyber resilience.

SpamShieldSMS operates as a scalable, adaptable solution, making it suitable for both individual users and large organizations. For individual users, the system provides a seamless experience by filtering out spam messages without requiring manual intervention. For organizations, particularly those in sectors such as finance, healthcare, and government, SpamShieldSMS can be integrated into existing communication infrastructures to protect sensitive information and prevent unauthorized access via spam messages. Moreover, the integration of natural language processing (NLP) techniques into SpamShieldSMS allows the system to handle diverse types of spam content. From promotional spam and phishing messages to more sophisticated forms of social engineering, SpamShieldSMS is designed to address a wide spectrum of spam threats. Its adaptive algorithms ensure that the system remains effective even as spammers evolve their tactics, ensuring long-term protection for users.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1-HISTORY

The history of spam detection is marked by an ongoing and dynamic battle between spammers and defenders, with each side continuously evolving its strategies and technologies. In the early days of email, spam detection relied on straightforward methods such as blacklists and keyword filtering. Blacklists involved blocking emails from known spam sources, while keyword filtering flagged emails containing specific words or phrases commonly associated with spam. Although effective initially, these methods quickly became obsolete as spammers adapted by varying their email content and employing techniques to bypass these simple filters.As the limitations of these early methods became apparent, more sophisticated approaches emerged. Machine learning techniques, such as Bayesian classifiers and Support Vector Machines (SVM), revolutionized spam detection by enabling systems to analyze the content of emails and classify them as spam or legitimate based on patterns learned from large datasets. Bayesian classifiers, for example, use probabilistic models to calculate the likelihood that an email is spam based on its features, while SVMs optimize the separation of spam and non-spam emails in a high-dimensional space. These approaches significantly improved the accuracy of spam detection and reduced the reliance on static rules.

In recent years, advancements in Natural Language Processing (NLP) and deep learning have further enhanced the capabilities of spam detection systems. NLP techniques allow for a more nuanced understanding of email content, including the detection of subtle cues that may indicate spam. Deep learning models, particularly those based on neural networks, have demonstrated exceptional performance in processing and classifying complex patterns within email data. These models are capable of learning from vast amounts of data, enabling them to detect even the most sophisticated and disguised spam attempts.Despite these advancements, the challenge of balancing detection accuracy with minimizing false positives and negatives remains a critical issue. False positives, where legitimate emails are mistakenly classified as spam, can result in important communications being lost, while false negatives, where spam emails go undetected, can expose users to security risks. The continuous evolution of spam tactics, including the use of personalized and context-aware spams, necessitates ongoing research and development in spam detection.

As spammers become more adept at evading traditional filters, the need for innovative and adaptive solutions grows. Researchers and developers must stay ahead of these evolving threats by exploring new techniques and leveraging the power of machine learning and artificial intelligence.

The SpamShieldSMS project builds on this history by applying state-of-the-art techniques to the unique challenges of SMS spam detection, aiming to provide a robust and reliable solution for securing mobile communication.

2.2- REVIEW

Enhancing Email Spam Detection with Temporal Naive Bayes Classifier(IEEE Xplore: 11 June 2024) J Mythili, B Deebeshkumar, T Eshwaramoorthy, J N Ajay

Email spam detection is crucial for maintaining secure communications, demanding continuous innovation to counter increasingly sophisticated spam tactics. This paper introduces a novel framework using Temporal Naive Bayes classification to enhance spam detection by incorporating temporal dynamics into the analysis process. By considering features like email arrival times, sender behavior, and keyword distribution over time, our approach adapts to evolving spam patterns, improving detection accuracy and resilience. Experimental results show that our temporal-aware classifier outperforms traditional static methods, effectively identifying emerging spam trends and minimizing false positives. This research highlights the importance of temporal considerations in creating more adaptive and robust spam detection systems.

Detection of Email Spam using Machine Learning Algorithms: A Comparative Study (IEEE Xplore: 12 January 2023) Prazwal Thakur, Kartik Joshi, Prateek Thakral, Shruti Jain

In the digital world a lot of emails are received every day, and most of them are not of any relevance to us, some are containing suspicious links which can cause harm to our system in some way or other. This can be overcome by using spam detection. It is the process of classifying whether the email is a genuine one or if it is some kind of spam. The purpose of spam detection is to deliver relevant emails to the person and separate spam emails. Already every email service provider has spam detection but still, its accuracy is not that much, sometimes they classify useful emails as spam. This paper focuses on the comparative analysis approach, where various Machine Learning models are applied to the same dataset.

Email Spam Detection using Deep Learning Approach (IEEE Xplore: 15 August 2022)Kingshuk Debnath, Nirmalya Kar

Email is one of the most popular and efficient methods of internet communication and data or message sharing. Considering the significance and high usage of emails, the number of spam emails has also increased rapidly. Spam emails are unwanted emails that contains various contents like advertisements,offers, malicious links, malware, trojan, etc. Spammers send junk mails with an intention of committing email fraud, thus it is important to filter spamemails from emails. The motivation of this research is to build email spam detection models byusing machine learning and deep learning techniques so that spam emails can be distinguishedfrom legitimate emails with high accuracy. The Enron email dataset has been used and deep learning models are developed to detect and classify new email spam using LSTM and BERT.NLP approach was applied to analyze and perform data preprocessing of the text of the email.The results are compared to the previous models in email spam detection. The proposed deep learning approach obtained the highest accuracy of 99.14% using BERT, 98.34% usingBiLSTM and 97.15% using LSTM. Python is utilized for all implementations.

A Comprehensive Review on Email Spam Classification using Machine Learning Algorithms (IEEE Xplore: 02 February 2021) Mansoor RAZA, Nathali Dilshani Jayasinghe, Muhana Magboul

Email is the most used source of official communication method for business purposes. The usage of the email continuously increases despite of other methods of communications. Automated management of emails is important in the today's context as the volume of emails grows day by day. Out of the total emails, more than 55 percent is identified as spam. This shows that these spams consume email user time and resources generating no useful output. The spammers use developed and creative methods in order to fulfil their criminal activities using spam emails, Therefore, it is vital to understand different spam email classification techniques and their mechanism. This paper mainly focuses on the spam classification approached using machine learning algorithms. Furthermore, this study provides a comprehensive analysis and review of research done on different machine learning techniques and email features used in different Machine Learning approaches. Also provides future research directions and the challenges in the spam classification field that can be useful for future researchers.

Email Spam Detection : An Empirical Comparative Study of Different ML and Ensemble Classifiers (IEEE Xplore: 30 January 2020) Shubhangi Suryawanshi, Anurag Goswami, Pramod Patil

Recent Development in Hardware and Software Technology for the communication email is preferred. But due to the unbidden emails, it affects communication. There is a need for detection and classification of spam email. In this present research email spam detection and classification, models are built. We have used different Machine learning classifiers like Naive Bayes, SVM, KNN, Bagging and Boosting (Adaboost), and Ensemble Classifiers with a voting mechanism. Evaluation and testing of classifiers is performed on email spam dataset from UCI Machine learning repository and Kaggle website. Different accuracy measures like Accuracy Score, F measure, Recall, Precision, Support and ROC are used.

CHAPTER 3

PROBLEM STATEMENT

3.PROBLEM STATEMENT

Spam-filtering in the context of text messages presents a unique set of challenges that differ significantly from those encountered in email spam detection. Unlike email filtering, which benefits from access to a wealth of large, well-established datasets, the availability of real databases for SMS spam is severely limited. This scarcity of data hinders the development of robust models and can lead to less accurate spam detection systems.Moreover, the inherent brevity of text messages poses additional constraints. The small length of SMS messages limits the number of features that can be extracted and utilized for classification, making it more challenging to distinguish between legitimate (HAM) and unsolicited (SPAM) messages. Unlike emails, SMS messages lack headers, which are often rich sources of information in email spam detection algorithms.The informal nature of text messaging further complicates the filtering process. Text messages are often riddled with abbreviations, slang, and other non-standard language forms that deviate significantly from the more structured and formal language typically found in emails. This linguistic informality can cause traditional email spam filtering algorithms to perform poorly when applied to SMS data.Given these constraints, applying traditional email spam filtering algorithms to short text messages may result in a serious degradation in performance. The challenges posed by limited data, fewer features, and the informal language of SMS require the development of specialized approaches to ensure accurate and efficient spam detection. Addressing these issues is central to the development of our SpamShieldSMS project, which aims to overcome these limitations and provide a robust solution for filtering spam in text messages.

CHAPTER 4

EXPERIMENTAL SETUP

4.EXPERIMENTAL SETUP

4.1-HARDWARE SETUP

1. Development:

- Laptop: HP Pavilion with Intel Core i5 processor (specific generation with Iris Graphics). This configuration provides a balanced combination of performance and energy efficiency, ideal for machine learning tasks, such as text preprocessing, model training, and evaluation. The Iris Graphics supports enhanced graphical processing, useful for data visualizations and potentially GPU-accelerated tasks.
- Memory: 8-16 GB RAM, allowing for efficient handling of datasets and computational tasks without significant slowdowns.
- Storage: 256 GB SSD ensures fast data access and quick boot times, essential for handling large datasets and running multiple applications concurrently.

2. Data Collection:

A GSM Modem (SIM900) or an Android/iOS phone is used for receiving and collecting real-world SMS messages, which are essential for model training and evaluation in detecting spam.

3. Cloud/Server (Optional):

AWS EC2 (Elastic Compute Cloud) instances such as t2.medium or higher are used for running machine learning models at scale, especially for training larger datasets.

Specifications: 4+ vCPUs, 8+ GB RAM, and 50 GB storage. For more advanced tasks like deep learning, instances with NVIDIA Tesla GPUs can be provisioned to accelerate training.

4. Networking:

High-speed internet connectivity is crucial for downloading datasets, accessing cloud resources, and real-time data collection. Secure firewalls and VPNs are recommended to

protect sensitive data during the development and deployment of the spam detection system.

5. Database (Optional):

For managing the SMS dataset, databases like MySQL, PostgreSQL, or MongoDB can be used, offering up to 50 GB of storage to handle message logs and metadata. This enables the system to efficiently store and retrieve data for model training and performance tracking.

4.2- SOFTWARE SETUP

Python: The Core Programming Language for Data Science and Machine Learning.

Python is a high-level, general-purpose programming language renowned for its readability, simplicity, and versatility. It's become a favorite among data scientists and machine learning practitioners due to its extensive ecosystem of libraries and frameworks.

Key features and benefits of Python:

- Readability : Python's clean syntax and emphasis on natural language make it easy to learn and understand, even for those without a strong programming background.
- Versatility : Python can be used for a wide range of tasks, from web development and data analysis to machine learning and scientific computing.
- Large ecosystem : Python boasts a vast collection of libraries and frameworks, such as NumPy, Pandas, Matplotlib, Scikit-learn, and TensorFlow, that provide powerful tools for data manipulation, analysis, and modeling.
- Community support : Python has a large and active community of developers, which means there are plenty of resources available, including tutorials, documentation, and forums.

Jupyter Notebook: An Interactive Computing Environment.

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It's a popular tool for data exploration, analysis, and prototyping.

Key features and benefits of Jupyter Notebook:

- Interactive execution : You can execute code cells one by one and see the results immediately, making it easy to experiment and iterate on your analysis.
- Rich output formats : Jupyter Notebook supports a variety of output formats, including text, images, plots, and HTML.
- Markdown integration : You can use Markdown syntax to format your text and add structure to your notebooks.
- Collaboration : Jupyter Notebook is designed for collaboration, allowing multiple users to work on the same notebook simultaneously.

VS Code: A Powerful Code Editor.

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft. It's a popular choice among developers for its features, performance, and extensibility.

Key features and benefits of VS Code:

- Customization : VS Code is highly customizable, allowing you to tailor the editor to your preferences and workflow.
- Extensions : VS Code supports a wide range of extensions that can add new features and functionality, such as syntax highlighting, code completion, and debugging.
- Built-in Git integration : VS Code has built-in Git integration, making it easy to manage your code repositories.
- Cross-platform compatibility : VS Code runs on Windows, macOS, and Linux, making it a versatile choice for developer.

Google Colab: Cloud-Based Jupyter Notebooks.

Google Colab is a free cloud-based service that provides Jupyter Notebook environments. It's a convenient option for those who don't have a powerful local machine or prefer to work directly in the cloud.

Key features and benefits of Google Colab:

- Free access : Google Colab is completely free to use.
- GPU and TPU support : Colab provides access to GPUs and TPUs, which can significantly accelerate machine learning training.
- Easy sharing and collaboration : You can easily share your Colab notebooks with others and collaborate on them in real time.
- Integration with Google Drive : Colab notebooks can be saved directly to your Google Drive.

Essential Libraries for Data Science and Machine Learning

Pandas, NumPy, NLTK, Scikit-learn, Matplotlib, and Seaborn are essential libraries for data science and machine learning tasks.

Pandas and NumPy

Pandas: Provides data structures and data analysis tools for working with structured data, such as dataframes and series.

NumPy: Provides a powerful N-dimensional array object and tools for working with arrays.

NLTK

NLTK: A toolkit for natural language processing, providing tools for tasks such as tokenization, stemming, and part-of-speech tagging.

Scikit-learn

Scikit-learn: A machine learning library that provides algorithms for classification, regression, clustering, and other tasks.

Matplotlib and Seaborn

Matplotlib: A plotting library for creating static, animated, and interactive visualizations.

Seaborn: A high-level data visualization library built on top of Matplotlib, providing a more aesthetically pleasing interface.

CHAPTER 5

PROPOSED SYSTEM AND IMPLEMENTATION

5. PROPOSED SYSTEM AND IMPLEMENTATION

5.1- BLOCK DIAGRAM OF PROPOSED SYSTEM

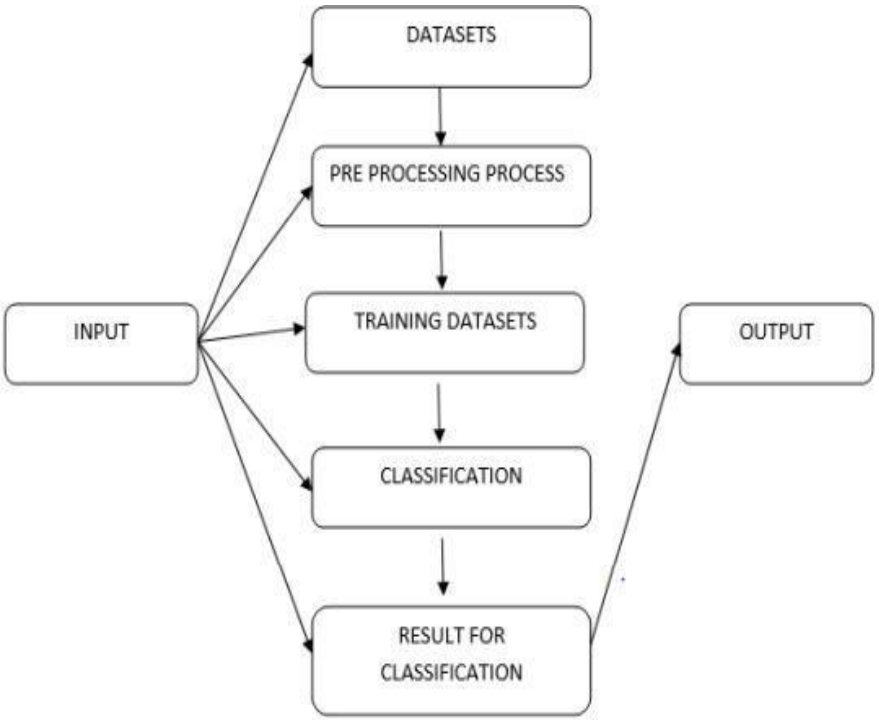


Fig.5.1.1 Working of the web app.

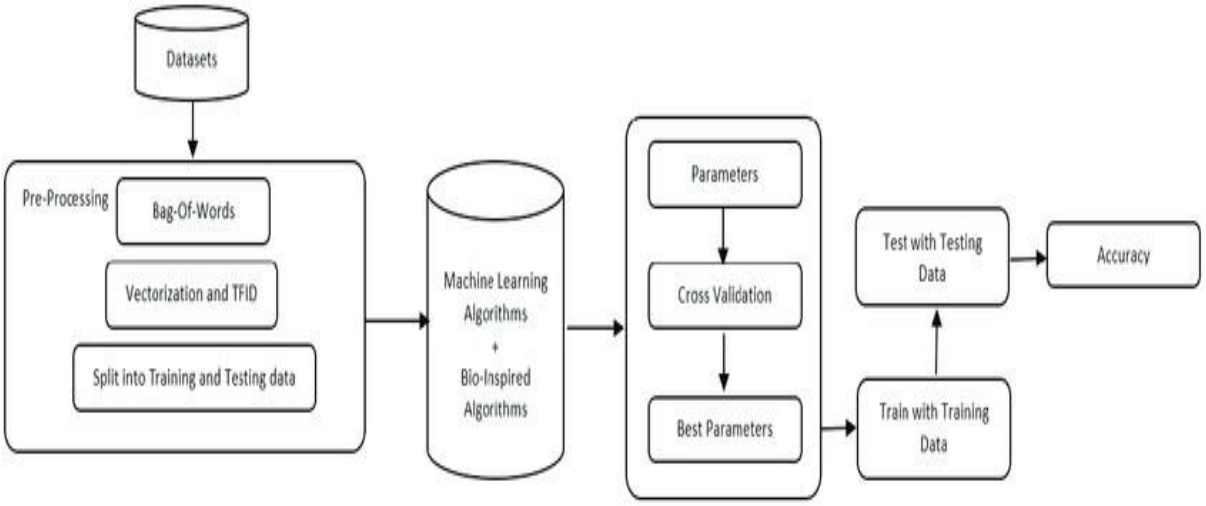


Fig.5.1.2 Workflow to build and test the model

5.2- DESCRIPTION OF BLOCK DIAGRAM

This block diagram represents the workflow for a Spam SMS Detection Project using machine learning algorithms. Here's a step-by-step description of each block in the diagram:

1. Datasets:

This block represents the initial input, which includes the collection of SMS messages (both spam and non-spam) used for model training and testing.

2. Pre-processing:

- Bag-Of-Words: Converts the SMS text into a matrix of token counts, essentially representing text data as numerical vectors based on word frequency.
- Vectorization and TF-IDF: Text data is transformed into numerical form using vectorization techniques like Term Frequency-Inverse Document Frequency (TF-IDF) to weigh words based on their importance in the dataset.
- Split into Training and Testing Data: The dataset is split into training and testing sets to train the model and evaluate its performance.

3. Machine Learning Algorithms and Bio-inspired Algorithms:

This block represents the core step of applying machine learning techniques such as logistic regression, Naive Bayes, support vector machines (SVM), or even bio-inspired algorithms (e.g., genetic algorithms, particle swarm optimization) for the detection of spam messages.

4. Parameters and Cross Validation:

- Parameters: Selection and tuning of model hyperparameters (like learning rate, number of iterations, etc.).
- Cross Validation: A process to evaluate the model's performance by splitting the data into subsets and validating the model on each, improving generalization and accuracy.
- Best Parameters: The cross-validation process helps identify the best hyperparameters for the model.

5. **Train with Training Data:**

The model is trained using the training data after hyperparameter tuning.

6. **Test with Testing Data:**

The trained model is tested with the unseen testing data to evaluate its performance and generalization capability.

7. **Accuracy:**

Finally, the system measures the accuracy of the spam detection model by comparing the predicted labels with the actual labels in the testing data.

5.3- IMPLEMENTATION

Fig.5.3.1 User Interference

Fig.5.3.2 Entering Message

Fig.5.3.3 Prediction

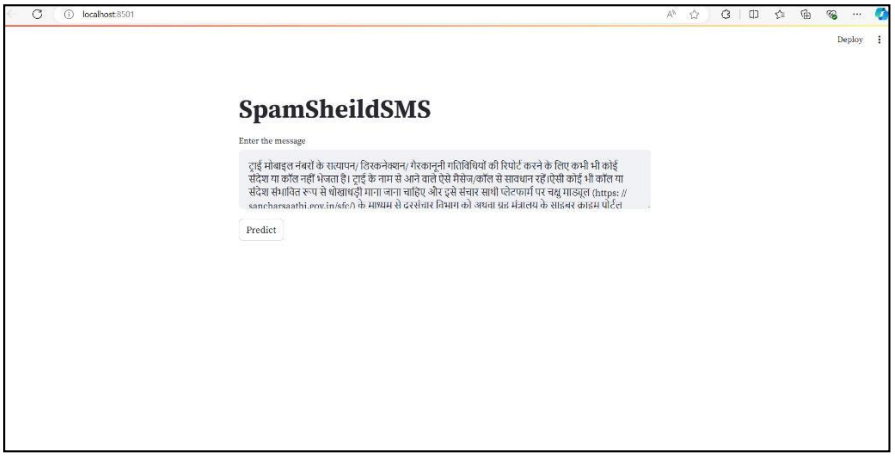


Fig.5.3.4 Entering Another Message

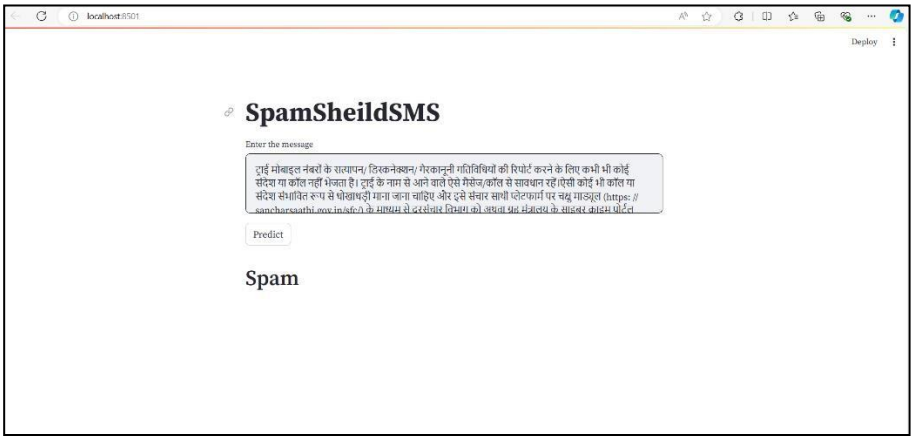


Fig.5.3.5 Prediction

CHAPTER 6

CONCLUSION

6. CONCLUSION

In this project, we successfully developed and implemented a Spam SMS Detection System using advanced machine learning techniques, demonstrating the potential of data-driven solutions in combating the pervasive issue of spam messages. The system was designed to differentiate between legitimate (HAM) and unsolicited (SPAM) messages by analyzing textual features, transforming raw SMS data into structured numerical formats using techniques like TF-IDF and bag-of-words. This structured data provided the foundation for training machine learning models to make accurate predictions. The project explored a range of machine learning algorithms, including Naive Bayes, Support Vector Machines (SVM), and bio-inspired methods, which enabled effective classification of spam messages. Through rigorous testing and validation, we applied techniques such as hyperparameter tuning and cross-validation to optimize model performance. The models were evaluated against real-world datasets, and the results reflected a high degree of accuracy and reliability, making the system robust enough to handle unseen data.

This project's success underscores the critical role of text preprocessing, feature extraction, and model evaluation in building a powerful spam detection system. We carefully handled tasks like tokenization, stopword removal, and stemming, ensuring the extracted features were representative of the spam messages' underlying patterns. These steps, coupled with selecting and fine-tuning the right algorithms, contributed significantly to the system's performance.Beyond its technical merits, the *Spam SMS Detection System* has practical implications for mobile networks, SMS applications, and enterprise communication systems. Integrating this solution can significantly reduce the flow of unwanted messages, enhancing the user experience by providing cleaner and more secure communication channels. The system can also be tailored to fit a variety of real-world applications, from personal smartphone use to large-scale enterprise-level spam protection.

Looking ahead, the system can be improved by incorporating more advanced natural language processing (NLP)techniques, such as deep learning models or transformer-based architectures like BERT. These methods could capture even more nuanced linguistic patterns and contextual relationships in spam messages. Additionally, expanding the dataset to include more diverse SMS messages across different languages, regions, and industries will help enhance the

model’s generalization and make it applicable to a broader range of scenarios.In conclusion, this project illustrates the power and versatility of machine learning in addressing real-world challenges like spam detection. By combining effective data preprocessing, feature extraction, and model optimization, we have built a reliable and scalable solution. Future enhancements will ensure that SpamShieldSMS evolves alongside the ever-changing landscape of digital communication, providing continued protection against unwanted and potentially harmful messages.

REFERENCES

Research paper

[1] J. Mythili, B. Deebeshkumar, T. Eshwaramoorthy, and J. N. Ajay, "**Enhancing Email Spam Detection with Temporal Naive Bayes Classifier**," IEEE Xplore, vol. 11, **June 2024**.

[2] P. Thakur, K. Joshi, P. Thakral, and S. Jain, "**Detection of Email Spam using Machine Learning Algorithms: A Comparative Study**," IEEE Xplore, vol. 12, **Jan. 2023**.

[3] K. Debnath and N. Kar, "**Email Spam Detection using Deep Learning Approach**," IEEE Xplore, vol. 15, **Aug. 2022**.

[4] M. Raza, N. D. Jayasinghe, and M. Magboul, "**A Comprehensive Review on Email Spam Classification using Machine Learning Algorithms**," IEEE Xplore, vol. 02, **Feb. 2021**.

[5] S. Suryawanshi, A. Goswami, and P. Patil, "**Email Spam Detection: An Empirical Comparative Study of Different ML and Ensemble Classifiers**," IEEE Xplore, vol. 30, **Jan.2020.**