JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

SMS SPAM FILTERING USING MACHINE LEARNING

[1]B Sai Deepthi, [2]K Sudheer Kumar, [3]CH B M Swaroop, [4]K Satya Sudheer

[1]Assistant Professor, [2]Student, [3]Student, [4]Student

Information Technology,
GIET Engineering College,Rajahmundry,Andhra pradesh,India

Abstract : **SMS spam has become a significant concern for mobile users, causing frustration and inconvenience. Machine learning has proven to be an effective solution for filtering out spam messages. However, implementing these methods in real-time scenarios comes with unique challenges. A recent study aims to address these challenges by developing a real-time SMS spam filtering system that leverages machine learning. The primary focus of this research is to optimize the system's performance in real-time classification by concentrating on data preparation, feature engineering, algorithm selection, and model deployment. By tailoring these aspects to the requirements of real-time classification, the system can efficiently combat SMS spam while maintaining a high level of accuracy and low latency. Another promising area of investigation is the integration of natural language processing (NLP) techniques to analyze the content of SMS messages more comprehensively. By identifying subtle spam characteristics, such as deceptive language or manipulative tactics, the system can improve its overall accuracy in filtering out spam messages. Expanding the system's applicability to other messaging platforms and languages can also broaden its impact in combating spam across various communication channels. This will not only benefit mobile users but also contribute to a safer and more secure digital environment.**

Keywords: SMS spam filtering, Real-time classification, Machine Learning

1. INTRODUCTION

SMS is still a popular communication method in today's world of continual connectivity. However, the pervasiveness of spam messages jeopardizes SMS's utility. Spam is more than simply an irritation; it may contain harmful links, phishing schemes, or false material, putting consumers at risk. Machine learning algorithms offer a strong alternative for creating intelligent spam filters, but tailoring these systems to the real-time demands of SMS filtering necessitates careful design and execution. In the ever-evolving landscape of communication technologies, Short Message Service (SMS) remains a stalwart, despite the proliferation of instant messaging platforms and social media. Its simplicity, universality across mobile devices, and reliability in areas with limited internet access make it a preferred choice for various personal and professional communications. However, the effectiveness of SMS as a communication method is under threat due to the rampant influx of spam messages. Spam in SMS is not merely an annoyance but a significant concern with far-reaching implications. Beyond cluttering inboxes and disrupting the user experience, spam messages often harbor malicious intent. They may contain harmful links leading to phishing websites, malware downloads, or solicitations for sensitive information under false pretenses. These threats not only compromise individual privacy and security but also undermine trust in SMS as a reliable communication channel. Addressing the issue of SMS spam requires a multifaceted approach that combines technological solutions with user awareness and regulatory measures. Among these solutions, machine learning (ML) algorithms have emerged as a powerful tool for creating intelligent spam filters. By leveraging vast datasets and sophisticated learning models, ML algorithms can analyze message content, sender behavior, and other contextual factors to differentiate between legitimate messages and spam with a high degree of accuracy.

2. Literature Survey

Pavas Navaney; Gaurav Dubey " SMS Spam Filtering Using Supervised Machine Learning Algorithms " , proposed a system in 2018. detection of Spam and Ham messages using various supervised machine learning algorithms like naive Bayes Algorithm, support vector machines algorithm, and the maximum entropy algorithm and compares their performance in filtering the Ham and Spam messages. As people indulge more in Web-based activities, and with rising sharing of private-data by companies, SMS spam is very common. SMS spam filter inherits much functionality from E-mail Spam Filtering. Comparing the performance of various supervised learning algorithms we find the support vector machine algorithm gives us the most accurate result.In the developing period of the Internet, individuals are increasingly in free online services.

pradeep kumar, jyothi " Deep learning to filter SMS Spam " , proposed a system in 2020.The popularity of short message service (SMS) has been growing over the last decade. For businesses, these text messages are more effective than even emails. This is because while 98% of mobile users read their SMS by the end of the day, about 80% of the emails remain unopened. The popularity of SMS has also given rise to SMS Spam, which refers to any irrelevant text messages delivered using mobile networks. They are severely annoying to users. Most existing research that has attempted to filter SMS Spam has relied on manually identified features. Extending the current literature, this paper uses deep learning to classify Spam and Not-Spam text messages. Specifically, Convolutional Neural Network and Long Short-Term Memory models were employed. The proposed models were based on text data only, and self-extracted the feature set. On a benchmark dataset consisting of 747 Spam and 4,827 Not-Spam text messages, a remarkable accuracy of 99.44% was achieved.

Hind Baaqeel; Rachid Zagrouba " Hybrid SMS Spam Filtering System Using Machine Learning Techniques " , proposed a system in 2020.Due to the massive proliferation of Short Message Service (SMS), Spammers got the interest to dig their way into it in the hope to reach more targets. Spam SMS can trick mobile users into giving away their confidential information which can result in severe consequences. The seriousness of this problem has raised the need to develop an accurate Spam filtration solution. Machine learning algorithms have emerged as a great tool to classify data into labels. This description fits our case perfectly as it classifies SMS into two labels: spam or ham. This paper will tackle the SMS spam filtration solutions by introducing a hybrid system using two types of machine learning techniques: supervised & unsupervised machine learning algorithms. The new hybrid system is designed to achieve better spam filtration accuracy and F-measures.

Muhammad Adeel Abid,Dr. Saleem Ullah " Spam SMS filtering based on text features and supervised machine learning techniques " , proposed a system in may 2022.The advancement in technology made a significant mark with time, which affects every field of life like medicine, music, office, traveling, and communication. Telephone lines are used as a communication medium in ancient times. Currently, wireless technology overrides telephone wire technology with much broader features. The advertisement agencies and spammers mostly use SMS as a medium of communication to convey their business brochures to the typical person. Due to this reason, more than 60% of spam SMS are received daily. These spam messages cause users' anger and sometimes scam with innocent users, but it creates large profits for the spammer and advertisement companies. This study proposed an approach for the classification of spam and ham SMS using supervised machine learning techniques. The feature extracting techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and bag-of-words are used to extract features from data.

3. OVERVIEW OF THE SYSTEM

3.1 Existing system

Existing SMS spam filtering systems frequently use a variety of methodologies, such as rule-based filtering, basic machine learning, or a combination of both. Rule-based systems rely on established criteria, such as keywords or blacklisted phone numbers, which makes them easier to overcome and necessitate frequent updates. Machine learning systems use methods such as Naive Bayes or Support Vector Machines, which improve accuracy but may sacrifice real-time efficiency. While these systems have advantages, they frequently struggle to successfully filter SMS within the milliseconds required for real-time settings. Furthermore, continuing spammer changes reduce system efficiency if datasets and models are not regularly updated.Spammers constantly alter their communications, utilizing fresh language, obfuscating tactics, and changing senders to avoid filters. Systems that do not actively update their datasets or retrain models quickly lose effectiveness. Limited Accuracy: Simpler systems, particularly those that rely largely on rules, frequently produce both false positives (genuine messages incorrectly designated as spam) and false negatives (spam messages that sneak through). This lowers the user experience. Real-time Performance Challenges: Machine learning models created for offline analysis may not be fast enough for real-time filtering. Complex feature engineering and computationally costly techniques limit the capacity to classify communications in milliseconds.Traditional SMS spam filtering systems employ various strategies. Some rely on rule-based approaches, where messages are blocked based on predetermined keywords or blacklists. While simple to implement, these systems are easily circumvented by spammers and require constant maintenance. Other systems utilize basic machine learning algorithms like Naive Bayes or Support Vector Machines. These provide improved accuracy but may not

deliver the lightning-fast processing speed needed for seamless real-time filtering. The challenge with existing systems is that they often struggle to effectively classify SMS messages within the time constraints of a real-time environment. Furthermore, as spammers evolve their tactics, these filters can become less effective unless they are continually retrained and updated with fresh data.

3.2 Proposed system

The proposed method detects spam using a Naive Bayes classifier trained on a carefully curated and annotated sample of SMS messages. Text preprocessing includes conventional cleaning and normalization. The bag-of-words concept allows for computationally efficient feature extraction. Naive Bayes is naturally fast, emphasizing real-time appropriateness. A server-side deployment technique offers greater flexibility and capacity for larger datasets. The use of focused feature selection is predicted to improve accuracy. Periodic dataset updates and model retraining are intended to offset the consequences of shifting spam strategies.The proposed method detects spam using a Naive Bayes classifier trained on a carefully curated and annotated sample of SMS messages. Our proposed system relies on a Naive Bayes classifier to identify spam. To prepare for training, a dataset of SMS messages is meticulously assembled and labeled to distinguish spam from legitimate messages. Preprocessing steps involve standard text cleaning and normalization procedures. We employ the bag-of-words model for efficient feature extraction, ensuring computational feasibility. Naive Bayes is chosen for its speed, making it well-suited for real-time classification. To provide more adaptability and accommodate larger datasets, the system adopts a server-side deployment model. Real-time Efficiency: Emphasize the speed and performance enhancements built into your system to identify SMS messages with low latency. This could include algorithm selection, faster feature extraction, and possible coding optimizations. Improved Accuracy: Explain how your chosen features and machine learning model will improve accuracy over simpler filters. For example, perhaps your feature engineering better captures the details that separate spam from regular SMS messaging. Adaptability: Explain how you intend to keep your system successful against evolving spam trends. This could include devising a plan for updating your dataset, retraining your model, or employing strategies that are intrinsically more resistant to novel spam patterns.

3.3 Proposed System Design

In this project work we used four modules and each module has own functions such as:

1. Scikit-Learn
2. NLTK (Natural Language Toolkit)
3. Random Forest
4. Scikit-Plot

3.3.1 Scikit-Learn

Scikit-learn, often abbreviated as sk learn, is a powerful and versatile machine learning library for the Python programming language. It is designed to provide simple and efficient tools for data analysis, machine learning modeling, and predictive analytics. Scikit-learn is built on top of other scientific computing libraries such as NumPy, SciPy, and Matplotlib, leveraging their capabilities to offer a comprehensive suite of machine learning algorithms, data preprocessing techniques, model evaluation tools, and visualization utilities. One of the key strengths of Scikit-learn is its ease of use and accessibility. The library provides a consistent and user-friendly API that makes it straightforward for users, including beginners and experienced data scientists, to implement various machine learning tasks. Scikit-learn's well-documented API and extensive collection of examples and tutorials make it an excellent choice for learning and practicing machine learning concepts. Scikit-learn offers a wide range of machine learning algorithms that cover supervised learning, unsupervised learning, and reinforcement learning tasks. In supervised learning, where the model learns from labeled data, Scikit-learn provides algorithms such as linear regression, logistic regression, support vector machines (SVM), decision trees, random forests, k-nearest neighbors (KNN), and neural networks.

3.3.2 NLTK (Natural Language Toolkit)

NLTK, which stands for Natural Language Toolkit, is a Python library widely used for natural language processing (NLP) tasks. It provides a comprehensive suite of tools, algorithms, and resources for processing and analyzing human language data. NLTK is designed to facilitate various NLP tasks such as tokenization, stemming, lemmatization, part-of-speech tagging, named entity recognition, sentiment analysis, and more. One of the key features of NLTK is its extensive collection of corpora and lexical resources, which include text datasets, word lists, and linguistic resources for different languages. These corpora serve as valuable datasets for training and testing NLP models and algorithms, allowing researchers and practitioners to experiment with various techniques and evaluate their performance on real-world language data. NLTK also provides functionalities for text preprocessing and normalization, making it easier to clean and prepare text data for analysis.This includes tokenization, which involves breaking

text into individual words or tokens; stemming, which reduces words to their base or root form; and lemmatization, which maps words to their canonical forms based on linguistic rules. These preprocessing steps are essential for improving the accuracy and effectiveness of NLP tasks.

3.3.3 Random Forest

Random Forest is a powerful ensemble learning technique widely used in machine learning for both classification and regression tasks. At its core, Random Forest is based on the concept of decision trees. Decision trees are hierarchical structures that make predictions by splitting the data based on features at each node, leading to leaf nodes that represent the final outcomes or predictions. However, decision trees can be prone to overfitting, especially when dealing with noisy or complex datasets. To address this issue, Random Forest employs ensemble learning, which involves creating multiple decision trees during training. Each decision tree in the Random Forest is trained on a random subset of the training data (bootstrap sampling) and a random subset of features (feature bagging). This randomness introduces diversity among the trees, reducing overfitting and improving the model's ability to generalize to unseen data. During the prediction phase, Random Forest combines the individual predictions of all decision trees through a process called voting (for classification tasks) or averaging (for regression tasks). In classification, the class with the most votes across the trees becomes the final prediction. In regression, the average of the predictions from all trees is taken as the final output.

3.3.4 Scikit Plot

Scikit-plot is a Python library built on top of the popular machine learning library Scikit-learn and the visualization library Matplotlib. It is designed to simplify the process of generating visualizations for machine learning tasks, particularly for model evaluation and data exploration. With Scikit-plot, data scientists and machine learning practitioners can create informative and insightful plots with just a few lines of code, enhancing their understanding of model performance, data distributions, and relationships within the dataset. One of the key features of Scikit-plot is its ability to generate various types of plots for model evaluation. This includes classification metrics such as confusion matrices, ROC curves, precision-recall curves, and calibration plots. Confusion matrices are useful for visualizing the performance of binary and multiclass classification models by displaying the true positive, false positive, true negative, and false negative predictions.

4. Architecture

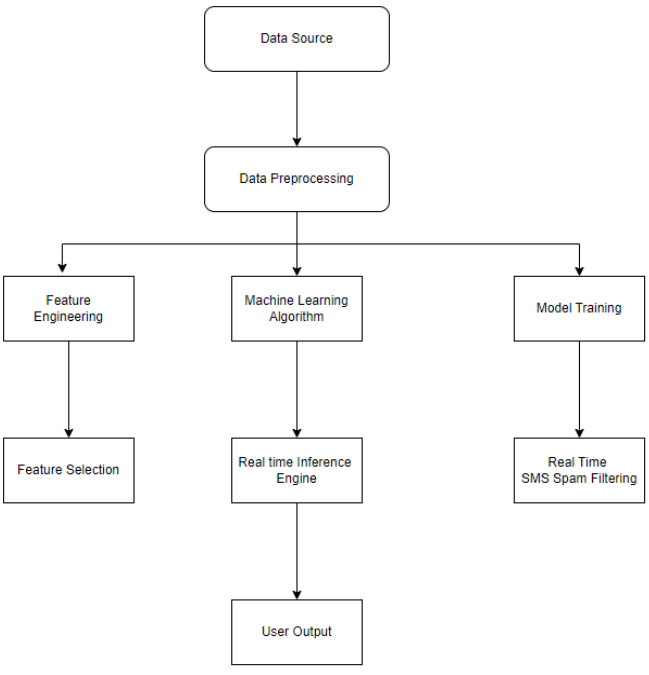

Fig 1: System Architecture

Economical Feasibility:

The real-time SMS spam filtering project is financially feasible since it may drastically lower the expenses that mobile users and service providers bear in relation to spam-related problems. The project's goal is to reduce the negative effects of SMS spam on user experience by implementing an effective machine learning-based solution, which will boost customer happiness and retention. The system's development and deployment costs may be offset in the long run by lower spam-related complaints, lower customer support costs, and the possibility of revenue losses from unhappy customers. Furthermore, by maximizing computational resources and reducing processing time, the system's low latency and high accuracy help to reduce operating costs. Overall, the project's

economic viability is reinforced by its capacity to yield observable advantages in the form of lower expenses, better user experiences, and increased operational effectiveness in the fight against SMS spam.

Technical Feasibility:

Developments in machine learning methods, cloud computing infrastructure, and natural language processing (NLP) approaches lend technical assistance to the real-time SMS spam filtering initiative. Model building and training are made easier by frameworks like TensorFlow and PyTorch, and cloud platforms provide scalable methods for processing massive amounts of SMS data quickly. Developments in NLP improve feature extraction, which raises the bar for spam detection precision. Real-time APIs facilitate smooth communication between mobile networks and the filtering system, guaranteeing prompt categorization of incoming communications. These technological elements work together with effective data pipelines and integration protocols to produce a reliable and strong system that can fight SMS spam in real time with minimal interference to user experience and high accuracy. The integration of real-time APIs and messaging protocols facilitates seamless communication between the spam filtering system and mobile networks, ensuring timely classification of incoming SMS messages.

Social Feasibility:

Because the real-time SMS spam filtering project has the potential to improve user pleasure, trust, and the entire mobile communication experience, it is socially feasible. The project makes mobile consumers' texting experience safer and more fun by efficiently eliminating SMS spam. Users may become more engaged with mobile services as a result of this enhanced experience, and they may become less irritated by unsolicited communications. Additionally, the initiative complies with social norms on data security and privacy because it shields users against fraudulent schemes and phishing scams, which are frequently linked to spam messages. Furthermore, the system's little impact on user experience guarantees that genuine messages are not mistakenly filtered out, preserving smooth communication. The project's overall positive influence on user well-being, faith in mobile services, and overall messaging platform satisfaction clearly demonstrate its social feasibility.

5. RESULTS SCREENSHOTS

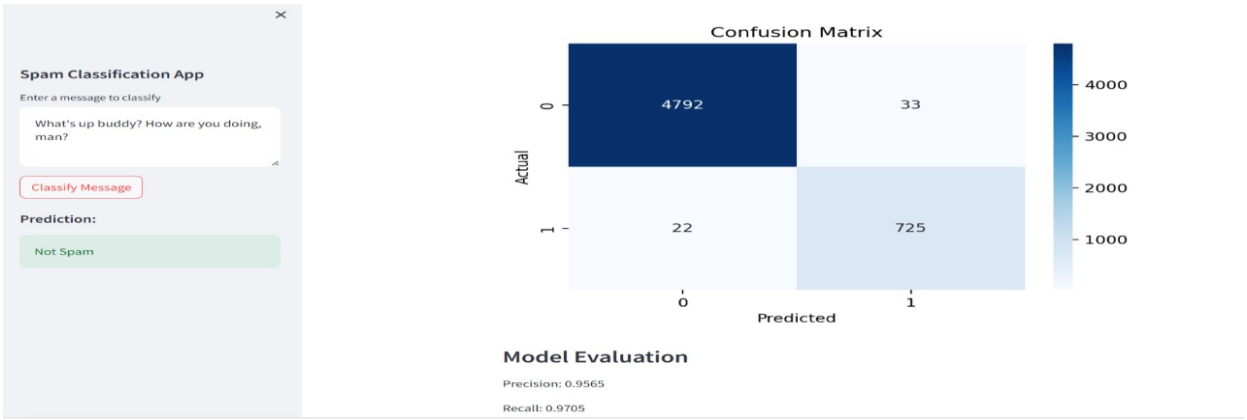

Fig 2: Prediction as not Spam

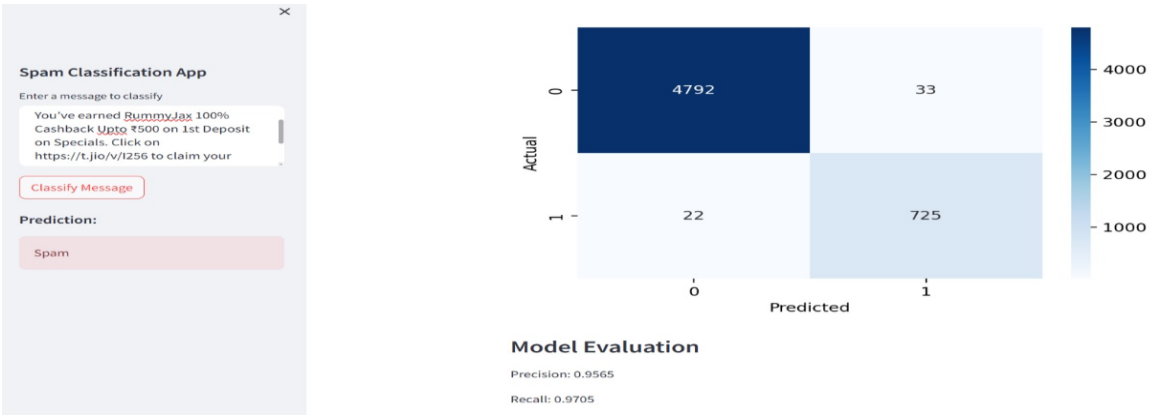

Fig 3: Prediction as Spam

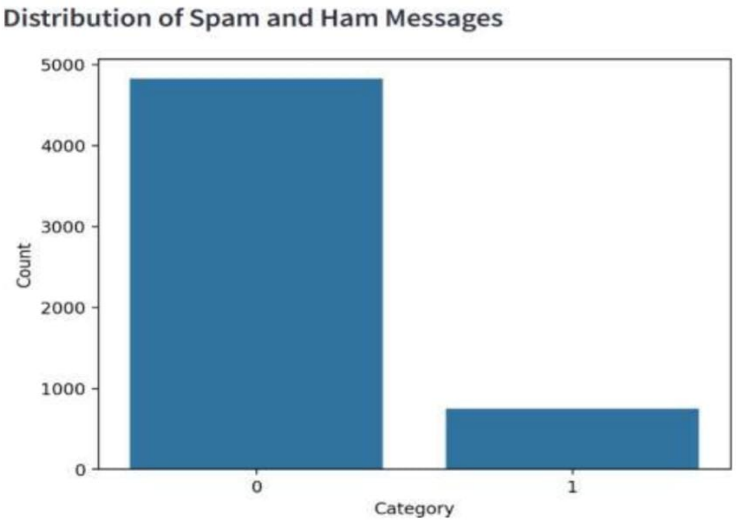

Fig 4: Graphical representation(SpamVs Ham messages)

6. CONCLUSION

The ongoing battle against SMS spam has been a persistent issue for mobile consumers, causing frustration and inconvenience. Traditional methods of spam filtering often fall short in real-time contexts, necessitating the use of advanced machine learning techniques. This project delves into the development of a real-time SMS spam filtering system that leverages machine learning to address this challenge effectively. One of the key areas of focus in this endeavor is data preparation, which involves collecting and preprocessing large volumes of SMS data to extract meaningful features for model training. Feature engineering plays a crucial role in transforming raw data into informative features that capture the essence of spam messages, such as keyword frequency, message length, and sender reputation. These engineered features are then used to train machine learning algorithms capable of distinguishing between legitimate messages and spam in real time.Algorithm selection is another critical aspect considered in this study, as different machine learning algorithms exhibit varying performance characteristics in terms of accuracy, speed, and resource efficiency.

Through thorough experimentation and evaluation, the most suitable algorithms for real-time SMS spam classification are identified and integrated into the filtering system. Model deployment is a crucial stage in the development process, where the trained machine learning models are deployed to production environments to handle incoming SMS messages in real time. The deployment phase involves optimizing the system for low latency and minimal disruption to the user experience. Efficient model deployment strategies, such as containerization or cloud-based solutions, are explored to ensure seamless integration with existing infrastructure. Additionally, mechanisms for continuous model monitoring and updating are implemented to adapt to evolving spam patterns and maintain high filtering accuracy over time. Throughout the project, the emphasis is not only on technical prowess but also on user-

centric design. User feedback mechanisms and usability testing are incorporated to gauge the system's effectiveness from the end-user perspective. This human-centered approach ensures that the SMS spam filtering system not only delivers on its technical promises but also enhances the overall messaging experience for mobile consumers.

7. FUTURE ENHANCEMENT

The future scope of this real-time SMS spam filtering system powered by machine learning lies in enhancing its efficiency and adaptability to tackle emerging challenges. This can be achieved through continuous improvement in data preparation techniques, feature engineering, and algorithm selection. Additionally, exploring advanced machine learning models, such as deep learning architectures, can further boost the system's accuracy and ability to handle complex spam patterns. Incorporating user feedback and preferences to personalize the filtering system will also be crucial for enhancing user experience. Moreover, expanding the system's applicability to other messaging platforms and languages can broaden its impact in combating spam across various communication channels. Another promising avenue for future research is the integration of natural language processing (NLP) techniques to analyze the content of SMS messages more effectively. This can help in identifying subtle spam characteristics, such as the use of deceptive language or manipulative tactics, which may currently evade the system. Lastly, exploring the potential of edge computing and distributed systems for real-time SMS spam filtering can help reduce latency and improve the overall performance of the system, making it more scalable and responsive to the dynamic nature of spam threats.

Integrating natural language processing (NLP) techniques into the system can further bolster its spam detection capabilities by analyzing the semantic meaning and context of SMS messages. By understanding the subtleties of language usage, the system can identify deceptive or manipulative spam content that may evade traditional keyword-based filters. Considering advancements in edge computing and distributed systems, exploring these technologies for real-time SMS spam filtering can lead to reduced latency and enhanced scalability. Edge computing enables processing data closer to the source, minimizing communication overhead and improving response times, while distributed systems can distribute the computational load efficiently across multiple nodes, ensuring robust performance even under high loads.Personalization is another key aspect of future development, where integrating user feedback and preferences can tailor the filtering system to individual users' needs and preferences. This personalized approach not only enhances the user experience but also improves the system's effectiveness by adapting to users' evolving communication patterns and spam tolerance levels. Expanding the system's scope to encompass other messaging platforms beyond SMS, such as instant messaging apps or social media platforms, can significantly broaden its impact in combating spam across diverse communication channels. This expansion may require customizing the system's algorithms and features to suit the characteristics of each platform and language.

8. REFERENCES

[1] Crawford Michael, Taghi M. Khoshgoftaar, Joseph D. Prusa, Aaron N. Richter and Hamzah Al Najada, "Survey of Review spam detection using machine learning techniques", Journal of Big Data, 2015.

[2] R. Deepa Lakshmi and N. Radha, "Spam Classification using supervised learning techniques", A2CWiC10 Proceedings of the 1 st Amrita ACM-W Celebration of Women in Computing in India Article No. 66 .

[3] Anju Radhakrishnan et al., " Classification using Machine learning algorithms", International journal of Engineering and technology(IJE T).

[4] Dea Delvia Arifin, Shaufiah Moch and Arif Bijaksana, "Enhancing Spam Detection on mobile phone short message service(SMS) performance using FP-Growth and naive bayes classifier", Wireless and Mobile (APWiMob) 2016 IEEE Asia Pacific Conference, 2016.

[5] J.M. Gomez Hidalgo, T.A. Almeida and A. Yamakamim, "On the Validity of a New SMS Spam Collection", Proceedings of the 11th IEEE International Conference on Machine Learning and Applications, 2012.

[6] H. Kaur, "Survey on E-mail spam detection using supervised approach with feature selection", International Journal of Engineering Sciences and Research Technology.

[7] Rekha and S. Negi, "A Review on Different Spam Detection Approaches", International Journal of Engineering Trends and Technology (IJETT), vol. 11, no. 6, 2014.

[8] A. S. Aski and N. K. Sourati, "Proposed efficient algorithm to filter spam using machine learning techniques", Pacific Science Review-A Natural Science Engineering-Elsevier, vol. 18, no. 2, pp. 145-149, 2016.

[9] S. P. Teli and S. K. Biradar, "Effective Classification for Spam and Non-spam", International Journal of Advanced Research in Computer and Software Engineering, vol. 4, no. 2014.

[10] Shafil Muhammad Abdulhamid, "A Review on Mobile SMS Spam Filtering Techniques", IEEE Access, 2017.

[11] Almeida. T. A., and J. M. G. Hidalgo. (2018) ―SMS Spam Collection.‖ Available from: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/. [Accessed: 11st April 2018].

[12] Gudkova, D., M. Vergelis, T. Shcherbakova, and N. Demidova. (2017) ―Spam and Phishing in Q3 2017.‖ Securelist - Kaspersky Lab‘s Cyberthreat Research and Reports. Available from: https://securelist.com/spam-and-phishing-in-q3-2017/82901/. [Accessed: 10th April 2018].

[13] Mobile Commons Blog.
https://www.mobilecommons.com/blog/2016/01/how-textmessaging-will-change-for-the-betterin-2016/

[14] http://www.academia.edu/2987380/SMS_Spam_Filtering_Methods_and_Data

[15] Kaggle (2016) Sms spam collection dataset https://www.kaggle.com/uciml/sms-spam-collection-dataset/. Accessed 20 Apr 2021.

[16] Kaggle (2021) Spam mails dataset. https://www.kaggle.com/venky73/spam-mails-dataset. Accessed 24 April 2021.