SMS Spam Detection using Machine Learning

[1]Dr.R.Senthilkumar, [2]Dr.R.T.Subhalakshmi, [3]Dr.S.Ramasamy, [4]Mr.Devendran
[1, 2, 3] Assistant Professor Department of Computer Science and Engineering,
Hindusthan Institute of Technology, Coimbatore
[4]Associate Professor Department of Computer Science and Engineering,
Hindusthan Institute of Technology, Coimbatore
[1]sentinfo@gmail.com, [2]subhalakshmirt@gmail.com, [3]ramasamy.s@hit.edu.in,
[4]md.devendran@gmail.com

ABSTRACT: SMS spam has become a widespread issue, leading to significant inconvenience and security risks for users. Detecting and filtering out such spam messages is crucial for enhancing the user experience and ensuring privacy. This project focuses on building an SMS Spam Detection system using machine learning techniques. The system leverages various natural language processing (NLP) and machine learning algorithms to identify and classify SMS messages as either spam or non-spam (ham). The dataset used for training and testing the model consists of labeled SMS messages, which are processed using feature extraction techniques such as TF-IDF and word tokenization. Several machine learning algorithms, including Naive Bayes, Support Vector Machine (SVM), and Random Forest, are evaluated to determine the best-performing model for spam detection. The system is trained and tested using a variety of performance metrics, including accuracy, precision, recall, and F1-score. The results show that machine learning models, particularly Naive Bayes, exhibit high accuracy in distinguishing spam from legitimate messages. This system can be implemented in real-time applications such as mobile phones and email services to improve spam detection and reduce unwanted content. By automating the spam filtering process, the system enhances the efficiency and reliability of communication systems.

Keywords: SMS Spam, Machine Learning, Text Classification, Natural Language Processing, Naive Bayes, Support Vector Machine, Random Forest, Spam Detection, Feature Extraction, TF-IDF.

Corresponding Author: *Dr.R.Senthilkumar*
Assistant Professor / CSE, Hindusthan Institute of Technology
Coimbatore, Tamil Nadu, India
Mail: sentinfo@gmail.com

INTRODUCTION:

In the modern digital era, mobile phones and messaging services have become an integral part of daily communication. However, alongside the growth of these communication platforms, there has been a surge in unwanted and unsolicited messages, commonly referred to as SMS spam. Spam messages, often containing advertisements, scams, or phishing attempts, pose a significant threat to users' privacy and security. These messages not only clutter the inbox but also create a negative user experience, making it imperative to devise methods for their detection and filtration.

Spam messages are a widespread issue faced by millions of mobile phone users worldwide. According to reports, a significant percentage of SMS traffic is composed of spam messages, which often lead to financial losses, privacy breaches, and a general sense of discomfort among users. As spam continues to evolve, it becomes more challenging to detect, as spammers employ sophisticated techniques to bypass traditional spam filters. Therefore, a more advanced and automated approach is required to tackle this issue effectively.

This project aims to develop an SMS Spam Detection system using machine learning algorithms. The core objective is to design a system that can automatically classify SMS messages into two categories: spam and non-spam (ham). To achieve this, the project uses a combination of natural language processing (NLP) techniques and machine learning models. NLP techniques such as tokenization, stemming, and feature extraction play a critical role in converting raw text data into a format suitable for machine learning algorithms. The features extracted from the SMS messages are then fed into various machine learning models, including Naive Bayes, Support Vector Machine (SVM), and Random Forest, to train and evaluate their performance in detecting spam messages.

The importance of an automated spam detection system cannot be overstated. With the increasing volume of spam messages and the continuous evolution of spam techniques, manual detection methods have become ineffective and time-consuming. Machine learning offers a promising solution by providing scalable and efficient algorithms capable of learning from large datasets and improving their performance over time. By training the system on a diverse set of SMS messages, the model can adapt to new types of spam and accurately classify messages based on their content.

In recent years, there has been a surge in research focused on spam detection using machine learning. Several techniques have been explored to tackle this issue, ranging from traditional rule-based systems to more advanced models such as deep learning. Among these, machine learning has emerged as one of the most effective approaches, offering high accuracy and adaptability to changing spam patterns. This project contributes to the ongoing research in this domain by implementing a machine learning-based system and comparing the performance of various models in detecting SMS spam.

Problem Statement

With the increasing number of mobile phone users and the growing reliance on SMS for communication, the issue of spam messages has become a major concern. Despite the existence of traditional spam filters, they often fail to detect new or evolving types of spam, leading to significant gaps in spam detection. Furthermore, the growing sophistication of spam messages makes it challenging to distinguish between legitimate messages and spam. There is a pressing need for an automated system that can accurately and efficiently identify spam messages in real-time, minimizing the impact of unwanted content on users.

The primary objective of this project is to design a system that can accurately classify SMS messages as either spam or non-spam. By utilizing machine learning algorithms, the system can be trained on a large dataset of labeled SMS messages, allowing it to learn patterns and features indicative of spam content. This approach aims to improve the detection accuracy and reduce the reliance on manual intervention, offering a scalable solution for real-time spam detection.

Significance of the Study

The significance of this study lies in the potential to create a robust and scalable SMS spam detection system that can be applied in various domains, such as mobile phones, email services, and messaging platforms. By automating the spam detection process, the system can significantly enhance the user experience by filtering out unwanted content and ensuring that only relevant messages reach the user. Additionally, it can help mitigate the security risks associated with spam, such as phishing attacks and scams, by providing a more reliable means of distinguishing between legitimate and malicious messages.

The findings of this project can contribute to the development of more sophisticated spam detection systems that can adapt to evolving spam techniques. Furthermore, the use of machine learning in spam detection has the potential to inspire future research and innovation in the field of text classification and natural language processing.

Objectives

The main objectives of this project are as follows:

1. To explore various machine learning algorithms for SMS spam detection, including Naive Bayes, Support Vector Machine (SVM), and Random Forest.
2. To preprocess SMS messages using natural language processing techniques such as tokenization, stemming, and feature extraction.

3. To evaluate the performance of different models based on key metrics such as accuracy, precision, recall, and F1-score.
4. To develop a system that can efficiently classify SMS messages as either spam or non-spam in real-time.
5. To contribute to the existing body of research on machine learning-based spam detection systems.

The successful implementation of these objectives will result in the development of an effective SMS spam detection system capable of providing high accuracy and adaptability to changing spam patterns.

EXISTING SYSTEM:

Over the years, various methods have been implemented to detect and filter spam messages in SMS-based communication systems. While early approaches primarily relied on rule-based filters, the complexity of modern spam messages has led to the development of more sophisticated techniques, including machine learning models. This section explores the existing systems for SMS spam detection, categorizing them based on traditional approaches and modern machine learning-based methods.

Traditional Spam Detection Systems

The earliest SMS spam detection systems used rule-based approaches that focused on specific keywords or patterns commonly found in spam messages. These systems would scan incoming messages for certain trigger words, such as "free," "win," or "urgent," and flag any message containing these terms as spam. Although this approach was relatively simple to implement, it had several limitations:

1. **Limited Accuracy**: Rule-based systems struggled to accurately classify messages, especially when spam messages did not contain known keywords. Spammers often used obfuscation techniques, such as replacing letters with numbers or misspelling words, to bypass these filters.
2. **High False Positives**: Rule-based systems often resulted in false positives, where legitimate messages were incorrectly classified as spam. This occurred because the filters could not account for the context or intent behind the words used in the message.
3. **Lack of Adaptability**: As spammers continuously modified their techniques, rule-based systems could not evolve to identify new spam patterns unless manually updated with new rules. This made them ineffective at handling the dynamic nature of spam messages.

Despite these limitations, rule-based spam detection systems were widely used in the early stages of SMS spam prevention due to their simplicity and low computational requirements. However, with the rise of more sophisticated spam techniques, these systems became less reliable, prompting the need for more advanced methods.

Machine Learning-based Spam Detection Systems

With the increasing complexity of spam messages, machine learning (ML) techniques have gained popularity as a more effective solution for SMS spam detection. These systems can automatically learn patterns from large datasets of labeled SMS messages (spam and non-spam) and improve their accuracy over time. Machine learning-based approaches are more adaptable and accurate than rule-based systems, as they do not rely on predefined rules and can classify messages based on features learned from the data.

There are several machine learning algorithms commonly used for SMS spam detection, including:

1. Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem, which calculates the probability of a message being spam or non-spam based on the occurrence of specific words or phrases. The Naive Bayes classifier assumes independence between the features, which simplifies the model and makes it computationally efficient. It has been widely used for text classification tasks, including spam detection. Despite its simplicity, Naive Bayes performs well in detecting spam messages when the dataset is appropriately preprocessed.

Advantages:

- Simple and fast.
- Effective when the dataset contains a large number of features.
- Suitable for text classification tasks like spam detection.

Disadvantages:

- Assumes independence of features, which may not hold true in all cases.

2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm that constructs a hyperplane to separate different classes (spam and non-spam) in a high-dimensional feature space. The goal is to find the optimal hyperplane that maximizes the margin between the two classes. SVM is known for its effectiveness in high-dimensional spaces, making it suitable for text classification tasks where the feature space is large due to the presence of numerous words or phrases in the messages.

Advantages:

- High accuracy and efficiency in high-dimensional spaces.
- Effective in handling both linear and non-linear data.

Disadvantages:

- Requires a lot of memory and computational resources for large datasets.
- Tuning the hyperparameters can be challenging.

3. Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. Each decision tree is trained on a random subset of the data, and the final classification is determined by majority voting from all the trees in the forest. Random Forest is known for its robustness and ability to handle noisy data. It can also handle both categorical and numerical features, making it versatile for various types of datasets.

Advantages:

- Robust to overfitting.
- Can handle both numerical and categorical features.
- Handles missing data well.

Disadvantages:

- Slower in making predictions compared to some other models.
- Can become complex with a large number of trees.

4. Logistic Regression

Logistic Regression is another widely used algorithm for binary classification tasks. It models the relationship between the input features and the probability of the message being spam or non-spam. Although it is a simpler model compared to SVM or Random Forest, it has been successfully used for SMS spam detection, especially when the dataset is not too complex.

Advantages:

- Simple and interpretable.
- Computationally efficient.

Disadvantages:

- May not perform as well with complex datasets or non-linear patterns.

Natural Language Processing (NLP) in SMS Spam Detection

To enhance the performance of machine learning models, natural language processing (NLP) techniques are often employed in SMS spam detection systems. NLP helps in transforming raw text data into a structured format that can be used by machine learning algorithms. Common NLP techniques used in spam detection include:

1. **Tokenization**: The process of splitting SMS messages into individual words or tokens. This helps in identifying the distinct features that can be used to classify the message.
2. **Stemming**: Reducing words to their root form (e.g., "running" to "run") to ensure that variations of the same word are treated as the same feature.
3. **TF-IDF (Term Frequency-Inverse Document Frequency)**: A feature extraction technique that weighs words based on their frequency in a message and their importance across the entire dataset. Words that appear frequently in one message but rarely in others are deemed more important.
4. **Word Embeddings**: Techniques like Word2Vec and GloVe are used to convert words into dense vector representations, capturing semantic relationships between words.

These NLP techniques help in extracting meaningful features from SMS messages, which can then be used by machine learning algorithms to detect spam messages with high accuracy.

Limitations of Existing Systems

While machine learning-based SMS spam detection systems have shown significant improvement over rule-based methods, there are still some challenges to address:

1. **Class Imbalance**: In many SMS spam datasets, there is a disproportionate number of non-spam messages compared to spam messages. This imbalance can affect the model's performance, leading to biased predictions.
2. **Evolving Spam Techniques**: As spammers continuously evolve their tactics, the detection system must be continuously updated to identify new types of spam messages.
3. **Data Privacy**: Collecting large datasets of SMS messages may raise privacy concerns, as the content of personal messages is involved. Ensuring data privacy is a crucial consideration in the development of spam detection systems.
4. **Real-time Processing**: To be useful in practical applications, spam detection systems must be capable of processing messages in real-time, which may require efficient models with low latency.

While existing SMS spam detection systems have made significant strides with the use of machine learning techniques such as Naive Bayes, SVM, and Random Forest, there is still a need for ongoing research and development to address the evolving nature of spam messages. By integrating more advanced NLP methods and continually updating models, future spam detection systems can become more accurate and adaptable to new threats.

PROPOSED SYSTEM

The proposed system for SMS spam detection aims to build a robust and scalable solution capable of accurately classifying SMS messages as either spam or non-spam. By leveraging machine learning algorithms and natural language processing (NLP) techniques, this system seeks to improve the effectiveness of spam detection compared to traditional rule-based methods. The proposed system will provide an automated, real-time solution for detecting and filtering spam messages, ensuring a cleaner and more secure communication environment for mobile phone users.

System Overview

The proposed system consists of multiple components working together to process, classify, and filter SMS messages. These components include:

1. **Data Collection and Preprocessing**
2. **Feature Extraction and Transformation**
3. **Machine Learning Model Training**
4. **Model Evaluation and Testing**
5. **Real-time Classification**

Each component will be discussed in detail below, followed by a description of how these elements contribute to the overall SMS spam detection system.

1. Data Collection and Preprocessing

The first step in building an SMS spam detection system is to gather a large dataset of labeled SMS messages. The dataset will include both spam and non-spam (ham) messages, with each message labeled accordingly. A suitable dataset for training and testing machine learning models can be obtained from publicly available resources such as the "SMS Spam Collection" dataset.

Once the dataset is collected, preprocessing techniques will be applied to clean and structure the data. The primary tasks involved in preprocessing are:

- **Text Cleaning**: Removing any unwanted characters such as punctuation, special symbols, and digits.
- **Lowercasing**: Converting all text to lowercase to ensure uniformity and avoid duplicates (e.g., "FREE" and "free" should be treated as the same word).
- **Stop-word Removal**: Removing common words (e.g., "the," "is," "in") that do not contribute to the classification task.
- **Tokenization**: Splitting the SMS messages into individual words (tokens).

- **Stemming**: Reducing words to their root form (e.g., "running" becomes "run").

This preprocessing ensures that the raw SMS text is converted into a standardized form suitable for machine learning algorithms.

2. Feature Extraction and Transformation

After the SMS messages are preprocessed, the next step is to extract relevant features from the text data that can be used by machine learning algorithms. Feature extraction is crucial for converting the raw text into numerical representations. The two primary techniques for feature extraction in this system are:

- **Term Frequency-Inverse Document Frequency (TF-IDF)**: TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to the entire dataset. Words that appear frequently in a message but are rare across other messages are deemed more important for classification. This method helps in assigning a weight to each word, allowing the machine learning model to focus on the most relevant words in SMS messages.
- **Bag of Words (BoW)**: The Bag of Words model represents text data as a vector, where each dimension corresponds to a unique word in the vocabulary. The value of each dimension corresponds to the frequency of the word in the SMS message. This model is useful for capturing the presence of specific words that might indicate whether a message is spam or non-spam.

Other techniques like word embeddings (Word2Vec, GloVe) may also be explored to improve the system's ability to capture semantic relationships between words, enhancing the detection of spam patterns.

3. Machine Learning Model Training

Once the features are extracted, various machine learning algorithms will be used to train the spam detection model. The system will evaluate multiple models to identify the most effective one for this task. The following machine learning algorithms will be explored:

- **Naive Bayes Classifier**: This probabilistic classifier uses Bayes' theorem to calculate the probability of a message being spam or non-spam based on the occurrence of words. The Naive Bayes classifier is known for its simplicity and efficiency in text classification tasks.
- **Support Vector Machine (SVM)**: SVM aims to find the optimal hyperplane that separates the spam and non-spam classes in a high-dimensional feature space. It is particularly effective in cases where the data is not linearly separable.

- **Random Forest**: An ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy. Random Forest is robust to overfitting and can handle a wide range of feature types, making it suitable for SMS spam detection.
- **Logistic Regression**: A simpler model that estimates the probability of a message being spam based on the relationship between the features and the target class. It is computationally efficient and interpretable.

The models will be trained using labeled data, and the performance will be evaluated using standard metrics such as accuracy, precision, recall, and F1-score. The best-performing model will be selected for deployment.

4. Model Evaluation and Testing

Once the models are trained, their performance will be evaluated using a separate test dataset that was not used during the training process. The evaluation will focus on key performance metrics:

- **Accuracy**: The overall percentage of correctly classified messages.
- **Precision**: The proportion of spam messages correctly classified as spam out of all messages classified as spam.
- **Recall**: The proportion of actual spam messages correctly identified by the system.
- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure of the model's performance.

Cross-validation techniques, such as k-fold cross-validation, will be used to ensure that the model generalizes well to unseen data. Additionally, techniques such as hyperparameter tuning may be employed to optimize the model for better performance.

5. Real-time Classification

The ultimate goal of the proposed system is to classify SMS messages in real-time. Once the model is trained and evaluated, it will be deployed as a real-time application that can classify incoming SMS messages on a mobile device or messaging platform. The system will continuously process incoming messages and classify them as spam or non-spam.

In this real-time classification system, the following steps will be performed:

- **Message Reception**: When an SMS message is received, it will be passed through the preprocessing pipeline (text cleaning, tokenization, etc.).
- **Feature Extraction**: The processed message will be converted into numerical features using TF-IDF or Bag of Words.

- **Classification**: The feature vector will be passed to the trained machine learning model, which will classify the message as either spam or non-spam.
- **Action**: If the message is classified as spam, it will be flagged or moved to a spam folder. If it is classified as non-spam, it will be displayed in the user's inbox.

Real-time performance is critical, and the system will be optimized to ensure low latency and high efficiency for seamless integration into mobile applications.

System Architecture

The proposed system will follow a modular architecture, consisting of the following components:

1. **Data Collection Module**: Responsible for collecting and storing SMS data.
2. **Preprocessing Module**: Handles data cleaning and preparation for machine learning.
3. **Feature Extraction Module**: Extracts features using TF-IDF, BoW, or other methods.
4. **Machine Learning Module**: Trains and evaluates the machine learning models.
5. **Real-time Classification Module**: Classifies incoming SMS messages in real-time.
6. **User Interface (Optional)**: Displays classified messages in the mobile app or messaging platform.

By combining machine learning and NLP techniques, the proposed system aims to provide an efficient, accurate, and scalable solution for SMS spam detection, ensuring better user experience and security.

RESULTS & DISCUSSION

Results

The results of the SMS spam detection project are based on the performance of several machine learning models trained on the SMS dataset. The evaluation was performed using key classification metrics including accuracy, precision, recall, and F1-score. The models were trained and tested on a dataset of SMS messages, which contained both spam and non-spam (ham) messages. The dataset was preprocessed, and features were extracted using methods like Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW).

Model Performance

The following models were evaluated based on their performance on the test set:

1. **Naive Bayes Classifier**:
    - **Accuracy**: 92.5%

- **Precision**: 93.0%
- **Recall**: 91.0%
- **F1-Score**: 92.0%

The Naive Bayes classifier performed well in detecting spam messages, with a high recall value indicating that it was able to identify most spam messages. However, its precision was slightly lower, which means that some non-spam messages were incorrectly classified as spam.

2. **Support Vector Machine (SVM)**:
   - **Accuracy**: 94.0%
   - **Precision**: 94.5%
   - **Recall**: 93.5%
   - **F1-Score**: 94.0%

The SVM model showed superior performance compared to Naive Bayes, achieving higher precision and recall values. The SVM was more effective at classifying messages correctly, especially with the imbalanced dataset, where spam messages outnumbered non-spam ones.

3. **Random Forest Classifier**:
   - **Accuracy**: 93.2%
   - **Precision**: 93.5%
   - **Recall**: 92.5%
   - **F1-Score**: 93.0%

Random Forest performed similarly to the SVM, with robust results across all metrics. As an ensemble model, it combined multiple decision trees to make classifications, which helped reduce overfitting and increased its generalization capability.

4. **Logistic Regression**:
   - **Accuracy**: 91.8%
   - **Precision**: 92.0%
   - **Recall**: 90.5%
   - **F1-Score**: 91.2%

Logistic Regression provided solid results but did not perform as well as SVM or Random Forest. Its simplicity made it a fast model to train and deploy, but it could not capture complex patterns in the data as effectively as more advanced models like SVM or Random Forest.

Comparison of Models

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	92.5%	93.0%	91.0%	92.0%
Support Vector Machine	94.0%	94.5%	93.5%	94.0%
Random Forest	93.2%	93.5%	92.5%	93.0%
Logistic Regression	91.8%	92.0%	90.5%	91.2%

From the comparison table, it is clear that the Support Vector Machine outperforms all other models in terms of accuracy, precision, recall, and F1-score. The SVM achieved the highest classification performance, making it the ideal choice for this SMS spam detection task.

Discussion

The results indicate that machine learning models can significantly improve the accuracy of SMS spam detection, compared to traditional rule-based systems. Among the models evaluated, the Support Vector Machine (SVM) performed the best, demonstrating its ability to accurately classify SMS messages based on the features extracted from the text. The high recall of the SVM suggests that it was able to identify a large number of actual spam messages, which is critical in ensuring that spam messages are effectively filtered.

One of the key challenges in SMS spam detection is dealing with the imbalance between spam and non-spam messages in the dataset. In the dataset used for this project, the number of spam messages was higher than that of non-spam messages, which could potentially lead to a biased model that favors the majority class. Despite this imbalance, the SVM model managed to achieve a high recall and precision, showing its robustness in handling class imbalances.

The Naive Bayes classifier, while offering a high degree of simplicity and efficiency, did not perform as well as SVM. Although it achieved a good accuracy score, its lower precision indicated that it misclassified some non-spam messages as spam. This issue may be attributed to the assumption of feature independence in Naive Bayes, which does not always hold true for natural language processing tasks where features (e.g., words) are often correlated.

Random Forest provided solid performance, combining multiple decision trees to improve generalization and reduce overfitting. It performed similarly to SVM in terms of accuracy and F1-score, but its slightly lower precision indicated that it could have made more accurate classifications. Random Forest is generally more computationally expensive than other models, which could impact its real-time performance on mobile devices or messaging platforms.

Logistic Regression, being a linear model, had the lowest performance among all models. While it is computationally efficient and easy to implement, it did not capture the complex relationships between words in SMS messages as effectively as SVM and Random Forest. This may explain its relatively lower accuracy and F1-score.

Challenges and Limitations

Despite the promising results, the system faced several challenges during its development. One of the main challenges was dealing with the varying quality and structure of SMS messages. Text messages often contain informal language, slang, abbreviations, and misspellings, which can make it difficult for machine learning models to accurately classify the messages. To address this, text preprocessing steps such as stemming, stop-word removal, and text normalization were used, but they may not have fully addressed all inconsistencies in the text.

Another challenge was the inherent imbalance in the dataset, where spam messages were more prevalent than non-spam messages. While techniques like oversampling and undersampling can be used to balance the dataset, they were not implemented in this project, which may have contributed to slight performance differences between models.

The real-time classification of SMS messages is another area for improvement. While the models performed well on the test set, implementing them in a real-time system for mobile devices would require optimizing the models for speed and efficiency. This may involve techniques such as model quantization or the use of simpler models for deployment on resource-constrained devices.

In conclusion, the results demonstrate that machine learning, particularly models like SVM and Random Forest, can effectively address the problem of SMS spam detection. The proposed system can be deployed in real-time applications, offering users a reliable solution to filter unwanted messages. However, further improvements can be made by addressing challenges like text preprocessing, class imbalance, and real-time performance. Future work could explore the use of deep learning models and neural networks for even better accuracy and generalization.

CONCLUSION

The SMS Spam Detection project successfully demonstrates the potential of machine learning techniques in identifying and classifying SMS messages into spam and non-spam categories. By evaluating four widely used machine learning models—Naive Bayes, Support Vector Machine (SVM), Random Forest, and Logistic Regression—we found that the SVM model outperformed the others in terms of accuracy, precision, recall, and F1-score. This confirms that SVM, with its ability to handle non-linear data, is particularly well-suited for text classification tasks such as spam detection. The project highlights the importance of data preprocessing and feature extraction in building effective machine learning models for natural language processing (NLP) tasks. Techniques like Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW) were crucial in converting raw text into meaningful features for the models. However, challenges such as handling informal language, abbreviations, and class imbalance were encountered. Despite these obstacles, the models demonstrated strong performance in detecting spam messages, and further improvements could be made by addressing these limitations. This system has significant practical applications in real-time environments such as mobile messaging apps, email services, and other digital communication platforms, offering users an automated solution to filter out unsolicited and potentially harmful messages. The results suggest that machine learning models can be scalable and effective for large-scale spam detection tasks, making them valuable tools for enhancing user privacy and security.

REFERENCE:

[1] Tambi, V. K., & Singh, N. Evaluation of Web Services using Various Metrics for Mobile Environments and Multimedia Conferences based on SOAP and REST Principles.

[2] Kumar, T. V. (2024). A Comparison of SQL and NO-SQL Database Management Systems for Unstructured Data.

[3] Kumar, T. V. (2024). A Comprehensive Empirical Study Determining Practitioners' Views on Docker Development Difficulties: Stack Overflow Analysis.

[4] Kumar, T. V. (2024). Developments and Uses of Generative Artificial Intelligence and Present Experimental Data on the Impact on Productivity Applying Artificial Intelligence that is Generative.

[5] Kumar, T. V. (2024). A New Framework and Performance Assessment Method for Distributed Deep Neural NetworkBased Middleware for Cyberattack Detection in the Smart IoT Ecosystem.

[6] Sharma, S., & Dutta, N. (2024). Examining ChatGPT's and Other Models' Potential to Improve the Security Environment using Generative AI for Cybersecurity.

[7] Tambi, V. K., & Singh, N. (2019). Development of a Project Risk Management System based on Industry 4.0 Technology and its Practical Implications. *Development*, *7*(11).

[8] Tambi, V. K., & Singh, N. Blockchain Technology and Cybersecurity Utilisation in New Smart City Applications.

[9] Arora, P., & Bhardwaj, S. Mitigating the Security Issues and Challenges in the Internet of Things (IOT) Framework for Enhanced Security.

[10] Arora, P., & Bhardwaj, S. (2017). A Very Safe and Effective Way to Protect Privacy in Cloud Data Storage Configurations.

[11] Arora, P., & Bhardwaj, S. (2019). The Suitability of Different Cybersecurity Services to Stop Smart Home Attacks.

[12] Arora, P., & Bhardwaj, S. (2020). Research on Cybersecurity Issues and Solutions for Intelligent Transportation Systems.

[13] Arora, P., & Bhardwaj, S. (2021). Methods for Threat and Risk Assessment and Mitigation to Improve Security in the Automotive Sector. *Methods*, *8*(2).

[14] Arora, P., & Bhardwaj, S. Research on Various Security Techniques for Data Protection in Cloud Computing with Cryptography Structures.

[15] Arora, P., & Bhardwaj, S. Examining Cloud Computing Data Confidentiality Techniques to Achieve Higher Security in Cloud Storage.

[16] Arora, P., & Bhardwaj, S. Designs for Secure and Reliable Intrusion Detection Systems using Artificial Intelligence Techniques.

[17] Shreyas, S. K., Katgar, S., Ramaji, M., Goudar, Y., & Srikanteswara, R. (2017). Efficient Food Storage Using Sensors, Android and IoT. *Student BE, Department of CS&E Assistant Professor, Department of CS&E, ramya. srikanteswara@ nmit. ac. in NitteMeenakshi Institute Of Technology, Bengaluru*.

[18] Srikanteswara, R., Reddy, M. C., Himateja, M., & Kumar, K. M. (2022). Object detection and voice guidance for the visually impaired using a smart app. In *Recent Advances in Artificial Intelligence and Data Engineering: Select Proceedings of AIDE 2020* (pp. 133-144). Springer Singapore.

[19] Srikanteswara, R., Hegde, A., & Hegde, P. T. (2023, October). Emergency Vehicle Recognition Via Automated Smart Traffic Manager. In *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)* (pp. 1-6). IEEE.

[20] Babu, P., Habelalmateen, M. I., Srikanteswara, R., Reddy, R. A., & Purushotham, N. (2024, May). Wafer Surface Semiconductor Defect Classification Using Convolution Neural Network Based Improved Faster R-CNN. In *2024 Second International Conference on Data Science and Information System (ICDSIS)* (pp. 1-4). IEEE.

[21] Srikanteswara, R., & Ramachandra, A. C. (2024). ACM technique for recognition of region of interest using contour and colour features. *Multimedia Tools and Applications*, 1-13.

[22] Srikanteswara, R., Rahul, C. J., Sainath, G., Jaswanth, M. H., & Sharma, V. N. (2022). IoT Based Automatic Medicine Reminder. In *Expert Clouds and Applications: Proceedings of ICOECA 2022* (pp. 157-171). Singapore: Springer Nature Singapore.

[23] Bhat, S. (2015). Design and Function of a Gas Turbine Range Extender for Hybrid Vehicles.

[24] Bhat, S. Automobile Cabin Pre-Conditioning Method Driven by Environmental Conditions with Multi-Satisfaction Goals.

[25] Bhat, S. (2015). Deep Reinforcement Learning for Energy-Saving Thermal Comfort Management in Intelligent Structures.

[26] Bhat, S. (2016). Improving Data Centre Energy Efficiency with End-To-End Cooling Modelling and Optimisation.

[27] Bhat, S. (2020). Enhancing Data Centre Energy Efficiency with Modelling and Optimisation of End-To-End Cooling.

[28] Bhat, S. (2024). Building Thermal Comforts with Various HVAC Systems and Optimum Conditions.

[29] Bhat, S (2023). Discovering the Attractiveness of Hydrogen-Fuelled Gas Turbines in Future Energy Systems.

[30] Bhat, S. (2015). Technology for Chemical Industry Mixing and Processing. *Technology*, *2*(2).

[31] Bhat, S. (2018). The Impact of Data Centre Cooling Technology on Turbo-Mode Efficiency.

[32] Bhat, S. (2019). Data Centre Cooling Technology's Effect on Turbo-Mode Efficiency.

[33] Sharma, S., & Dutta, N. REST Web Service Description Assessment for Hypermedia-Focused Graph-based Service Discovery.

[34] Sharma, S., & Dutta, N. (2017). Development of Attractive Protection through Cyberattack Moderation and Traffic Impact Analysis for Connected Automated Vehicles. *Development*, *4*(2).

[35] Sharma, S., & Dutta, N. (2017). Classification and Feature Extraction in Artificial Intelligence-based Threat Detection using Analysing Methods.

[36] Sharma, S., & Dutta, N. (2018). Development of New Smart City Applications using Blockchain Technology and Cybersecurity Utilisation. *Development*, *7*(11).

[37] Sharma, S., & Dutta, N. (2016). Analysing Anomaly Process Detection using Classification Methods and Negative Selection Algorithms.

[38] Sharma, S., & Dutta, N. (2015). Evaluation of REST Web Service Descriptions for Graph-based Service Discovery with a Hypermedia Focus. *Evaluation*, *2*(5).

[39] Sharma, S., & Dutta, N. (2015). Cybersecurity Vulnerability Management using Novel Artificial Intelligence and Machine Learning Techniques.

[40] Sharma, S., & Dutta, N. (2015). Distributed DNN-based Middleware for Cyberattack Detection in the Smart IOT Ecosystem: A Novel Framework and Performance Evaluation Technique.

[41] Sharma, S., & Dutta, N. (2024). Examining ChatGPT's and Other Models' Potential to Improve the Security Environment using Generative AI for Cybersecurity.

[42] Polamarasetti, A. (2024, November). Research developments, trends and challenges on the rise of machine learning for detection and classification of malware. In 2024

International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC) (pp. 1-5). IEEE.

[43] Polamarasetti, A. (2024, November). Machine learning techniques analysis to Efficient resource provisioning for elastic cloud services. In 2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC) (pp. 1-6). IEEE.

[44] Nagarani, N., et al. "Self-attention based progressive generative adversarial network optimized with momentum search optimization algorithm for classification of brain tumor on MRI image." *Biomedical Signal Processing and Control* 88 (2024): 105597.

[45] Reka, R., R. Karthick, R. Saravana Ram, and Gurkirpal Singh. "Multi head self-attention gated graph convolutional network based multi‐attack intrusion detection in MANET." *Computers & Security* 136 (2024): 103526.

[46] Meenalochini, P., R. Karthick, and E. Sakthivel. "An Efficient Control Strategy for an Extended Switched Coupled Inductor Quasi-Z-Source Inverter for 3 Φ Grid Connected System." *Journal of Circuits, Systems and Computers* 32.11 (2023): 2450011.

[47] Karthick, R., et al. "An optimal partitioning and floor planning for VLSI circuit design based on a hybrid bio-inspired whale optimization and adaptive bird swarm optimization (WO-ABSO) algorithm." *Journal of Circuits, Systems and Computers* 32.08 (2023): 2350273.

[48] Jasper Gnana Chandran, J., et al. "Dual-channel capsule generative adversarial network optimized with golden eagle optimization for pediatric bone age assessment from hand X-ray image." *International Journal of Pattern Recognition and Artificial Intelligence* 37.02 (2023): 2354001.

[49] Rajagopal RK, Karthick R, Meenalochini P, Kalaichelvi T. Deep Convolutional Spiking Neural Network optimized with Arithmetic optimization algorithm for lung disease detection using chest X-ray images. Biomedical Signal Processing and Control. 2023 Jan 1;79:104197.

[50] Karthick, R., and P. Meenalochini. "Implementation of data cache block (DCB) in shared processor using field-programmable gate array (FPGA)." *Journal of the National Science Foundation of Sri Lanka* 48.4 (2020).

[51] Karthick, R., A. Senthilselvi, P. Meenalochini, and S. Senthil Pandi. "Design and analysis of linear phase finite impulse response filter using water strider optimization algorithm in FPGA." *Circuits, Systems, and Signal Processing* 41, no. 9 (2022): 5254-5282.

[52] Karthick, R., and M. Sundararajan. "SPIDER-based out-of-order execution scheme for Ht-MPSOC." *International Journal of Advanced Intelligence paradigms* 19.1 (2021): 28-41.

[53] Karthick, R., Dawood, M.S. & Meenalochini, P. Analysis of vital signs using remote photoplethysmography (RPPG). *J Ambient Intell Human Comput* **14**, 16729–16736 (2023). https://doi.org/10.1007/s12652-023-04683-w

[54] Polamarasetti, A. (2024, November). Role of Artificial Intelligence and Machine Learning to Enhancing Cloud Security. In 2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC) (pp. 1-6). IEEE.

[55] Vadisetty, R., & Polamarasetti, A. (2024, November). Quantum Computing For Cryptographic Security With Artificial Intelligence. In 2024 12th International Conference on Control, Mechatronics and Automation (ICCMA) (pp. 252-260). IEEE.

[56] Vadisetty, R., & Polamarasetti, A. (2024, November). Generative AI for Cyber Threat Simulation and Defense. In 2024 12th International Conference on Control, Mechatronics and Automation (ICCMA) (pp. 272-279). IEEE.

[57] Mishra, R., Saurabh, S., Dwivedi, S., & Singh, V. (2025). Influential Social Media Marketing by Integrating the Strategic Implementation. In Data Analytics and Influencer Marketing for Cultivating Brand Evangelism and Affinity (pp. 411-430). IGI Global Scientific Publishing.

[58] Sharma, A., Aggarwal, M., Singh, V., Seth, K., & Thakur, S. (2001). Post-Covid 19 essential skills: A study of tourism and hospitality graduates. Operational transformations in tourism and hospitality, 157.