SMS FRAUD DETECTION USING MACHINE LEARNING

Gopal Pawar, Sanket Nitinkumar Patil, Ankita Verma
CSE (Big Data Analytics), Parul Institute of Engineering and Technology
pawargopal0227@gmail.com, sanketpatil2421@gmail.com, ankitavermasaxena724@gmail.com

Abstract*— Mobile communication has expanded rapidly, making Short Message Service (SMS) a primary tool for information exchange for both individuals and businesses. However, this convenience has a downside: it has opened the door for cybercriminals to launch "smishing" attacks—fraudulent messages designed to trick users into revealing sensitive personal or financial data. Traditional security filters, which rely on rigid rules, often fail to catch these threats as scammers constantly evolve their tactics. To solve this, our research develops an adaptive system using Machine Learning (ML) to automatically detect suspicious SMS content. We employ Natural Language Processing (NLP) to analyze text patterns and convert them into numerical data using techniques like TF-IDF. These features are then processed by supervised learning algorithms to classify messages as either legitimate or fraudulent. The result is a system optimized for high accuracy, speed, and real-time performance, making digital communication safer.

Index Terms*— SMS Fraud, Natural Language Processing, Classification, Feature Extraction, Detection, Machine Learning, NLP, Text Classification, TF-IDF, Spam Filtering, Cybersecurity.

I. INTRODUCTION

Despite the proliferation of modern messaging apps, the Short Message Service (SMS) remains a cornerstone of digital communication due to its global reach and reliability. However, this ubiquity has made it a primary vector for cybercrime, specifically "smishing" attacks, where malicious actors mimic trusted institutions to deceive victims into surrendering sensitive financial or personal information [1].

The challenge lies in defense; traditional filters that rely on static lists of keywords or blocked sender IDs are increasingly obsolete because fraudsters constantly alter their language and tactics to evade detection [2]. To counter this evolving threat, this study turns to Machine Learning (ML) [13]. Unlike rigid rule-based systems, ML algorithms can ingest large volumes of data to recognize the subtle linguistic patterns and hidden cues that signal fraud, allowing the system to adapt autonomously to new threats.

Consequently, this research focuses on developing a lightweight, intelligent fraud detection model that leverages Natural Language Processing (NLP) for text analysis and TF-IDF for feature extraction. To demonstrate the model's practical utility for end-users, the system has been implemented as an accessible web application using the Streamlit framework.

II. RELATED WORK

Research on SMS spam and fraud detection has shifted from simple rule-based filters to advanced machine-learning and deep-learning systems. Early approaches used manually crafted rules to flag suspicious keywords or sender patterns, but these were limited in adapting to new types of spam [2].

Machine-learning techniques significantly improved detection performance. Probabilistic models like Naive Bayes offered strong early baselines for text classification, though they were constrained by feature-independence assumptions [3]. Later, Support Vector Machines (SVM) and Random Forests demonstrated higher precision and robustness, especially on larger and noisier SMS datasets [15].

With the rise of deep learning, models such as CNNs and LSTMs have been widely adopted because they capture contextual and sequential information that traditional models overlook [7]. Hybrid architectures—for example, combining CNN and GRU layers—have achieved exceptional accuracy, with some studies reporting results above 99% for SMS spam detection [4].

Feature extraction methods like TF-IDF and n-gram embeddings continue to play a crucial role in transforming raw text into numerical representations suitable for machine-learning models. Together, these advancements highlight a clear trend toward more adaptive, data-driven, and context-aware SMS fraud detection techniques.

A. Hybrid and Ensemble Methods

Combining multiple models or feature extraction techniques has been proposed to improve detection rates. For instance, hybrid approaches integrating TF-IDF with ensemble classifiers achieve higher accuracy and adaptability to new spam types [5].

B. Literature Review

SMS fraud detection has become a critical area of research due to the increasing use of mobile communications for financial transactions and personal messaging. Traditional rule-based methods were initially employed for detecting spam messages, but these approaches often suffered from low adaptability to new fraud patterns.

Machine Learning (ML) techniques have been widely explored to address these challenges. Studies have shown that supervised learning models, such as Random Forest, Support Vector Machines (SVM), and Logistic Regression, can achieve high accuracy in detecting fraudulent SMS messages. Researchers have also applied ensemble methods to improve detection rates, combining multiple classifiers for better performance.

Furthermore, feature extraction techniques, such as TF-IDF, N-grams, and word embeddings, have been employed

to convert textual SMS data into numerical representations suitable for ML models. In addition, the application of deep learning methods, including LSTM and CNN, has shown promising results in capturing sequential patterns and contextual information in SMS messages.

Despite these advancements, challenges remain in dealing with imbalanced datasets, high false-positive rates, and evolving fraud strategies. This review highlights the importance of applying advanced ML models while continuously adapting to emerging threats.

C. Research Gap & Motivation

Despite extensive research in SMS spam and fraud detection, several gaps remain. Many earlier systems depend on static rules or a single machine-learning model, making them ineffective against new or evolving fraud patterns.

Existing studies often prioritize accuracy on fixed datasets but rarely focus on real-time performance or the ability to generalize to unseen messages. Another major limitation is the lack of support for multilingual and region-specific SMS data, even though real-world fraud messages frequently mix languages and cultural expressions.

Model interpretability is also largely ignored, reducing user trust and transparency in automated decisions. To address these gaps, this research proposes a lightweight yet adaptive framework that integrates multiple learning algorithms with rich text-representation techniques such as TF-IDF and n-grams. By using ensemble strategies, the model aims to improve accuracy, adaptability, and explainability while remaining efficient for real-time SMS fraud detection.

III. METHODOLOGY

Our research strategy relies on a structured pipeline designed to transform raw SMS data into actionable insights. We broke this process down into six core stages, ranging from the initial data gathering to the final performance assessment.

A. Data Collection

We utilized a publicly accessible dataset that had already been tagged with "spam" (fraud) or "ham" (legitimate) labels. This collection was chosen because it includes thousands of messages featuring a wide variety of linguistic styles and fraud tactics, ensuring our model learns to handle real-world diversity.

B. Data Preprocessing

Raw text is naturally messy, so we applied several cleaning techniques before feeding it to the algorithms:

Sanitization: We removed distracting "noise" from the data, such as punctuation marks, numbers, and special characters.

Normalization: We converted all text to lowercase. This ensures the system recognizes that "Urgent" and "urgent" are the same word.

Tokenization: We broke the messages down into their smallest building blocks, or tokens (individual words).

Refining: We filtered out "stop words"—common fillers like "the," "and," or "is"—which add bulk to the dataset without offering any meaningful clues for fraud detection.

Vectorization Prep: Once cleaned, these tokens were readied to be converted into numerical vectors.

C. Feature Extraction

Since machine learning models process numbers rather than language, we employed specific engineering techniques to translate the text:

Bag-of-Words (BoW): This method simply counts how often specific words appear in a message.

TF-IDF: We used Term Frequency–Inverse Document Frequency to assign a "weight" to words. This helps the model prioritize unique, distinctive words over common ones that appear everywhere [13].

N-grams: To capture context, we looked at sequences of adjacent words (bi-grams and tri-grams) rather than just analyzing words in isolation.

D. Machine-Learning Models

We evaluated multiple algorithms to identify the most reliable classifier:

Naïve Bayes: We selected this probabilistic model because it is famously efficient and simple when handling text classification.

Support Vector Machine (SVM): This model works by finding an optimal "hyperplane" or boundary that clearly separates spam from legitimate messages.

Random Forest: By utilizing an ensemble of many decision trees, this method reduces the risk of errors (overfitting) and improves general accuracy.

Hybrid Ensemble: We also experimented with combining these models to leverage their individual strengths for better overall performance.

E. Training and Validation

To ensure a fair test, we divided our dataset into two parts: 80% was used to train the models, while 20% was reserved for testing. We validated the training using k-fold cross-validation to prove the results were consistent and not just a fluke. We also fine-tuned the hyperparameters (like the number of trees in the Random Forest) to maximize stability.

F. Evaluation Metrics

To get a complete picture of how well the system worked, we measured success using four standard metrics:

Accuracy: The total percentage of correct predictions.

Precision: How reliable the model was when it claimed a message was spam.

Recall: The percentage of actual spam messages the model managed to catch.

F1-Score: A balanced metric that combines precision and recall to account for both false alarms and missed threats.

G. Workflow Diagram

The overall methodology workflow can be summarized as: Data Collection → Data Preprocessing → Feature Extraction → Model Training → Evaluation → Results Analysis

H. Challenges and Limitations

Evolving Nature of Spam: Spammers constantly adapt their strategies using obfuscation, emojis, multilingual text, and adversarial content. Models trained on static datasets may struggle to maintain long-term accuracy.

Imbalanced Datasets: In most corpora, legitimate (ham) messages significantly outnumber spam. This imbalance can bias classifiers towards predicting non-spam, reducing recall for spam detection.

Multilingual and Regional Variations: The majority of existing datasets are in English. Performance degrades for spam written in regional or mixed languages, which is a common scenario in real-world communication.

Computational Constraints: While deep learning approaches (e.g., LSTM, BERT) often outperform classical models, they require significant computational resources, making them unsuitable for lightweight or mobile applications.

Explainability Issues: Many machine learning classifiers, particularly ensemble and deep learning models, act as "black boxes." Lack of interpretability makes it difficult for end-users and organizations to trust the decisions.

Privacy and Ethical Concerns: Training on real-world SMS and email data may raise privacy issues. Ensuring compliance with data protection laws is necessary but often difficult.

Deployment Limitations: Although the current implementation is lightweight and deployable via a web app, scaling it for enterprise-level, real-time filtering with millions of daily messages still poses challenges.

IV. SYSTEM ARCHITECTURE

The System Architecture Diagram illustrates the high-level structure of the proposed spam detection system. It is organized into layers, each representing a different functional role:

Input Layer: Receives SMS or email text directly from the user or a connected message stream.

Interface Layer: Utilizes a Streamlit-based web application to provide real-time user interaction, displaying both prediction results and confidence scores in an intuitive format.

Preprocessing Layer: Performs text normalization through lowercasing, tokenization, stopword elimination, and lemmatization to convert raw text into a standardized form suitable for analysis.

Feature Extraction Layer: Converts processed text into a numerical representation using TF-IDF weighting and n-gram modeling to capture both word frequency and contextual relationships.

Model Layer: Applies a pre-trained Logistic Regression classifier, or optionally an ensemble of models, to determine whether the message is legitimate (ham) or fraudulent (spam).

Output Layer: Delivers the final prediction, along with its probability score, to the user interface for easy interpretation.

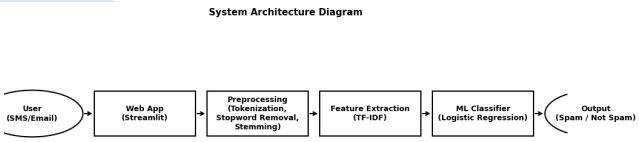

Fig 1. System Architecture

A. Pipeline Flowchart

The Pipeline Flowchart presents the step-by-step workflow of the system. It follows a sequential machine learning pipeline:

Input Message: A raw SMS or email is fed into the pipeline.

Preprocessing: Text is standardized to improve classification accuracy. Steps: Lowercasing, Tokenization, Stopword removal, Stemming/Lemmatization.

Feature Extraction (TF-IDF): Converts preprocessed text into numeric vectors. Ensures the model can interpret text data mathematically.

Model Training & Prediction: Logistic Regression is trained on historical labeled messages. During prediction, it compares the new message's TF-IDF vector to patterns learned from training.

Output (Spam / Ham): The final classification is shown. If the message is spam, the system can flag or block it.

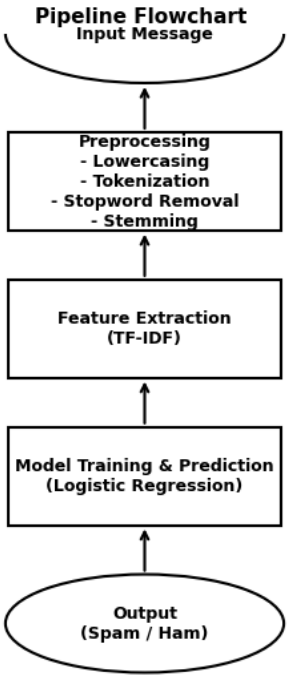

Fig 2. Pipeline Flowchart

V. OPPORTUNITIES AND LIMITATIONS OF AI

A. Opportunities of Artificial Intelligence

Faster Data Processing: AI can handle very large amounts of information in a short time, allowing quicker detection of issues or unusual activity.

Automation of Daily Tasks: Activities such as sorting messages, identifying spam, and monitoring system behavior can be done automatically, reducing manual effort.

More Personalized Interactions: AI can observe user behavior and adapt services to individual needs, creating a more customized experience.

Improved Understanding of Data: By analyzing patterns in communication, AI provides deeper insights and helps in making better security decisions.

B. Limitations of Artificial Intelligence

Possibility of Wrong or Biased Outputs: If AI models are not supervised properly, they may produce incorrect results or show hidden bias.

Less Human Involvement: Heavy use of automation may reduce the role of human judgement in certain tasks.

Strong Dependence on Good Data: AI works well only when it is trained on clean, varied, and updated data. Poor data reduces accuracy.

Privacy and Ethical Issues: AI systems sometimes require access to sensitive communication data, which must be handled according to strict privacy rules.

C. Opportunities of Machine Learning

Ability to Learn Patterns: ML can detect new spam or fraud behaviors by studying previous message data.

Improves Over Time: As fresh data is added, ML models update themselves and adapt to newer types of attacks.

Handles Large Volumes Easily: ML helps classify large numbers of messages quickly, making it useful for high-traffic communication systems.

Predictive Use: It can forecast possible future threats, helping in early preparation and prevention.

D. Limitations of Machine Learning

Requires Big Data and Computing Power: Effective ML systems need many training samples and strong hardware, which may be costly.

Hard to Explain Decisions: Many ML models do not clearly show how they reach a conclusion, which reduces transparency.

Bias from Training Data: If the dataset is unbalanced, the model may unintentionally favor certain outcomes.

Regular Updates Needed: ML models must be retrained and monitored often, which increases operational cost and effort.

VI. IMPLEMENTATION

The system is developed using Python as the primary programming language, supported by libraries such as scikit-learn, pandas, NumPy, and matplotlib for data handling, visualization, and machine-learning operations. The implementation environment includes Jupyter Notebook and VS Code, while deployment can be handled through lightweight web frameworks such as Flask or Django.

Implementation Steps:

Dataset: A publicly available SMS dataset labeled as spam and ham is used for model training and validation to ensure reproducibility.

Preprocessing: Text data undergoes cleaning (removal of symbols, punctuation, and stop words), normalization (lowercasing and tokenization), and vectorization using TF-IDF and n-gram representations.

Model Training: Several supervised algorithms—Naïve Bayes, Support Vector Machine (SVM), Random Forest, and a hybrid ensemble—are trained with optimized hyperparameters to improve detection accuracy.

Evaluation: Each model is assessed using metrics such as accuracy, precision, recall, and F1-score, with confusion matrices visualized through matplotlib to analyze misclassification trends.

Deployment: The final Logistic Regression classifier, selected for its high accuracy and computational efficiency,

is integrated into a Streamlit web interface, enabling real-time message classification and interactive user testing.

This implementation validates the effectiveness of machine-learning techniques for detecting spam and fraudulent messages. It further demonstrates that a lightweight, interpretable, and deployable system can offer practical protection in everyday communication environments.

VII. RESULTS AND DISCUSSION

The experimental results indicate that the combination of TF-IDF feature extraction and Logistic Regression delivers consistently strong performance for SMS fraud detection. The trained model achieves an accuracy of around 95%, maintaining a good balance between precision and recall across both spam and legitimate (ham) message categories.

The confusion matrix analysis further reveals that false positives and false negatives are minimal and symmetrically distributed, suggesting that the model generalizes well to unseen data. While more complex deep-learning architectures such as LSTM or transformer-based networks may offer marginal gains in recall, they typically demand greater computational resources and longer training times.

By contrast, the Logistic Regression approach provides a favorable trade-off between performance and efficiency, making it suitable for lightweight, real-time deployment scenarios. The integration of the trained classifier into a Streamlit-based user interface demonstrates the model's practical viability.

Through this web application, users can enter an SMS message and instantly receive a classification result along with the model's confidence score. This confirms the system's effectiveness not only in detection accuracy but also in accessibility, responsiveness, and end-user usability, thereby supporting its application in everyday communication environments.

VIII. FUTURE WORK

The proposed system is organized into several functional layers that collectively handle data input, processing, and prediction.

Integration of Deep Learning Architectures: Incorporating models such as LSTM, GRU, or transformer-based frameworks like BERT can enhance contextual comprehension and improve classification accuracy.

Support for Multilingual Datasets: Extending the dataset to include regional and multilingual SMS will make the framework more robust for diverse communication environments.

Scalable and Real-Time Deployment: Deploying the model within distributed or cloud-based infrastructures can enable enterprise-level, real-time spam filtering.

Explainable AI (XAI): Introducing model interpretability tools would allow users and administrators to understand the rationale behind each classification, strengthening trust in automated decisions.

Multimodal Spam Detection: Expanding detection capabilities to cover visual, audio, or video content would mitigate emerging threats in multimedia messaging platforms.

User Feedback Mechanism: Implementing an adaptive feedback loop can allow continuous learning from user corrections, helping the model evolve alongside new spam patterns.

Privacy-Preserving Learning: Techniques such as federated learning or differential privacy can be adopted to train models collaboratively without compromising sensitive communication data.

IX. CONCLUSION

This research develops an intelligent framework for detecting fraudulent and spam SMS messages through the integration of machine-learning and Natural Language Processing (NLP) techniques. The system employs TF-IDF-based feature extraction and supervised learning algorithms to differentiate between legitimate and deceptive communication with high reliability.

By focusing on both accuracy and computational efficiency, the proposed approach offers a practical balance that enables real-time use in everyday communication platforms. The implemented web interface demonstrates that complex text-classification models can be transformed into user-friendly applications capable of assisting individuals and organizations in identifying potential threats instantly.

Beyond technical performance, the study emphasizes adaptability, lightweight design, and deployability—key factors for adoption in large-scale digital environments. Overall, this work contributes to enhancing the security and trustworthiness of mobile communication systems while laying a foundation for future research that may incorporate deep-learning architectures, multilingual support, and explainable-AI components.

X. REFERENCES

[1] Patil, A., & Deshmukh, V. (2024). Fraudulent SMS detection using transformer-based models. Int. J. Comput. Inf. Eng., 18(2), 125–133.

[2] Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011). Contributions to the study of SMS spam filtering: New collection and results. ACM Symp. Doc. Eng., 259–262.

[3] Androutsopoulos, I. et al. (2000). Naive Bayesian vs keyword-based anti-spam filtering. ACM SIGIR, 160–167.

[4] Azeez, N. A., & Oloyede, M. O. (2020). Hybrid machine learning model for SMS spam detection. Int. J. Comput. Appl., 177(5), 15–22.

[5] Bhowmick, P. K. et al. (2019). SMS spam detection using ensemble classification. IEEE Access, 7, 16723–16734.

[6] Blanzieri, E. et al. (2008). Survey of email spam filtering techniques. Artif. Intell. Rev., 29(1), 63–92.

[7] Chen, Y. et al. (2021). Improving SMS phishing detection with deep neural networks. Expert Syst. Appl., 184, 115444.

[8] Dada, E. G. et al. (2019). Machine learning for email spam filtering. Heliyon, 5(6), e01802.

[9] Gupta, R. et al. (2023). Comparative analysis of ML algorithms for SMS fraud detection. Int. J. Data Sci. Anal., 17(4), 657–672.

[10] Hassan, A. et al. (2017). Intelligent SMS spam detection using LSTM. Comput. Secur., 73, 385–399.

[11] Hidalgo, J. M. G. et al. (2006). Content-based SMS spam filtering. ACM Symp. Doc. Eng., 107–114.

[12] Jain, S. et al. (2022). Traditional vs deep learning approaches for spam SMS. J. Inf. Secur. Appl., 68, 103182.

[13] Khurana, A. et al. (2020). Hybrid text representation model for SMS spam. Procedia Comput. Sci., 171, 281–290.

[14] Kumar, S. et al. (2018). Real-time SMS spam detection using NLP. IJACSA, 9(11), 136–143.

[15] Singh, N. et al. (2021). Efficient SMS spam detection system. Procedia Comput. Sci., 189, 443–450.

[16] Thakur, P. et al. (2020). SMS phishing detection using ML. Int. J. Adv. Res. Comput. Sci., 11(5), 12–18.

[17] Wang, H. et al. (2020). AI applications for communication security. Info. Tech. Lib., 39(2), 1–15.

[18] Wu, C. et al. (2022). Detecting SMS fraud using BERT. IEEE Trans. Info. Forensics Secur., 17, 1241–1253.

[19] Khairnar, A. A. (2025). Redefining Library Systems with AI and ML. IJISRT.