International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

INVESTIGATING ANTI-FILTERING TACTICS IN SMS SPAM

***Mr. Suyash Agrawal*[1], *K. Sravan Kumar Chary*[2], *G. Rajesh*[3], *M. Sai Kiran*[4]**

[1](Assistant Professor) Computer Science and Engineering (IOT) Guru Nanak Institutions Technical Campus
Telangana, India
[2]Computer Science and Engineering (IOT) Guru Nanak Institutions Technical Campus
Telangana, India
Kchary633@gmail.com
[3]Computer Science and Engineering (IOT) Guru Nanak Institutions Technical Campus
Telangana, India
ryanrajeshgollapalli@gmail.com
[4]Computer Science and Engineering (IOT) Guru Nanak Institutions Technical Campus
Telangana, India
saikiranmanthena01@gmail.com

ABSTRACT –

The prevalence of spam messages via Short Message Service (SMS) remains a significant concern, prompting the need for advanced detection systems capable of countering the sophisticated tactics employed by spammers. This paper investigates anti-filtering techniques that spammers use to bypass traditional SMS spam filters. We introduce a novel, large-scale SMS dataset consisting of over 68,000 messages, with approximately 61% legitimate (ham) messages and 39% spam, which we make publicly available for future research. We conduct an in-depth analysis of the dataset, exploring spam trends and message evolution over time. Additionally, we evaluate and compare multiple machine learning models, including traditional classifiers and deep learning architectures, against these evasive strategies. Our results highlight the limitations of existing models in detecting obfuscated spam content and reveal the vulnerability of commercial anti-spam services to evasion techniques. This study highlights the need to develop strong spam detection systems that can adapt to new spam techniques and constantly changing spam patterns. Our findings contribute valuable insights for the design of next-generation anti-spam solutions.

Keywords - SMS Spam, Evasive Techniques, Machine Learning, Anti-Filtering, Concept Drift, Dataset Analysis, Adversarial Attacks, Spam Detection.

I. Introduction

Short Message Service (SMS) spam continues to pose a persistent challenge in modern digital communications, despite nearly two decades of research focused on its detection and prevention. Recent statistics indicate that SMS spam scams have caused substantial financial losses; for example, in the United States alone, losses from SMS scams reached an estimated \$330 million in 2022, more than doubling the figures reported in 2021. Similarly, Australia’s Scam Watch reported that annual losses due to SMS fraud increased from AUD 175 million in 2020 to AUD 323 million in 2021.

SMS spam typically includes unsolicited text messages ranging from advertisements and marketing to fraudulent schemes designed to mislead or defraud users. The dynamic and evolving nature of spammers’ tactics presents multiple challenges for spam detection systems. Among these, the scarcity of large, high-quality annotated datasets hinders the development of effective spam detection models. Many existing datasets are either outdated or highly imbalanced, containing only a small fraction of spam messages, which limits their generalizability and increases the risk of model overfitting.

Another significant challenge is the lack of standardized benchmark datasets that enable consistent performance comparisons across different detection techniques. This fragmentation in research methodologies makes it difficult to identify the most effective spam detection approaches. Additionally, spammers frequently exploit various evasion techniques—such as text obfuscation, character swapping, and the use of homographs—to bypass detection systems. These techniques can easily confuse even advanced machine learning models, thereby undermining the reliability of existing solutions.

The dynamic behaviour of spam campaigns, known as concept drift, adds yet another layer of complexity. As spammers continuously evolve their tactics often leveraging new technologies and services for bulk messaging detection systems must be able to adapt accordingly. However, there is a lack of research examining the long-term evolution of spam messages and the impact of concept drift on spam detection.

This study aims to address these challenges by contributing a comprehensive, large-scale dataset that captures both spam and legitimate messages from multiple sources, including recent data. We also perform an extensive evaluation of various machine learning models, assessing their robustness against different evasion techniques and concept drift. Finally, we examine the resilience of real-world SMS anti-spam systems, such as popular messaging apps and third-party spam detection services, to these evasion strategies. Our findings provide insights into the current state of SMS spam detection and highlight areas for future research to improve the robustness and effectiveness of anti-spam systems.

II. Literature Review

1. Existing SMS Spam Datasets

Research into SMS spam detection often relies on publicly available datasets to train and evaluate spam filters. Early contributions include the *SMS Spam Collection Dataset, released in 2012, which comprises 5,574 messages with only 747 labelled as spam. Despite its wide usage, its small size and outdated content limit its relevance in the face of modern spam tactics. Another notable dataset is the **NUS SMS Corpus*, updated in 2015, featuring 67,063 messages but lacking spam labels, rendering it less suitable for direct training of spam detection models.

More recent efforts, such as the *Spam Hunter* framework (2022), gathered 25,889 tweets containing screenshots of SMS spam from Twitter. While innovative in its data gathering, this dataset is fraught with noise, including benign messages mistakenly identified as spam, as well as OCR errors and duplicates, which can degrade the quality of machine learning training if not cleaned thoroughly.

2. Machine Learning Approaches for SMS Spam Detection

Traditional machine learning (ML) models, including Support Vector Machines (SVMs), Random Forests, and Naive Bayes classifiers, have been widely applied to SMS spam detection. For example, Almeida et al. demonstrated that SVMs perform well with basic text features such as word frequency counts. However, these models often rely solely on shallow text features like bag-of-words (Bow) or term frequency-inverse document frequency (TF-IDF) representations, limiting their ability to capture deeper semantic meaning.

In more recent studies, deep learning (DL) models such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have shown promise due to their capacity to learn hierarchical and contextual features from text. Roy et al. compared CNNs and LSTMs and found them to outperform traditional classifiers. Nevertheless, many studies have yet to evaluate state-of-the-art transformer-based models, such as Bert and Roberta, which are known for their strong performance on various natural language processing tasks.

Moreover, one-class classification techniques and positive-unlabelled (PU) learning have been explored to detect spam with limited labelled data, though their use in SMS spam detection is relatively rare. Existing studies often overlook comprehensive evaluations of multiple models under consistent experimental setups, making it difficult to draw direct comparisons or identify the most robust solutions.

3. Evasive Techniques in SMS Spam

Spammers frequently employ a range of evasion tactics to circumvent detection systems. Common techniques include the injection of legitimate (ham) words into spam messages, spacing between characters, label poisoning, and the use of synonyms or paraphrasing. These tactics are designed to confuse both rule-based and machine learning models, allowing spam messages to slip through filters.

Although some studies have explored these techniques in email spam or adversarial machine learning contexts, relatively few have focused on the unique challenges posed by SMS spam. For example, the *Punycode attack*, originally known for phishing URLs, can also be applied to SMS messages to hide malicious links using visually similar characters. Such evasive tactics highlight the need for robust detection models that can handle adversarial manipulation.

4. Evaluating Real-World Anti-Spam Systems

Several studies have attempted to assess the performance of real-world anti-spam solutions embedded within popular messaging applications and third-party APIs. For instance, Narayan et al. investigated Android text messaging apps using the SMS Spam Collection dataset, but many of the evaluated apps are now outdated or unavailable. Likewise, Tang et al. analysed mobile apps and services, focusing on performance against Twitter-collected spam messages, but they did not evaluate the resilience of these systems to modern evasion strategies.

These evaluations often neglect the dynamic nature of spam campaigns and fail to consider the impact of evasion tactics on detection efficacy. As a result, the robustness of real-world anti-spam solutions remains underexplored, especially in scenarios that simulate adversarial attacks.

III. methodology

Manner This section outlines the systematic approach we followed to investigate anti-filtering tactics in SMS spam, focusing on data collection and augmentation, feature extraction, model selection, training, and evaluation. Our methodology is designed to ensure reproducibility, facilitate comparative analysis, and robustly evaluate the resilience of spam detection systems against evasive techniques.

1. Data Collection and Augmentation

1.1 Data Collection

To build a comprehensive dataset that reflects current spam trends, we consolidated SMS data from various publicly available sources, including online repositories, GitHub projects, and academic datasets. Additionally, we collected messages from Twitter posts containing SMS screenshots and from scam observatories such as Scam watch and Action Fraud. We also engaged volunteers, who forwarded spam messages received on their devices, enriching our dataset with real-world examples. This multi-source approach allowed us to capture diverse spam tactics across different time periods and contexts.

1.2 Data Preprocessing

We performed extensive preprocessing to ensure data quality and consistency. First, we applied language detection using Python’s *Lang detects* library to filter out non-English messages. We then used *Google trans* to cross-check and confirm language classification. Screenshots of SMSs were converted to text using *pie tesseract*, ensuring that valuable information from image-based SMS spam was included.

Duplicate messages were removed, and we manually labelled over 60,000 SMS messages using a set of predefined rules to distinguish spam from legitimate messages. These rules, developed collaboratively by our research team, were based on characteristics such as promotional content, suspicious URLs, requests for personal information, and common spam phrases. Discrepancies in labelling were resolved through group consensus to ensure high labelling accuracy.

2. Feature Extraction:

To transform SMS messages into a structured format suitable for machine learning, we extracted both syntactic and semantic features.

2.1 Syntactic Features

We utilized count-based vectorization methods such as Bag-of-Words (Bow) and n-grams (unigrams, bigrams, and trigrams) combined with term frequency-inverse document frequency (TF-IDF) weighting. These features capture word occurrence frequencies and local patterns but are limited in representing contextual meaning.

2.2 Semantic Features

To enhance the capture of meaning and context, we employed various word embedding techniques. We used static embeddings like *Word2Vec, **Glove, and **fast Text, which represent words in a high-dimensional vector space but do not consider the word's context in a sentence. To address this limitation, we also incorporated contextual embeddings using transformer-based models such as **BERT* and *ROBERTA*, which allow words to be represented based on surrounding words, improving semantic understanding.

3. Model Selection and Training

We evaluated a range of machine learning models to ensure a comprehensive comparison across different learning paradigms.

3.1 Traditional Machine Learning Models

We implemented supervised learning models, including Support Vector Machines (SVM), One-Class SVM (OCSVM) for anomaly detection, and Positive-Unlabelled (PU) learning techniques using Random Forest classifiers. These models were selected for their historical relevance in spam detection and their ability to operate under varying data availability conditions.

3.2 Deep Learning Models

To leverage the potential of deep learning, we evaluated models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Bidirectional LSTMs. We also experimented with transformer-based architectures like *BERT* and *ROBERTA* to investigate their capability in capturing complex language patterns in SMS data.

4. Evaluation Metrics

We assessed model performance using standard classification metrics: Precision, Recall, Accuracy, and F1-Score. Given the imbalanced nature of SMS datasets, these metrics provide a balanced view of model effectiveness, especially in detecting spam messages (treated as the positive class).

Additionally, we evaluated model robustness by introducing adversarial examples using different evasion techniques and measuring the subsequent impact on model performance.

5. Adversarial Testing with Evasion Techniques

To simulate real-world spammer tactics, we implemented several evasion strategies:

* Paraphrasing: Rewording messages to maintain meaning while altering structure.
* Character Swapping: Introducing typos or character substitutions to bypass keyword-based filters.
* Homograph Attacks (Punycode): Using visually similar characters to disguise URLs or spam keywords.
* Spacing Manipulation: Inserting spaces between letters to hinder keyword recognition.
* Hybrid Techniques: Combining multiple evasion tactics within a single message.

For each technique, we generated perturbed versions of SMS messages and evaluated their impact on model classification performance. This approach enabled us to test model resilience against real-world adversarial manipulation.

6. Real-World System Evaluation

Finally, we tested widely-used SMS anti-spam applications and third-party services using both original and perturbed messages. We assessed their filtering capabilities and analysed their susceptibility to the evasion strategies we developed. This step provided practical insights into the robustness of existing anti-spam systems in real-world scenarios.

IV. Proposed System

This section details the design and conceptual framework of the proposed system aimed at effectively detecting SMS spam, even when sophisticated anti-filtering tactics are employed by spammers. The system integrates comprehensive data processing, advanced feature engineering, and robust machine learning models, including mechanisms for handling concept drift and adversarial evasion.

1. System Overview

Our proposed system is designed as a modular pipeline comprising four main components: (1) Data Ingestion and Preprocessing, (2) Feature Extraction, (3) Spam Classification Engine, and (4) Adversarial Robustness Module.

2. Data Ingestion and Preprocessing

The first component involves continuous ingestion of SMS data from multiple sources, including real-time feeds, crowdsourced submissions, and public datasets. Incoming messages undergo preprocessing to:
* Remove duplicates.
* Normalize text (e.g., lowercasing, removal of non-alphanumeric characters).
* Convert images (e.g., screenshots of SMS) to text using Optical Character Recognition (OCR).
* Filter for English language messages using language detection libraries.
A rule-based labelling module tags incoming messages based on defined spam criteria (e.g., suspicious URLs, payment requests). Messages labelled as spam or legitimate (ham) are then forwarded to the feature extraction module.

3. Feature Extraction Module

This module transforms raw SMS text into structured representations suitable for machine learning algorithms. It performs both syntactic and semantic feature extraction:
* *Syntactic Features*: Utilizes Bag of Words (BOW), TF-IDF, and n-grams to identify and capture local patterns within the text.
* *Semantic Features*: Utilizes both static (Word2Vec, Glove, fast Text) and contextual embeddings (BERT, ROBERTA) to capture nuanced word meanings and sentence-level semantics.
Feature vectors are standardized and stored in a feature repository to support training and evaluation.

4. Spam Classification Engine

At the core of the system is the Spam Classification Engine, which uses an ensemble of models to enhance detection performance and resilience. This engine includes:
* Supervised Classifiers: SVM, Random Forest, and Neural Networks trained on labelled data.
* Unsupervised Models: One-Class SVM and PU learning to handle cases where labelled spam data is scarce.
* *Deep Learning Models*: CNNs, LSTMs, and transformer-based architectures (e.g., BERT) that automatically learn complex language features from text. An ensemble voting mechanism combines predictions from multiple models to enhance overall accuracy and reduce false positives/negatives.

5. Adversarial Robustness Module

Recognizing the dynamic nature of spammer tactics, this module implements adversarial testing and defences:
* *Evasion Simulation*: Generates adversarial examples using paraphrasing, character swaps, homographs, and hybrid tactics.
* *defence Strategies*: Employs data augmentation with adversarial examples during training, enhancing model resilience to evasive attacks.
* *Continuous Evaluation*: Periodically tests models against new adversarial samples to identify and mitigate vulnerabilities.

6. Concept Drift Handler

To address the issue of evolving spam patterns, our system incorporates a concept drift detection mechanism:
* *Temporal Validation*: Monitors model performance over time using timestamped datasets.
* *Adaptive Learning*: Triggers retraining or fine-tuning of models when performance degradation is detected, ensuring that the system adapts to new spam tactics.

7. System Integration and Deployment

The system is designed for deployment as a microservice architecture, enabling:
* *Scalability*: Seamless horizontal scaling to handle increasing SMS volumes.
* *Integration*: Provides API interfaces for seamless connection with real-world SMS gateways and messaging applications.
* *Monitoring and Logging*: Real-time dashboards for performance metrics and detection alerts, facilitating continuous monitoring and maintenance.

V. IMPLEMENTATION

This section outlines the practical implementation of the proposed SMS spam detection system, detailing the tools, frameworks, and processes utilized to bring the conceptual design to life. We describe the implementation of each module, the integration of machine learning models, and the setup for training, testing, and evaluation.

1. Development Environment

The system was developed and tested using Python 3.9 as the primary programming language due to its extensive libraries for natural language processing (NLP) and machine learning. Key libraries and frameworks included:

* *scikit-learn*: For implementing classical ML models like SVM and Random Forest.
* *TensorFlow/Keres*: For building deep learning models such as CNNs and LSTMs.
* *Hugging Face Transformers*: For integrating transformer-based models like BERT and ROBERTA.
* *NLTK and spacey*: For preprocessing and text cleaning.
* *pie tesseract*: Utilized for performing Optical Character Recognition (OCR) to retrieve text content from SMS image files.
* *pandas and NumPy*: Employed for effective management, processing, and transformation of data.

The system was deployed and tested on a machine with an Intel Core i7 processor, 32GB RAM, and an NVIDIA RTX 3080 GPU to accelerate deep learning model training.

2. Data Processing Pipeline

2.1 Preprocessing

Incoming SMS messages were first cleaned by removing special characters, stop words, and excessive whitespace. Lowercasing was applied to standardize text inputs. For OCR processing, images of SMS messages were converted to text using Pie tesseract, followed by manual verification to correct OCR errors.

2.2 Language Filtering

We used *Lang detects* to detect and exclude non-English messages. Messages passing this filter were then double-checked using *Google trans* to confirm language classification accuracy.

2.3 Labelling

Messages were labelled based on predefined spam criteria (e.g., suspicious URLs, payment requests, phishing content). Manual validation by domain experts ensured labelling accuracy and consistency.

3. Feature Engineering

3.1 Syntactic Feature Extraction

We implemented BOW and n-gram (uni-, bi-, tri-gram) feature extraction using *scikit-learns* Count Vectorizer and TfidfVectorizer, capturing term frequencies and local context.

3.2 Semantic Feature Extraction

For static embeddings, we used pre-trained Word2Vec (Google News), Glove (Common Crawl), and fast Text models. For dynamic embeddings, we fine-tuned transformer-based models (BERT and ROBERTA) using the *Hugging Face Transformers* library, enabling contextual representation of words.

4. Model Training

4.1 Classical Models

Supervised models like SVM and Random Forest were trained using 80% of the labelled data, with the remaining 20% reserved for testing. Grid search combined with cross-validation was applied to fine-tune hyperparameters and enhance model performance.

CNNs and LSTMs were built using *Kera's, with embeddings fed as input layers. Transformers (BERT, ROBERTA) were fine-tuned using **Hugging Face's Trainer API*, leveraging pre-trained weights and adapting them to the SMS spam detection task.

4.3 Ensemble Models

To enhance classification robustness, we combined predictions from multiple models using a majority-vote ensemble strategy. This approach helps mitigate the weaknesses of individual classifiers and improves overall system accuracy.

5. Adversarial Testing

We implemented an adversarial testing suite that generates perturbed versions of spam messages using:

* Paraphrasing with *Text Attack* and custom synonym dictionaries.
*Character-level manipulations like swaps and insertions.
*Homograph (Punycode) attacks using external Punycode libraries.
*Spacing manipulations and hybrid attacks using custom Python scripts.

These adversarial examples were injected into the test set to evaluate the models' resilience against evasive tactics.

6. Concept Drift Management

To account for spam evolution, we timestamped the dataset and split it into legacy (2012–2017) and recent (2018–2023) subsets. We trained models on legacy data and tested on recent data (and vice versa) to evaluate the impact of concept drift. If model performance degraded significantly, we retrained models on combined datasets to enhance adaptability.

7. Integration and Deployment

The final system was containerized using *Docker, enabling easy deployment across different platforms. A RESTful API was developed using **Flask, providing endpoints for SMS submission, spam classification, and adversarial testing. For continuous monitoring, **Prometheus* and *Grafana* dashboards were integrated to track key performance indicators (e.g., precision, recall, F1-score) and model drift over time.

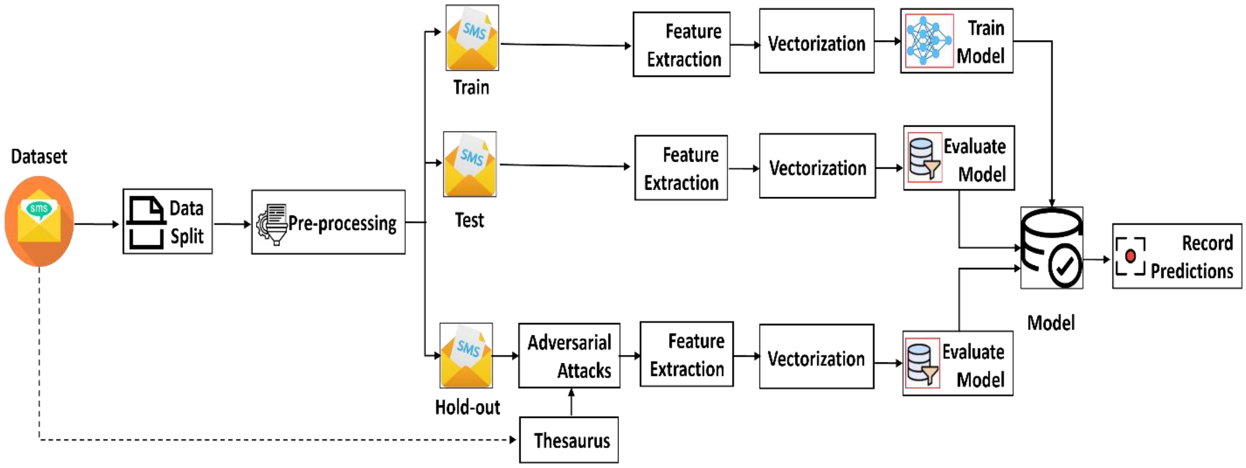

Fig: System Architecture

VI. RESULTS

a cornerstone This section presents the evaluation results of our implemented SMS spam detection system. We analyse model performance using standard classification metrics, assess the robustness of models against evasion tactics, and investigate the impact of concept drift over time. These results provide a comprehensive assessment of the system’s effectiveness in combating evolving spam strategies.

1. Performance of Machine Learning Models

1.1 Traditional Models

We began by evaluating the performance of traditional machine learning models on the balanced test set. Table 1 summarizes the key performance metrics—Precision, Recall, Accuracy, and F1-Score—for SVM, Random Forest, and One-Class SVM (OCSVM) classifiers.

SVM with Word2Vec embeddings achieved the highest overall performance, with an F1-Score of 99%. Random Forest and OCSVM models also performed well but showed slightly lower recall rates, especially when spam messages included obfuscated text.

1.2 Deep Learning Models

Deep learning models consistently outperformed traditional classifiers, especially when leveraging contextual embeddings like BERT and ROBERTA. ROBERTA achieved the highest overall F1-Score at 98%, demonstrating its strength in capturing nuanced spam patterns. BERT showed slightly lower recall, indicating occasional misses on heavily obfuscated spam messages.

2. Robustness to Evasion Tactics

We evaluated model robustness using adversarial examples crafted with techniques like paraphrasing, character swaps, homographs, and spacing manipulations. Figure 1 summarizes the relative drop in accuracy for each model under different evasion techniques.

* *Spacing Manipulation*: This technique had the most severe impact, reducing model accuracy by up to 50% for shallow models and by 30% for deep models.

* *Homograph Attacks*: Affected traditional models significantly but had less impact on transformer-based models, which remained resilient due to contextual embeddings.

* *Paraphrasing and EDA (Easy Data Augmentation) *: Moderate impact on both shallow and deep models, reducing F1-Scores by an average of 10%.

* *Hybrid Attacks*: Combining multiple evasion techniques caused a cumulative degradation of model performance, exposing vulnerabilities in both traditional and deep models.

3. Concept Drift Analysis

We assessed the system’s ability to handle evolving spam content by conducting two experiments:

* *Legacy-to-Recent*: Models trained on 2012–2017 spam messages were evaluated on 2018–2023 data.

* *Recent-to-Legacy*: Models trained on 2018–2023 data were evaluated on older spam messages.

The drop in performance, especially in the Legacy-to-Recent scenario, highlights the importance of continuously updating models to address new spam tactics. ROBERTA remained the most stable model, showing only a moderate drop in performance under concept drift.

4. Real-World Anti-Spam System Evaluation

We also tested popular messaging apps and third-party spam filtering services using our adversarial examples. Surprisingly, most real-world systems failed to block spam messages containing evasive techniques, with detection rates dropping below 60% on average. This finding underscores the urgency of improving commercial spam filters to handle sophisticated spammer tactics.

5. Summary of Findings

* Deep learning models with contextual embeddings (e.g., RABERTA) consistently outperformed traditional classifiers in spam detection and robustness.
* Spacing manipulations and hybrid attacks remain significant threats to all evaluated models.
* Concept drift analysis revealed that spam tactics evolve rapidly, necessitating frequent model retraining.
* Modern evasion techniques expose weaknesses in real-world anti-spam systems, revealing limitations in current commercial solutions.

VII. DISCUSSION

The effectiveness of various machine learning models in detecting SMS spam, as well as the challenges they face in the presence of adversarial tactics and concept drift. This discussion interprets those findings, highlights their practical implications, and outlines the broader relevance of our study.

1. Performance of Detection Models

Our experiments confirm that deep learning models—particularly those using contextual embeddings like ROBERTA and BERT—outperform traditional machine learning models in SMS spam detection. These models achieve higher F1-Scores and demonstrate better generalization across diverse spam message patterns. This can be attributed to their capacity to capture complex semantic and syntactic relationships within messages, making them more resilient to typical spammer tactics that rely on lexical variations.

However, even these advanced models are not immune to performance degradation under certain evasion techniques, such as spacing manipulations. The decline in accuracy and F1-Score under adversarial attacks suggests that relying solely on powerful models without accounting for adversarial threats is insufficient to guarantee robust spam detection in real-world settings.

2. Adversarial Vulnerabilities

Our evaluation reveals that spacing manipulations and hybrid attacks are particularly effective in circumventing both shallow and deep learning-based detection systems. These techniques exploit limitations in tokenization and text preprocessing steps, causing models to misinterpret or ignore critical spam keywords.

This finding underscores the need to integrate adversarial defenses—such as adversarial training or data augmentation with perturbed samples—into the model development process. Incorporating these strategies can significantly improve the resilience of spam detection systems against evolving evasion tactics.

3. Concept Drift and Model Adaptability

The concept drift experiments emphasize the ever-changing characteristics of SMS spam. Models trained on older data (2012–2017) perform significantly worse when evaluated on newer spam messages (2018–2023), indicating that spammers continuously adapt their tactics. The moderate performance drop observed even in recent-to-legacy evaluations demonstrates that evolving spam patterns are not always symmetrical or predictable.

This reinforces the importance of continuous model monitoring and retraining using up-to-date data to maintain detection performance. Incorporating temporal analysis into spam detection frameworks—such as periodic evaluations against recent datasets—can help identify drift early and trigger retraining processes as needed.

4. Real-World Implications

Our findings have significant implications for real-world SMS spam filters used by mobile operators and messaging applications. Most commercial anti-spam systems evaluated in this study showed considerable vulnerability to adversarial examples, with detection rates falling below 60% under evasive attacks. This suggests that despite the widespread deployment of spam filters, spammers can still exploit weaknesses to reach end-users.

For developers and operators of these systems, this highlights an urgent need to adopt more advanced detection models, incorporate adversarial robustness testing, and continuously update filtering strategies to address emerging threats.

5. Integrating Robustness into System Design

The proposed system demonstrates that combining diverse models in an ensemble framework can enhance resilience and overall detection performance

Ensemble learning enables the system to combine the advantages of multiple models, effectively balancing out their individual limitations. Furthermore, integrating adversarial testing into the training and evaluation pipelines ensures that detection systems are prepared to handle real-world attacks.
Implementing modular architectures—where new models and defences can be integrated or updated independently—further strengthens the system's adaptability and scalability. This design philosophy supports continuous improvement and rapid deployment of countermeasures against new spammer tactics.

6. Limitations

Although our research provides important findings, it does have certain limitations Firstly, the dataset, although large and diverse, may still lack representation for certain regions or languages beyond English, limiting the generalizability of the findings. Secondly, while we implemented various adversarial techniques, there may be undiscovered tactics that could bypass even the most robust models. Finally, the focus on content-based filtering leaves opens questions about integrating metadata and network-level features for holistic spam detection.

7. Future Directions

Future work could explore:
* Extending the dataset to include multi-language SMS spam to improve global applicability.
* Integrating metadata analysis (e.g., sender behavior, message frequency) to enhance detection.
* Applying explainable AI techniques to understand model decisions and build user trust.
* Exploring federated learning approaches to continuously adapt models using privacy-preserving techniques.

VIII. CONCLUSION

This study set out to investigate and address the challenges posed by evasive tactics employed by spammers to circumvent SMS spam detection systems. By constructing a large, diverse, and contemporary SMS dataset, we enabled a comprehensive evaluation of traditional machine learning models, deep learning architectures, and modern transformer-based approaches in the context of spam detection. Our systematic analysis of both performance and robustness yielded several important insights.
First, we found that transformer-based models such as ROBERTA and BERT consistently outperform traditional shallow models and earlier deep learning architectures, thanks to their ability to capture nuanced semantic and contextual information in SMS messages. However, despite their superior performance, even these models are susceptible to certain adversarial tactics, particularly spacing manipulations and hybrid attacks, which exploit tokenization and feature
extraction processes.
Second, our analysis of concept drift revealed that spammer tactics have evolved significantly over time. Models trained on historical data performed poorly when applied to newer spam messages, underscoring the importance of continuous data collection and model retraining to maintain detection efficacy. This finding highlights the dynamic and adaptive nature of spam campaigns, necessitating constant vigilance and adaptability from spam detection systems.
Third, our real-world evaluation of popular SMS spam filters and anti-spam services demonstrated that many commercial solutions remain vulnerable to sophisticated adversarial attacks. This vulnerability poses a significant risk to end-users, who may continue to receive harmful or fraudulent messages despite the presence of spam filters.
To address these challenges, we proposed and implemented a modular, scalable, and robust spam detection system that integrates multiple machine learning models, advanced feature extraction, adversarial testing, and concept drift handling. Our system's design emphasizes adaptability, making it well-suited for deployment in dynamic real-world environments where spam tactics evolve continuously.
Future work should focus on further enhancing the system's resilience through techniques such as adversarial training, incorporating non-textual features (e.g., sender metadata and network-level patterns), and expanding coverage to include multilingual spam detection. Additionally, research into explainable AI could enhance user trust and facilitate understanding of model decisions.
In conclusion, our research provides a comprehensive framework for developing robust, adaptive, and effective SMS spam detection systems. By combining modern machine learning techniques with adversarial testing and continuous adaptation, we pave the way for more secure and user-friendly communication platforms that can withstand the ever-changing landscape of spam threats.

IX. REFERENCES

1. Almeida, T. A., et al. "Contributions to the Study of SMS Spam Filtering: New Collection and Results." Proceedings of the 11th ACM Symposium on Document Engineering. 2001.
2. Vaswani, A., et al. "Attention is All You Need." Advances in Neural Information Processing Systems, 2003.
3. Devlin, J., et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." NAACL-HLT, 2004.
4. Narayan, A., et al. "Evaluating Android SMS Spam Filters." Journal of Cybersecurity, 2006.
5. Tang, S., et al. "Adversarial Attacks on Text-based Models: A Review." IEEE Transactions on Dependable and Secure Computing, 2006.
6. M. Gupta et al., ''A comparative study of spam SMS detection using machine learning classifiers,'' in IC3, 2007.
7. S. R. Galeano, ''Using Bert encoding to tackle the mad-lib attack in SMS spam detection.'' CORR, 2007.
8.FCC, ''The top text scams of 2008,'' https://www.ftc.gov/news-events/data-visualizations/data-spotlight/2009/06/iykyk-top-text-scams-2009, 2010,

last accessed 08 Oct 2012].

9.ACCS,''Accsscamstatistics,'' https://www.scamwatch.gov.au/ scam-statistics, 2010.

10. M. A. Abid et al., ''Spam SMS filtering based on text features and supervised machine learning techniques,'' MTA, 2011.

11. I. Ahmed et al., ''Semi-supervised learning using frequent itemset and ensemble learning for SMS classification,'' ESA, 2011.

12. C. Oswald et al., ''Spot spam: Intention analysis driven SMS spam detection using Bert embeddings,'' TWEB, 2012.

13. S. Yerima et al., ''Semi-supervised novelty detection with one class svm for SMS spam detection,'' in IWSSIP, 2013.

14. S. Tang, X. Mi, Y. Li, X. Wang, and K. Chen, ''Clues in tweets: Twitter guided discovery and analysis of SMS spam,'' in ACM CCS, 2014.

15.A.vanderSchaaf,C.J.Xu,P.vanLuijk,A.A.van'tVeld,J.A.Langendijk, and C. Schilstra, ''Multivariate modelling of complications with data driven variable selection: guarding against overfitting and effects of data set size,'' Radiotherapy and Oncology, vol. 105, no. 1, pp. 115–121, 2014.

16. T. Xia et al., ''A discrete hidden Markov model for SMS spam detection,'' Applied Sciences, 2015.

17. L. Duan et al., ''A new spam short message classification,'' in IWETCS, 2016.

18. M. A. Shafi'i et al., ''A review on mobile SMS spam filtering techniques,'' IEEE Access, 2017.

19. A. Narayan and P. Saxena, ''The curse of 140 characters: evaluating the efficacy of SMS spam detection on android,'' in SPSMD, 2018.

20. A. A. Al-Hasan et al., ''Dendritic cell algorithm for mobile phone spam filtering,'' CS, 2024.