Machine Learning-Based Optimization Algorithms for Spam SMS Classification

Lipsa Das
Amity Institute of Information Technology, Amity University, Uttar Pradesh, India
Lipsaentc9@gmail.com (corresponding author)

Laxmi Ahuja
Amity Institute of Information Technology, Amity University, Uttar Pradesh, India
lahuja@amitu.edu

Adesh Pandey
KIET Group of Institutions, Ghaziabad, Uttar Pradesh, India
ak.pandey@kiet.edu

Received: 10 September 2025 | Revised: 20 October 2025 and 2 November 2025 | Accepted: 3 November 2025

 DOI: https://doi.org/10.48084/etasr.14670

ABSTRACT

Modern communication networks are plagued by spam Short Message Service (SMS), which invade users' personal spaces and pose significant security threats. This study proposes a novel spam SMS categorization method that combines Machine Learning (ML) techniques with the Golden Search Optimizer (GSO) and the Whale Optimization Algorithm (WOA). The hybrid Golden Search Optimizer–Whale Optimization Algorithm (GSOWOA) optimization technique is integrated with Random Forest (RF) models to improve performance and convergence time. Experimental findings reveal that categorization accuracy is much higher than that of classical ML approaches. By fine-tuning model hyperparameters, the optimization process reduces both false positive and false negative classifications, enhancing overall performance. The results demonstrate an effective and systematic approach for improving spam SMS categorization systems. This strategy has been beneficial for optimally tuning hyperparameters, leading to high classification performance for models such as Extreme Gradient Boosting (XGBoost) and RF. The main experimental evaluation indicates that the proposed framework achieved a total accuracy of 98.9% and a precision of 97.85%, with a 35% faster convergence rate than conventional protocols and general metaheuristic methods. Notably, GSOWOA demonstrated a strong resistance against overfitting, coupled with computational efficiency, making it viable for real-time spam detection on edge devices. These results provide evidence of the practical benefits of implementing a hybrid optimization process to achieve high performance and optimal resource utilization for SMS filtering.

Keywords-spam SMS; Random Forest (RF); classification; optimization algorithm; Extreme Gradient Boosting (XGBoost)

I. INTRODUCTION

The present-day digital environment is characterized by the widespread presence of mobile devices and the dominant use of text message communication, which forms an essential component of everyday life. Nevertheless, the ease and convenience of Short Message Service (SMS) communication are being undermined by the widespread occurrence of spam messages [1]. These unsolicited texts not only overwhelm consumers with undesirable information but also present potential risks to their personal privacy and security. The accurate detection and categorization of spam SMS are now major issues in information security [2, 3].

Due to the increasing threat of SMS spam and its economic implications, attention has shifted to technical challenges in SMS-specific detection, including the dynamism of spam design, high false positive rates, and limitations of existing Machine Learning (ML)–based classifiers in handling such dynamism [4]. Unlike traditional ML and Deep Learning (DL) techniques, which have proven successful in SMS spam detection, their effectiveness is compromised when confronted with dynamic and adaptive spam evolution [5]. Current metaheuristic algorithms also have limitations, despite performing well for parameter optimization, such as premature convergence, imperfect parameter evolution, and excessive computational cost. Thus, a strong optimization algorithm that can maintain a balance between exploration and exploitation with minimal training time is required [6]. To address this problem, the hybrid Golden Search Optimizer–Whale Optimization Algorithm (GSOWOA) is proposed, providing an

efficient way to tune optimal classifier hyperparameters and thereby enhancing the adaptability, accuracy, and computational efficiency in SMS spam detection.

Text classification within ML has been motivated by the prevalence of unsolicited SMS messages, or spam. The effectiveness of conventional techniques, such as rule-based systems and simple heuristics, has been reported to be limited in coping with the dynamic and sophisticated nature of spam messages [7]. Therefore, there is an increasing demand for sophisticated methods that can learn autonomously and adapt to new patterns and changes within spam content. The goal of this paper is to address this challenging problem by developing a new hybrid framework combining ML with the optimization capabilities of GSOWOA. The purpose of this work is to achieve higher accuracy and effectiveness in spam SMS classification, offering users more reliable protection against new spam messages. This work extends previous research by introducing a new model in which the GSOWOA optimization technique is incorporated into the ML models. The rationale for this integration is that optimization algorithms can help optimize model parameters, leading to higher classification accuracy and faster convergence. The benefits of such a fusion are twofold: on one side, it improves the performance of spam SMS categorization, and on the other, it contributes to the broader understanding of the interplay between optimization techniques and ML models.

The major contributions of this paper are as follows:

- Present a hybrid optimization model (GSOWOA) for hyperparameter tuning of conventional ML classifiers to improve the performance of SMS content filtering for spam detection. This combined model takes advantage of the exploration capability of Golden Search Optimization (GSO) as well as the exploitation capability of Whale Optimization Algorithm (WOA), enhancing convergence speed and classification accuracy compared with existing single–optimizer approaches.
- Conduct extensive experimental studies using different ML classifiers (Naive Bayes Classifier (NBC), Krill Herd Optimization (KHO), Random Forest (RF), and Fuzzy-based Recurrent Neural Network-based Harris Hawk Optimization (FRNN–HHO)) to demonstrate the effectiveness, consistency, and generalization capability of the proposed optimization scheme on SMS spam datasets.
- Perform a comparative study with current state-of-the-art detectors. The proposed method outperforms the baseline ML and Large Language Model (LLM)–driven detectors such as SpaLLM-Guard and Mixtral 8×7B in terms of higher accuracy and lower computational complexity under very low false positive/false negative rates.

II. LITERATURE REVIEW

Unwanted SMS spam is pervasive in this information technology era, highlighting the need for effective methods for both classification and filtering. This paper is organized in a manner that allows a complete and methodical presentation of work regarding spam SMS message classification. The focus is primarily on ML tools, with a growing interest in optimization techniques such as WOA and KHO [8, 9].

Authors in [10] introduced an innovative methodology for filtering SMS spam using the Dendritic Cell Algorithm (DCA) and a bio-inspired KHO algorithm. Experiments showed that the suggested model outperformed well-known ML classifiers such as Extreme Gradient Boosting (XGBoost) and Naive Bayes (NB). Authors in [11] conducted an independent investigation in which they devised a multi-filter system incorporating an ensemble of different ML classifiers. The approach employed three distinct classification approaches: NB, Support Vector Machines (SVM), and Naive Bayes Multinomial (NBM). The results demonstrated the flexibility of their model, which was integrated partially or fully into servers and mobile applications.

In the study conducted by authors in [12], genetic programming was employed to reduce false positive errors in SMS spam filtering. Results indicated that the approach effectively improved SMS spam classification accuracy as generations progressed. Other evolutionary techniques have also been found effective in SMS spam classification. Authors in [13] proposed a new DCA-based approach applied on NB and SVM classifiers across different features sets. Inspired by an artificial immune system, authors in [14] proposed an iterative collaborative and adaptive filtering method, which also incorporated an artificial immune system for blocking SMS spam [15].

Authors in [16] compared various ML algorithms for SMS spam detection, employing four types of ML methods, finding that neural networks improved the performance of all other classifiers. Authors in [17] analyzed ML classifier performance for SMS categorization in Bahasa Indonesia. Authors in [18] proposed a computational model for classifying spam text messaging, including classifiers such as K-Nearest Neighbors (K-NN), Categorization and Regression Trees (CART), NB, and SVM, as well as ensemble classifiers such as RF, Adaboost, and voting methods. Their experimental results indicated that ensemble learning based on RF achieved the highest classification accuracy.

Authors in [19] presented a significant work implementing an enhanced Hidden Markov Model (HMM) with weighted word features. Their experimental results indicated that the weighted HMM approach outperformed the Long Short-Term Memory (LSTM) model in terms of accuracy and efficiency. In SMS spam classification, a DL-based solution combining LSTM and Convolutional Neural Networks (CNN) has been proposed by authors in [20]. The experimental results showed strong performance of DL models under three configurations. Moreover, regularization methods such as dropout improved classification performance [21].

Authors in [22] proposed an interesting hybrid model combining Recurrent Neural Networks (RNN) and LSTM. The results revealed that the model effectively predicted patterns based on vector sets, achieving improved accuracy with moderate runtime. Authors in [23] analyzed the importance of hyperparameters in achieving optimal results.

Authors in [24] introduced a swarm optimization–based approach for spam filtering of tweets. An ML model was trained on a dataset specifically tailored for spam detection. WOA was applied for feature selection, and Stochastic Gradient Descent (SGD) was incorporated into the WOA objective function. An Adaboost classifier trained with selected features achieved the best results, reaching a record accuracy of 99.85% with only seven features in approximately 17.9 s [25].

Authors in [26] integrated NB with Particle Swarm Optimization (PSO) for e-mail spam/ham classification. The model, trained on 1,000 emails from the Ling dataset, employed Correlation-based Feature Selection (CFS) and outperformed traditional NB in precision, recall, F-measure, and accuracy, achieving values above 94% for all metrics.

Authors in [27] proposed a hybrid fuzzy-based network architecture that integrates fuzzy logic with RNNs to improve accuracy in sentiment analysis tasks. The hybridization of RNNs with fuzzy logic provides a significant advantage, as the detection and classification of fake text data are strongly influenced by fuzzy reasoning. However, these methods have not achieved adequate performance in terms of accuracy and error rates. Hence, the suggested architecture incorporates a hybrid approach that effectively analyzes classified emails, resulting in improved efficiency.

Authors in [28] presented a methodology for emotion analysis on Chinese short messages, which is significant for monitoring and gaining insights from social media platforms. The proposed approach, Attention-of-Emoticons Based CNN (AEB-CNN), combines attention mechanisms and emoticon information with CNNs to improve accuracy. Authors in [29] examined the efficacy of three ML models for detecting spam images, using Canny edge detection as a supporting technique. The models included SVM, CNN, and Multi-Layer Perceptron (MLP).

Authors in [30] performed an in-depth comparison of ML-based spam detectors, considering both shallow and DL techniques using the novel Super SMS Dataset. They found that conventional ML and deep neural models achieved good performance on static corpora but were susceptible to evasion-based strategies like character obfuscation, homoglyph attacks that mimic human-like behavior, or concept drift over time, highlighting the limitations of current anti-spam systems. To mitigate these issues, authors in [31] proposed SpaLLM-Guard and empirically evaluated open-source and commercial LLMs, including GPT-4, DeepSeek, LLAMA-2, and Mixtral, across zero-shot, few-shot, and fine-tuning approaches for SMS spam detection. Their findings showed that fine-tuned LLMs with Mixtral (8×7B) reached accuracy up to 98.6% with balanced false positive and false negative rates below 2%, demonstrating higher resilience against adversarial manipulations and concept drift. Altogether, these studies exemplify a shift in paradigm from conventional ML-based spam filtering to LLM-driven adaptive schemes capable of maintaining high detection accuracy in the face of evolving spam strategies.

Table I presents the critical analysis and research gaps of existing methods. The literature review underscores the evolution of spam SMS classification techniques, emphasizing the transition from rule-based systems to ML-based approaches [32-35]. Existing methods such as NBC and RF provide reasonable accuracy and simplicity but lack flexibility in handling evolving spam, and are limited by overfitting and poor resistance to adaptive spam. Similarly, KHO and FRNN-HHO demonstrate good optimization and low error rates but suffer from slow convergence and high computational cost. In contrast, the proposed hybrid optimization framework naturally combines the exploration capability of GSO with the exploitation capability of WOA, resulting in faster convergence, superior accuracy, and better resistance against concept drift and adversarial SMS spam.

TABLE I. CRITICAL ANALYSIS OF EXISTING MODELS

Model	Strengths	Limitations	Research gap
NBC [26]	Simple and computationally efficient; performs well with small and balanced datasets	Assumes feature independence, reducing real-world accuracy; struggles with non-linear and noisy text data	Traditional ML models like NBC lack adaptive feature learning and require manual tuning; they cannot cope with evolving spam tactics or unbalanced data
KHO [10]	Effective bio-inspired optimization technique with global search ability and strong exploration strength	Slow convergence in high-dimensional search spaces; easily trapped in local optima	Requires an enhanced optimizer capable of faster convergence and better exploitation balance to avoid local minima and reduce tuning overhead
RF [35]	Robust ensemble model reducing overfitting; handles high-dimensional features effectively	Performance depends on hyperparameter tuning; lacks adaptive optimization for evolving data	Needs automated, intelligent hyperparameter tuning to optimize performance and reduce computational cost across diverse spam patterns
FRNN-HHO [27]	Combines fuzzy logic and DL for adaptive classification	High training complexity and longer runtime; requires high computational resources for convergence	Requires a hybrid optimization approach with lower computational cost and faster convergence suitable for real-time or edge-level SMS filtering

III. PROPOSED METHODOLOGY

This section presents the overall methodology used for SMS spam classification, including dataset preparation, preprocessing, the machine learning classifiers employed, and the overall strategy.

A. Dataset

Firstly, a balanced dataset comprising spam and non-spam messages is required. The dataset is used to train and evaluate ML models and should contain a variety of text messages. Collected for the purpose of studying SMS spam, the SMS Spam Collection consists of a database of tagged SMS texts. Each of the 5,574 English-language SMS messages in the collection has been classified as spam or ham, and the dataset was originally introduced by authors in [36]. The sourced

dataset was collected from the publicly available Kaggle repository [37]. This study's use of the SMS dataset was driven by its widespread acceptance in the scholarly community and its singularity as the only publicly accessible collection of genuine spam SMS data.

In addition, it is important to identify the sentiments expressed in the categorized SMS, as this allows for the evaluation of emotions conveyed in both spam and non-spam messages. An optimization-based ML technique is used to identify the feelings of detected spam and ham SMS messages. The SMS analysis task employs RF and XGBoost. To enhance classification accuracy, the proposed ML model is combined with a metaheuristic optimization technique, namely the GSOWOA algorithm. By modifying the weight parameter, the GSOWOA approach achieves the best or near-optimal solution, which enables rapid convergence.

B. Data Preprocessing

Preprocessing plays a crucial role in SMS text categorization systems. Improved text classification performance results from applying appropriate preprocessing methods to textual datasets. The removal of noise from datasets is a crucial step in the field of ML. The preprocessing stage encompasses various procedures for categorizing input SMS messages through the utilization of several techniques, including text cleaning, tokenization, stop word removal, and stemming or lemmatization. The preprocessing techniques are described in detail in the following subsections.

1) Tokenization

Paragraphs are often divided into sentences or phrases are broken down into meaningful components, such individual words, via the process of tokenization. Tokenization primarily focuses on alphabetic or alphanumeric characters, separating non-alphanumeric symbols such as punctuation marks and whitespace. Tokenization methods can be broadly categorized into isolating, agglutinative, and inflectional approaches. In isolating methods, words are not broken into smaller units, whereas agglutinative methods separate words into meaningful smaller units.

2) Stop Word Removal

Frequently used words that do not substantially add to the overall meaning are removed during this stage. Stop words primarily serve grammatical functions by linking other words to form coherent sentences. These words typically include determiners, prepositions, and articles and are characterized by their high frequency of occurrence. Removing such words helps reduce dimensionality.

3) Stemming/Lemmatization

Words are reduced to their root or base form to handle variations. Stemming is applied to convert words into a standardized representation by removing suffixes and reducing word variations. This process reduces text redundancy and improves model generalization by mapping related words to a common base.

C. Random Forest Classifier

Classification and regression are two common applications of the RF technique, which is a supervised ensemble learning classifier. The suggested method is an ensemble learning strategy that builds a network of decision trees and uses their combined predictions to produce a final forecast.

The RF technique uses a network of decision trees, each contributing to a forecast; the final prediction is the average of all the forecasts. This approach helps reduce overfitting and improves the model's overall accuracy. Two key parameters must be specified when constructing an RF model: $feature_n$ and $tree_n$. The parameter $feature_n$ represents the number of features randomly chosen as candidates for splitting at each decision tree node, whereas $tree_n$ represents the total number of decision trees within the RF. The $feature_n$ parameter is responsible for controlling the level of randomness included into the model. Reducing the value of $feature_n$ may increase model randomness and potentially improve accuracy. However, this comes at the expense of increased model complexity, which in turn raises the risk of overfitting. Increasing the value of $feature_n$ enhances the model's ability to handle noise present in the data. The $tree_n$ parameter plays a crucial role in determining the total complexity of the model. Increasing the value of $tree_n$ enhances the model's accuracy; however, this improvement comes at the expense of more computing resources and longer training time. According to authors in [38], the computing resources do not impose a limitation on the number of trees ($tree_n$) in a forest. However, it has been observed that the incremental performance gains achieved by increasing the number of trees in the forest are negligible. Nevertheless, according to the findings of authors in [39], it is argued that the availability of computing resources serves as the primary constraint on the quantity of trees inside a given forest. Mean Square Error (MSE) is a node-level metric used in RF to address regression tasks and is defined as:

$$\text{MSE} = \frac{1}{M}\sum_{i=1}^{M}(f_i - x_i)^2 \quad (1)$$

where f_i and x_i, represent the predicted and actual values, respectively, and M is the total number of data points.

D. Extreme Gradient Boosting Classifier

XGBoost is a supervised ensemble ML algorithm that follows the boosting approach. It sequentially trains a set of learners, typically Classification and Regression Trees (CART), where each subsequent model aims to optimize the residuals of the preceding models, hence enhancing predictive performance. The approach operates by computing the similarity score at each node of the CART to evaluate potential splits based on the residuals of SMS feature values, as defined in (2):

$$S_{score} = \frac{\sum_{i=1}^{\eta}(a_i - f(b_i))^2}{\eta * \gamma} \quad (2)$$

where S_{score} denotes the similarity score, a_i represents the predicted value of the i-th variable, $f(b_i)$ represents the independent feature values, η is the residual value, and γ denotes the regularization parameter. The information gain obtained from a split is calculated using (3):

$$Gain = S^{i}_{score} - S^{j}_{score} \quad (3)$$

where S^{i}_{score} and S^{j}_{score} represent the similarity scores after and before the split. The regularization parameter γ controls tree complexity by penalizing unnecessary splits and reducing the impact of outliers. Increasing γ lowers the similarity score and, consequently, reduces the information gain, as shown by (3). Another important parameter in XGBoost is β, which acts as a threshold for node splitting. If the computed gain exceeds β, a split will occur at the node; otherwise, the node remains unsplit. This mechanism helps mitigate overfitting by controlling tree depth. Larger values of β result in more aggressively pruned trees. In this study, the default values of $\beta = 0$ and $\gamma = 1$ were maintained throughout each phase.

The output prediction of a basic CART learner is computed as:

$$O = \frac{\sum_{i=1}^{\eta}(a_i - f(b_i))}{\eta * \gamma} \quad (4)$$

The prediction is then updated iteratively using (5):

$$Update_p = Prev_p + (\alpha * O) \quad (5)$$

where $Update_p$ is the updated prediction, $Prev_p$ represents the previous prediction, and α is the learning rate. The learning rate is a crucial parameter that plays a significant role in controlling convergence speed and reducing overfitting by scaling the contribution of each tree.

E. Proposed GSOWOA-Based Optimization Framework

Figure 1 illustrates the proposed GSOWOA-based optimization framework for SMS spam classification. The workflow starts with data preprocessing, which consists of cleaning the input SMS data, followed by tokenization, stop word removal, stemming, and Term Frequency–Inverse Document Frequency (TF–IDF) vectorization to transform SMS messages into appropriate numerical features. Then, the hyperparameter initialization phase defines the search space for key parameters of the RF and XGBoost classifiers, such as the number of estimators, learning rate, and tree depth. The GSOWOA-based optimization phase performs global exploration and local exploitation of the hyperparameter space.

Candidate solutions are assessed using a fitness function that takes into account classification accuracy and convergence speed. The optimized hyperparameters are then applied to train the RF and XGBoost models, allowing automated tuning that enhances convergence, generalization ability, and classification performance. GSOWOA is used to adaptively modify the critical parameters (i.e., number of estimators, learning rate, and tree depth) according to classification accuracy of spam/ham message samples via cross-validation. This adaptive relaxation enables the model to account for linguistic patterns, weighted importance of features, and frequency characteristics that separates spam from legitimate SMS messages.

IV. INTEGRATION OF GSOWOA WITH MACHINE LEARNING MODELS

GSO, a population-based optimization technique developed for numerical function optimization, is introduced in this section in conjunction with the WOA algorithm. This approach is straightforward and efficient, avoiding the need to solve more complex problems. The algorithm employs a simple yet effective method. At the outset, the fitness values representing potential GSO algorithm solutions are initialized in a completely arbitrary manner. All entities work together in accordance with a straightforward mathematical model to get the best possible outcome on a global scale. A transfer operator is included into an adaptive step-size adjustment technique in the proposed method to achieve a satisfactory exploration–exploitation balance in search.

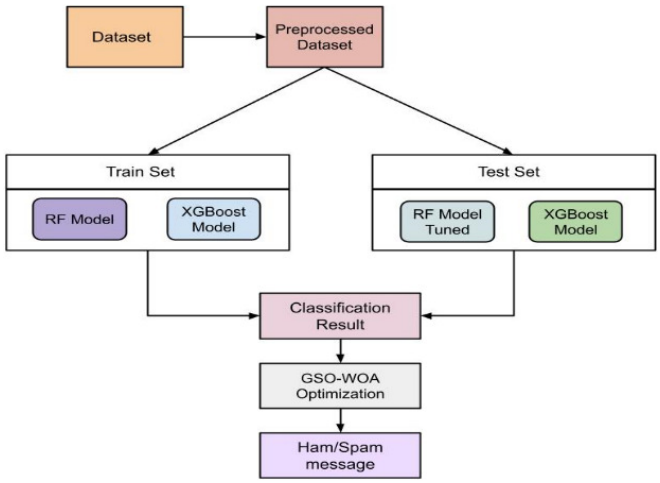

Fig. 1. Workflow of the proposed ML model for SMS spam classification.

Authors in [40] introduced a population-based metaheuristic approach, WOA, inspired by humpback whales' hunting behavior. The whales' bubble-net feeding strategy is a foraging approach that helps them in their predatory pursuits. Furthermore, it should be mentioned that whales in general display a behavior where they join forces with the nearest whale, and then proceed to swim towards a predetermined destination by progressively narrowing the diameter of their spiral path. The detailed mathematical model of WOA optimization is as follows.

A. Initialization

Within a Z-dimensional search space, the whale population is first determined. The search space can be mathematically represented as:

$$S = \{S_1, S_2, .., S_c, .., S_s\} \quad (6)$$

where s denotes the variables of a specific problem, S_c denotes the c-th potential solution, and S denotes the population.

B. Prey Encircling

The corresponding WOA formulation is:

$$P_\delta(c+1) = P_\delta(c) + Z_\delta(c+1) \quad (7)$$

Let:

$$P_\delta(c+1) = U_\delta(c+1) \quad (8)$$

$$P_\delta(c) = U_\delta(c) \quad (9)$$

Then, (7) becomes:

$$U_\delta(c+1) = U_\delta(c) + Z_\delta(c+1) \quad (10)$$

$$U_\delta(c) = U_\delta(c+1) - Z_\delta(c+1) \quad (11)$$

Humpback whales catch prey by identifying their exact location and circling around it. As soon as the best search agent is identified, the other search agents move their positions closer to it. The following is the GSOWOA's final update equation:

$$U_\delta(c+1) = \frac{(m_\delta - t_\delta(c))}{(m_\delta - t_\delta(c)) - X[m_\delta(c) - v_\delta]} \left[V^*(c)[1 - XW] + V\left[\frac{v_\delta(m_\delta - t_\delta(c)) - t_\delta[m_\delta(c) - v_\delta]}{(m_\delta - t_\delta(c))}\right]\right] \quad (12)$$

Here, v_δ represents the minimal random walk of the δ-th variable, $t_\delta(c)$ represents the minimal value of the δ-th variable at the c-th iteration, and $m_\delta(c)$ represents the maximum of the δ-th variable at the c-th iteration.

$$V = 2v * \hbar - v \quad (13)$$

$$t = 2 * \hbar \quad (14)$$

Here, v is bound between 0 and 2, and $\hbar$ is limited to values between 0 and 1. The ideal solution's position vector is represented by $V(c)$, during the c-th iteration, and $L_\delta(c)$ reflects the position of the δ-th solution.

$$Z_\delta(c+1) = O.Z_\delta(c) + t_1 \cos(\hbar_1)\left(E_{best\delta} - \chi_\delta(c)\right) + t_2 \sin(\hbar_2)\left(E_{\vartheta\, best\delta} - \chi_\delta(c)\right) \quad (15)$$

Here, $\hbar_1$ and $\hbar_2$ are selected randomly from [0, 1], and t_1 and t_2 are randomly chosen from [0, 2]. $Z_\delta(c)$ represents the step size at iteration c.

C. Bubble-Net Approach

Two separate methods are used to explain the mathematical modeling of whales' bubble-net behavior, as detailed below.

1) Encircling Approach

With v decreasing from 2 to 0 during the iterations, the updated search agent's position is calculated between the agent's current position and the position of the currently identified optimum agent by assigning random values to the variable r in the range [-1, 1].

2) Spiral Updating Position

When whales and their prey interact, a spiral pattern forms, mimicking the helical movement of the whales themselves:

$$V(c+1) = \vec{X}'.\lambda^{\beta ot}.\cos(2\pi o) + V^*(c) \quad (16)$$

The distance between a whale and its prey is represented by $\vec{X}' = \left|\overrightarrow{V^*}(c) - \vec{V}(c)\right|$. The constant β determines the shape of the logarithmic spiral, and o is a random number in the range [-1, 1].

A 50% chance is used to select one of the two mechanisms for updating a whale's position within the optimization framework. Mathematically, this is expressed as:

$$\vec{V}(c+1) = \begin{cases} \overrightarrow{V^*}(c) - \vec{X}.\vec{X}, & \text{if } \rho < 0.5 \\ \vec{X}.\lambda^{\beta o}.\cos(2\pi o) + \overrightarrow{V^*}(c), & \text{if } \rho \geq 0.5 \end{cases} \quad (17)$$

where ρ is a random number between 0 and 1.

3) Searching Prey

During the exploration phase, a random agent is picked to update the seeking agent's position rather than using the best agent found so far. The update equations are:

$$\vec{X} = \left|\vec{W}.\overrightarrow{V_{rand}} - \vec{V}\right| \quad (18)$$

$$\vec{V}(c+1) = \vec{V}_{rand} - \vec{X}.\vec{X} \quad (19)$$

The optimal classification of Ham or Spam SMS is achieved by iteratively performing the ML-GSOWOA steps. Algorithm 1 provides the pseudocode for ML-GSOWOA.

```
Algorithm 1: Proposed ML-GSOWOA Pseudocode
Input: SMS data
Output: Classified Ham or Spam SMS
Update CART learner's prediction value:
   Update_p = Prev_p + (α * O) using (5)
Initialize whales' population
Compute fitness using (12)
While c < Max_itr
   For each individual search agent
      Update v, X, W, o, ρ
      If (ρ < 0.5)
         If (|X| < 1)
            Relocate the current search
            agent using (16)
         Else if (|X| ≥ 1)
            Choose a random search agent V_rand
            Relocate the current search
            agent using (17)
         End if
      Else if (|ρ| ≥ 0.5)
         Relocate the current search agent
         using (19)
      End if
   End for
   Verify the seeking agent remains within
   the search space
   Evaluate fitness of individual search
   agents
   Update V*
   c = c + 1
End while
Return Ham or Spam SMS
Stop
```

The workflow of the hybrid GSOWOA method is illustrated in Figure 2. Once the system balance is ensured by calculating the standard deviation, Algorithm 1 applies the GSO algorithm to determine the optimal locations. Subsequently, the WOA algorithm is performed to optimize the best search agent for classifying the message as spam or ham.

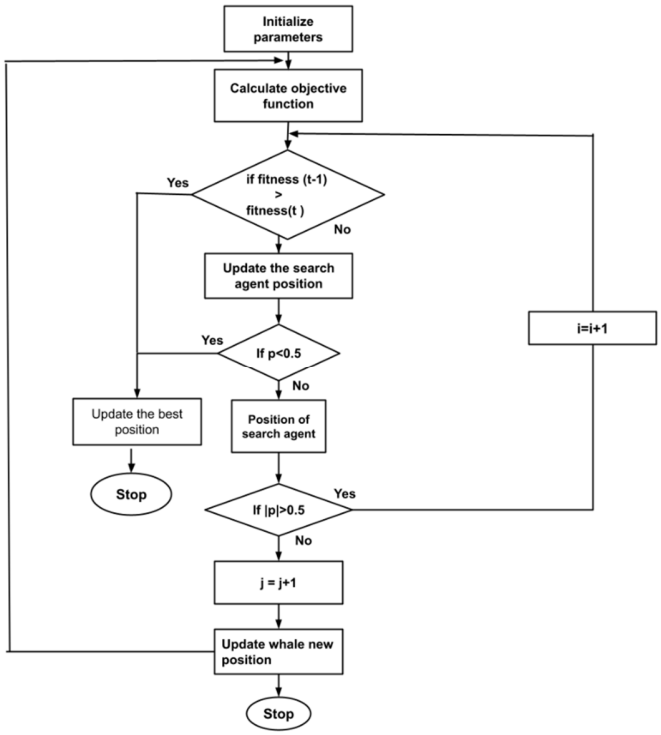

Fig. 2. Workflow of the GSOWOA method.

V. EXPERIMENTAL SETUP AND RESULTS DISCUSSION

This section delineates the implementation details of the ML-GSOWOA model for SMS spam detection. The model's capability to recognize spam messages was evaluated using data from the University of California, Irvine (UCI) repository [37] and compared with state-of-the-art ML techniques. An equitable distribution of classes is necessary to ensure accurate evaluation, as the dataset contains 5,574 English-language SMS messages, including both spam and ham. The dataset was split with 80% of messages used for training and the remaining 20% for validation and testing.

The ML-GSOWOA methodology is a comprehensive search method used to identify optimal hyperparameters within a predefined range through an extensive search process. Its selection in this study, among other parameter-tuning approaches such as random search, was motivated by its straightforward implementation and ability to execute in parallel. The random search methodology was also used to identify the most favorable parameter values, including the number of trees, number of iterations, and learning rate. Table II presents the experimental configuration used for exploring the parameter space of RF, XGBoost, and WOA.

A. Performance Evaluation Metrics

Evaluation metrics such as testing accuracy, recall, precision, and F1-score were used to assess the ML-GSOWOA model. The experiments show that spam messages can be effectively classified using a two-level binary classification scheme.

Precision measures the proportion of correctly identified positive messages among all messages classified as positive:

$$\text{Precision} = \frac{T_{Pos}}{T_{Pos} + F_{Pos}} \quad (20)$$

Recall measures the proportion of actual positive messages that are correctly identified:

$$\text{Recall} = \frac{T_{Pos}}{T_{Pos} + F_{Neg}} \quad (21)$$

The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both, especially useful in class-imbalanced scenarios:

$$\text{F1} - \text{score} = 2 * \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \quad (22)$$

Accuracy measures the proportion of correctly classified messages, both positive and negative, among all messages:

$$\text{Accuracy} = \frac{T_{Pos} + F_{Neg}}{T_{Pos} + F_{Pos} + T_{Neg} + F_{Neg}} \quad (23)$$

TABLE II. ML-GSOWOA PARAMETERS

Parameter	Value / Setting
Baseline model	NBC, KHO, RF, FRNN-HHO
Maximum depth	08
$\boldsymbol{\alpha}$	0.003
Tuned models	(500, 100) and (5, 10, 15, 20, 25, 30)
Max. depth	07, 08, 09
rounds	10, 20, 30
Population size	30
Coefficient vector	[0, 2]
Epochs	40
$\boldsymbol{r}$	[-1, 1]
$\hbar$	0 to 1
$\boldsymbol{v}$	0 to 2

B. Training Evaluation

The testing accuracy of the proposed ML-GSOWOA model and other existing models, including NBC, KHO, RF, and FRNN-HHO, is illustrated in Figure 3. Using 60% of the dataset for training, the corresponding testing accuracy values for NBC, KHO, RF, FRNN-HHO and ML-GSOWOA are 0.794, 0.781, 0.857, 0.874, and 0.89, respectively. Compared to these baseline models, the ML-GSOWOA approach demonstrates superior performance, achieving improvements of 10.288%, 11.751%, 3.163%, and 2.259%, respectively.

The precision results are presented in Figure 4. Using 70% of the available data for training the ML-GSOWOA model achieved a precision score of 0.972. The other methods, NBC, KHO, RF, and FRNN-HHO, obtained precision scores of 0.792, 0.788, 0.853, 0.86, and 0.869, respectively. Accordingly, the ML-GSOWOA method outperforms these approaches by 10.1808%, 11.261%, 3.941%, and 3.1533%, respectively.

Figure 5 illustrates recall evaluation. The ML-GSOWOA model achieved a recall of 0.941, outperforming NBC, KHO, RF, and FRNN-HHO, which scored 0.838, 0.824, 0.891, 0.904, and 0.941, respectively, using 80% of the training data The

proposed method demonstrates improvements of 12.34%, 13.34%, 6.23%, and 3.28% over the existing techniques.

Figure 6 presents the F1-score evaluation. With 90% of the data used for training, the ML-GSOWOA model achieved an F1-score of 0.935, compared to NBC, KHO, RF, and FRNN-HHO, which achieved F1-scores of 0.821, 0.809, 0.881, and 0.891, respectively. This corresponds to performance improvements of 12.192%, 13.475%, 5.775%, and 3.201%, respectively.

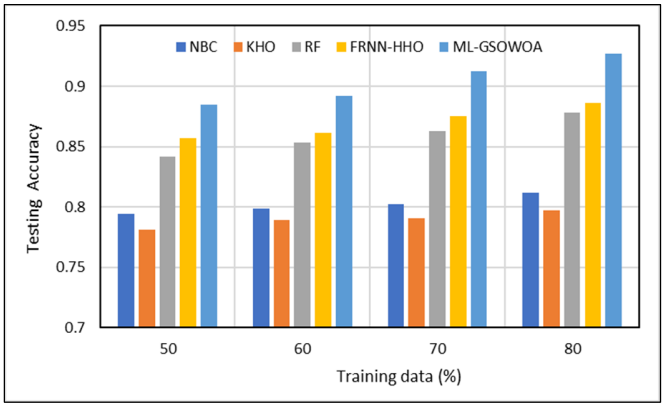

Fig. 3. Testing accuracy comparison of ML-GSOWOA and baseline models.

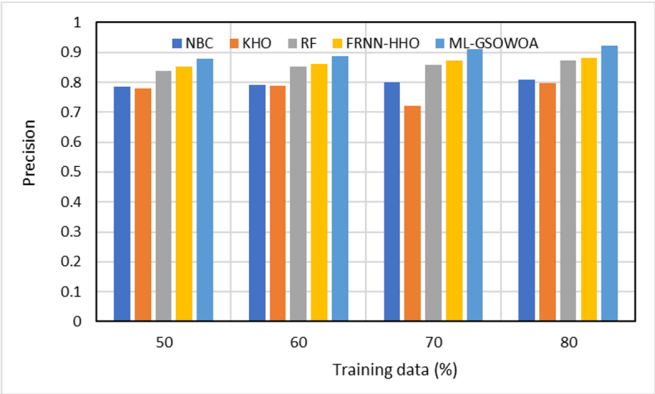

Fig. 4. Precision comparison of ML-GSOWOA and baseline models.

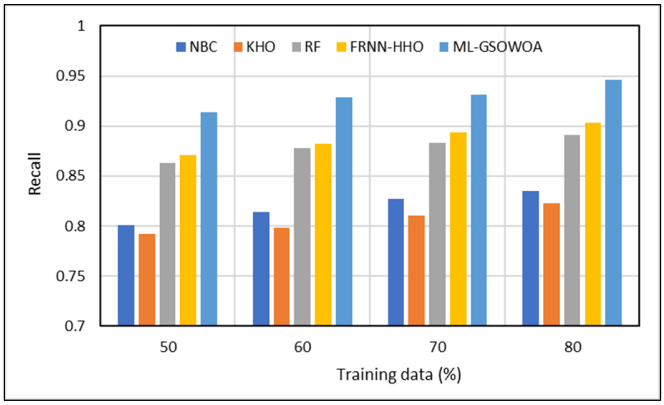

Fig. 5. Recall comparison of ML-GSOWOA and baseline models.

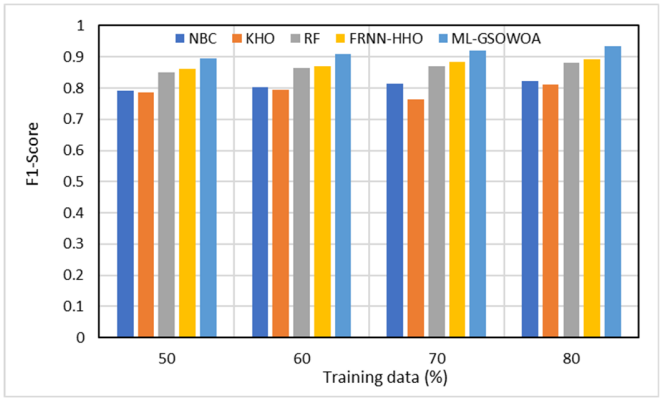

Fig. 6. F1-score comparison of ML-GSOWOA and baseline models.

Table III shows a comparison of the models on the spam label, confirming that ML-GSOWOA outperforms other approaches in precision, recall, and F1-score.

TABLE III. COMPARISON OF RESULTS FOR SPAM LABEL

Technique	Precision	F1-score	Recall	Label
NBC [26]	0.90	0.88	0.87	Spam
KHO [10]	0.88	0.85	0.83	Spam
RF [35]	0.91	0.90	0.90	Spam
FRNN-HHO [27]	0.92	0.90	0.89	Spam
ML-GSOWOA (proposed)	0.97	0.93	0.94	Spam

Figure 7 visualizes the performance of precision, recall, and F1-score for the spam label. The proposed method clearly demonstrates superior performance, with the KHO algorithm performing the worst.

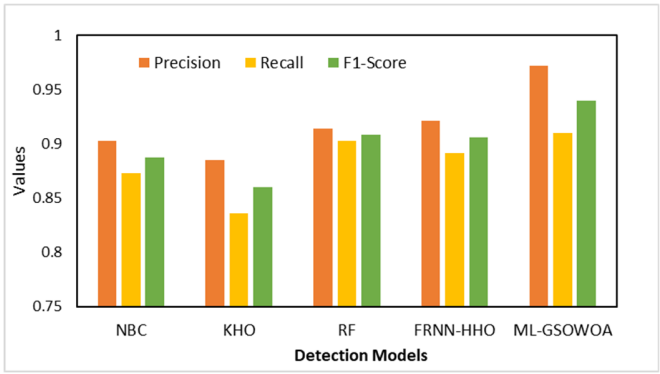

Fig. 7. Precision, recall, and F1-score evaluation for spam label.

Table IV presents a comparison for the non-spam (ham) label, showing that ML-GSOWOA achieves the highest precision, recall, and F1-score among the tested models.

TABLE IV. COMPARISON OF RESULTS FOR HAM LABEL

Technique	Precision	F1-score	Recall	Label
NBC [26]	0.9721	0.9645	0.9686	Ham
KHO [10]	0.9536	0.9572	0.9611	Ham
RF [35]	0.9778	0.9764	0.9721	Ham
FRNN-HHO [27]	0.9798	0.9809	0.9875	Ham
ML-GSOWOA (proposed)	0.9872	0.9889	0.9907	Ham

Figure 8 illustrates the results for the ham label, confirming that ML-GSOWOA outperforms NBC, KHO, RF, and FRNN-HHO, whereas the KHO algorithm again has the lowest performance.

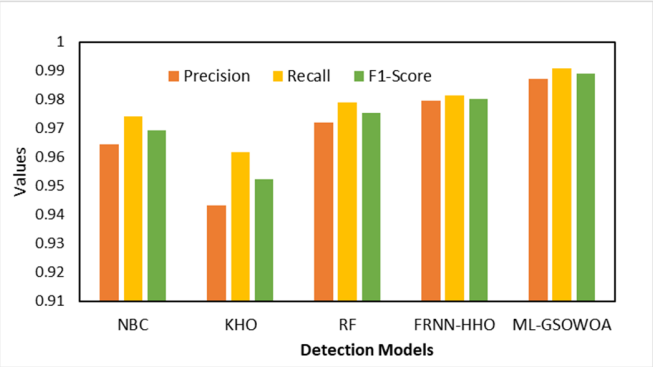

Fig. 8. Precision, recall, and F1-score evaluation for ham label.

Figures 9 and 10 compare training and validation accuracy of the evaluated models. Even when using a pre-trained module, the ML-GSOWOA method achieves superior performance. The pre-trained Inception module extracts basic features, whereas subsequent layers handle higher-dimensional data. After 40 epochs, the proposed method achieves a training accuracy of 98.9% and a validation accuracy of 92.2%.

Table V compares ML-GSOWOA with existing approaches. The proposed method achieves 98.9% accuracy, 97.85% precision, and 97.14% F1-score, with faster convergence for most datasets. The SpaLLM-Guard model [30], using fine-tuned Mixtral (8×7B) LLMs, slightly outperforms ML-GSOWOA in accuracy (98.61%) and shows stronger robustness to adversarial attacks and concept drift, maintaining both False Positive Rate (FPR) and False Negative Rate (FNR) below 2%. Traditional ML and DL ensembles [31] achieved lower accuracy (95.45%) and Area Under the Curve (AUC) (0.963) and were more vulnerable to homoglyph and spacing-based attacks. Overall, the findings confirm that optimization-based and LLM-driven approaches improve spam detection effectiveness and adaptability over ML techniques.

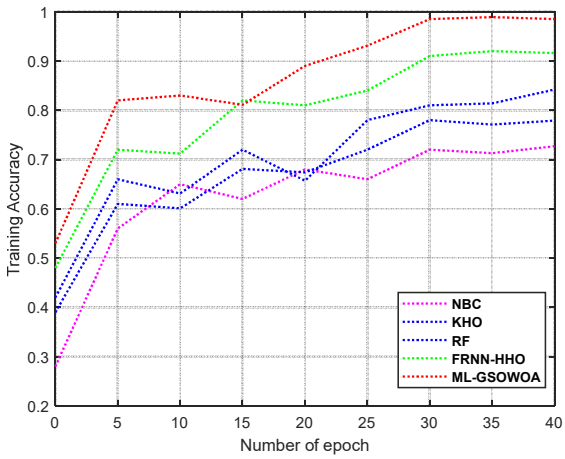

Fig. 9. Training accuracy comparison across models.

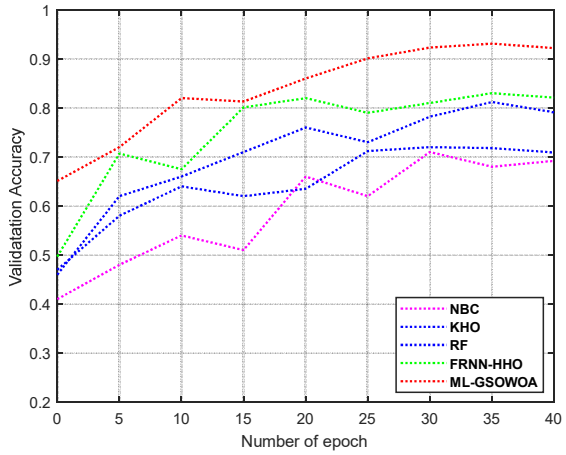

Fig. 10. Validation accuracy comparison across models.

TABLE V. COMPARISON OF PROPOSED METHOD WITH EXISTING MODELS FOR SMS SPAM DETECTION

Study	Proposed method	[30]	[31]
Model type	ML-GSOWOA	Fine-tuned LLM (Mixtral 8×7B)	Classical ML and DL
Dataset used	SMS Spam Collection (UCI and Kaggle)	Super SMS Dataset	Super SMS Dataset
Accuracy (%)	98.9	98.61	95.45
Precision (%)	97.85	98	94.8
Recall (%)	96.45	98.4	95.2
F1-score (%)	97.14	98.2	94.9
Remarks	Converges ≈ 35% faster; robust to noise; effective on legacy data	Balanced FPR < 2%, FNR < 2%; highest robustness under adversarial drift	AUC = 0.963; models vulnerable to homoglyph & spacing attacks

VI. CONCLUSION

This research addressed the challenges of spam Short Message Service (SMS) classification. Spam poses significant threats to user privacy and information security due to the widespread use of mobile devices in digital communication. The study of the proposed methodology has led to notable insights and advancements in the field. The effectiveness of integrating the Golden Search Optimizer–Whale Optimization Algorithm (GSOWOA) with Machine Learning (ML) classifiers for categorizing spam SMS was demonstrated through extensive experiments. The results consistently show improvements in classification accuracy, precision, recall, and F1-score compared to previous approaches. The hybrid optimization method contributed significantly to tuning model parameters, making the classification task more robust and adaptable. This work contributes to the broader academic discourse on spam SMS detection by combining ML and optimization techniques. The ML-GSOWOA methodology shows strong potential for enhancing the accuracy and efficiency of spam SMS detection systems.

However, the current study primarily focuses on static textual features of spam messages and does not account for variations that include multimedia content. Furthermore, although GSOWOA enhances accuracy and convergence, its performance in real-time large-scale simulations has not yet been fully investigated.

For future work, we aim to generalize the model to support multiple languages and larger SMS corpora. We also plan to investigate adaptive real-time message filtering and incorporate semantic and contextual embeddings to further strengthen spam detection in a diverse and evolving communication environment.

DATA AVAILABILITY

The dataset used in this study is publicly available at: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset/data.

REFERENCES

[1] O. Abayomi-Alli, S. Misra, and A. Abayomi-Alli, "A deep learning method for automatic SMS spam classification: Performance of learning algorithms on indigenous dataset," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 17, Aug. 2022, Art. no. e6989, https://doi.org/10.1002/cpe.6989.

[2] M. Raza, N. D. Jayasinghe, and M. M. A. Muslam, "A Comprehensive Review on Email Spam Classification using Machine Learning Algorithms," in *2021 International Conference on Information Networking*, Jeju Island, Korea, 2021, pp. 327–332, https://doi.org/10.1109/ICOIN50884.2021.9334020.

[3] A. Ghosh and A. Senthilrajan, "Comparison of machine learning techniques for spam detection," *Multimedia Tools and Applications*, vol. 82, no. 19, pp. 29227–29254, Aug. 2023, https://doi.org/10.1007/s11042-023-14689-3.

[4] S. Rao, A. K. Verma, and T. Bhatia, "A review on social spam detection: Challenges, open issues, and future directions," *Expert Systems with Applications*, vol. 186, Dec. 2021, Art. no. 115742, https://doi.org/10.1016/j.eswa.2021.115742.

[5] A. A. Akinyelu, "Advances in spam detection for email spam, web spam, social network spam, and review spam: ML-based and nature-inspired-based techniques," *Journal of Computer Security*, vol. 29, no. 5, pp. 473–529, Aug. 2021, https://doi.org/10.3233/JCS-210022.

[6] M. H. Alsuwit, M. A. Haq, and M. A. Aleisa, "Advancing Email Spam Classification using Machine Learning and Deep Learning Techniques," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 14994–15001, Aug. 2024, https://doi.org/10.48084/etasr.7631.

[7] M. Z. Gashti, "Detection of Spam Email by Combining Harmony Search Algorithm and Decision Tree," *Engineering, Technology & Applied Science Research*, vol. 7, no. 3, pp. 1713–1718, June 2017, https://doi.org/10.48084/etasr.1171.

[8] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, and B. Khan, "Golden Search Optimization Algorithm," *IEEE Access*, vol. 10, pp. 37515–37532, 2022, https://doi.org/10.1109/ACCESS.2022.3162853.

[9] R. K. Saidala and N. R. Devarakonda, "Bubble-net hunting strategy of whales based optimized feature selection for e-mail classification," in *2017 2nd International Conference for Convergence in Technology*, Mumbai, India, 2017, pp. 626–631, https://doi.org/10.1109/I2CT.2017.8226205.

[10] A. Sharaff, C. Kamal, S. Porwal, S. Bhatia, K. Kaur, and M. M. Hassan, "Spam message detection using Danger theory and Krill herd optimization," *Computer Networks*, vol. 199, Nov. 2021, Art. no. 108453, https://doi.org/10.1016/j.comnet.2021.108453.

[11] S. Bosaeed, I. Katib, and R. Mehmood, "A Fog-Augmented Machine Learning based SMS Spam Detection and Classification System," in *2020 Fifth International Conference on Fog and Mobile Edge Computing*, Paris, France, 2020, pp. 325–330, https://doi.org/10.1109/FMEC49853.2020.9144833.

[12] D. Sharma and A. Sharaff, "Identifying Spam Patterns in SMS using Genetic Programming Approach," in *2019 International Conference on Intelligent Computing and Control Systems*, Madurai, India, 2019, pp. 396–400, https://doi.org/10.1109/ICCS45141.2019.9065686.

[13] A. A. Al-Hasan and E.-S. M. El-Alfy, "Dendritic Cell Algorithm for Mobile Phone Spam Filtering," *Procedia Computer Science*, vol. 52, pp. 244–251, Jan. 2015, https://doi.org/10.1016/j.procs.2015.05.067.

[14] A. S. Onashoga, O. O. Abayomi-Alli, A. S. Sodiya, and D. A. Ojo, "An Adaptive and Collaborative Server-Side SMS Spam Filtering Scheme Using Artificial Immune System," *Information Security Journal: A Global Perspective*, vol. 24, no. 4–6, pp. 133–145, Dec. 2015, https://doi.org/10.1080/19393555.2015.1078017.

[15] T. M. Mahmoud and A. M. Mahfouz, "SMS Spam Filtering Technique Based on Artificial Immune System," *International Journal of Computer Science Issues*, vol. 9, no. 2, pp. 589–597, Mar. 2012.

[16] A. Alzahrani and D. B. Rawat, "Comparative Study of Machine Learning Algorithms for SMS Spam Detection," in *2019 SoutheastCon*, Huntsville, AL, USA, 2019, pp. 1–6, https://doi.org/10.1109/SoutheastCon42311.2019.9020530.

[17] A. Theodorus, T. K. Prasetyo, R. Hartono, and D. Suhartono, "Short Message Service (SMS) Spam Filtering using Machine Learning in Bahasa Indonesia," in *2021 3rd East Indonesia Conference on Computer and Information Technology*, Surabaya, Indonesia, 2021, pp. 199–203, https://doi.org/10.1109/EIConCIT50028.2021.9431859.

[18] D. S. Sisodia and A. K. Yogi, "Performance Evaluation of Ensemble Learners on Smartphone Sensor Generated Human Activity Data Set," in *Data, Engineering and Applications: Volume 2*, R. K. Shukla, J. Agrawal, S. Sharma, and G. Singh Tomer, Eds. Singapore: Springer, 2019, pp. 277–284, https://doi.org/10.1007/978-981-13-6351-1_22.

[19] T. Xia and X. Chen, "A weighted feature enhanced Hidden Markov Model for spam SMS filtering," *Neurocomputing*, vol. 444, pp. 48–58, July 2021, https://doi.org/10.1016/j.neucom.2021.02.075.

[20] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter SMS Spam," *Future Generation Computer Systems*, vol. 102, pp. 524–533, Jan. 2020, https://doi.org/10.1016/j.future.2019.09.001.

[21] S. Ouhame, Y. Hadi, and A. Ullah, "An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model," *Neural Computing and Applications*, vol. 33, no. 16, pp. 10043–10055, Aug. 2021, https://doi.org/10.1007/s00521-021-05770-9.

[22] A. Chandra and S. K. Khatri, "Spam SMS Filtering using Recurrent Neural Network and Long Short Term Memory," in *2019 4th International Conference on Information Systems and Computer Networks*, Mathura, India, 2019, pp. 118–122, https://doi.org/10.1109/ISCON47742.2019.9036269.

[23] T. O. Omotehinwa and D. O. Oyewola, "Hyperparameter Optimization of Ensemble Models for Spam Email Detection," *Applied Sciences*, vol. 13, no. 3, Feb. 2023, Art. no. 1971, https://doi.org/10.3390/app13031971.

[24] P. Manasa *et al.*, "Tweet Spam Detection Using Machine Learning and Swarm Optimization Techniques," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4870–4877, Aug. 2024, https://doi.org/10.1109/TCSS.2022.3230823.

[25] S. Bazzaz Abkenar, E. Mahdipour, S. M. Jameii, and M. Haghi Kashani, "A hybrid classification method for Twitter spam detection based on differential evolution and random forest," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 21, Nov. 2021, Art. no. e6381, https://doi.org/10.1002/cpe.6381.

[26] K. Agarwal and T. Kumar, "Email Spam Detection Using Integrated Approach of Naïve Bayes and Particle Swarm Optimization," in *2018 Second International Conference on Intelligent Computing and Control Systems*, Madurai, India, 2018, pp. 685–690, https://doi.org/10.1109/ICCONS.2018.8662957.

[27] U. Srinivasarao and A. Sharaff, "SMS sentiment classification using an evolutionary optimization based fuzzy recurrent neural network," *Multimedia Tools and Applications*, vol. 82, no. 27, pp. 42207–42238, Nov. 2023, https://doi.org/10.1007/s11042-023-15206-2.

[28] Y.-J. Su, C.-H. Chen, T.-Y. Chen, and C.-C. Cheng, "Chinese Microblog Sentiment Analysis by Adding Emoticons to Attention-Based CNN," *Journal of Internet Technology*, vol. 21, no. 3, pp. 821–829, May 2020.

[29] T. Sharmin, F. Di Troia, K. Potika, and M. Stamp, "Convolutional neural networks for image spam detection," *Information Security Journal: A Global Perspective*, vol. 29, no. 3, pp. 103–117, May 2020, https://doi.org/10.1080/19393555.2020.1722867.

[30] M. Salman, M. Ikram, N. Basta, and M. A. Kaafar, "SpaLLM-Guard: Pairing SMS Spam Detection Using Open-source and Commercial LLMs." arXiv, Jan. 09, 2025, https://doi.org/10.48550/arXiv.2501.04985.

[31] M. Salman, M. Ikram, and M. A. Kaafar, "Investigating Evasive Techniques in SMS Spam Filtering: A Comparative Analysis of Machine Learning Models," *IEEE Access*, vol. 12, pp. 24306–24324, 2024, https://doi.org/10.1109/ACCESS.2024.3364671.

[32] M. A. Abid, S. Ullah, M. A. Siddique, M. F. Mushtaq, W. Aljedaani, and F. Rustam, "Spam SMS filtering based on text features and supervised machine learning techniques," *Multimedia Tools and Applications*, vol. 81, no. 28, pp. 39853–39871, Nov. 2022, https://doi.org/10.1007/s11042-022-12991-0.

[33] S. B. S. Ahmad, M. Rafie, and S. M. Ghorabie, "Spam detection on Twitter using a support vector machine and users' features by identifying their interactions," *Multimedia Tools and Applications*, vol. 80, no. 8, pp. 11583–11605, Mar. 2021, https://doi.org/10.1007/s11042-020-10405-7.

[34] L. P. Lim and M. Mahinderjit Singh, "Resolving the imbalance issue in short messaging service spam dataset using cost-sensitive techniques," *Journal of Information Security and Applications*, vol. 54, Oct. 2020, Art. no. 102558, https://doi.org/10.1016/j.jisa.2020.102558.

[35] N. N. Amir Sjarif, N. F. Mohd Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, "SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm," *Procedia Computer Science*, vol. 161, pp. 509–515, Jan. 2019, https://doi.org/10.1016/j.procs.2019.11.150.

[36] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in *Proceedings of the 11th ACM symposium on Document engineering*, Mountain View, CA, USA, 2011, pp. 259–262, https://doi.org/10.1145/2034691.2034742.

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

[38] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How Many Trees in a Random Forest?," in *Machine Learning and Data Mining in Pattern Recognition: 8th International Conference*, Berlin, Germany, 2012, pp. 154–168, https://doi.org/10.1007/978-3-642-31537-4_13.

[39] Y. Fujiwara, Y. Ida, S. Kanai, A. Kumagai, J. Arai, and N. Ueda, "Fast Random Forest Algorithm via Incremental Upper Bound," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Beijing, China, 2019, pp. 2205–2208, https://doi.org/10.1145/3357384.3358092.

[40] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, https://doi.org/10.1016/j.advengsoft.2016.01.008.