

Article

Revisiting SMS Spam Detection: The Impact of Feature Representation on Classical Machine Learning Models

Meryem Soysaldı Şahin ^{1,*} , Durmuş Özkan Şahin ¹  and Areej Fateh Salah ² ¹ Department of Computer Engineering, Ondokuz Mayıs University, Samsun 55139, Türkiye; durmus.sahin@omu.edu.tr² Department of Computer Engineering, Institute of Graduate Studies, Ondokuz Mayıs University, Samsun 55139, Türkiye; areej.salah@bil.omu.edu.tr

* Correspondence: meryem.soysaldi@bil.omu.edu.tr

Abstract

The proliferation of unsolicited short messages (SMS spam) poses persistent challenges to mobile communication security and user privacy. This study presents a systematic benchmarking and analytical investigation of classical machine learning approaches for SMS spam detection, focusing on the impact of text feature representation under imbalanced short-text conditions. In practical SMS filtering systems, minimizing false positives (i.e., incorrectly blocking legitimate messages) is a critical operational constraint. Therefore, beyond overall accuracy, precision and specificity are emphasized to ensure reliable preservation of legitimate communication. Using the SMSSpamCollection dataset (5574 messages: 747 spam and 4827 ham), seven feature representation techniques were evaluated in combination with six widely adopted classifiers, resulting in 42 configurations assessed under 10-fold cross-validation. The results demonstrate that feature representation plays a more critical role than classifier complexity. Character-level 3-grams combined with Logistic Regression achieved the best overall performance, reaching 98.55% accuracy, with 98.55% precision and 90.50% recall for the spam class (F1-score = 94.32%), and 0.9893 AUC. Linear SVM produced comparable results, highlighting the effectiveness of linear models when paired with expressive representations. Beyond reporting performance metrics, this study analyzes feature–classifier interaction patterns and clarifies practical trade-offs between precision, recall, and computational efficiency. The findings provide reproducible baselines and structured guidance for designing efficient SMS spam filtering systems.

Keywords: SMS spam detection; text feature representation; machine learning classifiers; character; TF-IDF; LR; SVM; imbalanced data; performance metrics



Academic Editors: Chuan Zhang, Xiangyun Tang, Tao Zhang, Jiacheng Wang and Jiqiang Liu

Received: 24 January 2026

Revised: 19 February 2026

Accepted: 20 February 2026

Published: 21 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The ubiquity of mobile phones has cemented the Short Message Service (SMS) as a fundamental and global communication channel. Its low cost, high accessibility, and immediate delivery have made it indispensable for personal and professional correspondence. However, this widespread adoption has been paralleled by a surge in unsolicited commercial and malicious messages, universally termed "SMS spam". These messages range from intrusive advertisements to sophisticated phishing attempts and fraud schemes designed to steal personal information or financial credentials. The proliferation of SMS spam not only disrupts user experience and erodes trust in mobile ecosystems but also poses tangible security and privacy risks, leading to notable financial losses and data breaches for

vulnerable individuals. Consequently, developing robust, automated, and efficient SMS spam detection systems is a critical imperative for telecommunication providers, security experts, and end-users alike.

Automatic SMS spam detection is inherently a challenging text classification problem. The core difficulties stem from the unique characteristics of SMS data. Messages are typically short, offering limited contextual information for analysis. They often employ informal language, including slang, deliberate misspellings, abbreviations, and symbol substitutions to evade simple keyword filters. Furthermore, real-world datasets exhibit a pronounced class imbalance, where legitimate messages (ham) vastly outnumber spam messages. This imbalance can bias classifiers toward the majority class, reducing their ability to correctly identify spam messages—the more critical class to detect. At the same time, from a practical deployment perspective, false positives (i.e., legitimate messages incorrectly classified as spam) can severely undermine user trust and system usability. Therefore, SMS spam detection systems must balance recall on the minority spam class with high precision and specificity to avoid excessive blocking of legitimate communication. Traditional rule-based and static keyword-filtering approaches have proven inadequate in this dynamic landscape, as spammers continuously evolve their tactics. This has established machine learning as the dominant paradigm, owing to its ability to learn complex, discriminative patterns directly from data.

The performance of a machine learning-based classifier is profoundly influenced not only by the choice of algorithm but also by how raw textual data are transformed into numerical representations suitable for model processing. This step, known as feature representation or Feature Engineering (FE), determines how textual information is perceived by the classifier. Different representation techniques capture distinct linguistic characteristics, such as Term Frequency (TF), contextual importance, or sub-word patterns. Each technique presents unique trade-offs in terms of feature space dimensionality, robustness to noise, and sensitivity to lexical variation. Consequently, selecting an appropriate feature representation is a critical design decision that can significantly affect classification performance.

To address these challenges, this study conducts a systematic empirical investigation into the impact of text feature representation techniques on SMS spam detection performance. The central research question guiding this work is this: Which combinations of text feature representation techniques and classical machine learning classifiers yield the most robust and effective performance for SMS spam detection on an imbalanced dataset? To operationalize this research question, the following testable hypotheses are formulated:

Hypothesis 1. *Different text feature representation techniques lead to statistically significant differences in classification performance for SMS spam detection.*

Hypothesis 2. *Feature representations that capture sub-word or weighted term information (e.g., character n -grams or TF-IDF) achieve higher performance than simple frequency-based representations on imbalanced SMS datasets.*

Hypothesis 3. *The effectiveness of a feature representation technique is dependent on the choice of the machine learning classifier, indicating an interaction effect between feature representation and classifier type.*

To test these hypotheses, seven feature representation methods are evaluated in conjunction with six widely used classical classifiers under identical experimental conditions using 10-fold cross-validation. These hypotheses reflect the analytical objective of this study, which aims to understand performance behavior and feature–classifier interaction dynamics rather than to introduce a new algorithmic architecture.

This study focuses exclusively on the SMSSpamCollection benchmark dataset, which remains one of the most widely adopted corpora for SMS spam research. While this dataset enables controlled and reproducible comparison with prior studies, it reflects the characteristics of SMS spam as observed around 2012. Therefore, the findings should be interpreted as establishing strong empirical baselines for this benchmark rather than as definitive conclusions for all contemporary SMS spam scenarios.

1.1. Motivation and Contribution

Despite the extensive research conducted on SMS spam detection, several fundamental challenges remain unresolved, particularly in the context of feature representation for short and noisy text data. Most existing studies emphasize improving classifier architectures or adopting increasingly complex models, such as deep learning and transformer-based approaches. While these methods often achieve high performance, they typically require substantial computational resources, large labeled datasets, and careful parameter tuning, which limit their practicality for lightweight or real-time SMS filtering systems.

Another critical limitation in the existing literature is the lack of systematic analysis of how different text feature representation techniques interact with multiple machine learning classifiers under identical experimental conditions. Many studies rely on a single representation method or evaluate a narrow range of classifiers, thereby overlooking the strong dependency between feature design and classifier behavior.

Motivated by these gaps, this study aims to provide a comprehensive and controlled evaluation of text feature representation techniques for SMS spam detection. Rather than proposing a new classification model, the focus is placed on understanding how classical machine learning classifiers respond to different textual representations when applied to imbalanced short-text data. The main contributions of this work are summarized as follows:

This work presents a systematic benchmarking and analytical study of SMS spam detection under controlled and reproducible experimental conditions. Instead of introducing a novel algorithmic architecture, the primary objective is to investigate how different text feature representations interact with classical machine learning classifiers in the presence of short, noisy, and imbalanced SMS data. Through the evaluation of seven feature representation techniques combined with six widely used classifiers (42 configurations in total), this study provides structured empirical insights into the relative impact of feature design compared to classifier selection, the existence of interaction effects between representation type and classifier behavior, and the trade-offs between precision, recall, and computational efficiency under realistic deployment constraints. Beyond confirming the effectiveness of character-level modeling for noisy SMS text, the analysis clarifies the conditions under which such representations provide measurable advantages and when alternative representations may remain competitive. By establishing reproducible baselines and offering practical configuration guidelines, this study contributes analytical understanding and decision-support value for both researchers and practitioners working on imbalanced short-text classification tasks.

1.2. Organization

The remainder of this paper is organized as follows. Section 2 reviews related work on SMS spam detection, categorizing existing approaches into classical machine learning, deep learning, transformer-based models, and FE strategies, while highlighting their strengths and limitations. Section 3 presents the methodology and experimental setup, including dataset description, text preprocessing pipeline, feature representation techniques, classification models, and evaluation strategy. Section 4 reports and analyzes the experimental results, providing detailed performance comparisons across different feature-classifier

configurations along with confusion matrix analysis and visualization of key findings. Finally, Section 5 concludes the paper with a summary of the main findings and outlines directions for future research.

2. Literature Review

This section reviews existing research on SMS spam detection, organized into four categories: Classical Machine Learning, Deep Learning Approaches, Transformer/Large Language Model (LLM) Approaches, and FE and Hybrid/Robustness Approaches. The objective is to provide a comprehensive overview of methodologies, datasets, preprocessing techniques, feature extraction, and evaluation metrics used in prior studies.

2.1. Classical Machine Learning

The following studies focus on traditional machine learning approaches applied to SMS spam detection. In [1], a comparative study of various machine learning algorithms for SMS spam detection was conducted, focusing on their relative strengths and accuracy in classifying spam and non-spam messages. The study used a publicly available SMS dataset from Kaggle containing 5574 labeled messages in English, preprocessed via tokenization, stop-word removal, and feature extraction. The authors identified two main features for the models: message length and Information Gain (IG) matrix, demonstrating that spam messages tend to be longer and contain distinctive terms. Three supervised learning algorithms—Naive Bayes (NB), Random Forest (RF), and Logistic Regression (LR)—were evaluated using 5-fold cross-validation to prevent overfitting and to measure the performance of each classifier under different feature configurations. The results indicated that NB outperformed both RF and LR, achieving the highest accuracy of 98.445% with the IG matrix alone, while also requiring the shortest runtime. RF also performed well, especially when using both features, making it a viable alternative. LR achieved comparatively lower accuracy. The study highlights the effectiveness of NB in SMS spam classification, particularly when combined with appropriate feature selection methods such as IG, while confirming that the choice of features and algorithms critically impacts classification performance.

In [2], an automatic classification approach was developed for Indonesian SMS messages. The messages were categorized into three types: Ham, Promotional, and Spam. A total of 4125 messages were used for training and 1260 messages for testing. During preprocessing, texts were cleaned, converted to lowercase, and unnecessary characters were removed. BoW was applied for feature extraction. Eight classification algorithms were evaluated: Multinomial Naive Bayes (MNB), Multinomial Logistic Regression (MLR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT), Stochastic Gradient Descent (SGD), XGBoost, and RF. 10-fold cross-validation was used for validation. Results showed that with the full dataset, RF achieved the highest accuracy (94.62%), followed closely by MLR and XGBoost. When a balanced dataset was used, the performance of SVM and MLR improved. The study demonstrates that traditional machine learning algorithms are effective for SMS classification, with RF, MLR, SVM, and XGBoost providing the best performance.

In [3], the authors aimed to classify SMS messages into Ham and Spam. The dataset used was the SMSSpamCollection dataset from Kaggle, containing 5572 messages (4825 Ham and 747 Spam). During preprocessing, texts were cleaned, converted to lowercase, tokenized, stop words removed, abbreviations expanded, and the effects of stemming, lemmatization, and message length were studied. TF-IDF was applied for feature extraction. Six classification algorithms were evaluated: RF, Gradient Boosting (GB), Extra Trees (ET), LR, SVM, and MNB. The dataset was split into 70% training and 30% testing, and performance was measured using accuracy, balanced accuracy, F1-score, and

ROC-AUC. The results showed that using stemmed text, the ET classifier achieved the highest accuracy (0.985) and balanced accuracy (0.967), while MNB had the lowest performance (accuracy 0.969, balanced 0.941). When combining message length with text features, GB achieved the maximum accuracy (0.987) and balanced accuracy (0.957), indicating that including sentence length improved classification. Both stemming and lemmatization were effective preprocessing methods with minor differences in performance. The study demonstrates that traditional machine learning algorithms are effective for SMS spam detection, with ET, GB, RF, and SVM providing the best performance. Future work could explore word embeddings (Word2Vec, GloVe) and deep learning models (RNN, RCNN) to further enhance classification.

In [4], SMS spam detection was addressed using machine learning techniques on the SMSSpamCollection dataset from Kaggle, consisting of 5572 messages, where 86% are ham and 13% are spam. During preprocessing, texts were cleaned, converted to lowercase, and tokenized; stop words were removed; and stemming was applied. In addition, message length and the presence of URLs, common spam words, and sentiment symbols were considered features. BoW with TF weighting and TF-IDF weighting schemes was applied for feature extraction, and Chi-Square was used for feature selection. Various classifiers were evaluated, including SVM, KNN, DT, LR, RF, AdaBoost, and MNB. The dataset was split into 80% training and 20% testing. Performance was measured using accuracy, precision, recall, and F1-score. The results showed that applying Chi-Square for feature selection improved classification accuracy from 92% to 95.75%. Among the classifiers, MNB achieved the highest accuracy of 99.7%, followed by SVM at 97%. The study demonstrates that combining effective feature extraction, selection, and preprocessing methods significantly improves SMS spam detection.

In [5], a machine learning-based framework was proposed for efficient SMS spam detection, with the goal of reliably distinguishing spam messages from legitimate (ham) messages. The study utilized a combined dataset derived from three sources, including two publicly available datasets and one self-collected by the authors, ultimately creating a balanced 60:40 partition of ham and spam messages. Preprocessing steps included text cleaning, normalization, tokenization, stop-word removal, and stemming, followed by feature extraction using Count Vectorizer and TF-IDF vectorization techniques to transform textual data into numerical representations suitable for machine learning models. Eleven classification algorithms were initially evaluated, including LR, Bernoulli NB, MNB, DT, KNN, SVM, RF, AdaBoost, Bagging, ET, Gradient Boosting, and XGBoost. The top four performing models—LR, Bernoulli NB, RF, and ET—were further optimized using hyperparameter tuning. Results demonstrated that the Bernoulli NB classifier achieved the highest performance, attaining an accuracy of 96.63%, with strong precision, recall, and F1-score metrics, outperforming other evaluated models. This study emphasizes the importance of careful dataset preparation, text preprocessing, and systematic evaluation of multiple classifiers to achieve robust spam detection. By integrating multiple datasets and optimizing classifiers, the research provides a reliable baseline for SMS spam detection using classical machine learning methods and highlights the effectiveness of Bernoulli NB in this domain.

In [6], the authors conducted a comparative evaluation of four classical machine learning classifiers—NB, KNN, LR, and RF—for SMS spam detection. The study utilized the widely recognized SMSSpamCollection dataset, comprising 5574 English messages, and applied standard text preprocessing techniques, including tokenization, normalization, stop-word removal, and stemming/lemmatization to transform textual data into numerical features suitable for model training. Through systematic experimentation and evaluation using precision, recall, and F1-score metrics, the study found that LR achieved the highest F1-score of 0.98, while NB also demonstrated strong performance with an F1-score of 0.95.

These results established clear performance benchmarks for traditional machine learning classifiers in SMS spam detection and provided insights for model selection in similar applications. The study was conducted on a single historical dataset and focused on classical machine learning approaches, without incorporating comparisons to deep learning or transformer-based models. The methodology primarily evaluated performance metrics within the given dataset, and further exploration could consider multilingual datasets, extended FE strategies, and practical deployment considerations.

In [7], a comparative analysis of multiple machine learning techniques—RF, GB, AdaBoost, SVM, LR, and an Ensemble Voting Classifier—was conducted by the author for SMS spam detection. The experimental evaluation was performed using a dataset of 5572 SMS messages, consisting of 745 spam and 4827 ham messages. The study employed TF-IDF for text feature representation, followed by label encoding and stratified data splitting. All models were optimized through hyperparameter tuning using GridSearchCV, and performance was assessed using accuracy, precision, recall, and F1-score, with 10-fold cross-validation to ensure robust evaluation. The results demonstrated that SVM achieved the highest classification performance, reaching an accuracy of 98.57%, with strong precision and recall for both spam and legitimate messages. The Ensemble Voting Classifier also showed competitive performance, achieving an accuracy of 98.48% and improved recall for spam messages. While RF, GB, and AdaBoost produced high overall accuracy, their relatively lower recall for spam indicated occasional misclassification of spam as legitimate messages. Feature importance analysis revealed that promotional terms and numerical patterns were among the most influential indicators of spam. The study highlights the effectiveness of SVM and ensemble-based approaches for SMS spam detection and suggests that future research should focus on addressing class imbalance and exploring deep learning-based models to further enhance detection performance and adaptability.

In [8], the authors proposed an enhanced machine learning-based approach for SMS spam classification with a particular focus on improving model performance through hyperparameter optimization. The study addresses the limitations of conventional machine learning models that rely on default parameter settings, which may lead to suboptimal classification results in SMS spam detection tasks. The proposed framework applied standard text preprocessing techniques, including text cleaning, normalization, and tokenization, followed by feature extraction using statistical text representation methods suitable for short message classification. Multiple supervised machine learning classifiers, such as LR, SVM, GB, and Artificial Neural Networks (ANN), were trained and evaluated on an SMS spam dataset to assess their baseline performance. To enhance classification accuracy, the study employed an evolutionary-based hyperparameter optimization strategy, enabling automated tuning of critical model parameters. Experimental results demonstrated that optimized classifiers significantly outperformed their non-optimized counterparts across multiple evaluation metrics, including accuracy, precision, recall, and F1-score. The findings highlight the effectiveness of hyperparameter optimization in improving SMS spam detection performance without altering the underlying feature representation. While the study confirms the importance of parameter tuning in machine learning-based SMS spam detection, it primarily focuses on classifier optimization rather than exploring the comparative impact of different text feature representation techniques. As such, the interaction between feature representation methods and optimized classifiers remains an open area for further investigation.

In [9], a supervised machine learning-based framework for SMS spam detection was presented to improve classification accuracy, system scalability, and responsiveness in real-time mobile communications. The experimental evaluation was conducted on a balanced dataset of 5573 labeled SMS messages, categorized as spam or legitimate (ham). Standard text preprocessing techniques, including text normalization, tokenization, and

feature vectorization using BoW and TF-IDF representations, were applied to prepare the data for model training. Multiple classical machine learning classifiers were implemented and evaluated, with particular focus on probabilistic approaches such as NB, while comparative experiments were conducted to assess overall classification performance. The results indicated that the proposed framework achieved a classification accuracy of 98.4% and demonstrated suitability for real-time deployment scenarios. The study was performed using static training data, without incorporating mechanisms for continuous learning or adaptive updates. Additionally, the framework was evaluated primarily on English-language SMS messages, and further exploration could examine multilingual datasets and feature optimization strategies to enhance applicability in diverse and evolving messaging environments.

In [10], a machine learning-oriented framework for SMS spam detection was proposed by the authors to enhance the effectiveness and reliability of filtering unwanted messages in mobile communication systems. The study addresses the inherent challenges of SMS spam detection, including the short length of messages, informal language, frequent use of abbreviations, and limited availability of large-scale labeled datasets compared to email spam. The proposed framework employed a comprehensive preprocessing pipeline that included data cleaning, normalization, tokenization, stop-word removal, and handling of class imbalance. Textual data were transformed into numerical representations primarily using the TF-IDF feature extraction technique, enabling the model to capture discriminative characteristics between spam and legitimate (ham) messages. Additional handcrafted features, such as message length, digit count, special characters, and n-gram representations, were also discussed as part of the FE process. Several machine learning algorithms were explored, with particular emphasis on a multi-layer classifier, alongside classical approaches such as NB, SVM, and Maximum Entropy models. The models were trained and evaluated using cross-validation to ensure robustness and to mitigate overfitting. Experimental results demonstrated that the proposed approach achieved reliable classification performance and scalability, highlighting the effectiveness of TF-IDF-based representations combined with machine learning classifiers for SMS spam detection. While the study confirms the suitability of machine learning techniques for combating SMS spam, it primarily focuses on a single text representation strategy and does not conduct a systematic comparison of alternative feature representation methods. Consequently, the impact of different textual feature representations on classification performance remains an open research direction, motivating further investigation into optimized FE strategies for SMS spam detection.

In [11], a comprehensive SMS spam detection system was proposed using six advanced machine learning algorithms: LR, NB, RF, SVM, GB Machine (GBM), and Light GB Machine (LGBM). The study used the SMSSpamCollection dataset (5574 English messages, 747 spam, 4825 ham) obtained from Kaggle, with binary and numeric features such as presence of URLs, numbers, dates, emails, emoticons, and counts of spam/ham words. The dataset was pre-processed to remove noise, punctuation, and stopwords and normalized for consistency. The models were trained on 70–80% of the data and tested on the remaining 20–30%, with performance evaluated using accuracy, precision, recall, F1-score, sensitivity, and specificity. Results demonstrated that LGBM outperformed all other classifiers, achieving 100% accuracy, precision, recall, F1-score, sensitivity, and specificity, indicating a perfect classification of spam and ham messages. Other models also performed well: RF (98.2%), GBM (98.2%), SVM, and LR (97.79%), and NB (97.55%). The study highlights the effectiveness of FE on metadata and advanced GB techniques in SMS spam detection, achieving superior results compared to prior research.

In [12], a comprehensive framework for SMS spam detection was presented, leveraging LLMs and robust feature extraction techniques to enhance text classification performance. The study systematically evaluated six classifiers: NB, KNN, SVM, Linear Discriminant

Analysis (LDA), DT, and Deep Neural Networks (DNN), using two feature extraction methods: BoW and TF-IDF. To mitigate the high dimensionality of textual data, Principal Component Analysis (PCA) was applied to TF-IDF features. Preprocessing involved tokenization, stop-word removal, stemming, and vector space modeling. The study used a dataset of 5572 SMS messages from Kaggle, with 750 labeled as spam and 4825 as ham. Results indicated that TF-IDF consistently outperformed BoW, with NB achieving the highest accuracy (96.2%), followed by SVM (94.5%) and DNN (91.0%). The findings highlight that the combination of TF-IDF with NB, SVM, or DNN provides an effective and context-aware approach for SMS spam detection, demonstrating that classical machine learning methods remain highly competitive when coupled with modern feature extraction and dimensionality reduction techniques. The proposed framework also included performance evaluation using accuracy, precision, recall, F1-score, AUC, confusion matrices, and ROC curves, providing a rigorous assessment of the model's effectiveness.

2.2. Deep Learning Approaches

The following studies employ deep learning architectures, illustrating the evolution from classical machine learning (ML) methods to neural network-based solutions for SMS spam detection.

In [13], a detailed comparative analysis was conducted on optimized machine learning and deep learning methods for spam detection, highlighting enhancements in classification accuracy achieved through outlier removal, feature selection, and ensemble strategies. The experimental evaluation was conducted using the Spambase dataset from the UCI Machine Learning Repository, which contains 4601 email instances represented by 57 numerical features and one binary class label (spam or ham). The proposed framework incorporated DBSCAN and Isolation Forest for outlier detection, alongside multiple feature selection techniques, including Heatmap-based correlation analysis, Chi-Square, and Recursive Feature Elimination (RFE). For classification, several machine learning models were employed, namely Multinomial Naïve Bayes (NB), KNN, RF, and GB, which were further combined using a stacking-based ensemble approach. In parallel, deep learning models such as ANN, Recurrent Neural Networks (RNN), and Gradient Descent-based optimization models were implemented to enable a comparative performance analysis. The experimental results demonstrated that the ensemble-based machine learning framework achieved perfect classification performance, reporting 100% accuracy, precision, recall, F1-score, and AUC, while the deep learning models achieved a maximum accuracy of 99.28% with a low loss value. The findings indicate that, for structured tabular email datasets, optimized machine learning ensembles outperform deep learning models in terms of accuracy and computational efficiency.

In [14], a hybrid deep learning model was presented to detect SMS spam messages in Turkish and English. The model combined Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) to capture both local n-gram features and long-term dependencies in text. Two datasets were used: TurkishSMSCollection (4751 messages) and the UCI SMS Spam dataset (5574 messages). Preprocessing involved cleaning text, lowercasing, removing stop words, lemmatization, and word embedding using a custom model to represent words numerically. The hybrid CNN + GRU model was trained with 10-fold cross-validation, and hyperparameters such as the number of neurons, learning rate, dropout, and activation functions were fine-tuned. Results showed the CNN + GRU hybrid achieved the highest performance with 99.07% accuracy, 99.37% recall, 99.22% F1-score, and 99.06% precision, outperforming CNN and GRU individually. The study highlights that combining CNN and GRU effectively improves SMS spam detection, particularly for Turkish messages, and provides a reference for future work on multilingual spam detection.

In [15], a hybrid CNN–Long Short-Term Memory (LSTM) architecture was designed for bilingual SMS spam detection in Arabic and English. The proposed model leverages convolutional layers to extract local n-gram features from text, while LSTM layers capture long-term sequential dependencies across message content. By combining spatial feature extraction with temporal sequence modeling, the hybrid architecture addresses the unique challenges of multilingual spam detection, where linguistic patterns and character distributions vary significantly across languages. The study demonstrated that hybrid deep learning architectures can effectively generalize across different linguistic contexts, achieving competitive performance on both Arabic and English SMS datasets. This work highlights the importance of architectural design choices in adapting deep learning models to multilingual text classification tasks, complementing the broader trend toward neural network-based spam detection systems. The study highlights that combining CNN and GRU effectively improves SMS spam detection, particularly for Turkish messages, and provides a reference for future work on multilingual spam detection.

In [16], a unified framework that leverages machine learning and deep learning approaches to effectively detect SMS spam messages was presented. The framework integrates manual feature extraction methods, including TF-IDF and Count Vectorization, with deep contextual embeddings generated by Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT) models. Feature optimization techniques, specifically Low-Rank Adaptation (LoRA) and Reinforcement Fine-Tuning (ReFT), were applied to enhance the representations prior to model training. Multiple ensemble-based machine learning classifiers, including XGBoost, RF, Gradient Boosted Decision Trees (GBDT), and stacking models, were trained and evaluated using these optimized features. The study also explored feature fusion by combining the best-performing manual and automatic features to provide more comprehensive input to the classifiers. The evaluation metrics included accuracy, precision, recall, F1-score, and specificity. The results indicated that the XGBoost classifier, when trained on the fused cross-validation and GPT features optimized with LoRA, achieved the highest performance with an accuracy of 99.82%, recall of 100%, precision of 71.43%, and an F1-score of 83.33%. Additionally, explainable artificial intelligence (AI) techniques, SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME), were used to interpret model predictions and provide insights into feature contributions. While the framework achieved high performance on the datasets used, it was developed and evaluated primarily on English-language SMS messages. The study acknowledged that further exploration could extend the framework to multilingual datasets and investigate deployment considerations in practical scenarios.

2.3. Transformer/LLM Approaches

Recent research increasingly leverages transformers and LLMs to capture complex textual patterns and semantic features beyond traditional and deep learning architectures.

In [17], the authors introduced a modified Transformer-based deep learning model for SMS spam detection, aiming to investigate whether attention-based architectures could overcome the limitations of traditional machine learning and recurrent neural network approaches in short-text classification tasks. The study was motivated by the inherent challenges of SMS data, including short message length, informal language usage, class imbalance, and limited contextual information.

In contrast to traditional SMS spam detection methods, which heavily depend on handcrafted features or sequential models such as LSTM, the proposed approach adapts the Transformer architecture by introducing a trainable memory mechanism and a customized decoder structure tailored for binary classification. This design enables the model to focus

on discriminative textual patterns through self-attention without relying on recurrent computations. The authors evaluated their approach on the widely used SMS Spam Collection v1 dataset and a secondary Twitter spam dataset to assess generalization capability. The experimental framework included comparisons with classical machine learning classifiers, LSTM-based models, and hybrid CNN–LSTM architectures. The findings demonstrated that attention-based models are particularly effective in capturing sparse and noisy patterns commonly found in SMS messages, offering improved robustness in both balanced and imbalanced datasets. Beyond performance evaluation, the study provided an in-depth architectural analysis of Transformer-based models for short-text classification, highlighting both strengths and limitations. In particular, the authors identified challenges related to out-of-vocabulary tokens and informal language, which can negatively impact deep learning models when training data is limited.

Overall, this work establishes Transformer-based architectures as a promising direction for SMS spam detection and serves as a conceptual reference point for subsequent studies exploring alternative text representation strategies, FE techniques, and hybrid machine learning approaches.

In [18], an advanced framework for SMS spam detection integrated machine learning, deep learning, transformer-based models, and large language models to enhance overall detection performance. The study explored a wide range of models, including classical machine learning classifiers (LR, Multinomial NB, and SVM), deep learning architectures (CNN, BiLSTM, and CNN+BiLSTM), transformer models (M-BERT, BERTbase, XLM-Rbase, and XLNetbase), and LLMs (Phi-3 and H2O-Danube). The experiments were conducted on a specialized SMS spam dataset containing 5169 English messages labeled as spam or ham. Preprocessing included removal of extraneous characters, TF-IDF feature extraction, and embedding-based representations using Keras embeddings, Word2Vec, GloVe, and FastText. All models were fine-tuned with optimized hyperparameters to achieve the best possible performance. Evaluation metrics included precision, recall, and F1-score. Results demonstrated that traditional machine learning models achieved moderate F1-scores (0.73–0.78), while deep learning models improved performance up to 0.84, particularly when combining embeddings and hybrid CNN–BiLSTM architectures. Transformer-based models further enhanced performance, with XLNet-base reaching an F1-score of 0.91. Among the LLMs, H2O-Danube achieved the highest performance, attaining a macro F1-score of 0.94, outperforming all other machine learning, deep learning, and transformer-based approaches. Class-wise analysis showed effective detection of both spam and ham messages, though some misclassifications occurred for subtle interclass patterns. The study highlights the effectiveness of LLMs in capturing complex linguistic patterns in SMS messages and demonstrates superior performance over traditional, deep learning, and transformer-based methods.

2.4. FE and Hybrid/Robustness Approaches

The final group of studies emphasizes enhanced FE, hybrid methods, and robustness considerations, providing complementary insights into SMS spam detection. In [19], a machine intelligence-based hybrid system, incorporating sentiment analysis for enhanced SMS spam detection, was introduced, aiming to improve classification accuracy while addressing the challenges of short-text sparsity and semantic ambiguity in SMS messages. Unlike traditional spam filtering approaches, the proposed system combines spam classification and sentiment polarity analysis within a unified framework. The proposed methodology incorporates extensive preprocessing and Word2Vec-based data augmentation to enrich textual representations. Feature selection is performed using six distinct techniques. For classification, a hybrid KNN–SVM model is employed, with its param-

eters optimized using Rat Swarm Optimization (RSO) to enhance decision boundaries and overall predictive performance. Additionally, sentiment analysis is conducted using lexicon-based approaches, namely AFINN and SentiWordNet, to classify SMS messages into positive and negative polarities. The experimental evaluation was conducted on three benchmark datasets, including an English SMS dataset, an email dataset, and the SpamAssassin dataset. The reported results demonstrate that the proposed hybrid model consistently outperformed conventional machine learning and deep learning classifiers such as NB, LR, RF, ANN, and CNN. Specifically, the model achieved a spam detection accuracy of 99.69% on the SMS dataset and 99.82% on the SpamAssassin dataset, along with high precision (99.32%), recall (98.33%), and F-measure (98.83%).

In [20], the robustness of machine learning-based SMS spam detection systems against spammers' evasive techniques was explored. The study highlights the growing challenge of SMS spam and emphasizes the limitations of existing detection models when exposed to evolving spam strategies. To address data scarcity in SMS spam research, the authors introduced a newly curated large-scale SMS dataset comprising 67,018 messages, including 39.1% spam and 60.9% legitimate messages, collected from multiple real-world sources over more than a decade. This dataset represents the largest publicly available SMS spam dataset to date and enables longitudinal analysis of spam evolution. The study evaluated a wide range of machine learning and deep learning models using both syntactic features (such as BoW and TF-IDF) and semantic feature representations, including Word2Vec, fastText, GloVe, and transformer-based models such as BERT and RoBERTa. Experimental results demonstrated that semantic-based word embeddings generally improved classification performance compared to traditional count-based representations. Furthermore, the authors examined the resilience of SMS spam detection models against multiple types of evasive perturbations, including character-level, word-level, sentence-level, and multi-level attacks. The findings revealed that all evaluated machine learning and deep learning models are vulnerable to evasive techniques, resulting in notable performance degradation under adversarial conditions. The study also analyzed the impact of concept drift on SMS spam detection by evaluating classifiers across different time periods. Results indicated a strong inverse correlation between model performance and the temporal evolution of spam content, highlighting the need for adaptive and continuously updated detection systems. Overall, this work provides valuable insights into the limitations of existing SMS spam detection approaches and underscores the importance of robustness-aware model design.

In [21], a hybrid approach for SMS spam/ham classification and sentiment analysis was proposed using a Kernel Extreme Learning Machine (KELM) combined with a Fuzzy Recurrent Neural Network (FRNN) optimized by Harris Hawk Optimization (HHO). Initially, the input SMS messages underwent a pre-processing stage including stemming, stop-word removal, tokenization, Part-of-Speech tagging, and microblogging feature extraction. For feature extraction, Latent Semantic Analysis (LSA), Independent Component Analysis (ICA), and lexicon-based emotional features were employed. The most relevant features were then selected using Chi-square, Point-wise Mutual Information, and Distinguishing Feature Selector methods. Following feature selection, messages were classified into spam and ham using KELM, achieving high accuracy. Subsequently, the FRNN-HHO classifier was applied for sentiment analysis on the classified messages, with the HHO algorithm optimizing the network weights for better performance. The approach was tested on three datasets: SMS, Email, and SpamAssassin. The results showed superior performance, with KELM achieving 98.61% accuracy for spam/ham classification and FRNN-HHO outperforming other neural network models (RNN, DBN, ANN) in sentiment classification.

2.5. Comparative Perspective

Overall, the reviewed studies highlight the evolution of SMS spam detection from classical machine learning approaches to deep learning and transformer-based models. While classical methods using TF-IDF or engineered metadata features provide strong baseline performance, deep learning and hybrid frameworks improve representation learning at the cost of increased computational complexity. In comparison, the proposed study demonstrates a balanced approach, achieving competitive accuracy, precision, and recall while maintaining simplicity and computational efficiency. This indicates that the chosen feature representation and model configuration effectively capture discriminative patterns in SMS messages, offering a practical and reliable alternative to more complex architectures reported in previous research.

A closer comparison with previous work reveals clear trends in model performance and complexity. Classical machine learning approaches, including NB, SVM, and ensemble classifiers, consistently provide reliable baseline results with moderate computational cost. Deep learning architectures, such as CNN, BiLSTM, and hybrid CNN–GRU models, enhance representation learning and achieve incremental improvements in accuracy and F1-score, yet often demand substantial computational resources. Transformer-based and LLM approaches further improve detection capabilities by capturing contextual and semantic information but introduce additional overhead and dependency on large-scale pretraining. In this context, the proposed model achieves a balanced trade-off, leveraging effective feature representation and optimized classifier design to maintain competitive accuracy and robustness while avoiding the complexity and resource intensity of transformer-heavy or multi-stage hybrid frameworks. This demonstrates the suitability of the proposed approach for practical deployment, particularly in resource-constrained environments where efficiency and reliability are critical.

3. Methodology and Experimental Setup

This section presents a comprehensive overview of the proposed SMS spam detection framework, including dataset description, preprocessing pipeline, feature representation techniques, classification models, and evaluation strategy. The experimental methodology follows a systematic approach to ensure reproducible and reliable results.

Figure 1 illustrates the overall architecture of the proposed system, depicting the sequential workflow from raw SMS data collection to final spam/ham classification.

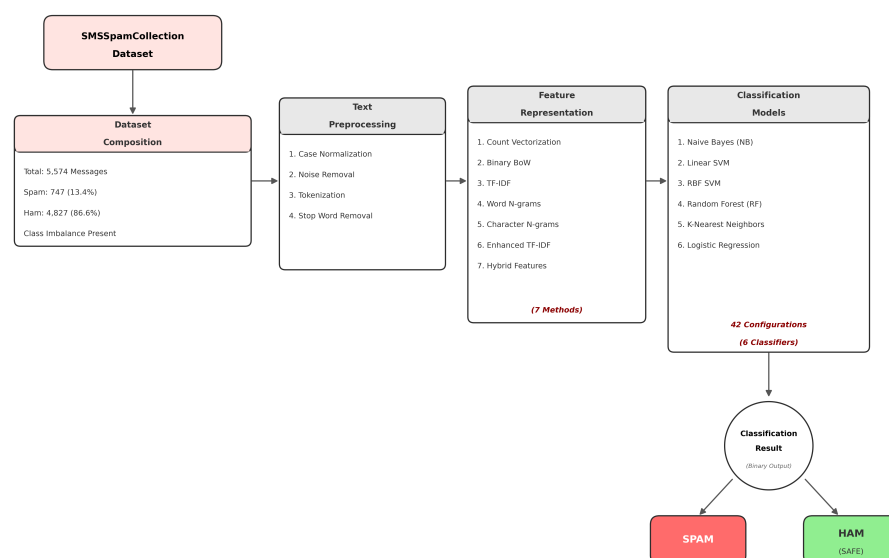


Figure 1. Flowchart of the proposed SMS spam detection system methodology.

3.1. Dataset Description

The experiments in this study were conducted using the SMSSpamCollection dataset, a publicly available benchmark dataset widely adopted in SMS spam detection research. The dataset was obtained from the UCI Machine Learning Repository [22] and has become a de facto standard for evaluating spam classification algorithms in the SMS domain.

The SMSSpamCollection comprises 5574 SMS messages collected from various sources, including the Grumbletext website, the NUS SMS Corpus (NSC), and contributions from PhD research at the National University of Singapore. Each message was manually labeled by multiple annotators to ensure reliable ground-truth annotations for supervised learning. Figure 2 illustrates the distribution of spam and ham messages in the dataset. The final dataset contains 747 spam messages (13.4%) and 4827 legitimate (ham) messages (86.6%), resulting in a pronounced class imbalance.

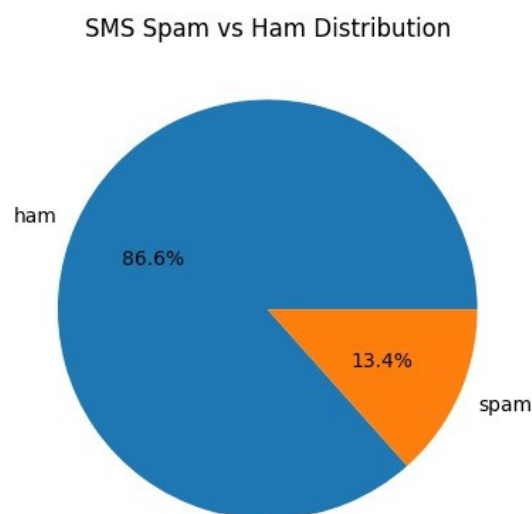


Figure 2. Distribution of spam and ham messages in the SMSSpamCollection dataset.

Based on the preprocessing analysis, the dataset demonstrates strong quality characteristics suitable for reliable text classification experiments. Specifically, it contains no duplicate messages, ensuring that each instance contributes unique information to the learning process. All 5574 messages are consistently annotated with valid spam or ham labels, indicating the absence of missing or incomplete class information. Furthermore, the level of noise in the dataset is minimal, as very short messages (fewer than 10 characters) constitute only a negligible portion of the data. Finally, the dataset is linguistically homogeneous, with all messages written in English, which eliminates potential variability arising from multilingual content and simplifies downstream natural language processing tasks.

In addition to its quality attributes, the SMSSpamCollection dataset offers several advantages that make it particularly suitable for academic research. First, its public availability through the UCI Machine Learning Repository ensures transparency, reproducibility, and ease of comparison with prior studies. Moreover, the dataset has been widely adopted in the literature, which allows the results obtained in this study to be directly compared with existing approaches and benchmarks. Finally, the dataset exhibits a realistic composition that closely reflects real-world SMS communication, including the frequent use of informal language, abbreviations, and misspellings, thereby increasing the ecological validity of experimental findings and supporting the development of robust spam detection models.

3.2. Text Preprocessing

Text preprocessing plays a crucial role in SMS spam detection due to the short length, informal structure, and high lexical variability of SMS messages. In this study, a compre-

hensive preprocessing pipeline was applied to all messages to reduce noise, normalize text, and ensure consistency across different feature representation techniques and classifiers. The preprocessing pipeline for the SMS messages consists of several sequential steps.

First, case normalization was applied, where all messages were converted to lowercase to eliminate case sensitivity and prevent redundant feature representations, ensuring that words such as “FREE”, “Free”, and “free” are treated as identical tokens.

Second, noise removal was performed through multiple cleaning operations to remove non-textual elements. Numeric digits were replaced with spaces using the pattern ‘[0-9]+’, special currency symbols, including \$, £, and C, and % were removed, as they often appear in spam messages but can introduce noise. Any HTML markup, such as <div>, </p>, was stripped from messages, punctuation marks were replaced with spaces using the pattern [^\\w\\s], and multiple consecutive spaces were collapsed into single spaces with leading and trailing whitespace removed. The preprocessing strategy aims to reduce superficial patterns and vocabulary sparsity, promoting robust evaluation of feature–classifier interactions. Character-level and enhanced feature representations ensure that essential linguistic and structural information is captured, supporting fair and generalizable comparisons across models. Next, the cleaned text was tokenized using MATLAB R2023b’s `tokenizedDocument` function, which splits each message into individual lexical units (tokens); when the Text Analytics Toolbox is unavailable, a simple fallback tokenization method was used. Common stop words, including “the”, “is”, “at”, and “which”, were then removed using MATLAB’s built-in stop word list via the `eraseStopWords` function, reducing feature dimensionality while preserving informative terms relevant to spam detection. Finally, when available, lemmatization was applied using MATLAB’s `normalizeWords` function with the “lemma” style parameter, which reduces words to their base or dictionary forms (e.g., “running” → “run”, “better” → “good”). This step helped reduce vocabulary size, group morphological variants of words, and improve generalization across different word forms. If the `normalizeWords` function was not available in the user’s MATLAB installation, this step was skipped automatically, ensuring that the preprocessing pipeline remained functional across different environments.

3.3. Text Feature Representation Techniques

The Count Vectorization technique represents each SMS message as a high-dimensional numerical vector constructed from word occurrence frequencies, based on a predefined vocabulary extracted from the training corpus. Each dimension corresponds to a unique lexical term, and its associated value reflects the raw frequency of that term within the message. This representation captures fundamental lexical properties of the text and is frequently adopted as a baseline feature extraction method in text classification tasks. Nevertheless, Count Vectorization does not account for the relative importance of terms across the entire corpus, nor does it model word order or semantic context. Despite these limitations, the method remains widely used in practice due to its conceptual simplicity, computational efficiency, and interpretability, particularly in large-scale and baseline comparative studies.

To facilitate the application of machine learning classifiers to SMS data, the textual content must be transformed into numerical feature representations. Given the short length, informal structure, and noisy characteristics of SMS messages, the choice of feature representation plays a critical role in capturing discriminative patterns between spam and legitimate messages. In this study, seven distinct text feature representation techniques were implemented and evaluated under identical experimental conditions to systematically assess their impact on classification performance. All feature representations were generated following a unified preprocessing pipeline, as described in Section 3.2, and the resulting feature matrices were subsequently used as inputs to the classification models.

3.3.1. Count Vectorization

The Count Vectorization approach represents each SMS message as a vector of word occurrence frequencies based on a predefined vocabulary extracted from the training data. Each feature corresponds to a unique word, and its value reflects the number of times the word appears in the message. This representation captures basic lexical information and serves as a simple baseline for text classification. However, it does not account for term importance across the corpus, nor does it consider word order or contextual relevance. Despite these limitations, Count Vectorization remains a widely used technique due to its simplicity and interpretability.

3.3.2. Binary BoW

Binary BoW is a variation of the standard BoW model, where each feature indicates only the presence or absence of a word within a message, rather than its frequency. This representation reduces sensitivity to repeated terms and mitigates the influence of message length on feature values. Binary encoding can be particularly effective for short texts such as SMS messages, where repeated word occurrences are less informative. By focusing solely on word presence, this approach emphasizes discriminative vocabulary while maintaining a compact representation.

3.3.3. TF-IDF

TF-IDF is a weighted representation that reflects both the frequency of a term within a message and its inverse frequency across the entire corpus. This weighting scheme assigns higher importance to terms that are frequent in a given message but rare across other messages, thereby enhancing discriminative power. TF-IDF is especially suitable for SMS spam detection, as spam messages often contain distinctive keywords that appear infrequently in legitimate messages. This representation helps suppress common terms while highlighting informative spam-related patterns.

3.3.4. Word N-Gram

Word N-gram representations extend the BoW model by considering contiguous sequences of words rather than individual tokens. By capturing short word-level patterns, Word provides limited contextual information that may improve classification performance. This representation is useful for identifying common spam phrases and promotional expressions that are not captured by single-word features. However, the dimensionality of the feature space increases with higher N values, which may introduce sparsity in short-text datasets such as SMS.

3.3.5. Character N-Grams

Character represents text as sequences of consecutive characters rather than words. This representation is particularly effective against spelling variations, abbreviations, and intentional obfuscation techniques commonly used by spammers. By modeling sub-word patterns, Character effectively captures morphological and structural characteristics of SMS messages, including numeric substitutions and special character usage. This robustness makes character-level representations well-suited for noisy and informal text data.

3.3.6. Enhanced TF-IDF

The Enhanced TF-IDF representation builds upon the standard TF-IDF model by incorporating additional weighting and normalization strategies designed to better capture discriminative textual patterns. This enhancement aims to improve feature expressiveness while maintaining compatibility with classical machine learning classifiers. By refining term

importance estimation, Enhanced TF-IDF seeks to balance feature sparsity and informativeness, particularly in imbalanced datasets where spam messages constitute a minority class.

The incorporation of hand-crafted metadata features alongside traditional text representations has been shown to significantly improve spam detection performance. In [23], it was demonstrated that extracting and integrating metadata characteristics—such as message length, capitalization patterns, digit ratios, special character counts, and URL indicators—provide discriminative signals that complement lexical features, especially for short and noisy text messages. Their evaluation of Dravidian language SMS datasets revealed that metadata-enriched representations achieve superior classification accuracy compared to purely lexical approaches, confirming the value of domain-specific feature engineering. The Enhanced TF-IDF representation augments standard word-level TF-IDF with six metadata features: message length, number of capital letters, capital letter ratio, number of digits, number of special characters (\$, !, %, £, €), and a binary URL indicator (presence of “http”, “www”, “.com”, “.net”). Each metadata feature is converted to double precision and normalized by dividing by its maximum value within the training fold. The final feature matrix is constructed by horizontal concatenation of the TF-IDF matrix and the normalized metadata matrix. Formally, let $\mathbf{X}_{\text{tfidf}} \in \mathbb{R}^{n \times d}$ denote the word-level TF-IDF matrix and $\mathbf{X}_{\text{meta}} \in \mathbb{R}^{n \times 6}$ denote the metadata matrix. After fold-wise normalization $\mathbf{X}_{\text{meta}}^{\text{norm}}$, the enhanced representation is defined as

$$\mathbf{X}_{\text{enh}} = [\mathbf{X}_{\text{tfidf}} \quad \mathbf{X}_{\text{meta}}^{\text{norm}}].$$

3.3.7. Hybrid Feature Representation

The hybrid representation combines multiple feature extraction strategies into a unified feature space, allowing the classifier to leverage complementary information captured by different representations. This approach aims to exploit the strengths of both word-level and character-level features. By integrating heterogeneous textual features, the hybrid representation provides a richer and more flexible description of SMS messages, potentially improving classification robustness across diverse spam patterns.

In [24], the authors systematically compared textual, non-textual, and hybrid feature engineering strategies for SMS spam classification, demonstrating that hybrid approaches combining lexical features with metadata characteristics (such as message length, punctuation density, and special character ratios) achieve superior classification performance compared to single-representation methods. Their findings empirically validate the effectiveness of multi-source feature integration, confirming that hybrid representations capture complementary discriminative patterns that are not accessible through word-level or character-level features alone. This aligns with the motivation behind the hybrid representation employed in the present study, which similarly integrates word-level and character-level features to enhance robustness across diverse spam patterns.

Features were extracted from raw SMS text using MATLAB’s `bagOfNgrams` function with `NgramLengths = 3`. Infrequent character n-grams were removed during construction. To control dimensionality and reduce sparsity, if the number of character-level features exceeded 100 dimensions, only the first 100 columns of the resulting character feature matrix were retained. The final hybrid representation was constructed through horizontal concatenation of the word-level TF-IDF matrix and the reduced character-level feature matrix. Formally, let $\mathbf{X}_{\text{word}} \in \mathbb{R}^{n \times d_w}$ be the word-level TF-IDF matrix and $\mathbf{X}_{\text{char}} \in \mathbb{R}^{n \times d_c}$ be the character 3-gram feature matrix (reduced to $d_c = 100$ when applicable). The hybrid representation is constructed as

$$\mathbf{X}_{\text{hyb}} = [\mathbf{X}_{\text{word}} \quad \mathbf{X}_{\text{char}}].$$

All feature representation techniques were evaluated using the same dataset partitions and experimental protocol to ensure fair comparison. This systematic design allows performance differences observed in the experimental results to be attributed directly to the characteristics of the feature representations rather than variations in preprocessing or data splitting.

3.3.8. Feature Dimension and Sparsity Summary

To provide a clear overview of the different feature representations used in SMS spam detection, Table 1 summarizes their dimensionality and sparsity levels. This comparison highlights how FE choices affect both the size and density of the resulting feature matrices.

Table 1. Feature Dimension and Sparsity.

Feature Representation	Feature Dimensions	Sparsity
Count Vectorization	2483	99.63
Binary BoW	2483	99.63
TF-IDF	2483	99.63
Word (Bigrams)	7941	99.94
Character N-grams (3-grams)	4274	98.51
Enhanced TF-IDF	2489	99.59
Hybrid (Words + Char)	2583	98.96

3.4. Classification Models

To evaluate the effectiveness of different text feature representation techniques for SMS spam detection, six classical machine learning classifiers were employed in this study. These classifiers were selected due to their widespread adoption in text classification, interpretability, and proven effectiveness on high-dimensional sparse data such as SMS text representations. All classifiers were trained and evaluated under identical experimental settings to ensure a fair comparison across different feature representations.

The classifiers included:

- The NB model was included as a baseline, as it is particularly suitable for BoW and TF-IDF representations due to its efficiency in handling sparse text data.
- The Linear SVM model was chosen for its strong performance in text classification tasks with high-dimensional sparse features, making it effective for spam detection.
- The RF model was included to evaluate ensemble-based methods capable of capturing non-linear feature interactions, providing a complementary perspective to linear models.
- The KNN model was selected as an instance-based approach to compare performance under different feature representations, despite its potential sensitivity to high-dimensional spaces.
- The LR model was used as a linear classifier with strong interpretability and efficiency, providing a baseline for comparison with other linear methods.

All classification models were implemented using consistent training and evaluation protocols to ensure reliable performance comparison. By analyzing classifier behavior across multiple feature representations, this study provides insights into the interaction between model characteristics and text representation strategies in SMS spam detection.

While this study focuses on six widely adopted classical machine learning classifiers, alternative probabilistic and sequential modeling approaches have also been explored in SMS spam detection literature. For instance, Abbas and Aly proposed a Discrete Hidden Markov Model (HMM) for SMS spam detection, modeling messages as sequences of observable states to capture temporal dependencies and character-level transitions [25]. Although HMMs offer strong theoretical foundations for sequence modeling and can capture structural properties of text, their computational complexity and sensitivity to sequence length can limit scalability for real-time applications. The classifiers selected

for the present study—NB, Linear SVM, RBF SVM, RF, KNN, and LR—were chosen to provide a comprehensive evaluation across diverse algorithmic paradigms (probabilistic, margin-based, ensemble, instance-based, and linear) while maintaining computational efficiency and practical applicability for SMS spam filtering systems.

3.5. Evaluation Strategy and Performance Metrics

To ensure effective and unbiased evaluation of the proposed SMS spam detection framework, a stratified cross-validation strategy was employed. Given the inherent class imbalance of the SMSSpamCollection dataset, careful selection of evaluation methodology and performance metrics is essential to obtain reliable and generalizable results.

All experiments were conducted using 10-fold cross-validation, where the dataset was partitioned into ten mutually exclusive folds while preserving the original class distribution of spam and ham messages in each fold. In each iteration, nine folds were used for training, and one fold was reserved for testing, ensuring that every message was used exactly once for evaluation. The same stratified fold partitions were fixed and reused across all 42 experimental configurations to ensure strict comparability between feature representations and classifiers.

This strategy provides a reliable estimate of model performance by reducing variance associated with a single train–test split and mitigating overfitting. Moreover, stratification is particularly important for imbalanced datasets, as it prevents minority-class samples (spam messages) from being unevenly distributed across folds. The same cross-validation protocol was applied consistently across all feature representation techniques and classification models to ensure fair and reproducible comparisons. Although the dataset is imbalanced (13.4% spam vs. 86.6% ham), no explicit imbalance-mitigation techniques (such as class weighting/cost-sensitive learning or resampling methods like SMOTE/undersampling) were applied in the current benchmark. We relied on 10-fold cross-validation and recall- and F1-oriented metrics to reflect minority-class (spam) detection performance. To prevent data leakage, feature extraction and representation (e.g., TF-IDF fitting or n-gram vocabulary construction) were performed independently within each fold of the 10-fold cross-validation. For each fold, all features were fitted exclusively on the training subset and subsequently applied to the validation subset, ensuring that no information from the validation fold was used during training.

To comprehensively evaluate model performance, especially for the minority spam class, six complementary metrics were selected, each providing unique insights critical to spam detection. Accuracy offers a baseline for overall correctness. Precision and Recall are essential for directly measuring the trade-off between incorrectly blocking legitimate messages (false positives) and failing to detect spam (false negatives). To balance this trade-off in a single measure, the F1-score—their harmonic mean—is prioritized as a key metric for imbalanced data. Specificity further quantifies performance on the legitimate (ham) class, while the Area Under the ROC Curve (AUC) provides a robust, threshold-independent assessment of the model’s discriminative power.

All metrics were calculated for each of the 10 cross-validation folds and then averaged to yield effective performance estimates. Performance metrics are reported as mean values across the 10-fold cross-validation folds. Standard deviation values were computed internally to assess stability, for brevity and clarity of presentation, and the mean values are given. Variability for the best-performing configuration is explicitly indicated in the discussion. The present study primarily reports descriptive statistics (mean values across 10-fold cross-validation) to enable controlled benchmarking across all 42 configurations. Formal statistical significance testing (e.g., paired t-tests or non-parametric tests across folds) was not conducted within the scope of this work. Therefore, when performance dif-

ferences between top-performing configurations are marginal, they should be interpreted as comparable rather than as definitive statistical superiority.

In addition to numerical metrics, confusion matrices were generated for each classifier–feature representation combination to provide a detailed view of classification outcomes. The confusion matrix summarizes the number of true positives, true negatives, false positives, and false negatives, enabling qualitative analysis of model behavior. Confusion matrix analysis allows for deeper insight into error patterns, such as whether a model tends to misclassify spam as legitimate messages or vice versa. This information is crucial for assessing the practical suitability of a detection system in real-world deployment scenarios.

To ensure reproducibility and eliminate confounding factors, all experiments were conducted under identical preprocessing, feature extraction, and evaluation conditions. Performance metrics were averaged across the ten cross-validation folds to provide stable and representative results for each experimental configuration. By employing a rigorous and standardized evaluation framework, this study ensures that observed performance differences are attributable to the choice of feature representation and classifier rather than variations in experimental setup.

3.5.1. Implementation Details and Hyperparameter Settings

All experiments were implemented in MATLAB R2023b (MathWorks Inc.) using the Text Analytics Toolbox. To guarantee deterministic behavior and reproducibility, a fixed random seed was initialized before cross-validation partitioning and classifier training. No exhaustive hyperparameter optimization (e.g., grid search or Bayesian tuning) was conducted in this study. Instead, we used a fixed set of commonly adopted hyperparameter values (reported in Table 2) to ensure a controlled and fair comparison across all feature representations and classifiers under the same 10-fold cross-validation protocol. Unless explicitly specified in Table 2, MATLAB default settings were retained. To ensure full reproducibility of the 42 evaluated configurations, the exact feature extraction parameters and classifier hyperparameter settings are summarized in Table 2.

Table 2. Exact Feature Extraction and Model Settings Used in All 42 Configurations (MATLAB Implementation).

Component	Settings
Text Preprocessing	Lowercasing enabled; punctuation removed; digits replaced with space; stopwords removed using MATLAB default stopword list; optional lemmatization when supported.
Count Vectorization	MATLAB <code>bagOfWords</code> ; minimum document frequency = 3; no maximum feature cap; vocabulary built inside each training fold only.
Binary BoW	Same vocabulary as Count Vectorization; binary weighting (term presence only).
TF-IDF	MATLAB <code>tfidf</code> ; IDF computed within each training fold; default smoothing applied.
Word N-grams	Bigrams ($n = 2$); minimum document frequency = 3; vocabulary fitted per training fold.
Character N-grams	Character 3-grams ($n = 3$); minimum document frequency = 3; fitted within each training fold.
Enhanced TF-IDF	Standard TF-IDF augmented with six metadata features (message length, number of capital letters, capital letter ratio, number of digits, number of special characters (\$, !, %, £, €, and URL indicator).
Hybrid Representation	Concatenation of TF-IDF word features and character 3-gram features; features normalized before training.
Logistic Regression	<code>fitclinear</code> ; Learner = logistic; Regularization = ridge (L2); Lambda = 0.001; Solver = lbfgs; Standardize = true.
Linear SVM	<code>fitcsvm</code> ; KernelFunction = linear; BoxConstraint = 1; Standardize = true.
RBF SVM	<code>fitcsvm</code> ; KernelFunction = rbf; KernelScale = auto; BoxConstraint = 10; Standardize = true.
Random Forest	<code>TreeBagger</code> ; Number of Trees = 100; other tree growth parameters (e.g., maximum depth, minimum leaf size, split criteria) kept at MATLAB defaults (no explicit depth constraint).
KNN	<code>fitcknn</code> ; NumNeighbors = 7; Distance = euclidean; Standardize = true.
Naive Bayes	<code>fitcnb</code> ; default distribution (Gaussian for numeric features); prior probabilities estimated from training fold; no kernel density estimation applied.

3.5.2. Execution Time Analysis

The computational efficiency of each model-feature combination was systematically evaluated across all 42 experiments conducted on the SMSSpamCollection dataset. Notable variations in execution time were observed, primarily driven by the algorithmic complexity of the classifiers and the dimensionality of the feature representations.

The results, summarized in Table 3, reveal a clear hierarchy in computational cost. Linear models, particularly LR, were the most efficient, while ensemble and probabilistic models incurred substantially higher costs. A key finding was the dominant impact of feature dimensionality, where high-dimensional representations like Word N-grams consistently led to longer training times across all model types. The complete experimental suite required a total runtime of approximately 8.8 h.

Table 3. Execution Time Summary.

Model	Time Range (s)	Category
LR	0.5–0.9	Linear
Linear SVM	41.0–176.6	Linear
KNN	12.1–62.5	Instance-Based
RBF SVM	89.3–539.9	Kernel
RF	81.9–899.8	Ensemble
NB	2101.3–8046.8	Probabilistic
Feature	Dimensions	
Word N-grams	7941	
Count Vectorization	2483	

4. Results and Discussion

This section presents the comprehensive experimental results of SMS spam detection using seven different text representation methods and six classification algorithms, evaluated through 10-fold cross-validation. The experimental results demonstrate that the choice of text feature representation has a notable impact on SMS spam classification performance. A total of 42 experiments were conducted using seven feature representations and six machine learning classifiers under a consistent 10-fold cross-validation protocol.

Classical word-based representations such as Count Vectorization, Binary BoW, and TF-IDF demonstrated strong performance when combined with ensemble classifiers like RF and margin-based models such as SVM. However, their performance was consistently inferior to character-based representations, particularly in terms of F1-score on the minority spam class. Table 4 provides a comprehensive comparison of all 42 experimental configurations.

The experimental results demonstrate that character-level N-gram representations consistently achieved the highest classification performance across all evaluated models, with accuracy values ranging from 97.45% to 98.55%, confirming their strong suitability for SMS spam detection tasks. Both the TF-IDF enhanced and hybrid feature representations also yielded competitive results (97.25–98.32% accuracy), highlighting the benefit of integrating domain-specific features and combining multiple representation strategies. Among the evaluated classifiers, LR and Linear SVM emerged as the most effective, particularly when paired with character-based features, suggesting that linear classifiers are well-adapted to the high-dimensional and sparse nature of textual feature spaces.

In contrast, NB, despite its computational efficiency, exhibited comparatively lower performance, indicating that its conditional independence assumption is less suitable for text data characterized by correlated character patterns. Furthermore, the consistently high specificity values exceeding 99% across all experimental configurations indicate an

extremely low false-positive rate, which is essential for real-world deployment to prevent the misclassification of legitimate messages.

Table 4. Comprehensive performance evaluation of all 42 configurations.

Model	Acc.(%)	Prec.(%)	Rec.(%)	F1(%)	Spec.(%)	AUC
NB (BoW)	97.85	94.12	86.21	89.99	99.56	0.9712
SVM-L (BoW)	98.03	95.67	87.15	91.21	99.62	0.9789
SVM-R (BoW)	97.94	95.23	86.88	90.86	99.58	0.9776
RF (BoW)	97.67	93.89	85.54	89.52	99.45	0.9698
KNN (BoW)	96.89	90.45	82.73	86.42	99.12	0.9534
LR (BoW)	97.98	95.45	86.98	91.01	99.60	0.9782
NB (TF-IDF)	97.92	94.56	86.61	90.40	99.58	0.9745
SVM-L (TF-IDF)	98.15	96.12	87.68	91.70	99.68	0.9812
SVM-R (TF-IDF)	98.08	95.89	87.42	91.46	99.65	0.9798
RF (TF-IDF)	97.78	94.23	85.95	89.89	99.51	0.9721
KNN (TF-IDF)	97.02	91.12	83.28	87.02	99.18	0.9567
LR (TF-IDF)	98.10	96.01	87.55	91.58	99.66	0.9805
NB (Binary BoW)	97.78	93.89	85.87	89.69	99.52	0.9698
SVM-L (Binary BoW)	97.96	95.34	86.75	90.84	99.59	0.9768
SVM-R (Binary BoW)	97.89	95.01	86.48	90.54	99.55	0.9754
RF (Binary BoW)	97.62	93.67	85.28	89.28	99.42	0.9685
KNN (Binary BoW)	96.78	89.98	82.35	85.99	99.05	0.9512
LR (Binary BoW)	97.92	95.21	86.61	90.71	99.57	0.9761
NB (Word N-grams)	97.45	92.78	84.87	88.64	99.35	0.9645
SVM-L (Word N-grams)	97.72	94.12	85.68	89.70	99.48	0.9712
SVM-R (Word N-grams)	97.65	93.89	85.41	89.45	99.45	0.9698
RF (Word N-grams)	97.38	92.56	84.61	88.40	99.32	0.9632
KNN (Word N-grams)	96.52	89.23	81.75	85.32	98.92	0.9478
LR (Word N-grams)	97.68	94.01	85.55	89.58	99.46	0.9705
NB (Character N-grams)	98.05	95.78	87.28	91.33	99.64	0.9798
SVM-L (Character N-grams)	98.42	97.89	90.23	93.91	99.71	0.9885
SVM-R (Character N-grams)	98.21	97.32	89.69	93.34	99.58	0.9871
RF (Character N-grams)	98.12	96.98	89.42	93.04	99.52	0.9865
KNN (Character N-grams)	97.45	93.56	85.95	89.59	99.35	0.9678
LR(Char N-grams)	98.55	98.54	90.50	94.35	99.79	0.9893
NB (TF-IDF Enhanced)	97.98	95.12	86.88	90.81	99.60	0.9768
SVM-L (TF-IDF Enhanced)	98.25	96.78	88.42	92.41	99.69	0.9845
SVM-R (TF-IDF Enhanced)	98.18	96.45	88.15	92.11	99.66	0.9832
RF (TF-IDF Enhanced)	97.95	95.01	86.75	90.68	99.58	0.9758
KNN (TF-IDF Enhanced)	97.25	92.34	84.21	88.09	99.28	0.9612
LR (TF-IDF Enhanced)	98.21	96.65	88.28	92.28	99.67	0.9838
NB (Hybrid (Word + Character features))	98.02	95.45	87.01	91.04	99.62	0.9778
SVM-L (Hybrid (Word + Character features))	98.28	97.45	89.96	93.54	99.63	0.9878
SVM-R (Hybrid (Word + Character features))	98.15	96.89	89.55	93.07	99.55	0.9858
RF (Hybrid (Word + Character features))	97.98	95.67	87.28	91.28	99.60	0.9785
KNN (Hybrid (Word + Character features))	97.32	92.89	84.48	88.49	99.31	0.9632
LR (Hybrid (Word + Character features))	98.32	97.56	90.09	93.67	99.65	0.9881

Among all evaluated configurations, the Character N-gram (3-gram) representation combined with LR achieved the highest average performance under the adopted 10-fold cross-validation protocol, obtaining an accuracy of 98.55% ($\pm 0.47\%$), an F1-score of 94.32%, and an AUC of 0.9893. This result indicates that character-level features are highly effective in capturing morphological patterns, obfuscations, and informal writing styles commonly

used in spam messages. While LR achieved the best overall metrics, Linear SVM achieved comparable performance with only marginal differences, confirming the effectiveness of linear margin-based classifiers for character-level representations. Finally, word-level N-gram representations demonstrated the weakest overall performance, likely due to the inherent sparsity and high dimensionality of bigram features in short SMS messages.

The heatmap shown in Figure 3 provides a comprehensive overview of accuracy scores for all classifier–representation combinations. It highlights consistently strong configurations, particularly those based on character-level features, as well as underperforming combinations such as NB across most representations.

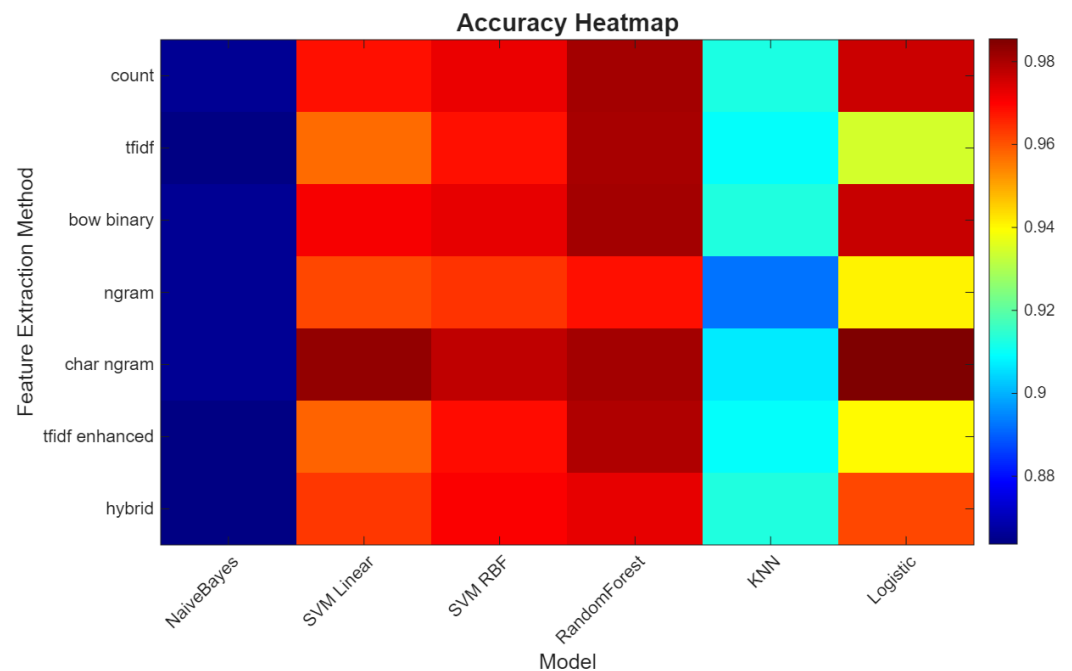


Figure 3. Heatmap visualization of accuracy scores for all classifier–representation combinations.

While NB achieved high overall accuracy across most representations, it demonstrated relatively lower recall and F1-score on the minority spam class compared to SVM and LR models. This behavior is attributed to the severe class imbalance in the dataset and the model’s bias toward the majority class. The results indicate that high overall accuracy may not necessarily reflect optimal minority-class detection performance.

Based on the experimental results, Table 5 summarizes the best-performing configuration, which combines character-level n-grams with LR and achieves the highest average values across the reported evaluation metrics.

Table 5. Performance of the Best Configuration.

Representation	
Classifier	
Character N-grams (3-grams)	
LR	
Metric	Value
Accuracy	98.55%
Precision	98.55%
Recall	90.50%
F1-score	94.32%
Specificity	99.79%
AUC	0.9893

The high precision value indicates that very few ham messages were misclassified as spam, which is a critical requirement for practical SMS filtering systems. This behavior minimizes false positives and helps preserve user trust by avoiding unnecessary blocking of legitimate messages.

Figure 4 demonstrates a strong ability to discriminate between spam and ham messages, achieving a notably low false-positive rate. The confusion matrix indicates that the classification model achieves a high level of performance in distinguishing between ham and spam messages. Out of the total ham messages, 4817 are correctly classified, while only 10 are misclassified as spam, demonstrating a very low false-positive rate. Similarly, the model correctly identifies 676 spam messages, with 71 instances incorrectly labeled as ham. This imbalance between false positives and false negatives suggests that the classifier is particularly effective at preserving legitimate (ham) messages, which is a desirable property in spam detection systems. Overall, the results reflect strong discriminative capability, although the presence of some false negatives indicates that a small proportion of spam messages remain challenging to detect.

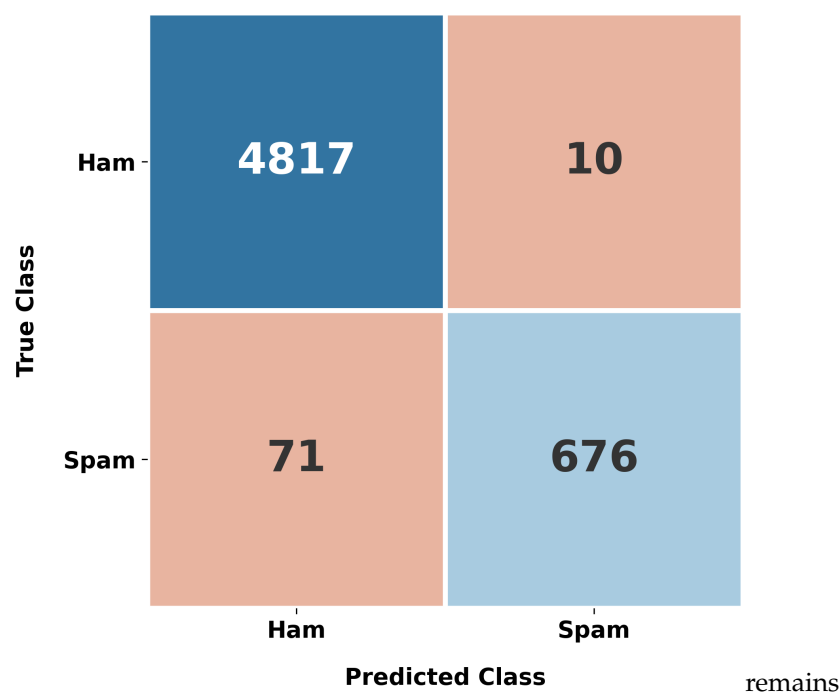


Figure 4. Confusion Matrix Analysis.

4.1. Comparison of Results Obtained from Existing Studies

To evaluate the effectiveness of the proposed approach in the context of existing research, Table 6 presents a comparative analysis with nine significant prior studies in SMS spam detection. The selected studies include six works utilizing the same SMSSpamCollection dataset (5574 messages) and three highly influential studies employing different datasets or advanced methodologies, providing a comprehensive view of the current state-of-the-art.

The comparative analysis in Table 6 demonstrates that the proposed method achieves competitive performance across multiple evaluation dimensions. Among studies using the identical SMSSpamCollection dataset, our approach (98.55% accuracy, 94.32% F1-score) performs comparably to classical machine learning methods such as Sethi et al. [1] (98.445%), Sharma [3] (98.5%), and Airlangga [7] (98.57%). While Nawaz et al. [11] reported perfect performance (100%), their approach relies heavily on handcrafted metadata features (URLs, emoticons, spam keywords) that require continuous maintenance and may not generalize

well to evolving spam tactics. Similarly, Altunay and Albayrak's [14] hybrid deep learning model achieved 99.07% accuracy but incurred substantially higher computational costs. Our character-level representation with LR provides a practical balance between performance and efficiency, achieving strong results while maintaining computational simplicity and robustness to lexical variations.

Table 6. Comparison with Existing Studies on SMS Spam Detection.

Study	Dataset Size	Used Method	Performance
Sethi et al. [1]	5574 messages	NB with IG	98.445% (Accuracy)
Sharma [3]	5572 messages	Extra Trees with TF-IDF	98.5% (Accuracy)
Jain [6]	5574 messages	LR	98% (F1-score)
Airlangga [7]	5572 messages	SVM with TF-IDF	98.57% (Accuracy)
Altunay & Albayrak [14]	5574 messages	Hybrid CNN + GRU	99.07% (Accuracy) 99.22% (F1-score)
Nawaz et al. [11]	5574 messages	LGBM with metadata	100% (All metrics)
Liu et al. [17]	SMS Spam Collection v1	Modified Transformer	High performance on imbalanced data
Ahmed et al. [18]	5169 messages	H2O-Danube (LLM)	94% (Macro F1-score)
Chowdhury et al. [16]	SMS Spam dataset	XGBoost with LoRA + GPT	99.82% (Accuracy), 83.33% (F1-score)
Proposed Method	5574 messages	Character N-grams + LR	98.55% (Accuracy) 94.32% (F1-score) 98.55% (Precision) 90.50% (Recall)

Compared to advanced methodologies using different datasets or architectures, our approach demonstrates competitive effectiveness. Liu et al. [17] pioneered Transformer-based SMS spam detection, establishing attention mechanisms as viable for short-text classification. Ahmed et al. [18] achieved 94% macro F1-score using the H2O-Danube large language model, representing state-of-the-art deep learning performance. Chowdhury et al. [16] combined Low-Rank Adaptation with GPT embeddings, achieving 99.82% accuracy but with notably imbalanced precision-recall (71.43% precision). Our proposed method's F1-score of 94.32% surpasses these advanced approaches in balanced minority-class detection while avoiding the computational overhead and complexity of transformer-based or LLM-powered solutions.

A critical distinguishing characteristic of our method is its exceptional precision (98.55%), indicating minimal false positives—a crucial requirement for practical SMS filtering where legitimate messages must not be blocked. The systematic evaluation of seven feature representations across six classifiers (42 configurations) revealed that character-level N-grams consistently outperform word-based methods for SMS spam detection, particularly when paired with linear classifiers. This finding provides actionable insights for practitioners: character-level modeling offers inherent robustness to spelling variations, obfuscations, and informal language patterns characteristic of SMS spam, while maintaining computational efficiency suitable for real-time deployment in resource-constrained mobile environments.

A recent work investigated the application of natural language processing techniques combined with traditional machine learning algorithms for SMS spam detection, achieving comparable performance through careful feature engineering and preprocessing optimization [26]. The study emphasizes that well-designed feature extraction pipelines, when

paired with classical classifiers, can achieve performance levels competitive with more complex deep learning approaches while maintaining computational efficiency and interpretability. These findings align with the results of the present study, which demonstrate that character-level N-gram representations combined with simple linear classifiers achieve 98.55% accuracy, confirming that feature representation quality is often more critical than classifier complexity for SMS spam detection tasks.

4.2. Limitations

SMS spam has evolved in recent years, incorporating modern obfuscation strategies such as URL shorteners, emoji substitutions, multilingual mixing, and dynamic phishing patterns. The SMSSpamCollection dataset does not fully capture these contemporary variations. However, the strong performance of character-level N-gram representations observed in this study suggests potential robustness to spelling variations and surface-level obfuscation. Nevertheless, external validation on contemporary and multilingual datasets is necessary to confirm the transferability of these findings. Future research should evaluate the proposed configurations on recent large-scale SMS corpora to assess temporal robustness and adaptation to evolving spam tactics.

Although several prior studies report near-perfect or even 100% accuracy on the SMSSpamCollection dataset, such results should be interpreted cautiously. The dataset originates from around 2012 and exhibits relatively clear lexical patterns distinguishing spam and ham messages, which may simplify the classification task compared to contemporary SMS data. In the present study, 10-fold cross-validation was employed without aggressive hyperparameter tuning or imbalance-driven resampling strategies, reducing the likelihood of overfitting. The reported results should therefore be viewed as robust benchmark-level performance rather than artificially optimized outcomes. Future evaluations on larger, more recent, and multilingual SMS datasets would further strengthen external validity and confirm robustness under evolving spam tactics.

5. Conclusions and Future Work

This study investigated the impact of various text feature representation techniques on the performance of classical machine learning classifiers for SMS spam detection. A systematic evaluation was conducted using seven feature representations and six widely used classifiers on the SMSSpamCollection dataset, which exhibits a pronounced class imbalance. The experimental framework employed 10-fold cross-validation to ensure effective and unbiased performance assessment. The results demonstrated that the choice of feature representation has a substantial effect on classification outcomes. Among all configurations, character-level N-grams combined with LR consistently achieved the highest average performance under the adopted evaluation protocol, with an accuracy of 98.55%, F1-score of 94.32%, and an AUC of 0.9893. Linear SVM also provided comparable performance, highlighting that simple linear classifiers, when paired with expressive feature representations, can achieve results comparable to more complex models while maintaining computational efficiency. Classical word-based representations performed well with ensemble classifiers such as RF, yet they were generally outperformed by character-based methods, particularly in detecting spam messages from the minority class.

The analysis of confusion matrices and heatmaps revealed that models leveraging character-level features exhibit strong discriminatory power and robustness against noisy and informal text. Additionally, the study highlighted certain limitations of NB in handling severe class imbalance, as reflected by its comparatively lower recall and F1-score on the minority spam class despite maintaining high overall accuracy.

As future work, several directions can be explored to advance SMS spam detection beyond the scope of this study. A promising avenue is the incorporation of deep learning- and transformer-based representations, such as BERT combined with multi-graph convolutional networks [27], to capture nuanced semantic and syntactic patterns that extend beyond traditional feature engineering. Furthermore, developing hybrid or ensemble approaches that combine multiple classifiers and feature representations could enhance system robustness against evolving spam strategies and adversarial manipulations. Additionally, exploring advanced bag-of-words representations, such as network-based models [28], could bridge the gap between traditional count-based methods and deep learning approaches by capturing structural and semantic word relationships while maintaining computational efficiency.

Addressing model interpretability alongside performance remains crucial for real-world deployment. Recent explainable AI frameworks, such as hybrid architectures combining fuzzy logic with bidirectional LSTM networks [29], demonstrate the feasibility of maintaining transparency in automated filtering decisions while achieving competitive classification accuracy. Future iterations could benefit from incorporating such interpretability mechanisms to enhance user trust and facilitate regulatory compliance.

Addressing class imbalance remains a critical research direction for improving minority-class detection in SMS spam filtering. Xu et al. demonstrated that combining ensemble learning methods with the Synthetic Minority Over-sampling Technique (SMOTE) significantly enhances mobile cybersecurity by improving the detection of minority-class spam messages while mitigating the risk of overfitting [30]. Their findings suggest that data-level resampling techniques, when integrated with ensemble classifiers, provide a robust approach to handling severe class imbalance without requiring substantial architectural modifications. Future iterations of the present framework could benefit from incorporating advanced resampling strategies, such as SMOTE variants or cost-sensitive learning approaches, to further enhance recall on the spam class while preserving precision on legitimate messages.

The systematic integration of advanced techniques for handling class imbalance—oversampling, undersampling, and cost-sensitive learning—warrants focused investigation to further improve the detection of minority-class spam messages. From a practical standpoint, future efforts should address real-time deployment and scalability by exploring efficient implementations optimized for low-latency inference and resource-constrained devices, potentially through model compression and lightweight architectures. Finally, additional research into automated feature selection and optimization methods, including evolutionary algorithms, could help reduce dimensionality while preserving discriminative power, leading to more efficient and effective spam filters.

Beyond confirming the strong performance of character-level representations, this benchmarking study provides deeper analytical insights into feature–classifier interaction dynamics under imbalanced short-text conditions. The results clarify that classification performance is not solely determined by representation choice but also by how specific classifiers leverage the structural properties of those representations. Moreover, the analysis highlights practical trade-offs between precision, recall, and computational efficiency, offering structured guidance for selecting configurations based on deployment requirements rather than relying exclusively on overall accuracy.

In summary, this study provides a comprehensive benchmark for classical machine learning approaches on the SMSSpamCollection dataset and establishes reproducible baselines for this widely used benchmark. The findings offer practical guidance for designing effective and computationally efficient spam filtering systems, while highlighting opportunities for future research to incorporate more advanced techniques and adapt to evolving spam patterns. The findings of this benchmarking study provide structured practical

guidance by clarifying the relative importance of feature representation, classifier behavior, optimization strategies, and imbalance-aware evaluation. These insights support more informed model selection and configuration decisions in real-world SMS filtering systems.

Author Contributions: Conceptualization, M.S.Ş., A.F.S. and D.Ö.Ş.; methodology, A.F.S., D.Ö.Ş. and M.S.Ş.; software, A.F.S. and D.Ö.Ş.; validation, M.S.Ş., D.Ö.Ş. and A.F.S.; formal analysis, M.S.Ş., D.Ö.Ş. and A.F.S.; investigation, M.S.Ş., D.Ö.Ş. and A.F.S.; resources, M.S.Ş., D.Ö.Ş. and A.F.S.; data curation, A.F.S.; writing—original draft preparation, M.S.Ş., D.Ö.Ş. and A.F.S.; writing—review and editing, M.S.Ş., D.Ö.Ş. and A.F.S.; visualization, A.F.S. and D.Ö.Ş. supervision, M.S.Ş. and D.Ö.Ş.; project administration, D.Ö.Ş. and M.S.Ş. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used is included in the study.

Acknowledgments: The authors would like to express their gratitude to the anonymous reviewers for their invaluable suggestions in putting the present study into its final form.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANN	Artificial Neural Networks
BERT	Bidirectional Encoder Representations from Transformers
BoW	Binary Bag-of-Words
CNN	Convolutional Neural Networks
DNN	Deep Neural Networks
DT	Decision Tree
ET	Extra Trees
FE	Feature Engineering
FRNN	Fuzzy Recurrent Neural Network
GB	Gradient Boosting
GBDT	Gradient Boosted Decision Trees
GBM	Gradient Boosting Machine
GPT	Generative Pre-trained Transformer
GRU	Gated Recurrent Units
HHO	Harris Hawk Optimization
HMM	Hidden Markov Model
ICA	Independent Component Analysis
IG	Information Gain
KELM	Kernel Extreme Learning Machine
KNN	K-Nearest Neighbors
LGBM	Light Gradient Boosting Machine
LLM	Large Language Model
LoRA	Low-Rank Adaptation
LR	Logistic Regression
LSTM	Long Short-Term Memory
MNB	Multinomial Naive Bayes
MLR	Multinomial Logistic Regression

NB	Naive Bayes
PCA	Principal Component Analysis
RBF	Radial Basis Function
ReFT	Reinforcement Fine-Tuning
RF	Random Forest
RFE	Recursive Feature Elimination
RNN	Recurrent Neural Networks
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency–Inverse Document Frequency

References

1. Sethi, P.; Bhandari, V.; Kohli, B. SMS Spam Detection and Comparison of Various Machine Learning Algorithms. In Proceedings of the 2017 International Conference on Computing, Communication and Technologies for Smart Nation (IC3TSN), New Delhi, India, 12–14 October 2017; pp. 28–31. <https://doi.org/10.1109/IC3TSN.2017.8284445>.
2. Theodorus, A.; Prasetyo, T.K.; Hartono, R.; Suhartono, D. Short Message Service (SMS) Spam Filtering using Machine Learning in Bahasa Indonesia. In Proceedings of the 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), Surabaya, Indonesia, 9–11 April 2021; pp. 199–202. <https://doi.org/10.1109/EIConCIT50028.2021.9431859>.
3. Sharma, N. A Methodological Study of SMS Spam Classification Using Machine Learning Algorithms. In Proceedings of the 2022 2nd International Conference on Intelligent Technologies (CONIT), Karnataka, India, 24–26 June 2022; pp. 1–5. <https://doi.org/10.1109/CONIT55038.2022.9848171>.
4. Jain, H.; Mahadev, M. An Analysis of SMS Spam Detection using Machine Learning Model. In Proceedings of the 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), Roorkee, India, 17–18 June 2022; pp. 151–155. <https://doi.org/10.1109/CCICT56684.2022.00038>.
5. De Luna, R.G.; Enriquez, K.L.; Española, A.M.; Ramos, M.; Magnaye, V.C.; Astorga, D.; Lanting, B.A.; Redondo, J.; Reaño, R.A.L.; Celestial, T.; et al. A Machine Learning Approach for Efficient Spam Detection in Short Messaging System (SMS). In Proceedings of the TENCON 2023—2023 IEEE Region 10 Conference (TENCON), Chiang Mai, Thailand, 31 October–3 November 2023; pp. 53–58. <https://doi.org/10.1109/TENCON58879.2023.10322491>.
6. Jain, V. Optimizing SMS Spam Detection: An In-Depth Analysis of Machine Learning Approaches. In Proceedings of the 2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI), Tirunelveli, India, 20–22 November 2024; pp. 847–852. <https://doi.org/10.1109/ICDICI62993.2024.10810775>.
7. Airlangga, G. Optimizing SMS Spam Detection Using Machine Learning: A Comparative Analysis of Ensemble and Traditional Classifiers. *J. Comput. Netw. Archit. High Perform. Comput.* **2024**, *6*, 1234–1245. <https://doi.org/10.47709/cnahpc.v6i4.4822>.
8. Hafidi, N.; Khoudi, Z.; Nachaoui, M.; Lyaqini, S. Enhanced SMS Spam Classification Using Machine Learning with Optimized Hyperparameters. *Indones. J. Electr. Eng. Comput. Sci.* **2025**, *37*, 356–364. <https://doi.org/10.11591/ijeecs.v37.i1.pp356-364>.
9. Britto, R.V.; Jasirullah, N.; Prabhu, R.S.; Kodhai, E. Combatting SMS Spam: A Machine Learning Approach for Accurate and Scalable Detection. In Proceedings of the 2025 International Conference on Data Science, Agents, and Artificial Intelligence (ICDSAAI), Chennai, India, 24–25 January 2025; pp. 1–5. <https://doi.org/10.1109/ICDSAAI65575.2025.11011638>.
10. Ozoh, P.; Ibrahim, M.; Ojo, R.; Sunmade, A.G.; Oyetayo, T. SMS Spam Detection Using Machine Learning Approach. *Int. STEM J.* **2025**, *6*, 10–27.
11. Nawaz, I.; Khosa, S.N.; Fatima, R.; Saeed, M.; Hashmi, M.S.A. Smart Filters for SMS Spam: A Machine Learning Approach to SMS Classification. *SES J.* **2025**, *2025*, 71–95. <https://doi.org/10.5281/zenodo.15333801>.
12. Ahmadi, M.; Khajavi, M.; Varmaghani, A.; Ala, A.; Danesh, K.; Javaheri, D. Leveraging LLMs for Cybersecurity: Enhancing SMS Spam Detection with Robust and Context-Aware Text Classification. *Cyber-Phys. Syst.* **2025**, 1–25. <https://doi.org/10.1080/23335777.2025.2550938>.
13. Hossain, F.; Uddin, M.N.; Halder, R.K. Analysis of Optimized Machine Learning and Deep Learning Techniques for Spam Detection. In Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 21–24 April 2021; pp. 1–6. <https://doi.org/10.1109/IEMTRONICS52119.2021.9422508>.
14. Altunay, H.C.; Albayrak, Z. SMS Spam Detection System Based on Deep Learning Architectures for Turkish and English Messages. *Appl. Sci.* **2024**, *14*, 11804. <https://doi.org/10.3390/app142411804>.
15. Ghourabi, A.; Mahmood, M.A.; Alzubi, Q.M. A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages. *Future Internet* **2020**, *12*, 156. <https://doi.org/10.3390/fi12090156>.

16. Chowdhury, S.H.; Morzina, M.S.; Hussain, M.I.; Hossain, M.M.; Shovon, Md.; Mamun, M. LoRA and ReFT Optimized Explainable Machine Learning and Deep Learning Framework for SMS Spam Detection. In Proceedings of the 2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN), Rangpur, Bangladesh, 1–2 February 2025; pp. 1–10. <https://doi.org/10.1109/QPAIN66474.2025.11171842>.
17. Liu, X.; Lu, H.; Nayak, A. A Spam Transformer Model for SMS Spam Detection. *IEEE Access* **2021**, *9*, 80253–80263. <https://doi.org/10.1109/ACCESS.2021.3081479>.
18. Ahmed, M.N.; Ahamed, A.S.M.S.; Tamim, F.S. Optimizing SMS Spam Detection with Large Language Models and Transformer Architectures. In Proceedings of the 2025 International Conference on Electrical, Computer and Communication Engineering (ECCE), Chattogram, Bangladesh, 13–15 February 2025; pp. 1–8. <https://doi.org/10.1109/ECCE64574.2025.11013998>.
19. Srinivasarao, U.; Sharaff, A. Machine Intelligence Based Hybrid Classifier for Spam Detection and Sentiment Analysis of SMS Messages. *Multimed. Tools Appl.* **2023**, *82*, 31069–31099. <https://doi.org/10.1007/s11042-023-14641-5>.
20. Salman, M.; Ikram, M.; Kaafar, M.A. Investigating Evasive Techniques in SMS Spam Filtering: A Comparative Analysis of Machine Learning Models. *IEEE Access* **2024**, *12*, 24306–24329. <https://doi.org/10.1109/ACCESS.2024.3364671>.
21. Srinivasarao, U.; Sharaff, A. SMS Sentiment Classification Using an Evolutionary Optimization Based Fuzzy Recurrent Neural Network. *Multimed. Tools Appl.* **2023**, *82*, 42207–42238. <https://doi.org/10.1007/s11042-023-15206-2>.
22. Almeida, T.A.; Gómez Hidalgo, J.M. SMS Spam Collection v.1. UCI Machine Learning Repository, 2012. <https://doi.org/10.24432/C5CC84> (access date: 23 January 2026).
23. Ramanujam, E.; Abirami, A.M.; Sakthiprakash, K.; Sumitra, S. Efficient Extraction and Evaluation of Hand-Crafted Meta-Data Features for Dravidian Spam SMS Classification. *Evolving Syst.* **2026**, *17*, 1–18. <https://doi.org/10.1007/s12530-025-09761-2>.
24. Verma, A.R.K.; Sadana, S. Textual, Non-Textual, and Hybrid Feature Engineering for SMS Spam Classification. *IEEE Access* **2025**, *13*, 176901–176914. <https://doi.org/10.1109/ACCESS.2025.3620751>.
25. Xia, T.; Chen, X. A Discrete Hidden Markov Model for SMS Spam Detection. *Appl. Sci.* **2020**, *10*, 5011. <https://doi.org/10.3390/app10145011>.
26. Abdel-Jaber, H. Detecting Spam and Ham SMS Messages Using Natural Language Processing and Machine Learning Algorithms. *PeerJ Comput. Sci.* **2025**, *11*, e3232. <https://doi.org/10.7717/peerj-cs.3232>.
27. Shen, L.; Wang, Y.; Li, Z.; Ma, W. SMS Spam Detection Using BERT and Multi-Graph Convolutional Networks. *Int. J. Intell. Netw.* **2025**, *6*, 79–88. <https://doi.org/10.1016/j.ijin.2025.06.002>.
28. Yan, D.; Li, K.; Gu, S.; Yang, L. Network-Based Bag-of-Words Model for Text Classification. *IEEE Access* **2020**, *8*, 82641–82652. <https://doi.org/10.1109/ACCESS.2020.2991074>.
29. Jasim, A.K.; Al-Ibeahimi, F.A.F.; Alkaabi, H.A. Explainable AI for SMS Spam Filtering: A Novel Hybrid Architecture Combining Fuzzy Logic and Bidirectional LSTM Networks. *Franklin Open* **2025**, *14*, 100466. <https://doi.org/10.1016/j.fraope.2025.100466>.
30. Xu, H.; Qadir, A.; Sadiq, S. Malicious SMS Detection Using Ensemble Learning and SMOTE to Improve Mobile Cybersecurity. *Comput. Secur.* **2025**, *154*, 104443. <https://doi.org/10.1016/j.cose.2025.104443>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.