Recognizing Fake Documents by Instance-Based ML Algorithm Tuning with Neighborhood Size

S. Prakash[1], B. Kalaiselvi[2,*], K. Sivachandar[3], M. Batumalay[4]

[1]*Department of EEE, Bharath Institute of Higher Education and Research, Chennai, India*

[2]*Department of ECE, Bharath Institute of Higher Education and Research, Chennai, India*

[3]*Department of ECE, RMK Engineering College, Chennai, India*

[4]*Faculty of Data Science and Information Technology, INTI International University Nilai, Malaysia*

(Received: November 21, 2024; Revised: December 11, 2024; Accepted: January 21, 2025; Available online: March 15, 2025)

Abstract

The primary objective of this research paper is to classify spam SMS messages for scamming threats as soon as they are received on a device. The study focuses on evaluating the performance of K-Nearest Neighbors (KNN) classifiers with different neighborhood sizes to determine the most effective machine learning technique for improving accuracy and predictions in SMS spam detection. SMS is a short text messages service that permits mobile phone users to exchange messages. In today's world, people are so much tending towards mobile phones and it has become easy to spread spam content through them. One can easily access any person's details through these social networking websites. No information which is shared and stored in the device is not secure. Numerous anti-spam systems have been developed. In this paper, we compare the classification results against spam SMS data to estimate the effectiveness of the KNN classifiers at different k levels and the comparisons shown. An effective method of classifying spam SMS, based on the metrics like F-measures, Precision, and recall score is recognized from the experiment results. The best performance was achieved with K = 4, where the classifier provided a high accuracy of 94.78% and strong results across all key performance metrics. The research highlights that feature selection plays a crucial role in improving classification efficiency by eliminating irrelevant or redundant features. Although KNN is a simple and effective approach, its scalability and real-time processing limitations suggest that future work should explore deep learning, ensemble models, or heuristic-based optimization for further improvements and support process innovation.

Keywords: SPAM, SMS, Feature Selection, KNN Neighbourhood Size, Classifier's Performance, Process Innovation

1. Introduction

Message service became a really necessary platform and a growing weakness for people and organizations as a result of its liable to misuse. The blind sending of unwanted messages is stated as spam [1]. As a result, users can bound to waste valuable time deleting spam emails. Presently, the heavy work on spam SMS filtering has persecuted the methods like decision trees, neural networks, Naive theorem classifiers, etc. Since SMS is a consistent source of communication without the internet, Spammers attempt to send fraudulent messages to gain financial benefits by unlawfully accessing credit card information. etc. so it has become very important to restrict the user to receive such SMS content [2]. If there is a structure to detect whether the receiving message is a SPAM or HAM, it is very useful for users to avoid scammers encroaching into the personal information stored on our devices. To develop such a system, an expert must identify whether the message's content is ham or SPAM using machine learning techniques [3]. To avoid these issues, people should be aware of scam messages, and at the same time, it will be very helpful if we identify the SMS as spam or not by any technique.

Filtering could be an extensive solution to the matter of spam [4]. There are so many methods proposed to detect spam messages so far. Machine learning techniques were effective in email spam filtering and offer safety to the users. The

***Corresponding author: B. Kalaiselvi (kalaigopal1973@gmail.com)**
DOI: https://doi.org/10.47738/jads.v6i2.654

 ISSN 2723-6471

Same method is being used for mobile devices in order to prevent SMS Spam issues. but due to the size of the text message and less formal language, it is challenging to recognize the spam content in SMS [5].

In this paper, we are using machine learning techniques like rule-based classifiers to classify SMS. The aim of this research work is to predict the best technique to improve the accuracy and predictions of SMS Spam Detection the SMS spam collection data is used to experiment with this method [6].

The role of KNN in SMS spam detection by following a structured approach. It begins with preprocessing SMS data, which involves cleaning and organizing messages into a structured feature set. Next, the study applies KNN with different values of K (ranging from 1 to 5) to identify the most effective configuration for classification. The performance of the classifier is then evaluated using key metrics such as accuracy, precision, recall, and F-measure, ensuring a comprehensive assessment of its effectiveness. By integrating KNN into SMS spam detection, the study highlights its practical application in real-world spam filtering systems, helping protect users from fraudulent and unsolicited messages.

2. Literature Review

With the increasing volume of spam messages, various machine learning-based SMS spam detection techniques have been developed to improve classification performance. One of the most effective approaches is the use of feature selection methods to refine the attributes used in training machine learning models, thereby enhancing accuracy and reducing computational complexity. This section reviews several related works that utilize K-Nearest Neighbors (KNN) and its variations in classification tasks, including feature selection, hybrid models, and ensemble learning techniques.

One study [6] explores a KNN-based approach for fake document detection, providing a practical and interpretable framework for classification. The method achieves notable accuracy improvements compared to baseline models, demonstrating the effectiveness of KNN in classification tasks. However, the study also identifies several limitations, particularly in terms of scalability and computational efficiency when handling large datasets. The study suggests that future research should integrate deep learning techniques to enhance detection accuracy further and improve model robustness in real-world applications. Despite these limitations, the proposed method remains a valuable contribution to the field of document forensics and lays the groundwork for future advancements in classification models.

Another study [7] highlights the effectiveness of ensemble learning techniques for classification tasks, particularly in fake document detection. By integrating multiple classifiers, the proposed method significantly enhances model accuracy compared to traditional approaches using individual classifiers. The results indicate that ensemble learning effectively mitigates the weaknesses of standalone classification models, leading to improved generalization and robustness. However, the study also identifies real-time efficiency as a challenge, suggesting that future research should focus on optimizing computational performance. Additionally, exploring deep learning-based ensemble techniques could further enhance classification accuracy and adaptability in large-scale datasets.

A study [8] focuses on feature selection in conjunction with KNN classification for improving classification performance. The proposed method employs a feature selection algorithm to identify the most relevant attributes before training the KNN classifier. The experimental results demonstrate that reducing the number of input features enhances both accuracy and computational efficiency, confirming the importance of feature selection in classification tasks. The study emphasizes that proper feature selection eliminates redundant and irrelevant attributes, preventing overfitting and ensuring that the classifier generalizes well across different datasets. This research highlights the need for more efficient feature selection techniques, which could be further improved using metaheuristic search algorithms or deep learning-based feature extraction methods.

A different approach [9] combines KNN with decision trees to create a hybrid classification model for fake document detection. In this method, KNN is first used to extract features and classify documents into initial categories, after which decision trees refine the classification results. The experimental evaluation reveals that this hybrid model outperforms both KNN and decision trees when used individually, demonstrating the advantages of combining multiple classification algorithms. The study further suggests that hybrid models enhance classification accuracy by leveraging

the strengths of different classifiers, thereby reducing classification errors. However, the research also notes that parameter tuning remains a critical factor in optimizing hybrid models, suggesting that future work should explore automated hyperparameter tuning methods to improve model performance further.

Another study [10] introduces a KNN-based classification method that incorporates Local Binary Patterns (LBP) for feature extraction. This approach utilizes LBP to extract texture-based features from documents, which are then classified using KNN to determine their authenticity. The experimental results confirm that integrating LBP with KNN significantly enhances classification accuracy, especially in cases where texture-based features play a crucial role in distinguishing between classes. The study highlights the potential of combining feature extraction techniques with machine learning classifiers to improve detection capabilities. However, the research also suggests that future work should explore deep learning-based feature extraction methods, such as Convolutional Neural Networks (CNNs), to further improve classification accuracy and feature representation.

These related works collectively demonstrate the wide applicability of KNN-based classifiers in fake document detection and classification tasks. They explore multiple aspects of classification enhancement, including feature extraction, ensemble learning, feature selection, hybrid models, and the integration of KNN with complementary techniques. The studies confirm that feature selection plays a crucial role in improving classification performance, as reducing irrelevant attributes enhances computational efficiency and prevents overfitting. Additionally, combining KNN with other classification techniques, such as decision trees and ensemble models, has proven to significantly enhance classification accuracy. The integration of texture-based feature extraction techniques, such as LBP, further expands the scope of KNN-based models, enabling more refined feature representation and improved classification results.

While these studies provide valuable insights into KNN-based classification, several challenges remain, including scalability issues, hyperparameter optimization, and real-time efficiency. Future research should explore the use of deep learning techniques to improve feature extraction, classification accuracy, and model scalability. Additionally, automated feature selection and hyperparameter tuning methods could further optimize classification performance, making KNN-based models more adaptable to real-world applications such as SMS spam detection and document forensics.

3. Material and Methods

The main objective of our method is to classify spam SMS messages for scamming threats as soon as the SMS is received on the device. For that, first, we collected a dataset that extracted the features from the messages (ham and spam) to generate a feature set. The total set is divided into training and testing purposes. We take the decision based on the performances of the classifiers on the feature sets. Figure 1 shows the system planning of our proposed approach. In the training phase, a binary classifier is generated by applying the feature vectors of spam and ham messages.

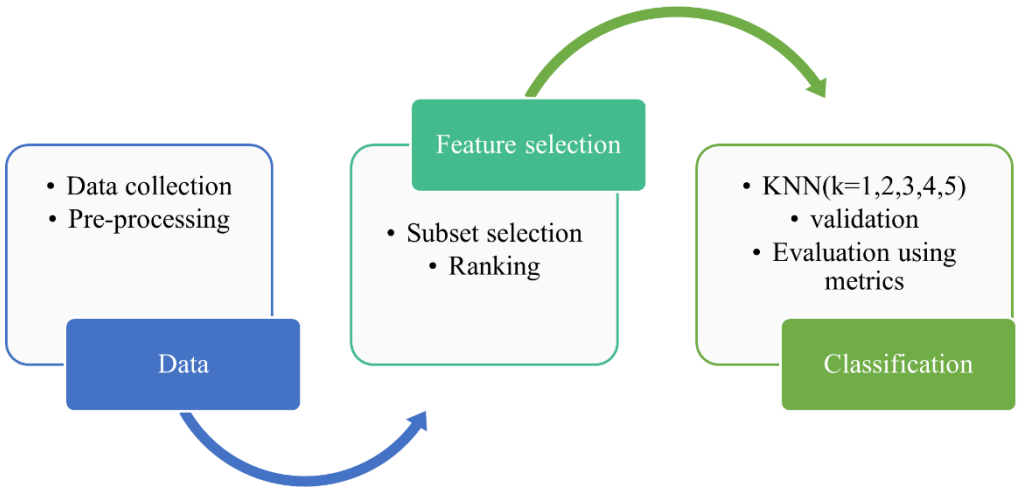

Figure 1. Flowchart of methodology

 ISSN 2723-6471

The KNN classifier is a supervised machine learning algorithm used for classification and regression tasks. It is an instance-based learning method that classifies data points based on their similarity to neighboring points. The classification is determined by a majority vote among the K nearest neighbors of a given instance, using distance metrics such as Euclidean distance. KNN is non-parametric and does not require a training phase, making it simple and easy to implement. However, its performance depends on the choice of K and can become computationally expensive for large datasets. In SMS spam detection, KNN helps classify messages as spam or ham by comparing new messages to previously labeled ones based on text similarity. In the testing phase, the classifier determines whether a new message is spam or not. At the end, we get classification results for different machine learning algorithms and performance is evaluated for each machine learning algorithm such that we can get the best algorithm for our proposed approach.

The practical implementation of this algorithm involves key decisions, such as selecting the number of exemplars, defining the method for measuring distance between a new data point and its nearest neighbors, and determining how to compute the final prediction when multiple neighbors (K-nearest neighbors) contribute to the classification ass shown in the figure 2. The final prediction can be derived through techniques like majority voting or averaging, depending on the task. Due to its reliance on storing and comparing instances from the dataset during classification, this method is often referred to as a "memory-based learner."

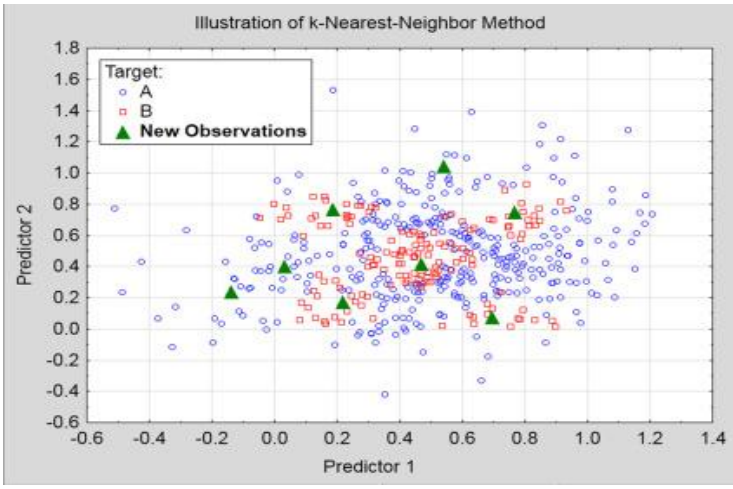

Figure 2. Classifying new observations using KNN method

3.1.Data Description

On the subject of mobile spam filtering, though there are few databases of valid SMS messages available on the Internet, it is not easy to find real samples of mobile phone spam. Thus, in this research to create the quantity for the purposes, we use data derived from Kaggle [11]. This SMS Spam Collection is a set of SMS-tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged according to being ham (legitimate) or spam. Out of 5,574, 425 are from the source The Grumble text Web site, 3375 are from NUS SMS Corpus (NSC), and 450 are from Caroline Tag's Ph.D. The thesis and 1324 are from Spam Corpus v.0 (figure 3). The files contain one message per line. Each line is composed of two columns: v1 contains the label (ham or spam) and v2 contains the raw text.

 ISSN 2723-6471

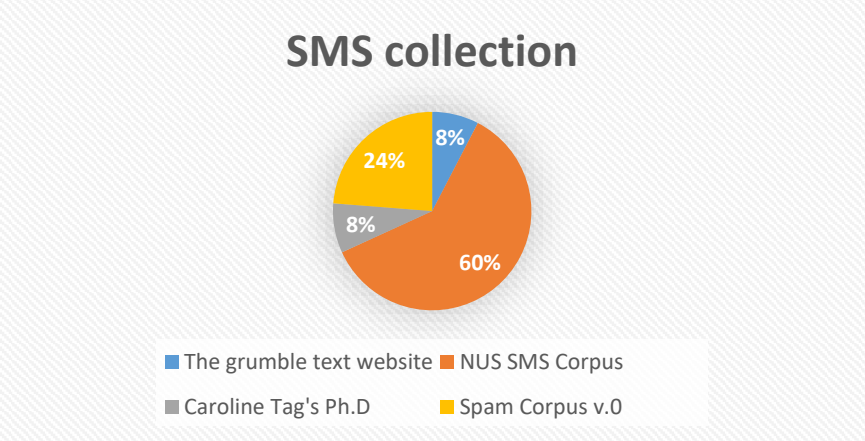

Figure 3. Pie chart of SMS Collection

3.2.Data Pre-processing

In this process, the data has to be cleaned by removing unwanted white spaces, and punctuations and tokenizing each word by stemming. The cleansed data is a list of words that helps to identify the similarities between the scamming keywords. The data processing phase in SMS spam detection involves systematically transforming raw text into a structured format for classification. The dataset, sourced from Kaggle, contains 5,574 messages labeled as spam or ham. The process begins with data preprocessing, where unnecessary elements such as punctuation and extra whitespace are removed, followed by tokenization and stemming to standardize text representation. Next, feature extraction and selection are performed using correlation-based ranking to retain only the most relevant attributes, improving classification accuracy. The dataset is then split into training and testing subsets, allowing the KNN classifier to learn from past examples and classify new messages based on similarity. Finally, performance evaluation is conducted using accuracy, precision, recall, F-measure, and ROC values to determine the optimal K value and feature set. This structured pipeline ensures efficient spam detection, achieving 94.78% accuracy through effective feature selection and KNN tuning.

3.3. Methods Based on Feature/Attribute Selection for Pre-processing

The feature selection and methods that do not use it, highlighting the advantages of reducing irrelevant or redundant features. However, the explanation could be more concrete by providing a direct comparison of results from both approaches rather than relying solely on theoretical descriptions. Including performance metrics, such as accuracy, precision, and computation time for models with and without feature selection, would strengthen the argument and offer clearer insights into its impact on SMS spam detection. Another major drawback is increased computational complexity, as processing a large number of features demands more memory and computational power, ultimately slowing down the classification process. Furthermore, reduced efficiency can make the classifier struggle with large datasets, making real-time spam detection impractical. By selecting only the most relevant features, the study optimizes KNN performance, achieving a higher accuracy of 94.78% while reducing computational costs and improving overall efficiency.

Feature Selection: These methods aim to identify and select the most relevant features from the dataset to improve classification performance [5]. By reducing the dimensionality of the data, feature selection techniques can eliminate irrelevant or redundant features, resulting in a more efficient and effective classifier. Examples of feature selection techniques include filter methods (e.g., correlation-based feature selection), wrapper methods (e.g., recursive feature elimination), and embedded methods (e.g., LASSO regularization).

Attribute Selection: Similar to feature selection, attribute selection focuses on selecting relevant attributes from the dataset. However, in this context, attributes typically refer to metadata or characteristics associated with the documents rather than the specific features extracted from them. Attribute selection techniques can help identify the most informative attributes that contribute to the classification task, thereby improving the accuracy and efficiency of the classifier.

ISSN 2723-6471

3.4. Methods Without Feature/Attribute Selection Pre-processing

Full Feature Set: In methods without feature or attribute selection pre-processing, the entire set of available features or attributes is used for classification. This means that all features or attributes are considered, without any attempt to remove irrelevant or redundant ones. While this approach may capture all potential information from the dataset, it can also lead to increased dimensionality, computational complexity, and potential noise or irrelevant information, which may negatively impact classification performance.

Dimensionality Reduction: In methods without explicit feature or attribute selection, dimensionality reduction techniques may be employed as a form of pre-processing. Dimensionality reduction methods aim to transform the dataset into a lower-dimensional space while preserving the most important information. Techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) can be used to reduce the dimensionality of the dataset without explicitly selecting specific features or attributes.

Methods based on feature/attribute selection for pre-processing involve explicitly selecting relevant features or attributes from the dataset, aiming to improve classification performance and reduce computational complexity. On the other hand, methods without this type of pre-processing may either consider the full set of features/attributes or employ dimensionality reduction techniques to reduce the dimensionality of the dataset. Both approaches have their advantages and considerations depending on the specific context and requirements of the classification task.

3.5. KNN Classifier

KNN is a type of instance-based learning that all computation is overdue until classification. It is the simplest of all machine learning algorithms. An instance is classified by majority votes of its neighbors by the object being assigned to the class most common among its k nearest neighbor (k is a positive small integer). The nearest neighbor is determined using similarity measure usually distance functions are user. Following are the distance function used by KNN [12].

We run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while keeping the algorithm's capability to precisely make predictions when it's given data it hasn't seen before. As we decrease the value of K to 1, our predictions become less stable. Inversely, as we increase the value of K, our predictions become more stable due to the majority choosing, and thus, more likely to make more accurate predictions. Eventually, we begin to witness an increasing number of errors. It is at this point we know we have pushed the value of K too far [13].

KNN is simple and easy to implement where there is no need to build a model, tune several parameters, or make additional assumptions. It can be used for classification, regression, and search. KNN's main disadvantage of becoming significantly slower as the volume of data increases makes it an impractical choice in environments where predictions need to be made rapidly. Moreover, there are faster algorithms that can produce more accurate classification and regression results.

4. Results and Discussion

This section elaborates on the evaluation of the proposed system using SMS datasets and the evaluation results follow. The first step in a Text Mining process is pre-processing [14] which aims to convert the unstructured information of the text files into a structured and ordered form, which can then be interpreted by the machine learning algorithms and also reduce the ultimate noise present in a collection and the space needed for storing it. In Pre-processing- It is used to distill unstructured data to a structured format. There are different pre-processing steps performed in Text mining such as tokenization, stop word removal, and stemming. In this section, we experimented with the cleaned data with different tree-based classifiers and examined the results.

A confusion matrix is a table used to summarize the performance of a classification model. Classification models are used to solve problems that have a definite outcome, such as predicting whether SMS is spam or not. We can measure the classification model quality metrics such as accuracy, precision, recall, F-measure and ROC [15].

ISSN 2723-6471

Metrics like accuracy, precision, recall, and F-measure are crucial for evaluating SMS spam classification. Accuracy measures overall correctness but can be misleading if the dataset is imbalanced. Precision ensures spam classifications are correct, reducing false positives, while recall focuses on detecting actual spam to minimize false negatives. F-measure (F1-score) balances precision and recall, providing a reliable performance metric [16], [17]. These metrics help assess the KNN classifier, ensuring it effectively distinguishes between spam and ham while minimizing errors.

Accuracy is a metric that measures how often a machine learning model correctly predicts the outcome. It can calculate accuracy by dividing the number of correct predictions by the total number of predictions [18]. Table 1 evaluates the accuracy performance of the KNN classifier at different values of K (1, 2, 3, 4, 5) using a correlation-based attribute selection method. The results indicate that K = 1 provides high accuracy but is more sensitive to noise, while K = 2 and K = 3 show slight improvements but with some fluctuations. The highest accuracy, 94.78%, is achieved at K = 4, suggesting that a moderate neighborhood size helps balance generalization and noise reduction. This confirms that feature selection improves accuracy by eliminating irrelevant attributes and that KNN = 4 is the most effective configuration for SMS spam detection in this study.

Table 1. Accuracy performance of KNN classifier (k=1,2,3,4,5) with Ranker attribute selection method

No.	List of Attribute selected	Number of attributes selected	Attribute selection method	KNN Classifier's accuracy percentage				
			Attribute evaluator/search method	K =1	K =2	K = 3	K = 4	K = 5
1.	Selected Attributes I: 6, 20, 57, 43, 56, 35	6	Correlation attribute eval / Ranker	93.36%	93.56%	93.18%	93.72%	93.68%
2.	Selected Attributes II: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64	11	Correlation attribute eval / Ranker	94.10%	94.72%	94.13%	94.78%	94.42%
3.	Selected Attributes III: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64, 87, 2, 71, 22	15	Correlation attribute eval / Ranker	94.35%	94.62%	93.79%	94.69%	94.17%

Figure 4 shows that the accuracy for the selected attribute II is comparatively higher in the KNN classifier at K = 4 and that is 95.05%. Table 2 presents the precision performance of the K-Nearest Neighbors (KNN) classifier using different values of K (1, 2, 3, 4, and 5), evaluated based on three different sets of selected attributes. The attributes were selected using the Correlation Attribute Evaluator with the Ranker search method. The table provides a comparative analysis of how the number of selected attributes affects the precision of the KNN classifier at different values of K.

ISSN 2723-6471

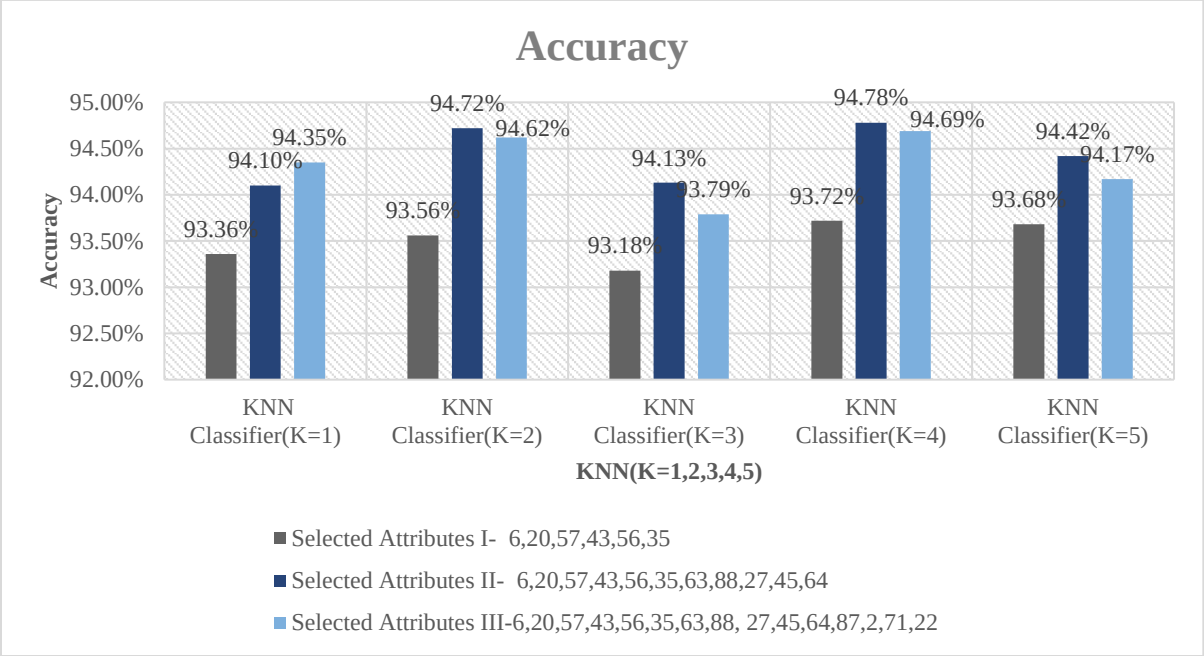

Figure 4. KNN classifier (k=1, 2, 3, 4, 5) vs. Accuracy

The first set, Selected Attributes I, consists of six attributes and yields the lowest precision among the three attribute selections, with values ranging from 0.933 to 0.939 across different K values. The second set, Selected Attributes II, includes eleven attributes, showing a notable improvement in precision, particularly at K = 1 (0.947) and K = 4, 5 (0.950). Finally, the third set, Selected Attributes III, comprises fifteen attributes and achieves the highest precision, especially at K = 1 (0.952). However, for K ≥ 2, the precision stabilizes around 0.947 - 0.950, indicating that adding more attributes does not necessarily yield further improvements beyond a certain point.

Table 2. Precision performance of KNN classifier (k=1,2,3,4,5) with Ranker attribute selection method

No.	List of Attribute selected	Number of attributes selected	Attribute selection method	KNN Classifier's Precision value				
			Attribute evaluator/search method	K = 1	K = 2	K = 3	K = 4	K = 5
1.	Selected Attributes I: 6, 20, 57, 43, 56, 35	6	Correlation attribute eval / Ranker	0.939	0.933	0.934	0.936	0.939
2.	Selected Attributes II: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64	11	Correlation attribute eval / Ranker	0.947	0.948	0.948	0.95	0.950
3.	Selected Attributes III: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64, 87, 2, 71, 22	15	Correlation attribute eval / Ranker	0.952	0.948	0.947	0.95	0.949

Figure 5 visually represents the precision values of the KNN classifier for the three selected attribute sets at different K values. The graph highlights that Selected Attributes I (gray bars) consistently produce the lowest precision, whereas Selected Attributes II (dark blue bars) and Selected Attributes III (light blue bars) show a significant improvement. The highest precision value, 0.952, is observed at K = 1 for Selected Attributes III, confirming that including a broader set of attributes enhances classification performance when using a small K value. However, for K = 2 to 5, precision remains relatively stable at approximately 0.95, regardless of the number of attributes.

ISSN 2723-6471

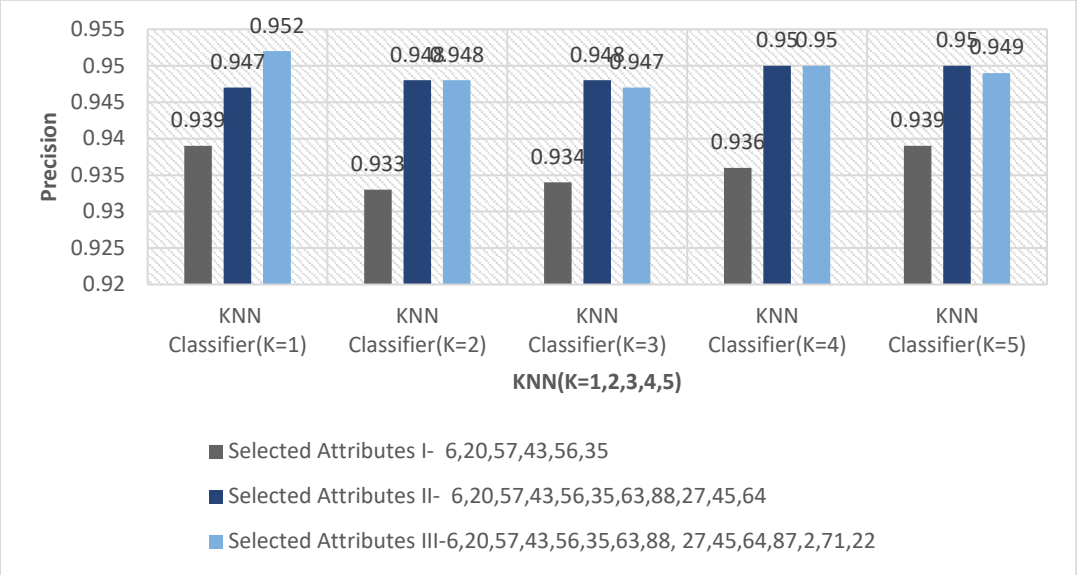

Figure 5. KNN classifier (k=1, 2, 3, 4, 5) vs. Precision

The results suggest that while increasing the number of selected attributes can enhance precision, the effect diminishes after reaching an optimal set. The best-performing configuration occurs when K = 1 and Selected Attributes III, demonstrating that a larger set of relevant attributes contributes to better classification performance at lower K values. However, at higher K values (K ≥ 2), the precision gain is minimal, indicating that beyond a certain number of attributes, additional features do not necessarily improve the model's precision. These findings emphasize the importance of optimal feature selection in maximizing classification accuracy while maintaining computational efficiency.

Table 3 presents the recall performance of the K-Nearest Neighbors (KNN) classifier across different values of K (1, 2, 3, 4, and 5). The evaluation is based on three different sets of selected attributes using the Correlation Attribute Evaluator with the Ranker search method. Recall measures the model's ability to correctly identify positive instances from all actual positive cases in the dataset, making it a critical metric in applications where missing positive instances could have significant consequences, such as medical diagnostics or fraud detection [19], [20].

Table 3. Recall performance of KNN classifier (k=1,2,3,4,5) with Ranker attribute selection method

No.	List of Attribute selected	Number of attributes selected	Attribute selection method	KNN Classifier's Recall value				
			Attribute evaluator/search method	K = 1	K = 2	K = 3	K = 4	K = 5
1.	Selected Attributes I: 6, 20, 57, 43, 56, 35	6	Correlation attribute eval / Ranker	0.934	0.936	0.932	0.937	0.937
2.	Selected Attributes II: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64	11	Correlation attribute eval / Ranker	0.941	0.947	0.941	0.948	0.948
3.	Selected Attributes III: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64, 87, 2, 71, 22	15	Correlation attribute eval / Ranker	0.943	0.946	0.938	0.947	0.942

The results indicate that Selected Attributes I (6 attributes) yield the lowest recall values, ranging from 0.932 to 0.937 across different K values. This suggests that a smaller set of features limits the model's ability to detect positive instances. Selected Attributes II (11 attributes) demonstrates a significant improvement, achieving the highest recall values at K = 2 (0.947), K = 4 (0.948), and K = 5 (0.948). Meanwhile, Selected Attributes III (15 attributes), despite containing additional features, does not consistently outperform Selected Attributes II. While it achieves the highest

ISSN 2723-6471

recall at K = 1 (0.943), its recall values at K = 3 and K = 5 are slightly lower than those of Selected Attributes II. This suggests that adding more features beyond a certain point may introduce redundancy, leading to marginal improvements or slight reductions in performance.

Figure 6 provides a visual representation of the recall performance for the different attribute sets across varying values of K. The results confirm that Selected Attributes I consistently achieve the lowest recall, while Selected Attributes II achieves the highest recall, particularly at K = 4 and K = 5, where it reaches 0.948. Although Selected Attributes III maintains high recall values, its performance fluctuates slightly and does not consistently surpass that of Selected Attributes II. Notably, Selected Attributes II outperforms Selected Attributes III at K = 3 (0.941 vs. 0.938) and K = 5 (0.948 vs. 0.942), reinforcing the idea that selecting an optimal set of attributes is more effective than simply increasing the number of features.

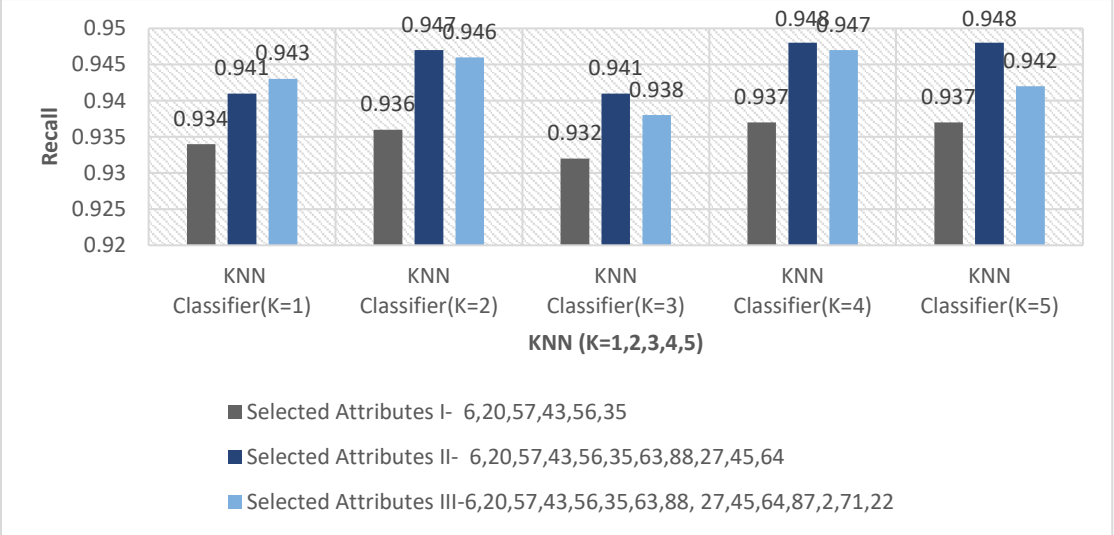

Figure 6. KNN classifier (k=1, 2, 3, 4, 5) vs. Precision

Overall, the results suggest that recall performance improves with an increased number of selected attributes, but the benefit diminishes beyond a certain threshold. The best-performing configuration is Selected Attributes II at K = 4 and K = 5, achieving the highest recall of 0.948. This finding highlights the importance of optimal feature selection, as excessive attributes may not necessarily contribute to better model performance. Additionally, selecting the appropriate K value is crucial, as K = 4 and K = 5 yield the best recall values, indicating a balance between sensitivity and classification effectiveness.

Table 4 presents the F-measure (F1-score) performance of the K-Nearest Neighbors (KNN) classifier across different values of K (1, 2, 3, 4, and 5). The F-measure is a crucial metric that balances precision and recall by calculating their harmonic mean, making it particularly useful for evaluating classification models where both false positives and false negatives must be minimized. The study evaluates the classifier's performance based on three different sets of selected attributes, chosen using the Correlation Attribute Evaluator with the Ranker search method. The goal is to determine the impact of feature selection on the classifier's ability to achieve an optimal balance between precision and recall.

Table 4. F- Measure performance of KNN classifier (k=1,2,3,4,5) with Ranker attribute selection method

No.	List of Attribute selected	Number of attributes selected	Attribute selection method	KNN Classifier's F-measure value				
			Attribute evaluator/search method	K = 1	K = 2	K = 3	K = 4	K = 5
1.	Selected Attributes I: 6, 20, 57, 43, 56, 35	6	Correlation attribute eval / Ranker	0.936	0.934	0.933	0.937	0.938

2.	Selected Attributes II: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64	11	Correlation attribute eval / Ranker	0.943	0.948	0.944	0.949	0.949
3.	Selected Attributes III: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64, 87, 2, 71, 22	15	Correlation attribute eval / Ranker	0.946	0.947	0.941	0.948	0.944

The results in Table 4 indicate that Selected Attributes I (6 attributes) produce the lowest F-measure values, ranging between 0.933 and 0.938, suggesting that a limited number of features restricts the classifier's performance. In contrast, Selected Attributes II (11 attributes) achieves significant improvements, attaining the highest F-measure scores at K = 2 (0.948), K = 4 (0.949), and K = 5 (0.949). These results suggest that adding more relevant attributes enhances the classifier's overall effectiveness. However, Selected Attributes III (15 attributes) does not consistently outperform Selected Attributes II, despite including additional features. While it achieves the highest F-measure at K = 1 (0.946), its scores at K = 3 (0.941) and K = 5 (0.944) are slightly lower than those of Selected Attributes II. This finding suggests that increasing the number of features beyond a certain threshold may introduce redundancy or noise, leading to marginal reductions in performance rather than further improvements.

Figure 7 provides a visual representation of the F-measure performance of the KNN classifier for different K values and attribute selection sets. The graph confirms that Selected Attributes I consistently yields the lowest F-measure values, while Selected Attributes II achieves the highest scores at K = 4 and K = 5, reaching 0.949. Although Selected Attributes III demonstrates strong performance, it does not consistently surpass Selected Attributes II, particularly at K = 3 (0.941) and K = 5 (0.944). The pattern observed in the figure indicates that while increasing the number of selected attributes generally improves F-measure performance, the benefit diminishes beyond an optimal set (11 attributes in this case).

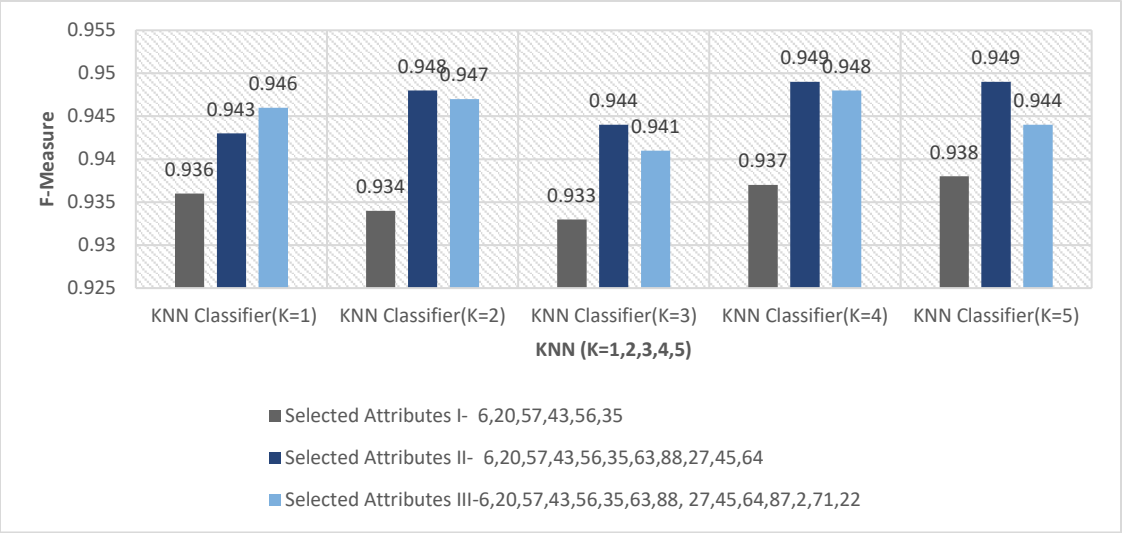

Figure 7. KNN classifier (k=1, 2, 3, 4, 5) vs. F-measure

The findings from Table 4 and Figure 7 highlight the importance of optimal feature selection in classification tasks. The best-performing configuration is Selected Attributes II at K = 4 and K = 5, where the F-measure reaches 0.949, confirming that an optimal balance exists between feature selection and model performance. Adding more attributes beyond this threshold (Selected Attributes III) does not necessarily lead to further improvements and, in some cases, may slightly reduce the model's effectiveness due to feature redundancy. Additionally, selecting an appropriate K value is critical, as K = 4 and K = 5 provide the best results, demonstrating that an optimal combination of feature selection and KNN hyperparameter tuning can significantly enhance classification accuracy.

Table 5 presents the Receiver Operating Characteristic (ROC) performance of the K-Nearest Neighbors (KNN) classifier across different values of K (1, 2, 3, 4, and 5). The ROC metric evaluates a classifier's ability to distinguish between positive and negative instances, with higher values indicating better performance. The ROC value represents the area under the ROC curve (AUC), where a value closer to 1.0 signifies a highly effective classification model. In

ISSN 2723-6471

this study, three different sets of selected attributes were analyzed using the Correlation Attribute Evaluator with the Ranker search method to assess the impact of feature selection on classification performance.

Table 5. ROC performance of KNN classifier (k=1,2,3,4,5) with Ranker attribute selection method

No.	List of Attribute selected	Number of attributes selected	Attribute selection method	KNN Classifier's ROC value				
			Attribute evaluator/search method	K = 1	K = 2	K = 3	K = 4	K = 5
1.	Selected Attributes I: 6, 20, 57, 43, 56, 35	6	Correlation attribute eval / Ranker	0.890	0.919	0.932	0.939	0.942
2.	Selected Attributes II: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64	11	Correlation attribute eval / Ranker	0.912	0.938	0.954	0.962	0.962
3.	Selected Attributes III: 6, 20, 57, 43, 56, 35, 63, 88, 27, 45, 64, 87, 2, 71, 22	15	Correlation attribute eval / Ranker	0.926	0.947	0.957	0.962	0.962

The results in Table 5 indicate that Selected Attributes I (6 attributes) achieve the lowest ROC values, ranging from 0.890 to 0.942, across all values of K. While performance improves as K increases, it remains significantly lower than the other two attribute selection sets. In contrast, Selected Attributes II (11 attributes) demonstrates substantial improvement, achieving higher ROC values at K = 3 (0.954), K = 4 (0.962), and K = 5 (0.962). This suggests that increasing the number of selected attributes enhances the classifier's ability to distinguish between positive and negative instances. Selected Attributes III (15 attributes) achieves slightly higher ROC values than Selected Attributes II at K = 1, 2, and 3, but beyond K = 4 and K = 5, both sets reach the same maximum ROC value of 0.962. This finding suggests that adding more attributes beyond a certain threshold does not necessarily contribute to further improvements in classification performance.

Figure 8 provides a visual representation of the ROC performance of the KNN classifier across different values of K and different sets of selected attributes. The results confirm that Selected Attributes I consistently produce the lowest ROC values, whereas Selected Attributes II and III achieve the highest values at K = 4 and K = 5, reaching 0.962. While Selected Attributes III slightly outperforms Selected Attributes II at K = 1, 2, and 3, the difference is minor, and their performance converges at K = 4 and K = 5. This indicates that increasing the number of attributes can enhance classification performance up to a certain point, beyond which additional attributes may introduce redundancy without significant benefits.

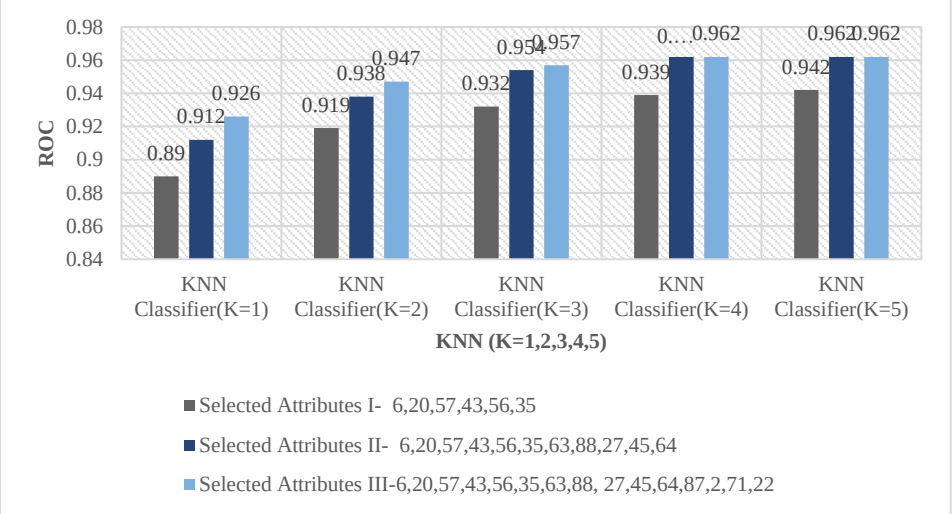

Figure 8. KNN classifier (k=1, 2, 3, 4, 5) vs. ROC

 ISSN 2723-6471

The findings from Table 5 and Figure 8 highlight the importance of selecting an optimal number of attributes to maximize classifier performance. The best-performing configuration is Selected Attributes II and III at K = 4 and K = 5, achieving the highest ROC value of 0.962. This suggests that feature selection plays a critical role in optimizing classification performance, as selecting only the most relevant attributes leads to better results than simply increasing the number of attributes. Additionally, choosing an appropriate K value is essential, as K = 4 and K = 5 consistently yield the best ROC values, indicating an optimal balance between classifier complexity and performance. These findings emphasize the importance of both feature selection and hyperparameter tuning in enhancing the effectiveness of the KNN classifier.

5. Conclusion

This study demonstrates the effectiveness of the KNN classifier in SMS spam detection, with K = 4 achieving the highest accuracy of 94.78% when using an optimal feature selection approach. The results emphasize the importance of feature selection in improving classification performance by reducing noise and enhancing computational efficiency. The findings confirm that KNN is a reliable method for spam filtering, but its performance is significantly influenced by the choice of K and the selected attributes.

Furthermore, the study shows that fine-tuning the neighborhood size and selecting an optimal set of attributes (11 features) leads to a peak accuracy of 94.78%. While this approach balances efficiency and complexity, further improvements can be achieved by exploring advanced search techniques or heuristic methods for hyperparameter tuning.

For future work, ensemble learning techniques such as Random Forest or hybrid models could be explored to further enhance accuracy and robustness. Additionally, deep learning approaches, including Recurrent Neural Networks (RNNs) and Transformer-based models, could be applied to handle complex text structures in spam messages. Implementing real-time spam filtering with optimized models can further improve user security and experience. Moreover, expanding the dataset to include multilingual SMS spam messages would help develop a more generalized and adaptable spam detection system for diverse user bases.

6. Declarations

6.1. Author Contributions

Conceptualization: S.P., B.K., K.S., and M.B.; Methodology: B.K.; Software: S.P.; Validation: S.P., B.K., and M.B.; Formal Analysis: S.P., B.K., and M.B.; Investigation: S.P.; Resources: B.K.; Data Curation: B.K.; Writing Original Draft Preparation: S.P., B.K., and M.B.; Writing Review and Editing: B.K., S.P., and M.B.; Visualization: S.P. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Gadde, A. Lakshmanarao and S. Satyanarayana, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021*, vol. 7, no. Jun., pp. 358-362, doi: 10.1109/ICACCS51430.2021.9441783.

[2] V. K. Singh and S. Bhardwaj, "Spam mail detection using classification techniques and global training set," *in Intelligent Computing and Information and Communication, S. Bhalla, V. Bhateja, A. Chandavale, A. Hiwale, and S. Satapathy, Eds. Singapore: Springer, 2018*, vol. 673, no. Jan., pp. 623–632. doi: 10.1007/978-981-10-7245-1_61.

[3] B. Zhou, Y. Yao, and J. Luo, "Cost-sensitive threeway email spam filtering," *J. Intell. Inf. Syst.*, vol. 42, no. 1, pp. 19–45, 2014.

[4] A. Kapoor, D. Saikia, and I. Dhawan, "SMS spam detection using machine learning approach," International Journal of Research in Science and Technology, vol. 14, no. 1, pp. 10–17, Jan.–Mar. 2024. doi: 10.37648/ijrst.v14i01.002

[5] Y. Y. Loong and C. Lee, "Detection of Fake Documents by Tree-Based Machine Learning Algorithms," Knowl. Trans. Appl. Mach. Learn., vol. 1, no. 3, pp. 1–17, Jul. 2023, doi: 10.59567/ktAML.V1.03.01.

[6] K. S. F. Azam, F. F. Riya, S. Al Mamun and M. S. Kaiser, "A Novel Approach of Detecting Image Forgery Using GLCM and KNN," *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh*, vol. 2021, no. Apr., pp. 125-129, 2021, doi: 10.1109/ICICT4SD50815.2021.9397057.

[7] O. J. Ogunsuyi and A. K. Ojo, "K-Nearest Neighbors Bayesian approach to false news detection from text on social media," International Journal of Engineering and Management Engineering, vol. 4, no. Aug., pp. 22–32, Aug. 2022. doi: 10.5815/ijeme.2022.04.03.

[8] S. K. W. Haidar, A. Sharma, S. Dahiya, V. Venkateswaran, and K. Kumar, "Machine learning based framework for unmasking bogus reviews in online shopping," *International Journal of Communication Networks and Information Security*, vol. 16, no. 4, pp. 238-248, 2024.

[9] S. Mohammed, N. Al-Aaraji, and A. Al-Saleh, "Knowledge rules-based decision tree classifier model for effective fake accounts detection in social networks," *International Journal of Safety and Security Engineering*, vol. 14, no. 4, pp. 1243–1251, Aug. 2024. doi: 10.18280/ijsse.140421.

[10] P. Gonasagi, S. S. Rumma, and M. Hangarge, "Text-Source Identification Using a Knowledge-Based Decision Tree Classifier," *Proc. 1st Int. Conf. Adv. Comput. Vis. Artif. Intell. Technol. (ACVAIT 2022)*, vol. 1, no. Aug., pp. 250–261, 2022. doi: 10.2991/978-94-6463-196-8_20.

[11] "SMS Spam Collection Dataset," *Kaggle*, Available: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset.

[12] K. Kaur and M. Kumar, "Spam Detection using KNN, Back Propagation and Recurrent Neural Network," *Int. J. Eng. Res. Technol. (IJERT)*, vol. 4, no. 9, pp. 557-564, 2015.

[13] S. Suryawanshi, A. Goswami and P. Patil, "Email Spam Detection : An Empirical Comparative Study of Different ML and Ensemble Classifiers," *2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India*, vol. 9, no. Jan., pp. 69-74, 2019. doi: 10.1109/IACC48062.2019.8971582.

[14] M. Sahami, S. Dumais, D. Heckerman, dan E. Horvitz, "A Bayesian approach to filtering junk e-mail," *AAAI Technical Report WS-98-05*, vol. 98, no. 1, pp. 55-62, 1998.

[15] S. J. Delany, P. Cunningham, and B. Smith, "ECUE: a spam filter that uses machine learning to track concept drift," *in Proceedings of the 2006 Conference on ECAI: 17th European Conference on Artificial Intelligence, Riva del Garda, Italy*, vol. 17, no. Sept., pp. 1-6, August 29 – September 1, 2006. doi: 10.21427/740tje03.

[16] A. M. Wahid, T. Hariguna and G. Karyono, "Optimizing Feature Extraction for Website Visuals: A Comparative Study of AlexNet and Inception V3," *2024 12th International Conference on Cyber and IT Service Management (CITSM), Batam, Indonesia*, vol. 12, no. 1, pp. 1-6, doi: 10.1109/CITSM64103.2024.10775681.

[17] H. Hanafi, A. Pranolo, Y. Mao, T. Hariguna, L. Hernandez, and N. F. Kurniawan, "IDSX-Attention: Intrusion detection system (IDS) based hybrid MADE-SDAE and LSTM-Attention mechanism," *International Journal of Advances in Intelligent Informatics*, vol. 9, no. 1, pp. 121-135, Mar. 2023. doi: 10.26555/ijain.v9i1.942

ISSN 2723-6471

[18] A. Ruangkanjanases and T. Hariguna, "Examining user satisfaction and continuous usage intention of digital financial advisory platforms," *HighTech and Innovation Journal*, vol. 6, no. 1, pp. 216–235, Mar. 2025. doi: 10.28991/HIJ-2025-06-01-015.

[19] B. H. Hayadi and T. Hariguna, "Determinants of student engagement and behavioral intention towards mobile learning platforms," *Contemporary Educational Technology*, vol. 17, no. 1, pp. 1-22, 2025. doi: 10.30935/cedtech/15774.

[20] T. T. Tin, K. J. Xin, A. Aitizaz, L. K. Tiung, T. C. Keat, and H. Sarwar, "Machine learning based predictive modelling of cybersecurity threats utilising behavioural data," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 9, pp. 832-840, 2023, doi: 10.14569/IJACSA.2023.0140987.