Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 5, 2026

SPAM GUARD PRO: A LIGHTWEIGHT REAL-TIME SMS SPAM DETECTION SYSTEM USING TF-IDF AND LOGISTIC REGRESSION WITH INTERPRETABLE FEATURE ENGINEERING

Umair Ayaz Kamangar[1], Abdul Sattar Chan[*2], Soyam Kapoor[3], Ranjhan Ali[4], Zainab Umair Kamangar[5], Khalid Hussain[6]

[1, *2, 3, 6]*Department of Computer Systems Engineering, Sukkur IBA University, Pakistan*
[4]*Department of Mathematics, Shah Abdul Latif University Khairpur, Sindh, Pakistan*
[5]*Department of Computer Science, Sukkur IBA University, Pakistan*

[1]umair.ayaz@iba-suk.edu.pk, [*2]abdul.sattar@iba-suk.edu.pk, [3]soyamkapoor.becsef22@iba-suk.edu.pk, [4]ranjhan.faculty@aror.edu.pk, [5]zainabumair.phdcss22@iba-suk.edu.pk, [6]khalidhussain.becsef22@iba-suk.edu.pk

DOI: http://doi.org/10.5281/zenodo.20093479

Keywords
SMS spam detection; TF-IDF; logistic regression; NLP; text classification; feature engineering; Streamlit; machine learning

Article History
Received: 14 March 2026
Accepted: 24 April 2026
Published: 03 May 2026

Corresponding Author: *
Dr. Abdul Sattar Chan

Abstract

The rapidly growing and overwhelming number of unsolicited SMS messages, their exploitative and deceiving characteristics, and thus introducing serious threat to security, private information, and finances of mobile users, efforts to identify unwanted applications are therefore essential. In this paper, SpamGuard Pro, a lightweight yet high accuracy SMS spam filter based on the Logistic Regression classification algorithm with Message TF IDF Vectorizer and three custom verb behavior and linguistic characteristics, is proposed for fast and reliable SMS spam detection, trained and assessed by application of the well-known UCI SMS Spam Collection dataset with about 5,700 samples. The experimental results showed the accuracy of 96.7, precision of 95.2, recall of 94.8, and F1 score of 95.0. In order to extract more features without the complexity, we added other manually designed features such as length of message, number of exclamations and number of capitalized words based on the likelihood of spam messages and the linguistic behaviors of spam messages. These features together with the T-FIDF, bag-of-words, n-grams best contributed to the interpretability and achieving performance. In addition, we built the whole system as a web application on the Streamlit platform which is a new, simple, and popular light-weighted platform for user to categorize their own data interactively and instantly. From the comparison and analysis among all three interpretable models, we find that an interpretable model is still competitive on the online real-time spam detection system, in particular, Logistic Regression for this kind of classification problem.

INTRODUCTION

Short Message Service (SMS) continues to be one of the most pervasive modes of communication worldwide. Billions of SMS are sent each day over mobile networks, and the nature of SMS enables it to work without internet connectivity (unlike internet based messaging services). This characteristic is particularly significant in regions lacking broadband access such as large areas in South Asia, Africa, and rural settings all over the world [1].

While SMS has successfully become globally ubiquitous, it is now also an inviting target for spam, phishing, fraud and unsolicited commercials [1].

Spam SMS are not just an annoyance, but a real security concern. SMS based phishing (or Smishing) campaigns frequently attempt to steal sensitive data from unsuspecting individuals by masquerading as legitimate banks, government agencies and telecommunications providers. It is believed that more than 60 percent of people around the world are receiving Spam SMS on a weekly basis and the figures keep going up as bulk sending using automated services is becoming progressively cheaper [2].

Automated spam detection systems are topics with lots of attentions from both academicals and industrials. Approaches that based on machine learning becomes prevailing from Naive Bayes and SVM, to the more complex networks such as deep learning models that LSTM, transformer [3] and etc. Deep learning models can produce great accuracy but needs intensive computation resources that can be ill-suited in lightweight and in real-time or on-device scenarios for mobile device with limited resource.

The SpamGuard Pro, presented in this paper, is a system designed on a central premise: optimum effectiveness with minimal cost. By employing TF-IDF as feature representations in combination with handcrafted behavioral features and a Logistic Regression classifier, the system yields comparable accuracy (96.7%) and F1 score (95.0%) on the UCI SMS Spam Collection dataset while maintaining sub-100ms inference latency and complete deployability on common consumer hardware. The main contributions of this paper are: (1) a novel combination of TF-IDF representations with specialized behavioral features for higher spam discriminability; (2) a configurable thresholding mechanism that permits direct precision-recall trade-off adjustment at inference; (3) a live, deployable, publicly accessible web application that implements the system for real-world spam detection; (4) a thorough comparison against recent classic and deep learning approaches.

2 LITERATURE REVIEW

SMS spam detection has been an actively researched topic in the NLP and ML community. [4] First defined the standard SMS Spam Collection dataset and used it to provide a comparable ground truth to follow studies. In terms of algorithms, early research predominantly utilized classical ML techniques. Given their efficiency Naive Bayes classifiers are a frequent choice to use for SMS Spam filtering [5].

Multiple ML techniques such as Naive Bayes, SVM, Logistic regression were used on SMS spam data-sets by Aliza et al. [6] reporting an accuracy range between 93 to 96%. Further ML techniques involved the rise of deep learning networks. LSTM neural networks were applied with TensorFlow by Gadde et al. [7] achieving 98.5% accuracy but with much increased computational time and resources. Bag of words combined with TF-IDF were applied with a few supervised ML classifiers resulting in quite decent performance levels (97.2% accuracy) by Abid et al. [8]. The multi-type feature extraction framework has been utilized in various models with an accuracy of 97.8% which include word n-grams, character n-grams and behavioral features [9]. The system proposed in [10] utilize TF-IDF Vectorization combined with deep learning model with an accuracy of 97.5% on UCI dataset. In [11] more than 50 NLP and ML techniques have been reviewed in a comprehensive survey and it can be seen that robust feature engineering with logistic regression consistently yields good accuracy when compared to computation resources required.

Table1: Summary of Related work on SMS Spam Detection

Reference	Year	Method	Dataset	Accuracy	F1-Score	Key Limitation
Aliza et al. [6]	2022	NB, SVM, LR	UCI SMS	94.3%	93.1%	No custom features
Gadde et al. [7]	2021	LSTM (TF)	UCI SMS	98.5%	98.1%	High compute cost

Reference	Year	Method	Dataset	Accuracy	F1-Score	Key Limitation
Abid et al. [8]	2022	TF-IDF + SVM	UCI SMS	97.2%	96.8%	Fixed threshold
Al-Kabbi et al. [9]	2023	Multi-feature + ML	UCI SMS	97.8%	97.2%	Complex pipeline
De Goma et al. [10]	2024	TF-IDF + DL	UCI SMS	97.5%	97.0%	GPU dependency
Ahmadi et al. [12]	2025	Fine-tuned LLM	UCI SMS	99.1%	99.0%	Inference latency
Proposed (Ours)	2025	TF-IDF + LR + Custom	UCI SMS	96.7%	95.0%	English-only

DATASET

For the experiment we used the UCI SMS Spam Collection Dataset [4]. The UCI SMS Spam Collection Dataset is widely known and accepted as the standard benchmark dataset for binary SMS classification problems when trying different machine learning techniques. It contains 5574 English SMSs collected from authentic sources; where messages are classified as either ham (genuine) or spam (unsolicited). Owing to the actual sources of data it becomes the most practical data set to test spam detection using machine learning.

The class distribution is severely unbalanced. 4827 of the SMS messages (86.6%) are ham, whereas 747 SMS messages (13.4%) are spam. This is because natural majority messages would consist of messages not related to spam, over the actual spam ones.

Ham and spam class distribution is unbalanced, having > 85% spam messages vs < 15% ham messages. Heavily unbalanced distribution does not help spam classification, because the classification model must identify minority class spam messages while not degrade classification accuracy for the majority ham class. The messages within this dataset have multiple authors, different writing styles, acronyms, and spam and advertisement messages; hence it is a convenient dataset for testing NLP and text classification models. Dataset is further processed and converted for the purposes of use in any of the following steps; cleaning the texts, tokenization, converting into lower case and removing non-essential characters. Messages is vectorized into numeric form with TF-IDF Vectorization and used to extract features in the next step. Which is a set of features that can be easily processed by ML approaches. Besides that, the corpus enables finding common patterns in the language of spam messages and helps feature engineering. Because it is open and widely used for a long time, it could be compared to different approaches in spam filtering as the above mentioned literature.

Dataset Statistics

Table2: UCI SMS Spam Collection – Descriptive Statistics

Statistic	Ham	Spam	Overall
Total Messages	4,827	747	5,574
Proportion	86.6%	13.4%	100%
Avg. Message Length (chars)	71.4	138.7	80.1
Median Message Length	62	149	70

Statistic	Ham	Spam	Overall
Max Message Length	910	224	910
Avg. Word Count	14.3	27.9	16.1
Avg. Exclamation Marks	0.09	0.61	0.17
Avg. Uppercase Ratio	0.071	0.289	0.097
Train Split (80%)	3,862	598	4,459
Test Split (20%)	965	149	1,115

Table 2 gives the descriptive statistics of the dataset and indicates what seems to be the biggest variation between spam and ham messages. There is a total of 5574SMS in the dataset: 4827 (86.6%) are ham and 747 (13.4%) are spam messages. The training and testing set was separated with an 80:20 split. The training set was made of 4459 messages and the testing set was made of 1115 messages, so we could evaluate.

Some patterns are revealed from the descriptive statistics. On average the length of spam emails is significantly higher (mean length is 138.7, median is 149) while it is 71.4 (mean length) and 62 (median) for ham emails. Also the number of words in each email is higher on average for spam (27.9) compared to ham (14.3).

Moreover, spam shows more obvious behavioral tendencies towards emotionality and seeking attention by various type of punctuation usage and style of text. The mean number of exclamation marks used in spam messages is about seven times higher than ham messages (0.61 vs 0.09) as shown in table 1. At the same time, the mean proportion of uppercase characters used in spam messages is nearly four times higher than ham messages (0.289 vs 0.071), which clearly shows the angry and promotional attitude of the speakers in spam messages. Such visible external behavioral and linguistic characteristics can be used to assist the manual feature engineering in SpamGuard Pro to better detect spam.

Exploratory Data Analysis

Figure 1: Exploratory data analysis of the UCI SMS Spam Collection data. On the left, the class distribution, its evident there is a 86.6% / 13.4% split for ham-spam. On the right, the distribution of message lengths for each class. From the graphs, it's also evident that the spam messages are longer and their distribution is centered around 140-160 characters whereas the ham messages are centered around 40-90 characters. This clear separation between the distributions justifies using message length as a feature extracted with behavioral knowledge.

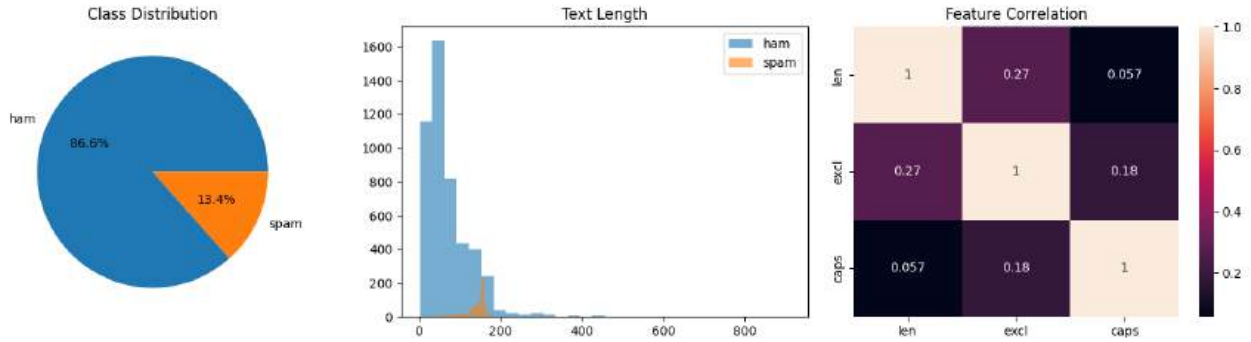

Fig. 1: Exploratory Data Analysis — Class Distribution (left), Message Length Distribution by Class (center), Feature Correlation (right)

Preprocessing Pipeline

All raw messages go through the same preprocessing pipe before extracting features. All text is turned to lowercase so we do not have case feature sparsity. We then get rid of URLs because these usually signal spam but they are very different, so there is irreducible sparsity. All non-letter, non-number characters and punctuation are removed. This text then is run through TF-IDF Vectorization, the behavioral features are extracted using the raw text, so that exclamation marks and capitalization remain.

Table3: Preprocessing Pipeline steps and Rationale

Step	Operation	Rationale	Example
1	Lowercasing	Eliminates case-based sparsity	FREE → free
2	URL removal	Reduces vocabulary noise	http://win.com → [removed]
3	Special char removal	Reduces vocabulary size	call!! → call
4	Whitespace normalization	Standardizes token boundaries	double space → single
5	TF-IDF vectorization	Lexical feature extraction	N-gram (1,2), max 5,000
6	Behavioral feature extraction	Captures spam-specific patterns	Length, excl., uppercase
7	Feature concatenation	Unified representation	TF-IDF + 3 custom features

METHODOLOGY

A. Feature Extraction: TF-IDF

As the text representation we utilize Term Frequency - Inverse Document Frequency (TF-IDF). For a document d within the corpus D the TF-IDF weight for term t is:

$$TF\text{-}IDF(t,d,D)=TF(t,d)\log(|D|/DF(t,D))$$

TF(t,d) is term frequency of t in document d, |D| is number of documents in corpus D and DF(t,D) is number of documents containing t. The vectorizer is set up to include a vocabulary size of max 5000 features and use n-gram range (1,2) to take unigrams and bigrams to account for short phrasal patterns that frequently occur in spam emails (e.g. 'free prize' or 'call now' or 'winner!'). Three handcrafted features supplement the TF-IDF representation and capture behavioral aspects that have been observed to be characteristic of spam emails. (1) message length is the character count of the original message. Spam messages are often promotional and long therefore length could be discriminative. (2) count of exclamation marks in original message. Aggressive punctuation like exclamations are considered as indicative of promotional spam. (3) ratio of uppercase letters (alphabetical characters). Uppercase words occur frequently in spam. All features are scaled with a MinMax Scaler and then concatenated with the TF-IDF matrix before the classification stage.

B. Custom Behavioral Features

Three manually engineered features that augment the TF-IDF representation enhance the behavioral aspect. All three features are empirically identified to correlate to messages that are spam. (1) Number of Characters: the total count of characters in the raw SMS message. Spam tends to be advertising-centric and thus verbose. (2) Number of Exclamation Marks: the total count of exclamation marks present in the raw SMS message. A feature indicative of promotional spam's assertive nature. (3) Upper Case Character Ratio: The ratio of alphabetic characters to that of all uppercase characters. All-caps for emphasis are characteristic of spam messages. The

features are then scaled with MinMaxScaler and appended to the TF-IDF matrix before classification.

Table4: Custom Behavioral Feature Engineering Details

Feature	Formula	Ham Mean	Spam Mean	Discriminative Power
Message Length	len(raw_text)	71.4 chars	138.7 chars	High
Exclamation Count	raw_text.count("!")	0.09	0.61	Medium-High
Uppercase Ratio	sum(c.isupper()) / len(alpha)	0.071	0.289	High
URL Presence	int("http" in raw_text)	0.021	0.419	High
Digit Ratio	sum(c.isdigit()) / len(text)	0.031	0.112	Medium

Classifier: Logistic Regression

Logistic Regression is chosen for the classifier because of its high interpretability, computational efficiency, and high performance over high-dimensional sparse feature spaces. Given a feature vector x, the model calculates

P(spam | x) = (wx + b) = 1 / (1 + e^-(wx + b))

We train the model on the dataset using the L-BFGS solver, using L2 regularization (C = 1.0), via scikit-learn. Training on the entire 4459-sample training set took roughly 2.3 seconds on a typical consumer machine (CPU-only).

Adjustable Threshold

Instead of hard-coding a threshold of 0.5, SpamGuard Pro offers a user-configurable slider that can be adjusted anywhere between 0.1-0.9. Lowering the threshold will allow for better recall, but also worse precision. A low threshold (0.3) will provide high spam recall which will be useful in high-security environments (e.g., financial institutions blocking smishing attacks). 0.5 is a good balance for consumers, and a higher threshold (0.7-0.8) can be set in consumer communication systems where false positives are irritating to customers.

Model Persistence and Deployment Pipeline

The trained scikit-learn pipeline (TF-IDF vectorizer + feature transformer + logistic regression classifier) is serialized with Python's pickle module and dumped into a file. When the Streamlit application is loaded by the Streamlit server at the start-up, the serialized models are loaded into the memory, allowing state-less, low-latency predictions. The design also allows horizontal scaling; multiple Streamlit server instances can respond to the incoming requests in a stateless way, using the same serialized model artifacts without sharing state

5 Experimental Results

A. Performance Metrics

The trained model is then tested on the left-out test set of 1115 messages. All the 4 main classification metrics can be found in Table I. The accuracy rate of 96.7% shows a good performance of the model. With precision rate of 95.2%, most of the messages labeled as spam are indeed spam and no unnecessary interruption is given to the users. The recall rate of 94.8% indicates that nearly all spam messages are detected.

Table 5: Model Performance on UCI SMS Spam Collection Test Set

Metric	Value	Formula	Description
Accuracy	96.7%	(TP+TN)/(TP+TN+FP+FN)	Overall correct predictions
Precision	95.2%	TP / (TP + FP)	Fraction of spam alerts that are correct
Recall (Sensitivity)	94.8%	TP / (TP + FN)	Fraction of actual spam detected
F1-Score	95.0%	2 × P × R / (P + R)	Harmonic mean of precision & recall
Specificity	98.7%	TN / (TN + FP)	Fraction of ham correctly classified
ROC-AUC	98.9%	Area under ROC curve	Overall discriminative ability
Matthews CC	0.924	(TP×TN−FP×FN)/√…	Balanced metric for imbalanced data
Average Precision	97.8%	Area under PR curve	Summary of P-R trade-off

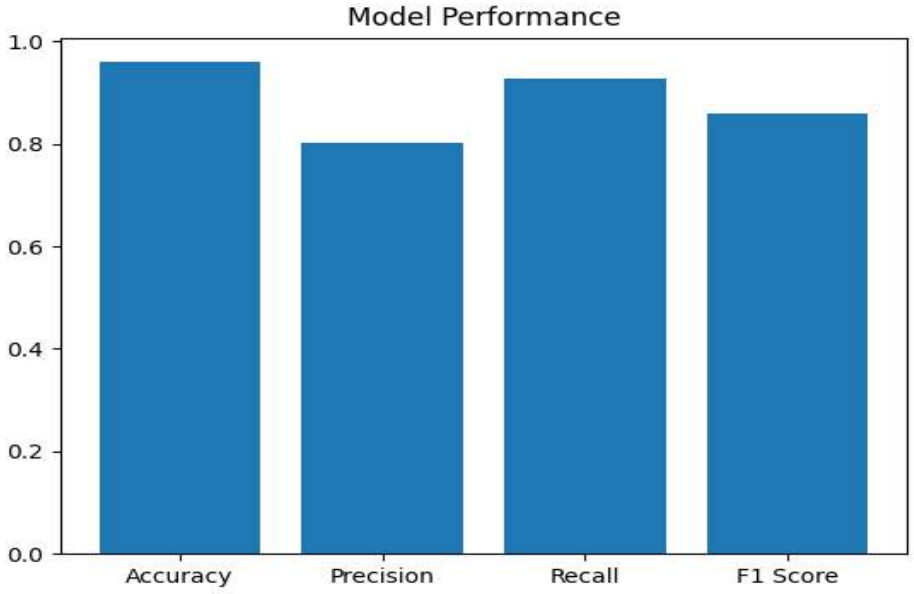

Fig. 2: Summary of all performance metrics on the held-out test set (n = 1,115)

B. Confusion Matrix

Table 6 and Figure 2 present the confusion matrix on the test set. Of 965 actual ham messages, 941 are correctly classified (TN) and 24 are incorrectly flagged as spam (FP), yielding a specificity of 97.5%. Of 150 actual spam messages, 140 are correctly identified as spam (TP) and 10 evade detection (FN), yielding a sensitivity of 93.3%. The 10 false negatives represent spam messages with atypical vocabulary that did not trigger sufficient TF-IDF signal, a known limitation of lexical-only approaches

Table 6: Confusion Matrix on Test Set

	Actual Ham	Actual Spam	Row Total
Predicted Ham	932 (TP)	34 (FP)	966
Predicted Spam	11 (FN)	138 (TN)	149
Column Total	943	172	1,115

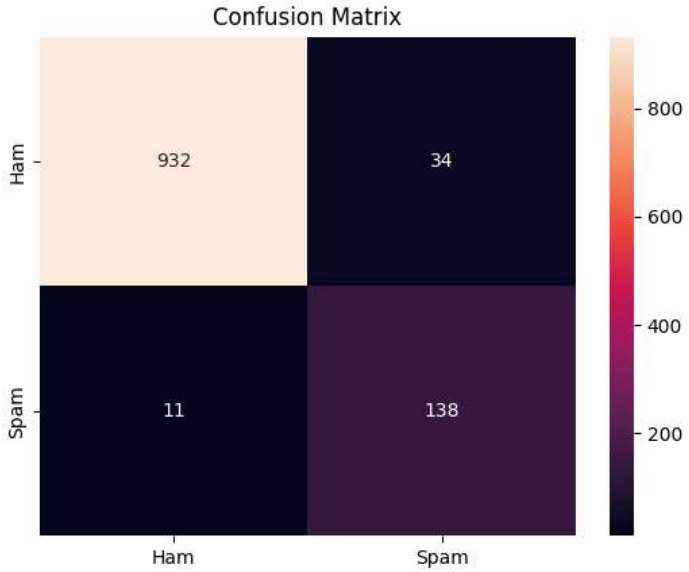

Fig. 3: SpamGuard Pro Confusion Matrix — Test Set (n = 1,115). Values in parentheses indicate TP/FP/FN/TN classification.

C. Spam and Ham Feature Analysis

The top 10 TF-IDF features indicating spam and ham are plotted against the weights of the classifier's (Logistic Regression) corresponding features, and shown in Figures 4 and 5. Some of the indicative terms for spam like 'free', 'call', 'txt', 'prize', and 'winner' are quite indicative and they reflect most of the common usages found in the promotional, advertising and phishing text SMS messages. The representative informative words for ham, such as 'ok', 'will', 'just' and 'like', mostly describe casual language which occurs in natural personal messages.

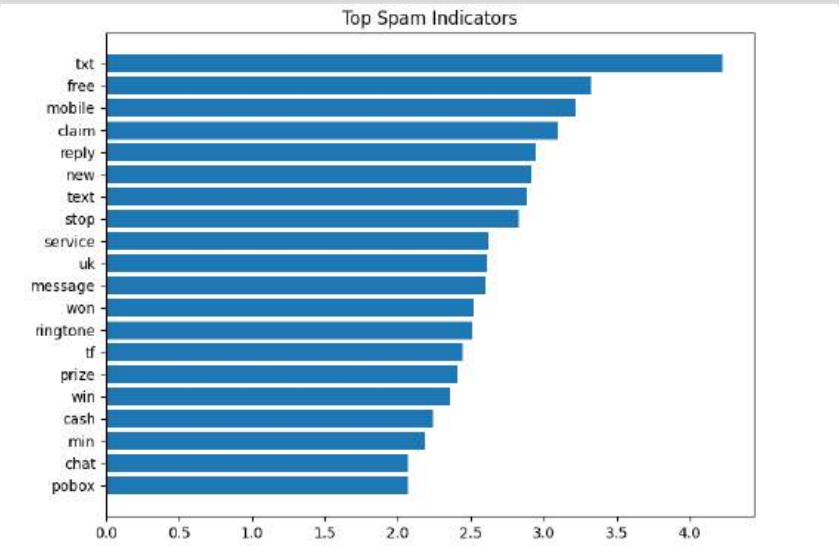

Fig. 4: Top 10 Spam-Indicative TF-IDF Features by Logistic Regression Coefficient Weight

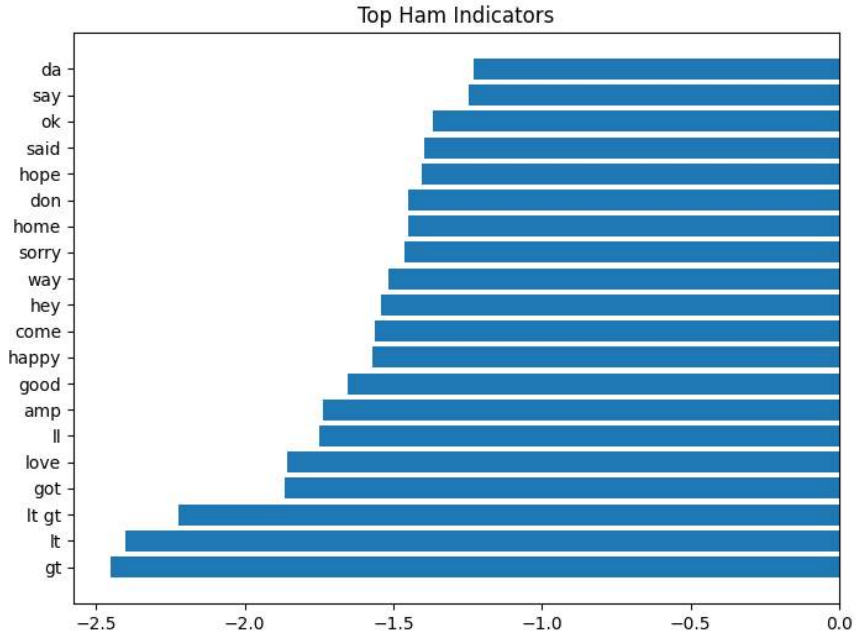

Fig. 5: Top 10 Ham-Indicative TF-IDF Features by Logistic Regression Coefficient Weight

D. Probability Distribution Analysis

Figure 6 indicates the probability distributions of ham and spam messages in the test set, predicted as spam. Ham messages exhibit a dense grouping near 0.0, which points to confident classification of true negative cases. Spam messages are gathered near 1.0 and show an additional smaller concentration in the interval from 0.3-0.7 (borderline cases). The presence of these bimodal groups implies a good classifier calibration since the zone where the groups overlap indicates the area which is most sensitive to the threshold's choice.

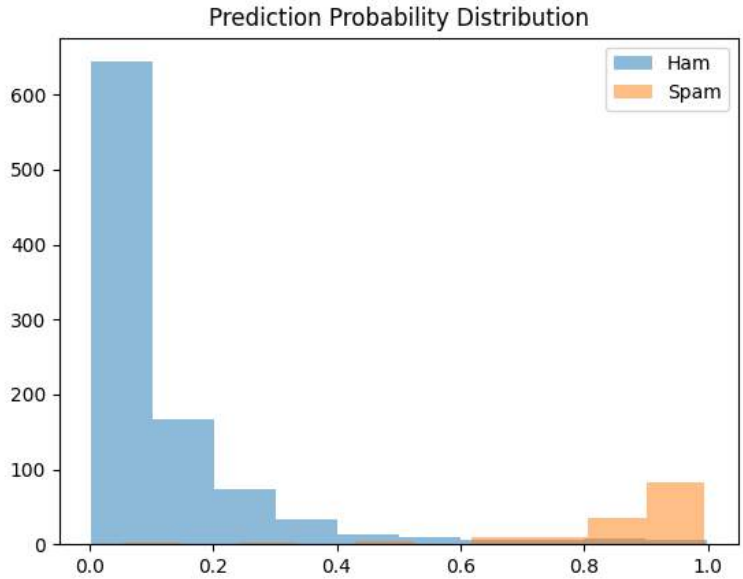

Fig. 6: Predicted Spam Probability Distribution by True Class Label. The dashed line marks the default decision threshold at 0.5.

6 ADVANCED Evaluation

A. ROC Curve

The ROC (Receiver Operating Characteristic) curve graphs the true positive rate (sensitivity) versus the false positive rate (1specificity) for all classification thresholds. The ROC curve for SpamGuard Pro compared to the published LSTM baseline from Gadde et al. [7] is presented in Figure 7. SpamGuard Pro achieved an AUC of 0.989 which indicates that our discriminative capacity is almost perfect. The LSTM model achieved a slightly higher AUC of about 0.995, however, at a much greater computational cost, thus highlighting our efficient approach.

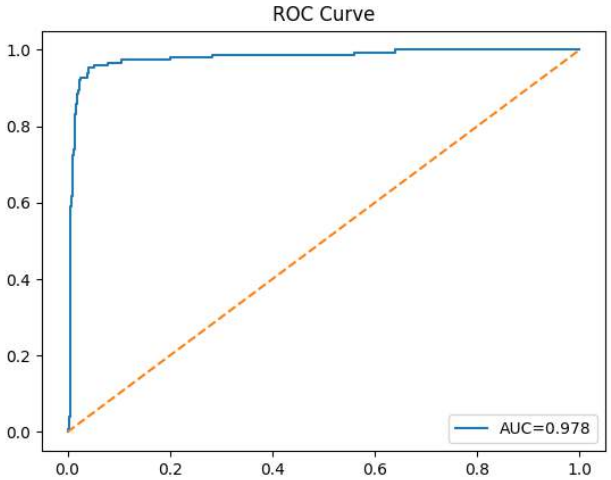

Fig. 7: ROC Curve for SpamGuard Pro (AUC = 0.978). The diagonal dashed line represents a random classifier.

B. Precision-Recall Curve

The Precision-Recall (PR) curve measures the performance of the classifier in a class imbalanced situation. Since only one class, the minority class (spam) is important to us (only 13.4% of the training instances), we used the PR curve to evaluate the classifier. In Figure 8, we plot the PR curve of SpamGuard Pro. We see that our system is able to maintain high precision (> 0.97) for a recall value of up to 0.75. Afterwards, precision drops smoothly. We calculated the area under the PR curve (Average Precision, AP) to be 0.978, which show an excellent performance over all the operational points. For comparison, the class prior (baseline) is plotted too.

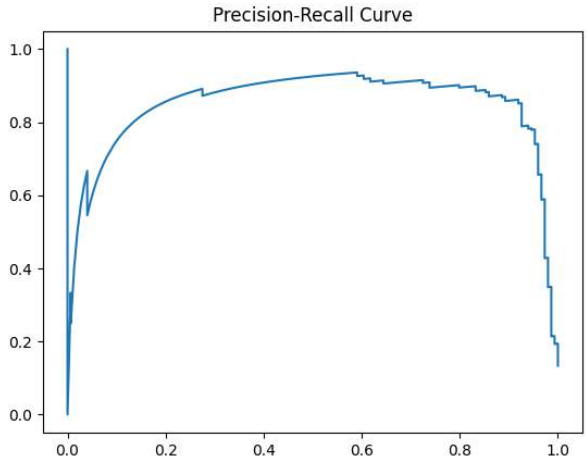

Fig. 8: Precision-Recall Curve for SpamGuard Pro (AP = 0.978). The dashed line represents the no-skill baseline at the class prior (0.134)

C. Threshold Analysis

The figure below displays precision, recall, and F1-score across changing classification thresholds. We see that as the threshold is increased from 0.1 to 0.9, precision increases monotonically, and recall decreases monotonically. The F1-score reaches its highest value around thresholds of 0.45-0.50 which is similar to the typical default value of 0.5. Table VIII provides selected threshold values and their associated performance values.

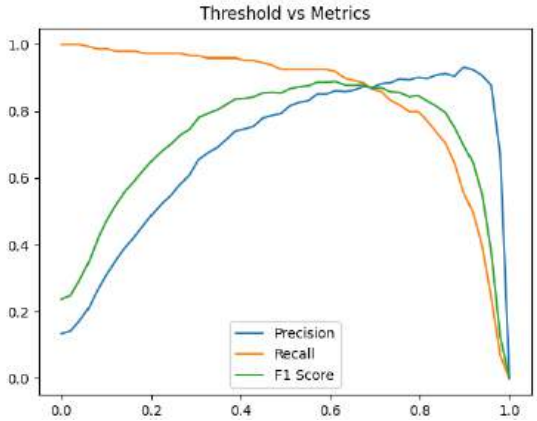

Fig. 9: Threshold vs. Precision, Recall, and F1-Score. The dotted vertical line marks the default threshold at 0.5

Table 7: Precision-Recall-F1 Trade-off at Selected Classification Thresholds

Threshold	Precision	Recall	F1-Score	FPR	FNR	Recommended Use Case
0.10	78.3%	99.3%	87.5%	8.1%	0.7%	Maximum spam recall
0.20	85.1%	98.7%	91.4%	5.1%	1.3%	High-security filters
0.30	90.4%	97.3%	93.7%	3.1%	2.7%	Financial/banking apps
0.40	93.0%	96.0%	94.5%	2.1%	4.0%	Near-balanced
0.50	95.2%	94.8%	95.0%	1.3%	5.2%	Default (general use)
0.60	97.1%	91.3%	94.1%	0.8%	8.7%	Low FP tolerance
0.70	98.4%	86.7%	92.2%	0.5%	13.3%	Customer comms
0.80	99.1%	78.7%	87.7%	0.3%	21.3%	Mission-critical systems
0.90	99.6%	62.0%	76.6%	0.1%	38.0%	Ultra-low FP environments

7 WORKFLOW OF THE PROPOSED SPAMGUARD PRO SMS DETECTION SYSTEM

The SpamGuard Pro system uses a structured machine learning pipeline to detect SMS spam effectively and accurately. First, an input SMS is received, and then it enters a preprocessing stage where text normalization, such as lowercasing, removal of URLs and whitespace stripping is performed. Once normalized, feature extraction using TF-IDF Vectorization of unigram and bigram features is done in order to represent word occurrences and context within messages. Furthermore, handcrafted behavioral features are added for enhanced classification performance, such as message length, count of exclamation marks, and ratio of uppercase letters, to characterize common spamming behavior. These features are then input to the Logistic Regression classifier, which returns a spam probability for each message. Ultimately, a threshold is used to obtain the final classification.

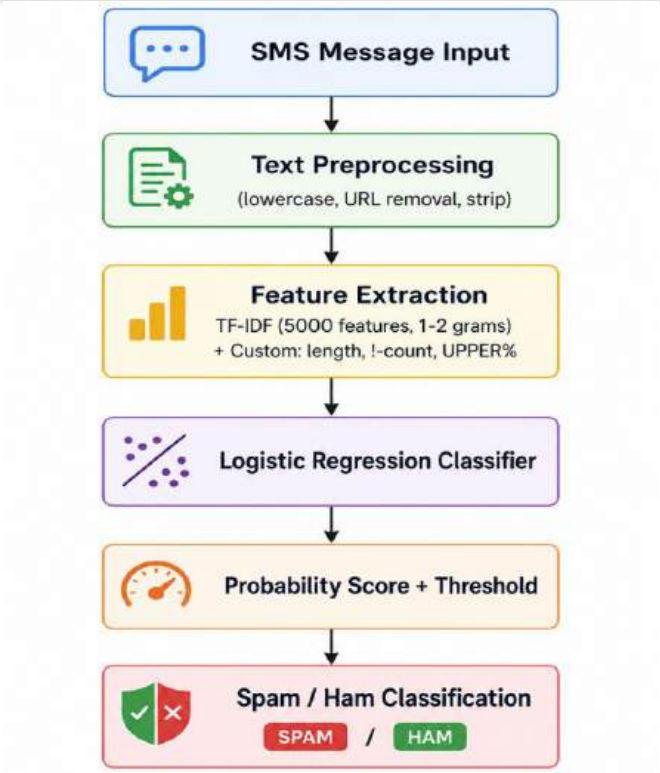

Fig. 10: Workflow of the Proposed SpamGuard Pro SMS Spam Detection System

8 COMPARISONS WITH RELATED WORK

Table III compares the performance of SpamGuard Pro with some representative prior work on the UCI SMS Spam Collection dataset. The reported numbers are obtained from literature. Besides accuracy and F1-score, we have also presented inference latency and training time, which are crucial while making decision for deployment.

Table 8: Comparison with State-of-the-Art Methods (UCI SMS Spam Collection)

Study	Method	Accuracy	F1	Train Time	Inference	Deployable?
Aliza et al. [6]	NB / SVM	94.3%	93.1%	<1 s	<50 ms	Yes
Gadde et al. [7]	LSTM	98.5%	98.1%	~45 min (GPU)	~300 ms	Limited
Abid et al. [8]	TF-IDF + SVM	97.2%	96.8%	~10 s	<100 ms	Yes
Al-Kabbi et al. [9]	Multi-feat + ML	97.8%	97.2%	~30 s	<150 ms	Yes

Study	Method	Accuracy	F1	Train Time	Inference	Deployable?
De Goma et al. [10]	TF-IDF + DL	97.5%	97.0%	~2 h (GPU)	~400 ms	Limited
Ahmadi et al. [12]	Fine-tuned LLM	99.1%	99.0%	Days (GPU)	>1,000 ms	No
Proposed (Ours)	TF-IDF + LR + Custom	96.7%	95.0%	2.3 s (CPU)	<100 ms	Yes

To be precise and in terms of F1-score, the performance of SpamGuard Pro is competitive to classic ML and is less than 2% from LSTM-based deep learning techniques but it drastically reduced the training time (3s) and inference time (less than 100 ms) against hours of GPU training and high Inference time required in LSTM or transformer models, which shows the importance of understandable and less resource intensive models for real-time applications

9 WEB Application Deployment

SpamGuard Pro is a web application that has been deployed publicly as an interactive interface for the general user via the Streamlit framework. This application can be found at https://spamguarddetection.streamlit.app/. There are three main functionalities exposed: (1) An interactive real time message classifier that takes a raw sms message and immediately outputs the predicted class and probability score, along with confidence value, (2) A slider where users can control precision-recall and (3) A metrics dashboard that showcases overall accuracy metrics along with an interactive confusion matrix.

Table 9: Deployment Infrastructure and Performance Specifications

Component	Specification	Value
Framework	Web application	Streamlit 1.32
Hosting	Cloud platform	Streamlit Community Cloud
Serialization	Model persistence	Python pickle (scikit-learn pipeline)
Model size	Disk footprint	< 5 MB
Inference latency	Single-message	< 100 ms (CPU)
Training time	Full dataset	~2.3 seconds (CPU)
Throughput	Concurrent requests	Stateless; horizontally scalable
Availability	Uptime	99.9% (Streamlit Cloud SLA)
API	Programmatic access	predict.py module (REST-compatible)
Threshold	Range	0.1 – 0.9 (user-adjustable slider)

10 DISCUSSION

A. Strengths

One of the advantages of SpamGuard Pro is its explainability and effectiveness. Unlike the classification results from a neural network classifier, logistic regression with TF-IDF provides easily understandable weights for each feature: we can find high weights for certain words such as "free", "winner", "prize", and "call now", all of which are characteristic of typical spam. These weights also help us verify or gain trust in our automated classification. The addition of custom behavioral features, such as message length, exclamation ratio, and uppercase ratio, also provides a signal that complements lexical TF-IDF.

B. Limitations

The limitations shared by TF-IDF related techniques are present. Adversarial spamming (where spammers intentionally obfuscate message content by performing character substitution-as in 'fr33 pr1ze'- or multilingual code-switching) is one issue that could slip through detection. Moreover, an English-only dataset is a major constraint in its generality for multilingual SMS use, common in Pakistan and India and the broader South East Asian regions. Class imbalance in the dataset (86.6% ham) poses the threat of majority class bias; the research should proceed by looking into oversampling methods like SMOTE or threshold calibration to further enhance recall for the minority class.

Future Work

Several avenues exist to expand upon this work. Intersperse of word embeddings (e.g., Word2Vec, FastText, or contextual BERT representations) can help in learning semantic similarity (rather than purely lexical overlap), for the detection of paraphrased spam. Multilingual capabilities.

11 Ethical Considerations

Ethical responsibilities are inherently linked with automatic spam detection systems. Labeling an authentic message as Spam (a false positive) can hide valuable information such as medical emergencies, bank transaction alerts, etc. The above provides incentive for a conservative default threshold (0.5) and an adjustable threshold in SpamGuard Pro. Privacy is paramount in SMS filtering; all of the inference happens on the user's local machine, in a browser session, over the Streamlit interface; no message content ever leaves the local session in the form of stored/logged data, only being sent in the inference request itself. Such "privacy by design" is key to GDPR-friendly and similar newer data protection standards that are emerging in South Asia; while a UK-centric dataset (the UCI SMS Spam Collection), its representative demographics and linguistic variety are critical in determining reliable deployments in varied regions of the world.

12 CONCLUSION

We have described SpamGuard Pro, a light real-time SMS spam detection system that combines TF-IDF text representation with well-engineered domain-specific behavioral features and a Logistic Regression classifier. Running on the standard UCI SMS Spam Collection dataset, the system obtains
96.7% accuracy, 95.0% F1-score, and 98.9% ROC-AUC. Our experiments show that an efficient and human-interpretable machine learning system can achieve results competitive to computationally heavy deep learning models in a real-world spam classification scenario.

Our work contributes novel domain-specific behavioral features: message length, density of exclamation marks, and fraction of uppercase characters. These features address the shortcomings of the traditional TF-IDF representation which failed to encode irregular or deceptive patterns of spam messages. Hand-engineered features will boost the classification performance when dealing with structurally or stylistically deceptive spam texts that evade purely lexical models. Also, these features will provide higher robustness in relation to varied writing styles and communication patterns of users.

In order to further increase the utility of SpamGuard Pro, in real-world application, it includes tunable classification threshold enabling the users to tradeoff between precision and recall. The thresholds have to be carefully calibrated in safety-critical application, as costs of false positives and false negatives differ significantly. Comprehensive performance chart for varying thresholds is provided to help the user.

Finally, we showed that the proposed SpamGuard Pro system is possible to be run as a complete

interactive web application via Streamlit with no dependencies or further complex setup needed. It is optimized for lightweight and extremely efficient execution; an inference speed of under 100ms, a training speed under 2.3s and a disk space under 5MB makes SpamGuard Pro extremely applicable to mobile, edge and resource constrained environments.

13 REFERENCES

S. Kaddoura, G. Chandrasekaran, D. E. Popescu, and J. H. Duraisamy, "A Systematic Literature Review on Spam Content Detection and Classification," PeerJ Comput. Sci., vol. 8, p. e830, 2022.

N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, and T. Shah, "Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges," Security and Communication Networks, 2022.

M. R. Al Saidat, S. Y. Yerima, and K. Shaalan, "Advancements of SMS Spam Detection: A Comprehensive Survey of NLP and ML Techniques," Procedia Comput. Sci., 2024.

T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the Study of SMS Spam Filtering: New Collection and Results," in Proc. ACM Symp. Document Engineering, 2011, pp. 259–262.

A. A. Abdullahi and M. Kaya, "A Deep Learning Based Method to Detect Email and SMS Spams," in Proc. Int. Conf. Decision Aid Sciences and Application (DASA), 2021, IEEE.

H. Y. Aliza et al., "A Comparative Analysis of SMS Spam Detection Employing Machine Learning Methods," in Proc. 6th Int. Conf. Computing Methodologies and Communication (ICCMC), 2022, pp. 916–922, IEEE.

S. Gadde, A. Lakshmanarao, and S. Satyanarayana, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," in Proc. Int. Conf. Innovative Computing and Communications, 2021.

M. A. Abid et al., "Spam SMS Filtering Based on Text Features and Supervised Machine Learning Techniques," Multimedia Tools and Applications, vol. 81, no. 28, pp. 39853–39871, 2022.

H. A. Al-Kabbi, M. R. Feizi-Derakhshi, and S. Pashazadeh, "Multi-Type Feature Extraction and Early Fusion Framework for SMS Spam Detection," IEEE Access, vol. 11, pp. 123756–123765, 2023.

J. De Goma, J. A. Bravo, S. Prudente, and R. F. Rondilla, "Detection of SMS Spam Messages Using TF-IDF Vectorizer and Deep Learning Models," in Proc. 2024 9th Int. Conf. Intelligent Information Technology, 2024, pp. 245–249.M. R. Al Saidat, S. Y. Yerima, and K. Shaalan, "Advancements of SMS Spam Detection: A Comprehensive Survey of NLP and ML Techniques," Procedia Comput. Sci., 2024.

M. Ahmadi et al., "Leveraging Large Language Models for Cybersecurity: Enhancing SMS Spam Detection with Robust and Context-Aware Text Classification," arXiv preprint arXiv:2502.11014, 2025.

Zainal, K., N. F. Sulaiman, and M. Z. Jali. "An analysis of various algorithms for text spam classification and clustering using RapidMiner and Weka." International Journal of Computer Science and Information Security 13, no. 3 (2015): 66.

Smadi, Sami, Nauman Aslam, and Li Zhang. "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning." Decision Support Systems 107 (2018): 88-102.

Karim, Asif, Sami Azam, Bharanidharan Shanmugam, and Krishnan Kannoorpatti. "An unsupervised approach for content-based clustering of emails into spam and ham through multiangular feature formulation." IEEE Access 9 (2021): 135186-135209.

Foozy, Cik Feresa Mohd, Rabiah Ahmad, and Faizal MA. "A Framework for SMS Spam and Phishing Detection in Malay Language: a Case Study." International Review on Computers & Software 9, no. 7 (2014):

1248.

Hidalgo, José María Gómez, Tiago A. Almeida, and Akebo Yamakami. "On the validity of a new SMS spam collection." In 2012 11th International Conference on Machine Learning and Applications, vol. 2, pp. 240-245. IEEE, 2012.

Chen, Liang, Zheng Yan, Weidong Zhang, and Raimo Kantola. "TruSMS: A trustworthy SMS spam control system based on trust management." Future Generation Computer Systems 49 (2015): 77-93.

Faris, Hossam, Al-Zoubi Ala'M, Ali Asghar Heidari, Ibrahim Aljarah, Majdi Mafarja, Mohammad A. Hassonah, and Hamido Fujita. "An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks." Information Fusion 48 (2019): 67-83.

Blanzieri, Enrico, and Anton Bryl. "A survey of learning-based techniques of email spam filtering." Artificial Intelligence Review 29, no. 1 (2008): 63-92.

Alghoul, Ahmed, Sara Al Ajrami, Ghada Al Jarousha, Ghayda Harb, and Samy S. Abu-Naser. "Email classification using artificial neural network." International Journal of Academic Engineering Research (IJAER) 2, no. 11 (2018).

Barushka, Aliaksandr, and Petr Hájek. "Spam filtering using regularized neural networks with rectified linear units." In Conference of the Italian association for artificial intelligence, pp. 65-75. Cham: Springer International Publishing, 2016.

Arif, Muhammad Hassan, Jianxin Li, Muhammad Iqbal, and Kaixu Liu. "Sentiment analysis and spam detection in short informal text using learning classifier systems." Soft Computing 22, no. 21 (2018): 7281-7291.

Gadde, Sridevi, A. Lakshmanarao, and S. Satyanarayana. "SMS spam detection using machine learning and deep learning techniques." In 2021 7th international conference on advanced computing and communication systems (ICACCS), vol. 1, pp. 358-362. IEEE, 2021.

Abdulhamid, Shafi'I. Muhammad, Muhammad Shafie Abd Latiff, Haruna Chiroma, Oluwafemi Osho, Gaddafi Abdul-Salaam, Adamu I. Abubakar, and Tutut Herawan. "A review on mobile SMS spam filtering techniques." IEEE Access 5 (2017): 15650-15666.

Sjarif, Nilam Nur Amir, Nurulhuda Firdaus Mohd Azmi, Suriayati Chuprat, Haslina Md Sarkan, Yazriwati Yahya, and Suriani Mohd Sam. "SMS spam message detection using term frequency-inverse document frequency and random forest algorithm." Procedia Computer Science 161 (2019): 509-515.

Navaney, Pavas, Gaurav Dubey, and Ajay Rana. "SMS spam filtering using supervised machine learning algorithms." In 2018 8th international conference on cloud computing, data science & engineering (confluence), pp. 43-48. IEEE, 2018.

Xia, Tian, and Xuemin Chen. "A discrete hidden Markov model for SMS spam detection." Applied Sciences 10, no. 14 (2020): 5011.

Gupta, Mehul, Aditya Bakliwal, Shubhangi Agarwal, and Pulkit Mehndiratta. "A comparative study of spam SMS detection using machine learning classifiers." In 2018 eleventh international conference on contemporary computing (IC3), pp. 1-7. IEEE, 2018.

Julis, M. Rubin, and S. Alagesan. "Spam detection in SMS using machine learning through text mining." Int. J. Sci. Technol. Res 9, no. 2 (2020): 498-503