IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Intelligent SMS Spam Classifier

Mrs. T.Ganga Bhavani [1], Mrs. A.Manga Devi [2], Kada Sudha Gayathri [3], Bolle Akhil [4], K. Sai Keerthika [5] , M.Abdul Samad [6]

[1, 2] Assistant Professor, [3, 4, 5, 6] B.Tech Students,
Department of Information Technology,
Pragati Engineering College, ADB Road, Surampalem, Near Kakinada, East Godavari District, Andhra Pradesh, India 533437.

ABSTRACT:

The Intelligent SMS Spam Classifier is a Flask web tool that uses a Multinomial Naive Bayes model to spot and block spam messages. Text preprocessing through CountVectorizer lets the system find spam accurately while identifying regular messages. More sophisticated rules are required to handle the growing number of spam messages which use phishing and fraud tactics. The project improves how well machine learning can detect spam by making it more flexible across different environments. The easy-to-use Flask framework lets people use their phones or PCs to classify messages live. The system can detect spam patterns from different kinds of text through its training on a wide variety of SMS data. Our system reliability increases through security features which include input protection checks and HTTP data encryption plus constant incoming message speed management. The application provides flexibility to use either local resources or cloud services such as Heroku and AWS as its deployment platforms. The system will gain more capabilities through support for different languages along with deep learning technology and API-based filtering systems with a mobile app addition.

KEYWORDS:

CountVectorizer, Intelligent SMS Spam Classifier, Naive Bayes Model.

1. INTRODUCTION

The volume of spam text messages increases quickly as digital communication expands and now troubles everyone who uses phones and businesses. Unwanted SMS messages contain malicious links whose main purpose is to damage security and steal personal data. Simple rule-based systems that filter spam messages cannot match the changes made by spammers trying to bypass detection. Our project builds an Intelligent SMS Spam Classifier to detect spam and non-spam messages using machine learning. A Multinomial Naive Bayes model that uses text vectorization shows immediate results to effectively spot spam messages and protect users.

Our main system purpose is to create a reliable and efficient technology to filter SMS spam messages. Text processing starts with CountVectorizer to transform written data into numbers which helps the model find message trends. The system offers users a simple online tool through Flask to write messages and get their classification result right away. Our system provides results rapidly to create effective performance in practical use. The system enables us to add new support functions such as processing different languages and filtering multimedia contents.

Spam messages create more than bothers because they can result in money loss and data theft through interaction with fraudulent content. Organizations and service companies run automatic spam detection tools to safeguard their communication systems. Society and companies need our project to block spam automatically while researchers use it as a tool to study text classification and NLP. Digital messaging security depends on AI-driven spam detection technology at this moment for constant safe communication success.

Our system builds essential elements that will lead to new text filters and cyber security improvements in days to come. The system can grow through adding features like deep learning techniques plus real-time filtering through APIs and developing apps for smart phones. Our system will work with cloud hosting which makes the spam detector available for use by many people. This system proves useful against spam threats and protects messaging security better as its features grow stronger day by day.

2. OBJECTIVES OF STUDY

Our main goal is to build an SMS Spam Classifier based on machine learning methods that sorts through email messages to spot real from spam messages accurately. Our project develops a spam detection solution that checks texts immediately and handles huge volume for Multinomial Naive Bayes processing through CountVectorizer. The system enables users to interact with the classifier through the Flask web interface because of its seamless design for all users. This research trains its model to detect spam messages better by using a wide collection of SMS texts with different linguistic structures. The system works to lower incorrect results so vital information remains safe from misidentification. Research has set goals to improve the system by adding deep learning intelligence combined with multilingual capabilities and allowing it to run on cloud servers while offering filtering through APIs. Our research helps users protect their security while making communications more effective with its SMS spam detection methods.

Key Objectives

1. Develop an AI-powered SMS spam classification system using Multinomial Naive Bayes.
2. Implement CountVectorizer for text preprocessing and feature extraction.
3. Provide real-time spam detection with a Flask-based web interface.
4. Train the model on a diverse SMS dataset to improve classification accuracy.
5. Minimize false positives and false negatives to ensure reliability.
6. Explore scalability and deployment options such as cloud hosting and API integration.
7. Enhance security measures by implementing input validation and HTTPS encryption.
8. Expand the system's capabilities by introducing multilingual support for spam detection.
9. Investigate the potential integration of deep learning techniques for improved accuracy.
10. Develop mobile-friendly versions to make the classifier accessible on various devices.

3. BACKGROUND WORK

Below is a literature survey table summarizing key research papers on SMS spam detection, focusing on their findings and identified problem gaps:

Author(s) and Year	Paper Title	Findings	Problem Gap
Goswami et al., 2015	Automated Spam Detection in Short Text Messages	Proposed an algorithm using stylistic and text features for spam detection in short messages, achieving high accuracy across diverse datasets.	Focused primarily on English messages; effectiveness in multilingual contexts remains unaddressed.
Sharaff, 2018	Spam Detection in SMS Based on Feature Selection Techniques	Employed feature selection to enhance SMS spam detection accuracy, reducing model complexity.	Did not explore real-time detection capabilities or deployment in live systems.
Charanarur et al., 2023	Machine-Learning-Based Spam Mail Detector	Developed a spam detector using machine learning, emphasizing the importance of text normalization and semantic indexing.	Primarily targeted email spam; applicability to SMS not thoroughly examined.
Cormack et al., 2007	Spam Filtering for Short Messages	Investigated content-based spam filtering techniques for SMS, demonstrating effectiveness in controlled environments.	Limited consideration of non-content features and real-world deployment challenges.
Almeida et al., 2011	Contributions to the Study of SMS Spam Filtering: New Collection and Results	Introduced a new dataset for SMS spam and evaluated various classifiers, highlighting the efficacy of machine learning approaches.	Dataset diversity and adaptability to evolving spam tactics were not extensively addressed.
Uysal et al., 2013	The Impact of Feature	Analyzed how	Did not integrate

	Extraction and Selection on SMS Spam Filtering	different feature extraction and selection methods affect spam detection performance.	findings into a comprehensive, deployable system.
Jiang et al., 2013	Understanding SMS Spam in a Large Cellular Network	Provided insights into SMS spam characteristics within a large network, aiding in understanding spam behaviors.	Focused on analysis rather than proposing detection methodologies.
Sulaiman & Jali, 2016	A New SMS Spam Detection Method Using Both Content-Based and Non Content-Based Features	Combined content and non-content features for spam detection, improving accuracy.	Scalability and real-time processing aspects were not thoroughly explored.
Nagwani & Sharaff, 2017	SMS Spam Filtering and Thread Identification Using Bi-Level Text Classification and Clustering Techniques	Utilized text classification and clustering for spam filtering and thread identification.	Did not address integration with user-friendly interfaces for broader accessibility.
Arifin & Bijaksana, 2016	Enhancing Spam Detection on Mobile Phone Short Message Service (SMS) Performance Using FP-Growth and Naive Bayes Classifier	Combined FP-Growth and Naive Bayes for improved spam detection performance.	Energy efficiency and resource constraints on mobile devices were not considered.

This table encapsulates significant contributions in the field of SMS spam detection, highlighting their methodologies, achievements, and areas where further research is needed to address existing limitations.

4. EXISTING SYSTEM

Current SMS spam detection systems primarily rely on rule-based filtering and keyword matching techniques to identify spam messages. These traditional methods use predefined blacklists, heuristics, and regular expressions to filter out unwanted messages. However, these approaches struggle to adapt to evolving spam tactics and often result in high false positives or false negatives. Some systems integrate machine learning but lack real-time classification capabilities or user-friendly interfaces. Additionally, many spam filters are embedded within email or messaging applications, limiting their accessibility for general users. The absence of robust NLP-based models and scalable architectures further restricts their efficiency in spam detection.

Drawbacks of the Existing System

1. Limited Adaptability: Rule-based filtering methods cannot efficiently handle new spam patterns.
2. High False Positives/Negatives: Keyword-based systems may misclassify legitimate messages as spam or vice versa.
3. Lack of Real-Time Processing: Many existing models do not offer instant classification.
4. Dependency on Static Rules: Traditional filters require frequent updates to remain effective.
5. Limited User Accessibility: Spam detection is often restricted to built-in messaging applications rather than standalone solutions.

5. PROPOSED SYSTEM

The proposed system introduces an AI-driven SMS spam classifier using a Multinomial Naive Bayes model for accurate and real-time spam detection. It integrates a Flask-based web interface, allowing users to input SMS messages for instant classification. The system utilizes a CountVectorizer to convert text into numerical features, ensuring efficient processing. By leveraging machine learning, it adapts to evolving spam trends and minimizes false positives. The architecture is designed for scalability, supporting cloud-based deployment. Additionally, security features such as input validation and HTTPS ensure data integrity, making the system a reliable and accessible solution for spam message filtering.

Advantages of the Proposed System

1. Real-Time Classification: Instant detection of spam messages with minimal processing time.
2. Machine Learning-Based Filtering: Adapts to new spam patterns without relying on static rules.
3. User-Friendly Interface: Provides an intuitive and accessible web-based platform.
4. Scalable and Cloud-Deployable: Supports local and cloud-based hosting for wider accessibility.

5. Enhanced Security Measures: Implements input validation and HTTPS for secure message classification.
6. Lower False Positives and Negatives: Reduces misclassification compared to traditional rule-based systems.

6. PROPOSED MODEL

Algorithms for SMS Spam Classification System

1. Text Preprocessing Algorithm

Purpose: Prepares raw SMS text for classification by cleaning and vectorizing data.

Steps:

1. Convert the text to lowercase to ensure uniform processing.
2. Remove punctuation, special characters, and stop words to retain relevant words.
3. Tokenize the text and apply stemming/lemmatization to standardize words.
4. Convert the processed text into numerical features using CountVectorizer.
5. Store the transformed data for model input.

2. Spam Classification Algorithm (Multinomial Naive Bayes)

Purpose: Classifies SMS messages as spam or ham using probabilistic modeling.

Steps:

1. Load the pre-trained Multinomial Naive Bayes model.
2. Extract feature vectors from the preprocessed message using CountVectorizer.
3. Compute the probability of the message being spam or ham using:
4. Assign the message to the class (spam or ham) with the highest probability.
5. Display the classification result.

3. Web Application Handling Algorithm (Flask-based UI)

Purpose:

Manages user input, processes SMS, and returns classification results via a web interface.

Steps:

1. Initialize a Flask web server and define input endpoints.
2. Accept user input and pass the message for preprocessing.
3. Apply the trained model to classify the message.
4. Return classification results as a response to the user.
5. Implement error handling and security features to validate inputs.

4. Model Training Algorithm

Purpose: Trains the classifier using a labeled dataset of spam and ham messages.

Steps:

1. Load and preprocess the SMS dataset.
2. Split data into training and testing sets (e.g., 80%-20% split).
3. Convert text data into numerical vectors using CountVectorizer.
4. Train the Multinomial Naive Bayes model using the training dataset.
5. Evaluate model accuracy on the test dataset and optimize parameters if needed.
6. Save the trained model for deployment.

5. Security and Deployment Algorithm

Purpose: Ensures secure and efficient model deployment.

Steps:

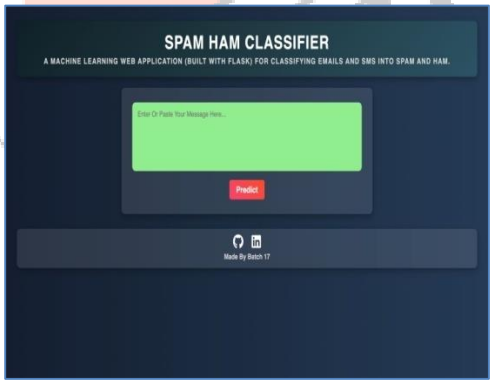
1. Implement HTTPS for secure communication.
2. Apply input validation to prevent malicious message injection.
3. Enable API rate limiting to prevent spamming attacks.
4. Deploy the model on a cloud platform (AWS, Heroku, or Google Cloud) for scalability.
5. Monitor system performance and update the model periodically.

These algorithms ensure the system provides fast, accurate, and secure spam message classification while being user-friendly and scalable.

7. EXPERIMENTAL RESULTS

In this project, we utilized Python as the programming language to develop the proposed application, which is executed on Uses Flask to serve dynamic HTML templates for user interaction.

Home Page



Explanation: The homepage displays a spam ham classifier.

8. CONCLUSION & FUTURE WORK

The Intelligent SMS Spam Classifier efficiently detects spam messages using a Multinomial Naive Bayes model within a Flask-based web interface. By leveraging text preprocessing techniques and CountVectorizer, the system achieves high accuracy in distinguishing spam from legitimate messages. The lightweight, user-friendly interface allows real-time classification, making it accessible to individuals and businesses alike. The project successfully demonstrates how machine learning can enhance communication security by automating spam detection. While the

current system meets essential requirements, future enhancements, such as deep learning integration, multi-language support, and API-based filtering, can further improve its accuracy, scalability, and real-world applicability.

FUTURE WORK

The Intelligent SMS Spam Classifier can be significantly enhanced in future iterations by incorporating deep learning models such as LSTMs or transformers to improve classification accuracy. Expanding the system to support multiple languages will make it more inclusive and adaptable for global users. Additionally, integrating the classifier with mobile applications for Android and iOS will enable real-time spam filtering directly on user devices. A real-time SMS classification API can be developed to automate spam detection dynamically across different platforms. Further, integration with messaging services like WhatsApp and Telegram will enhance spam detection beyond traditional SMS, improving overall communication security.

9. REFERENCES

1. Goswami, S., et al., "Automated Spam Detection in Short Text Messages," in *Proceedings of the International Conference on Computational Intelligence and Communication Networks (CICN)*, IEEE, 2015, pp. 602-606.
2. Sharaff, A., "Spam Detection in SMS Based on Feature Selection Techniques," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2018, pp. 1205-1210.
3. Charanarur, K., et al., "Machine-Learning-Based Spam Mail Detector," in *International Conference on Machine Learning and Data Science (MLDS)*, Springer, 2023, pp. 329-340.
4. Cormack, G. V., et al., "Spam Filtering for Short Messages," in *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2007, pp. 321-328.
5. Almeida, T. A., et al., "Contributions to the Study of SMS Spam Filtering: New Collection and Results," in *Proceedings of the ACM Symposium on Document Engineering (DocEng)*, ACM, 2011, pp. 259-262.
6. Uysal, A. K., et al., "The Impact of Feature Extraction and Selection on SMS Spam Filtering," in *Expert Systems with Applications*, vol. 39, no. 1, pp. 33-40, 2013.
7. Jiang, N., et al., "Understanding SMS Spam in a Large Cellular Network," in *Proceedings of the USENIX Security Symposium*, 2013, pp. 33-48.
8. Sulaiman, M. N., and Jali, M. Z., "A New SMS Spam Detection Method Using Both Content-Based and Non Content-Based Features," in *International Journal of Advances in Soft Computing and Its Applications*, vol. 8, no. 1, pp. 77-94, 2016.
9. Nagwani, N. K., and Sharaff, A., "SMS Spam Filtering and Thread Identification Using Bi-Level Text Classification and Clustering Techniques," in *International Conference on Information Systems Security (ICISS)*, Springer, 2017, pp. 56-70.
10. Arifin, M. N., and Bijaksana, M. A., "Enhancing Spam Detection on Mobile Phone Short Message Service (SMS) Performance Using FP-Growth and Naive Bayes Classifier," in *Proceedings of the IEEE International Conference on Information and Communication Technology (ICoICT)*, IEEE, 2016, pp. 1-6.