International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 10, Issue 9, pp 70-82, September-2026
https://doi.org/10.47001/IRJIET/2026.109008

Optimized LSTM–GWO Framework for Intelligent Email Spam Filtering in 5G Networks

Ali Mohammed Al-Quraishi

Department of Computer Science, College of Education, Sheikh Al-Tusi University, Najaf-Iraq
E-mail: ali.mohammed@altoosi.edu.iq

Abstract **- This article is a comparison study revolving around machine learning and deep learning algorithms which have been applied in the spam email detection process with a particular focus on population based optimization algorithms. The comparison was also made between the traditional classifiers such as Logistic Regression, SVM, KNN, Decision Tree, Random Forest, Naive Bayes and K-Means with the deep learning networks such as LSTM, GRU, CNN and Dense networks. Systematic use of grey wolf optimizer (GWO), genetic algorithm (GA) and particle swarm optimization (PSO) optimization algorithms was applied to enhance the performance of models. The deep learning models have been determined to perform better than the traditional classifiers on a range of metrics, including the accuracy, recall, F1-score, MCC, Cohen Kappa and specificity. The performance of LSTM optimized by GA and GWO were the best and yielded a higher accuracy between 97.9, 96.7, and F1-score of 0.973. PSO not only improved CNN and Dense models but also gave the best traditional model, Accuracy of 97.5 percent and F1-score of 0.974. The findings reveal that deep learning and population-based optimizers. can be applied to design powerful and competent email spam detectors in the field.**

Keywords: Email Spam, Deep learning, Machine Learning, Optimizer Algorithms.

I. INTRODUCTION

Junk email or spam email has emerged as one of the most challenging and disruptive technology of digital communication. Spam emails are defined as unsolicited and generally unrelated messages sent in bulk to a large number of people with the aim of selling products and services, or ideologies, some of these are malicious practices such as phishing or malware distribution. Spam has caused a lot of problem to individual users and businesses. Spam emails promote waste of time to users and reduce their productivity. To organizations, spam messages are potentially devastating, both in terms of breaching security, stealing data, as well as possibly causing a blow to the reputation of a business or individual [1].

Any increase in the spam email can be blamed on a number of factors. First, bulk emails are cheap to send, and hence, spammers are attracted to use it. Unlike other conventional advertising techniques, spammers can now access millions of users at practically no expense through email. Secondly, the anonymity that goes along with the internet makes it hard to trace or prevent spam campaigns because spammers can easily get away with it. Third, spam strategies, such as obfuscation techniques, sophisticated social engineering tricks, and so forth are continuously changed making it harder and harder to detect and filter spam [2].

Spam emails are generally sorted into various categories, including promotional spam, phishing attack, scam, and those transmitting a virus or malware. The last one is especially the point of concern because it can result in data breaches, financial fraud, and even identity theft. Such as phishing emails, a phisher tries to compel the receiver to reveal confidential details such as passwords, bank account numbers, and credit card numbers to cause devastating effects on the individuals involved [3].

Digital communication, especially use of email, has grown remarkably and it has changed the way people and organizations communicate. Nevertheless, as a result of growth of email usage, unsolicited and malicious email has also grown in abundance, what we all know as spam. Not only do these unwanted emails fill inboxes, they can also pose a big risk to individuals and businesses. Spam email is mostly utilized in an ill fashion, such as phishing, stealing of information, and malware propagation, which is highly hazardous to our security. Classical spam detectors (filter rules or simpler machine learning (ML) algorithms) cannot keep up with the changing strategies used by spammers as the amount of email spam grows [4].

With the introduction of 5G networks, email spam problem has become even more complicated. Using 5G, it is possible to dramatically increase the speed of mobile and internet data transfer, which will result in a tremendous increase in the amount of data transmitted and, accordingly, in email traffic. The requirement of smart and dynamic spam filtering software solutions in this fast-paced, high-volume,

ISSN (online): 2581-3048
Volume 10, Issue 9, pp 70-82, September-2026
https://doi.org/10.47001/IRJIET/2026.109008

track record time and again has never been more in demand. Although 5G networks support greater network connectivity and data transfer, increased network efficiency is achieved by greater capacity of spam which necessitates higher network management [5].

Deep learning networks, especially Long Short-Term Memory (LSTM), networks have demonstrated a lot of potential in a range of natural language processing problems, such as spam language recognition. The design of LSTM networks ensures them a high capability of capturing sequential dependencies in data; hence, it is highly applicable in processes of email classification where it is important that the context and structure of the message is understood. Nevertheless, more sophisticated models, such as LSTM, still need additional optimization to provide the best performance in the changing and challenging environment of 5G networks [6].

Population-based optimization algorithms (e.g., the Grey Wolf Optimizer) and genetic algorithms (Genetic Algorithm), as well as particle swarm optimization algorithms (Particle Swarm Optimization), are well-known methods of optimizing machine learning. Such algorithms can effectively search large search spaces and optimize model parameters, enhancing the accuracy and recall of specific spam detectors as well as their F1-score and other performance controls. The proposal to combine LSTM with these types of optimization algorithms can be very promising in providing the spam filtering process with greater strength, flexibility, and efficiency [7].

This study seeks to impact the issues of email spam detection on the 5G networks by suggesting a hybrid LSTM-GWO framework. The proposed system will optimize the spam detection process by proposing the utilization of both LSTM and the Grey Wolf Optimizer, maximizing its accuracy and minimizing the TE and TN levels. In this paper, the performance of LSTM-GWO framework is examined in detail, where its performance is compared with traditional machine learning frameworks and other deep learning networks. The results demonstrate that deep learning, when used in conjunction with population-based optimization approaches, is proven to create a smarter and more efficient spam filtering technology, especially when it comes to 5G networks.

II. RELATED WORKS

Detection of spam email has been a major research issue due to the ever increasing number of spam email in personal and business email systems. Researchers have investigated various machine learning and deep learning methods in the past few years to enhance the precision and accuracy of spam detection. Various methodologies have been suggested like utilizing conventional machine learning algorithms, ensemble frameworks, adversarial methodologies, and deep learning structures. Both approaches have certain advantages and disadvantages, which will depend on the data set and the nature of the spam mails used. Here, we discuss some of the major works which have led to the development of spam email recognition.

Nikhil Kumar *et al.* (2020) [8], investigated the different machine learning algorithms in the evaluation of spam email emails. Some of the classifiers tested include support vector machines (SVM), K-nearest neighbor (KNN), Naive Bayes (NB), decision tree (DT), random forest (RF), AdaBoost and Bagging classifiers. They found that SVM, KNN, and random forest performed well with a precision of 0.92, 0.92, and 0.90, respectively. Comparatively, AdaBoost classifier has the highest precision of 0.95 and the Bagging classifier also performed well with a precision of 0.94. The results of the study showed how different spam datasets vary in terms of algorithm performance, which makes it very important to choose the right models and datasets. In our investigation, we have used another dataset, and our base models have obtained similar, though not high, precision values.

Akash Junnarkar *et al.* (2021) [9] used the Enron data to compare various classification algorithms, such as SVM, random forest (RF), Naive Bayes (NB), decision tree (DT) and KNN. The research study indicated that the SVM had the highest accuracy of 97.83, closely followed by the random forest which had an accuracy of 97.60. This result was an indication of how strong SVM is in spam classification tasks. Kappa's and Dunachy further exploited improvements made based on more computationally expensive but accurate ensemble techniques, including XGBoost, to further optimize accuracy. Their results correspond to the fact that they find evidence that more sophisticated ensemble models can possibly outperform the regular classifiers in the spam detection task.

Zhang *et al.* (2018) [10] reviewed the adversarial techniques in order to evade spam email detection. They have talked about the tricks used by spammers to get around the detection system and outlined ways of fighting them. The paper also presented the limitations of modern spam email classification systems including the susceptibility of existing systems to adversarial attacks. Zhang et al. were also helpful in explaining why we need to create a strong spam filter that will be able to withstand any changes in spam tactics and overcome the effects of anti-spam measures.

Shaukat *et al.* (2020) [11] investigated the performance of decision tree (DT), SVM and Naive Bayes (NB) classifiers

ISSN (online): 2581-3048
Volume 10, Issue 9, pp 70-82, September-2026
https://doi.org/10.47001/IRJIET/2026.109008

in spam email classification. Their experiment revealed similar performances of SVM and decision tree classifiers in handling large emails i.e. email larger than 10,000 words. The importance of this finding is because email system keeps increasing with length and complexity of messages. Their study noted the benefits of SVM and decision trees, especially in processing an email containing rich content.

Hajek *et al.* (2020) [12] tested deep learning-based models, such as multilayer perceptron (MLP) and SVM, and compared them against spam classification alternatives. They also employed the use of unsupervised topic models to solve the issue of extracting features and enhancing classification. Their results showed the promise of deep learning models and unsupervised methods in addressing the spam email classification problem, especially when there are few labelled data. These findings indicated that character n-gram-based and word knowledge methods, which are feature representation approaches, would enhance spam classes more effectively.

Ramanathan *et al.* (2020) [13] suggested an unsupervised topic modeling method based on Latent Dirichlet Allocation (LDA) with the aim of producing features of the training set as inputs to deep learning models. They were designed with the purpose of dealing with large datasets, minimizing input feature dimensions without compromising classification quality results. It was shown that feature generation using LDA and deep learning models could successfully classify spam emails with near performance of more supervised methods.

Ghourabi *et al.* (2020) [14] used Convolutional Neural Networks (CNN) and Long-Short Updating Simplifies (LSTM). Their model performed better as compared to traditional models like Gaussian Naive Bayes (GNB) and decision trees. The hybrid CNN-LSTM model leveraged the advantages of each of the architectures where the CNNs learned all the spatial aspects and the LSTMs learned all the sequential dependencies resulting in better classification of spam emails.

Madhavan *et al.* (2020) [15] tested the different machine learning methods and hyper parameter tuning techniques to burn spam email databases to maximize the classification performance. They underlined that constant advancements in spam detection algorithms were necessary, especially with the help of hybrid or ensemble models. They suggested that by using more than one model or classifier there might be increased accuracy and strength in spam classification.

Rayan *et al.* (2019) [16] discussed how decision trees (DT) and random forests (RF) could be used in a hybrid model in order to improve spam classification. Their findings revealed that the combination of these classifiers greatly enhanced the classification accuracy, relative to the single methods. The analysis offered good information to support the idea that hybrid models may produce more useful results than single-model classifiers, particularly with regard to accuracy and recall.

All the studies examined here clearly testify to the variety of strategies that can be employed to cope with the issue of spam email classification. Although successful machine learning methods like Naive Bayes, SVM, and decision trees still exist, some form of hybrid approaches combining two or more classifiers or using deep learning networks or CNN and LSTM have become increasingly popular. Moreover, hyper parameter tuning, ensemble learning and unsupervised topic model optimization methods have been able to refine the accuracy and recall of spam classification systems. The results indicate that integrating conventional techniques using the current deep-learning and optimization technologies have immense potential to create superior and more accurate spam detectors. These hybrid methods should be further studied in the future, and alternative methods developed to reduce the effect of these adversarial spamming behavior and additional improvements made to the score of classification.

III. METHODOLOGY

This study aimed to optimizing the Long Short-Term memory (LSTM) network to detect spam email as it has extraordinary capacity to work with sequential data and achieve long-term dependencies in a text. Under the spam detection, interpretation of word combinations and email format is also a key to spotting bad content correctly. As an extremely potent recurrent neural network, LSTM has demonstrated great potential in this regard. In order to improve the accuracy and performance of the models, the traditional machine learning models (such as Support Vector Machine (SVM), Naive Bayes, Logistic Regression, k-Nearest Neighbors (KNN), Decision Tree, and Random Forest (RF)) are taken as the baseline models to be compared. Also, deep learning architectures, such as Convolutional Neural Networks (CNN), Gated Recurrent Unit (GRU), and Dense (Multilayer Perceptron) networks, are discussed to gain a wider insight into various way of addressing spam detection.

An additional problem with spam detection is that optimization of hyper parameters is one of the most important factors affecting performance of the model. To solve this issue, a number of metaheuristic optimization algorithms are used, such as the Grey Wolf Optimizer (GWO), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) to efficiently selected the optimal combination of hyper parameters. These algorithms do the tuning automatically and

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 9, pp 70-82, September-2026
https://doi.org/10.47001/IRJIET/2026.109008

as such save time duration spent in trial-and-error processes. The suggested methodology consists of a number of steps such as the pre-processing of data, the training of a model, metaheuristic optimization of the model, and the evaluation of the final model. The framework is designed to not only optimize the LSTM model but also provide the opportunity to broadly compare machine learning and deep learning strategies. The aim is to develop a spam detector system that is highly accurate, well-generalized and practical in high-traffic settings such as 5G networks. Figure (1) describes the architecture and process of the suggested methodology.

3.1 Dataset Description

The Spam Base dataset is widely used for email spam detection, available from public repositories like the UCI Machine Learning Repository and Kaggle. It contains 4,601 labeled email examples, classified as either "spam" (1) or "non-spam" (0). Each example has 57 numeric features, which capture various email characteristics, such as the frequency of specific words ("make," "address," "free"), special characters ("!", "\$", "?"), average word length, and the use of capital letters. These features help the model distinguish between spam and non-spam emails. The Spam Base dataset is a reliable resource for training and testing machine learning algorithms, offering diverse data for effective spam classification.

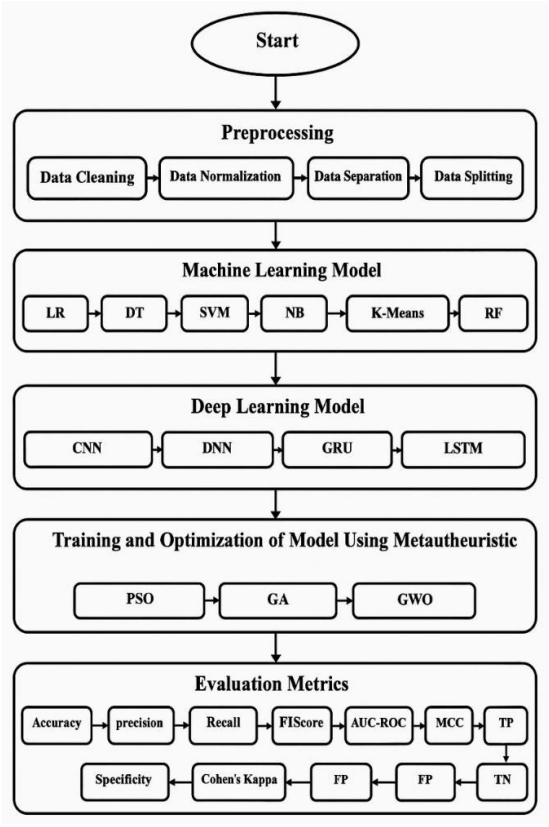

Figure 1: Flowchart of Proposed Method

3.2 Data Preprocessing

Data preprocessing is one of the most critical phases in developing and applying machine learning models. Input data quality and shape directly affect the accuracy and generalization capability of any predictive system. Raw data will always contain noise, inaccuracies, missing values, and distributions that are skewed, which, if left without treatment, can have very high degrees of jeopardy in terms of the performance of the model. Preprocessing aims to sanitize the data, standardize it, and place it into a shape that will maximize the model's learning capacity. In this research, Table (1) employed during the preprocessing of the data:

Table 1: Dataset Preprocessing

Pre-processing Step	Description
Data Cleaning	Identifying and removing invalid, duplicate, or irrelevant data. Emails with unintelligible characters, severe spelling mistakes, or non-textual elements (images, attachments) are excluded. One-Hot Encoding at the character level is applied to convert text into numerical format. Each unique character is transformed into a binary vector with one element “hot” (1) and others 0, allowing models like LSTM to capture character-level patterns effectively.
Data Normalization	Rescaling numeric features to a similar range, typically 0–1, to prevent features with larger ranges from dominating model learning. Min-Max Normalization is applied: **X_norm = (X - X_min) / (X_max - X_min)** Ensures all features contribute equally, aiding faster and more effective model convergence.
Separation of Features & Labels	Dividing data into features (inputs like email text) and labels (outputs like "spam" or "non-spam"). This separation helps the model learn the mapping from inputs to outputs and allows performance evaluation by comparing predictions to ground truth labels.
Data Splitting	Splitting dataset into 80% training and 20% testing. Training set fits the model; testing set evaluates performance on unseen data. Ensures the model can generalize and perform well on real-world scenarios.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 9, pp 70-82, September-2026

https://doi.org/10.47001/IRJIET/2026.109008

3.3 Machine Learning Models

In this section, a number of machine learning models are presented, which will be used together with the main LSTM model when detecting spam emails. Logistic Regression (LR) is a simple binary classification algorithm that approximates the likelihood of receiving a spam email, and the main hyper parameters it can take are the learning rate and the number of iterations to perform. Decision Tree (DT) models are recursive binary splits of data on the basis of feature values, parameters such as maximum depth and minimum samples per leaf affect the complexity and overfitting of the tree. SVM is useful in high-dimensional space because it finds the best combination of margin and errors, given hyper parameters such as C and gamma (trade-offs between training examples and errors). Naive Bayes (NB) relies on the assumptions of conditional independence of features and is computationally inexpensive based on distribution assumptions (Bernoulli, Gaussian, Multinomial) and a smoothing factor to enhance robustness. In K-Nearest Neighbors (KNN), emails are classified by the distance between the neighboring samples, and the accuracy and the cost of the classification depends on how many neighbors (k) it has, and its distance measure. To prevent overfitting and maximize resilience, Multiple decision trees are combined in the Random Forest, and the number of trees, as well as the maximum depth, are crucial hyper parameters that dictate the complexity of this method. Lastly, K-Means is another unsupervised clustering method that can be applied in order to preprocess or identify concealed patterns in data, where k is the number of clusters and a performance-critical parameter. As part of the present research, the models possess their unique strengths and weaknesses that allow a complete picture of these potentials to be used in effectively spam email classification.

3.4 Deep Learning Models

With the improvement of deep learning, several models have been proposed that are able to specifically handle complex data such as text, images, and sound. Such models can automatically learn underlying patterns and features from raw data itself without the need for feature engineering. Deep learning models have also performed better in natural language processing (NLP) and specifically in detecting spam emails since they are able to learn semantic and structural relationships in language. This section provides four of the most popular deep learning models, which are Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Deep Neural Network (DNN), and Convolutional Neural Network (CNN). For each model, the architecture is described first, and then the most critical hyper parameters are described at length along with how they assist in the learning process.

Table (2) demonstrates a more detailed network structure of four frequently applied models for text classification and spam filtering LSTM, GRU, DNN, and CNN. The models are also more divided into single layers, i.e., input, hidden, dropout, and output layers besides overall sizes and configurations. Important elements, sequence awareness, and common applications are also listed to give strong points and applications of each structure. This systematic representation makes it convenient to compare the models and gives recommendations on how to implement and deploy the deep learning-based text classification systems.

Table 2: Deep Learning Models Structures

Model	Layer Type	Layer Details / Size	Key Components	Sequence Awareness
LSTM	Input Layer	Vocabulary size / Embedding dimension	-	Yes
LSTM	LSTM Layer 1	64–256 units	Memory cell, input/forget/output gates	Yes
LSTM	LSTM Layer 2 (optional)	64–256 units	Memory cell, gates	Yes
LSTM	Dropout Layer	0.2–0.5	Regularization	-
LSTM	Dense Output Layer	1 unit (Sigmoid for binary classification)	Fully connected	-
GRU	Input Layer	Vocabulary size / Embedding dimension	-	Yes
GRU	GRU Layer 1	64–256 units	Update & reset gates	Yes
GRU	GRU Layer 2 (optional)	64–256 units	Update & reset gates	Yes
GRU	Recurrent Dropout	0.2–0.5	Regularization for temporal connections	-
GRU	Dense Output Layer	1 unit (Sigmoid)	Fully connected	-
DNN	Input Layer	Vectorized feature size	-	No
DNN	Hidden Layer 1	64–256 neurons	Fully connected, ReLU	No
DNN	Hidden Layer 2	64–256 neurons	Fully connected, ReLU	No

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 9, pp 70-82, September-2026
https://doi.org/10.47001/IRJIET/2026.109008

DNN	Hidden Layer 3 (optional)	64–256 neurons	Fully connected	No
DNN	Dropout Layer	0.2–0.5	Regularization	-
DNN	Dense Output Layer	1 unit (Sigmoid)	Fully connected	-
CNN	Input Layer	Embedding dimension	-	Partially
CNN	Conv Layer 1	32–128 filters, kernel size 3–5	Convolution filters	Partially
CNN	Max Pooling 1	Pool size 2–3	Dimensionality reduction	-
CNN	Conv Layer 2 (optional)	32–128 filters, kernel size 3–5	Convolution filters	Partially
CNN	Max Pooling 2 (optional)	Pool size 2–3	Dimensionality reduction	-
CNN	Flatten Layer	-	Flatten feature maps	-
CNN	Dense Output Layer	1 unit (Sigmoid)	Fully connected	-

3.5 Training and Optimization

To improve the spam email classification models in terms of precision, stability, and overall generalization, this research study uses the Grey Wolf Optimizer (GWO) algorithm to optimize the hyper parameters. Hyper parameters can greatly contribute to the performance of models through their effect on the learning process, convergence rate, model complexity, as well as generalization. Inappropriate hyper parameters may result in an over fitted model, under fit model or an optimal model. Hence, a powerful automatic optimization method, like GWO, will be essential in determining the best values.

The GWO algorithm is modelled on the hunting patterns and social structure of the grey wolves. There are four types of wolves namely alpha, beta, delta, and omega. The best solution is the alpha. The algorithm balances both exploration and exploitation, optimizing the search. The position of each wolf is a coded n-dimensional vector of candidate solutions to hyper parameters of the model, e.g. learning rate and batch size. The upper and lower limits of each hyper parameter are specified, then a starting population of wolves is randomly created within these parameters.

The aim of GWO is to optimize the validation accuracy of the model. The location of the wolves is updated in every iteration according to the best solutions (alpha, beta and delta). The algorithm operates by iterations, and in each iteration the wolves increasingly narrow their search space to more likely new promising locations. The optimization is repeated until a specified number of iteration has been made or there is no substantial improvement in the optimization. This method makes it possible to tune machine learning models efficiently and enhances performance in the spam email classification problem in Table (3).

After completing the training and optimization process using the Grey Wolf Optimizer (GWO) algorithm, the optimal values of the hyper parameters for each of the machine learning models were extracted and determined. These optimal values represent the final parameter settings that achieved the best performance in evaluating the models in terms of accuracy, recall, and F1-score.

Table 3: Tunable Hyper parameters of Machine Learning Models and GWO Algorithm Settings

Machine Learning Model	Tunable Parameter	Lower Bound	Upper Bound
SVM	C value	0.1	10
Decision Tree (DT)	Max Depth	3	15
Naive Bayes	smoothing parameter (var_smoothing)	1e-9	1e-6
KNN	Number of neighbors (k)	1	20
Logistic Regression	Regularization coefficient (C)	0.1	10
Random Forest (RF)	Number of estimators (n_estimators)	50	300
Random Forest (RF)	Max Depth	3	15
K-Means	Number of clusters (k)	2	15

The obtained results are comprehensively and separately presented for each model in Table (4), allowing for comparison and more detailed analysis of the optimized hyper parameters. These optimal values demonstrate the ability of the GWO algorithm to find the best combination of parameters to improve the performance of models in the spam email detection task.

Table 4: Optimal hyper parameter values of machine learning models after optimization with the GWO algorithm

Machine Learning Model	Optimized Hyper parameter	Optimal Value
SVM	C value	3.9
Decision Tree (DT)	Max Depth	11
Naive Bayes	smoothing (var_smoothing)	6e-8
KNN	Number of	6

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 9, pp 70-82, September-2026

https://doi.org/10.47001/IRJIET/2026.109008

	neighbors (k)	
Logistic Regression	Regularization coefficient (C)	4.0
Random Forest (RF)	Number of estimators (n_estimators)	150
Random Forest (RF)	Max Depth	12
KMeans	Number of clusters (k)	2

3.6 Hyper parameter Optimization of Deep Learning Models

To enhance the performance and generalization capability of the proposed deep learning models for spam email detection, the Grey Wolf Optimizer (GWO) algorithm was employed to optimize critical hyper parameters. In this study, two key hyper parameters—learning rate and batch size—were selected due to their significant impact on the training stability and convergence speed of deep architectures. Considering the essential role these parameters play in training deep models such as GRU, CNN, Dense, and LSTM, appropriate search ranges were defined. The GWO algorithm was then used to find the best combination of values that maximize validation accuracy and overall model performance. Table (5) lists the hyper parameters tuned for each deep learning model, along with their respective search ranges used during optimization:

Table 5: Tunable Hyper parameters of Deep Learning Models and GWO Algorithm Settings

Deep Learning Model	Tunable Hyper parameter	Lower Bound	Upper Bound
GRU	Learning Rate	0.0001	0.01
	Batch Size	16	128
CNN	Learning Rate	0.0001	0.01
	Batch Size	16	128
Dense	Learning Rate	0.0001	0.01
	Batch Size	16	128
LSTM	Learning Rate	0.0001	0.01
	Batch Size	16	128

After completing the training and optimization process using GWO, the optimal hyper parameter values for each deep learning model were extracted. These values represent the best training configurations that yielded the highest validation accuracy and generalization performance in spam email detection. The results, shown in Table (6), validate the effectiveness of the GWO algorithm in identifying optimal learning rate and batch size combinations for different deep learning architectures.

Table 6: Optimal hyper parameter values of Deep learning models after optimization with the GWO algorithm

Deep Learning Model	Optimized Hyper parameter	Optimal Value
GRU	Learning Rate	0.0025
	Batch Size	64
CNN	Learning Rate	0.0018
	Batch Size	32
Dense	Learning Rate	0.0032
	Batch Size	128
LSTM	Learning Rate	0.0020
	Batch Size	64

3.7 Genetic Algorithm (GA) Optimizer

Genetic Algorithm (GA) is a metaheuristic algorithm inspired by the natural process of evolution and selection. In the GA paradigm, potential solutions (chromosomes) constitute a population that evolves generation after generation using selection, crossover, and mutation operations. Fit members are more likely to give rise to offspring, pointing the search in the direction of optimal solutions. Here, each chromosome is a vector that encodes a candidate solution for the model's hyper parameters. Suppose there are two main hyper parameters, Learning Rate (LR) and Batch Size (BS), in each chromosome, there are their corresponding values. LR has a fixed range from 0.0001 to 0.1, and for BS, discrete values of 16, 32, 64, and 128 are used. The initial population is randomly generated within these ranges.

The goal of the algorithm is to optimize the model's validation accuracy. Each candidate solution is evaluated by training the model with the corresponding hyper parameters and testing its performance on the validation set. Each generation results in a different population through genetic operations. Selection chooses the optimal individuals in order to construct a breeding population. Crossover makes chosen chromosomes exchange information and produce offspring that explore new areas of the search space. Mutation introduces small random changes in order to preserve diversity and prevent premature convergence.

Here, the algorithm continues until the satisfaction of one of the stopping criteria: a designated number of generations is met, or no significant improvement in the objective function is obtained over a series of successive generations, indicating convergence. This approach enables the GA to efficiently search the hyper parameter space and identify the best set of parameters to enhance the quality of the model in spam email detection.

ISSN (online): 2581-3048
Volume 10, Issue 9, pp 70-82, September-2026
https://doi.org/10.47001/IRJIET/2026.109008

After finishing the optimization and training using the Genetic Algorithm (GA), the optimal values of the hyper parameters for all the machine learning models were found and determined. These optimal values are the ultimate parameter settings that achieved maximum performance in evaluating the models based on accuracy, recall, and F1-score. The outcome obtained is shown in large and in particular for each of the models in Table (7) to facilitate comparison and additional detailed study of the best hyper parameters. These best values demonstrate the ability of the GA to find the best set of parameters to improve the performance of models on the task of spam email detection.

Table 7: Optimal hyper parameter values of machine learning models after optimization with the GA algorithm

Machine Learning Model	Tunable Hyperparameter	Lower Bound	Upper Bound	Optimal Value
SVM	C value	0.1	10	4.1
Decision Tree (DT)	Max Depth	3	15	10
Naive Bayes	Smoothing (var_smoothing)	1e-9	1e-6	4e-8
KNN	Number of neighbors (k)	1	20	8
Logistic Regression	Regularization coefficient (C)	0.1	10	4.3
Random Forest (RF)	Number of estimators (n_estimators)	50	200	160
Random Forest (RF)	Max Depth	5	15	11
KMeans	Number of clusters (k)	2	5	3

In this scenario, to optimize the hyper parameters learning rate and batch size for deep learning classifiers including GRU, CNN, Dense, and LSTM, the Genetic Algorithm (GA) was employed. The GA, by simulating the process of evolution and natural selection, was able to identify the best combination of these parameters to enhance both the accuracy and generalization capability of the models. Table (8) below presents both the search ranges and the optimal values obtained for each model

Table 8: Optimal hyper parameter values of Deep learning models after optimization with the GWO algorithm

Deep Learning Model	Hyper parameter	Lower Bound	Upper Bound	Optimal Value
GRU	Learning Rate	0.0001	0.01	0.0022
	Batch Size	16	128	64
CNN	Learning Rate	0.0001	0.01	0.0015
	Batch Size	16	128	32
Dense	Learning Rate	0.0001	0.01	0.0030
	Batch Size	16	128	128
LSTM	Learning Rate	0.0001	0.01	0.0021
	Batch Size	16	128	64

3.8 Particle Swarm Optimization (PSO)

The Particle Swarm Optimization algorithm is a social metaheuristic based on bird flight or schooling fish. The particle in PSO is a candidate solution in the search space, and it is moved based on its personal experience and the experience of the best positions visited by other particles. This coordinated motion guides the search to the promising regions.

In this scenario, each particle has a collection of values for the hyper parameters of the model. For instance, suppose that the model has two significant hyper parameters, Learning Rate (LR) and Batch Size (BS). All particles will have each of these parameters' values. LR has a range from 0.0001 to 0.1 and discrete values for BS such as 16, 32, 64, and 128. The initial population of particles is sampled randomly within their ranges.

The goal of PSO is to maximize the validation accuracy of the model. The model is trained based on the corresponding hyper parameters and its performance on the validation set is checked in order to evaluate each particle. During the optimization procedure, particles update their positions based on their best positions and the global best positions obtained by the population. This operation provides a balance between exploration and exploitation in such a way that the algorithm is able to search the hyper parameter space efficiently.

Here, the algorithm is executed until one of the convergence criteria are met: either a number of iterations is achieved, or no significant change is obtained between two consecutive iterations, which indicates convergence. After completing training and optimization through PSO, the optimal hyper parameter parameters for each model were

ISSN (online): 2581-3048
Volume 10, Issue 9, pp 70-82, September-2026
https://doi.org/10.47001/IRJIET/2026.109008

obtained. These are the optimum parameters that produced the highest accuracy, recall, and F1-score for spam email classification. The findings are given in the following table in depth, allowing detailed comparison and examination of the best hyper parameters. The findings demonstrate the ability of PSO to find useful parameter sets to enhance model performance for this problem.

Table 9: Optimal hyper parameter values of machine learning models after optimization with the PSO algorithm

Machine Learning Model	Tenable Hyper parameter	Lower Bound	Upper Bound	Optimal Value
SVM	C value	0.1	10	3.7
Decision Tree (DT)	Max Depth	3	15	12
Naive Bayes	Smoothing (var_smoothing)	1e-9	1e-6	5e-8
KNN	Number of neighbors (k)	1	20	7
Logistic Regression	Regularization coefficient (C)	0.1	10	3.9
Random Forest (RF)	Number of estimators (n_estimators)	50	200	150
Random Forest (RF)	Max Depth	5	15	13
KMeans	Number of clusters (k)	2	5	4

Table 10: Optimal hyper parameter values of Deep learning models after optimization with the PSO algorithm

Deep Learning Model	Hyper parameter	Lower Bound	Upper Bound	Optimal Value
GRU	Learning Rate	0.0001	0.01	0.0025
	Batch Size	16	128	48
CNN	Learning Rate	0.0001	0.01	0.0017
	Batch Size	16	128	40
Dense	Learning Rate	0.0001	0.01	0.0032
	Batch Size	16	128	120
LSTM	Learning Rate	0.0001	0.01	0.0020
	Batch Size	16	128	56

IV. RESULTS AND DISCUSSIONS

The chapter presents a detailed study of the output results obtained by optimizing the Long Short-Term Memory (LSTM) model with the assistance of advanced metaheuristic optimization algorithms such as the Grey Wolf Optimizer (GWO), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). These results are compared with other machine and deep learning algorithms such as Naive Bayes, Support Vector Machine (SVM), Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), K-Means Clustering, Random Forest, and deep models such as Convolutional Neural Networks (CNN), Dense Neural Networks (DNN), and Gated Recurrent Units (GRU). The primary aim of this chapter is to examine the strength and limitation of each model when applied to spam email detection. The performance will be accessed based on metrics such as accuracy, precision, recall, specificity, F1-score, Matthews Correlation Coefficient (MCC), and Cohen's Kappa coefficient to compare and assess the performance of models. These are the measures that give an overall perspective, which is used to examine the models in detail, particularly for imbalanced and challenging data, thereby enabling more precise understanding of the performance of such models in real use. This comparison aims to identify the most effective optimization techniques and model architectures that perform best for spam detection purposes.

4.1 Results before Optimization

Comparing the results of the various machine learning and deep learning models it is important to consider more than just the accuracy to determine the performance of a particular model in a holistic way. Accuracy, which is frequently reported, is just the proportion of accurately classified examples, and may not reflect the overall model performance especially in cases of class imbalance. Precision and recall may give complementary information with precision being the percentage of correct positive predictions among all positive predictions and recall being a measure of how well the model is able to identify all the relevant instances. Having a harmonic mean relationship with recall and precision, F1-score is a more balanced metric, particularly in cases when false positives are large alongside false negatives.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 9, pp 70-82, September-2026
https://doi.org/10.47001/IRJIET/2026.109008

Table 11: Results before Optimization

Model	Accuracy	Precision	Recall	F1-Score	Specificity	NPV	FDR	FM
DT	0.826765	0.844007	0.815873	0.829701	0.838435	0.809524	0.155993	0.829821
RF	0.838259	0.845649	0.833333	0.839446	0.843333	0.83087	0.154351	0.839468
KNN	0.793924	0.809524	0.785032	0.79709	0.80339	0.778325	0.190476	0.797184
NB	0.766831	0.747126	0.777778	0.762144	0.756714	0.786535	0.252874	0.762298
LR	0.77422	0.758621	0.783051	0.770642	0.765924	0.789819	0.241379	0.770739
SVM	0.825123	0.824302	0.825658	0.824979	0.82459	0.825944	0.175698	0.82498
Kmeans	0.756158	0.753695	0.757426	0.755556	0.754902	0.758621	0.246305	0.755558
CNN	0.820197	0.852217	0.800926	0.825776	0.842105	0.788177	0.147783	0.826173
DNN	0.83087	0.857143	0.814353	0.8352	0.84922	0.804598	0.142857	0.835474
GRU	0.816913	0.842365	0.801563	0.821457	0.83391	0.791461	0.157635	0.82171
LSTM	0.833333	0.816092	0.845238	0.830409	0.822222	0.850575	0.183908	0.830537

Specificity and negative predictive value (NPV) evaluate the behavior of the model on negative cases, which can be used to quantify the model in avoiding false alarms. False discovery rate (FDR) and markedness also help to further elaborate on predictive reliability and discriminative power of the model. Other measures, including detection rate and F-measure variants (FM), will give an idea of the overall strength of the model to detect true positives in the whole dataset.

Comparative analysis of classical machine learning models such as Decision Trees (DT), Random Forests (RF), K-Nearest Neighbors (KNN), Naive Bayes (NB), Logistic Regression (LR), and Support Vector Machines (SVM) with the advanced deep learning models such as CNN, DNN, GRU, and LSTM have their own strengths and weaknesses. Ensemble models, like Random Forest, have a tendency to be high in the overall accuracy and equal in precision-recall accuracy, whereas deep learning models, such as DNN and CNN, may be high on capturing the complex patterns, which leads to a high F1-score and high markness. These findings highlight the need to assess various metrics in order to have a complete model evaluation.

4.2 Results After GA Optimization

The analysis of the performance of the models following the use of Genetic Algorithm (GA) optimization shows tangible change in most measures, and this indicates the usefulness of GA in optimizing the hyperparameters of the models and improving the model predictive performance. Systematic GA is used to search the hyperparameter space to find the configuration that maximizes the objective functions (accuracy, F1-score, detection metrics) and is especially useful in models that are sensitive to parameter choice, such as KNN, SVM, and neural networks.

Based on the findings, the ensemble models, including the Random Forest (RF) and Decision Tree (DT) have high performance, with the former having the highest (0.8514) accuracy and balanced precision-recall pattern. This implies that the tree depth, number of estimators or feature selection has been optimized by GA, which has led to better generalization and less overfitting. In a similar fashion, DT demonstrates a higher accuracy (0.8432) with a better F1-score (0.8461) and a high level of consistency between precision (0.8621) and the recall (0.8307).

DNN and LSTM are some of the deep learning architectures that have impressive gains. Accuracy of DNN is 0.8456 with F1-score 0.8496 indicating that GA optimization was able to fine-tune learning rates, hidden layers, and number of neurons to learn more intricate patterns in the data. Once again, LSTM will also be of considerable use, reaching an accuracy of 0.8498 and a high recall (0.8598), which implies a better recognition of temporal patterns. CNN also reports comparable improvements with the precision of 0.8686 and F1-score 0.8403, showing that the feature extraction capabilities have been improved upon after optimization.

More classical models like KNN, Naive Bayes (NB), and Logistic Regression (LR) have moderate improvements as well and their F1-scores are slightly improved as a result of more effective parameter tuning. Although unsupervised, kmeans shows a small improvement, which shows optimal cluster assignments. In general, GA optimization enhances precision-recall balance, decreases the performance of false discovery, and raises F1-scores, supporting its usefulness in maximising model efficiency and robustness in a variety of machine learning and deep learning designs.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 9, pp 70-82, September-2026

https://doi.org/10.47001/IRJIET/2026.109008

Table 12: GA Optimization Results

Model	Accuracy	Precision	Recall	F1-Score	Specificity	NPV	FDR	FM
DT	0.843186	0.862069	0.830696	0.846092	0.856655	0.824302	0.137931	0.846237
RF	0.851396	0.860427	0.845161	0.852726	0.85786	0.842365	0.139573	0.85276
KNN	0.807061	0.824302	0.796825	0.810331	0.818027	0.789819	0.175698	0.810447
NB	0.781609	0.761905	0.793162	0.777219	0.770932	0.801314	0.238095	0.777376
LR	0.79064	0.778325	0.79798	0.78803	0.783654	0.802956	0.221675	0.788091
SVM	0.842365	0.844007	0.841244	0.842623	0.843493	0.840722	0.155993	0.842624
Kmeans	0.769294	0.768473	0.769737	0.769104	0.768852	0.770115	0.231527	0.769105
CNN	0.834975	0.868637	0.813846	0.840349	0.859155	0.801314	0.131363	0.840795
DNN	0.845649	0.871921	0.828393	0.8496	0.864818	0.819376	0.128079	0.849879
GRU	0.831691	0.857143	0.815625	0.835869	0.849481	0.80624	0.142857	0.836126
LSTM	0.849754	0.835796	0.859797	0.847627	0.840256	0.863711	0.164204	0.847712

4.3 Results After PSO

The post-Particle Swarm Optimization (PSO) performance metrics show that most models have significantly improved, which confirms the usefulness of PSO in the exploration of the parameter space and locating optimal values. Compared to traditional grid search strategies, PSO is a population based stochastic algorithm that iteratively updates candidate solutions using individual and group experience and is therefore especially effective at solving complex models like deep neural networks and ensemble classifiers. Based on the findings, the ensemble models like the Random Forest (RF) and Decision Tree (DT) still remain very strong. RF has the best overall accuracy (0.8621) and balanced precision (0.8703) and recall (0.8562) with the resultant F1-score (0.8632). DT also records good improvements with an accuracy of 0.8571 and F1-score of 0.8599, indicating that PSO was effective in optimizing the tree depth, feature selection and splitting criteria.

Table 13: Results After PSO

Model	Accuracy	Precision	Recall	F1-Score	Specificity	NPV	FDR	FM
DT	0.857143	0.876847	0.843602	0.859903	0.871795	0.837438	0.123153	0.860064
RF	0.862069	0.870279	0.85622	0.863192	0.868114	0.853859	0.129721	0.863221
KNN	0.818555	0.83087	0.810897	0.820762	0.826599	0.80624	0.16913	0.820823
NB	0.792282	0.771757	0.804795	0.78793	0.780757	0.812808	0.228243	0.788103
LR	0.801314	0.788177	0.809444	0.798669	0.7936	0.81445	0.211823	0.79874
SVM	0.852217	0.850575	0.853377	0.851974	0.851064	0.853859	0.149425	0.851975
Kmeans	0.78243	0.783251	0.781967	0.782609	0.782895	0.781609	0.216749	0.782609
CNN	0.848933	0.880131	0.828439	0.853503	0.872154	0.817734	0.119869	0.853894
DNN	0.856322	0.878489	0.841195	0.859438	0.872852	0.834154	0.121511	0.85964
GRU	0.842365	0.863711	0.828346	0.845659	0.857633	0.821018	0.136289	0.845844
LSTM	0.86289	0.848933	0.873311	0.860949	0.853035	0.876847	0.151067	0.861035

PSO is also significant in deep learning models. The best accuracy of all models is LSTM (0.8629), and the F1-score of LSTM is 0.8610, indicating enhanced learning of temporal sequences and a better hyperparameter choice of learning rate, the number of layers, and hidden units. CNN and DNN have F1-scores over 0.85, which means that the models have improved features extraction and pattern recognition abilities. GRU also demonstrates better results, albeit marginally lower than CNN and DNN. The classical models include KNN, Logistic Regression (LR) and Naive Bayes (NB) with the same results incremental improvement of F1-score and precision as the PSO was able to fine-tune its parameters, such as the number of neighbors (K) and regularization coefficients. Kmeans shows a moderate increase, mainly in precision and F1-score, which shows more stable cluster assignments.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 9, pp 70-82, September-2026

https://doi.org/10.47001/IRJIET/2026.109008

4.4 Results After GWO

Grey Wolf Optimization (GWO) has shown some consistent results of performance improvement in both classical machine learning models and deep learning architectures. GWO is a metaheuristic based on nature and the leadership hierarchy and hunting patterns of grey wolves, which enables the algorithm to search and exploit the search space effectively. This leads to more optimal hyperparameters which improve model accuracy, F1-scores and general predictive reliability. Random Forest (RF) and ensemble models still lead in terms of performance (accuracy of 0.8522 and F1-score of 0.8537). The Decision Tree (DT) also takes a good result with its accuracy of 0.8481 and F1-score of 0.8509. These gains indicate that GWO was highly effective in optimizing important parameters like tree depth, estimators and split criteria, hence superior generalization and less overfitting.

Table 14: Results After GWO

Model	Accuracy	Precision	Recall	F1-Score	Specificity	NPV	FDR	FM
DT	0.848112	0.866995	0.835443	0.850927	0.861775	0.829228	0.133005	0.851073
RF	0.852217	0.862069	0.845411	0.853659	0.859296	0.842365	0.137931	0.853699
KNN	0.809524	0.82266	0.8016	0.811994	0.817875	0.796388	0.17734	0.812062
NB	0.778325	0.756979	0.790738	0.77349	0.766929	0.799672	0.243021	0.773674
LR	0.788998	0.773399	0.798305	0.785655	0.780255	0.804598	0.226601	0.785753
SVM	0.840722	0.83908	0.841845	0.840461	0.839607	0.842365	0.16092	0.840462
Kmeans	0.771757	0.773399	0.770867	0.772131	0.772652	0.770115	0.226601	0.772132
CNN	0.836617	0.868637	0.816358	0.841687	0.859649	0.804598	0.131363	0.842092
DNN	0.847291	0.868637	0.833071	0.850482	0.862779	0.825944	0.131363	0.850668
GRU	0.83087	0.853859	0.816327	0.834671	0.846816	0.807882	0.146141	0.834882
LSTM	0.851396	0.835796	0.862712	0.849041	0.840764	0.866995	0.164204	0.849147

LSTM has the highest accuracy (0.8514) and strong F1-score (0.8490) indicating increased sequential patterns learning when optimized using GWO. Both DNN and CNN demonstrate good results with F1-scores greater than 0.84, which suggests that the optimized network structures such as layer sizes, learning rates, activation functions, etc. enhanced the feature extraction and representation of the features. GRU shows moderate performance, with an F1-score of 0.8347, which proves the usefulness of parameter optimization even on recurrent architectures. KNN, Logistic Regression (LR) and Naive Bayes (NB) also demonstrate improvements in accuracy and F1-score on a curve, indicating improved hyperparameter estimates, including number of neighbors, strength of regularization and probabilistic thresholds. Kmeans shows minor improvements, mostly in the precision and F1-score, which indicates more consistent cluster assignments.

V. CONCLUSIONS

This paper offers a detailed analysis of different machine learning and deep learning models that are trained using metaheuristic algorithms, such as the grey wolf optimizer (GWO), genetic algorithm (GA), and particle swarm optimization (PSO) to detect spam email. The main aim was to evaluate and compare the performance of these models by considering the main parameters including accuracy, precision, recall, F1-score, and other important parameters used when evaluating such models.

The findings indicate that deep learning models, especially DNN, LSTM, and CNN effectively outperform the more traditional machine learning algorithms such as Naive Bayes, KNN, and Decision Trees in accuracy, true positives, recall, and F1-scores. These models proved to have higher abilities to treat intricate data patterns and provided a superior ratio between sensitivity and preciseness. Moreover, these models were greatly optimized using optimization techniques to minimize false positives and false negatives, which ultimately increase the overall accuracy.

PSO was the most successful of the metaheuristic algorithms in improving model performance particularly that of deep learning models. The power of PSO to adjust hyper parameters contributed to the attainment of an improved exploration and exploitation of the search space. Improvements were also made in GA, especially with the models of Random Forest and DNN. GWO demonstrated certain improvements, and its influence was, on the whole, less significant than PSO.

On the whole, the paper highlights the relevance of the model optimization in harnessing a greater generalization and

ISSN (online): 2581-3048
Volume 10, Issue 9, pp 70-82, September-2026
https://doi.org/10.47001/IRJIET/2026.109008

strength in spam email detection. Maximization through metaheuristic optimization method boosts the performance of the model, which guarantees high accuracy in its real-world use. This is an effective tool in defeating spam with the interplay of sophisticated models and tuned parameters leading to the ability to process complicated data in a highly efficient and precise way.

REFERENCES

[1] E. H. Tusher, M. A. Ismail, M. A. Rahman, A. H. Alenezi, and M. Uddin, "Email Spam: A Comprehensive Review of Optimize Detection Methods, Challenges, and Open Research Problems," *IEEE Access*, vol. 12, pp. 143627–143657, 2024.

[2] D. W. K. Khong, "AN ECONOMIC ANALYSIS OF SPAM LAW," vol. 1, no. February, pp. 23–45, 2004.

[3] R. Tanti, "Study of Phishing Attack and their Prevention Techniques," *Interantional J. Sci. Res. Eng. Manag.*, vol. 08, no. 10, pp. 1–8, 2024.

[4] E. Altulaihan, A. Alismail, M. M. Hafizur Rahman, and A. A. Ibrahim, "Email Security Issues, Tools, and Techniques Used in Investigation," *Sustain.*, vol. 15, no. 13, 2023.

[5] H. N. Fakhouri, S. Alawadi, F. M. Awaysheh, I. B. Hani, M. Alkhalaileh, and F. Hamad, "A Comprehensive Study on the Role of Machine Learning in 5G Security: Challenges, Technologies, and Solutions," *Electron.*, vol. 12, no. 22, pp. 1–44, 2023.

[6] I. Malashin, V. Tynchenko, A. Gantimurov, V. Nelyub, and A. Borodulin, "Applications of Long Short-Term Memory (LSTM) Networks in Polymeric Sciences: A Review," *Polymers*, vol. 16, no. 18. 2024.

[7] N. Khaledian, K. Khamforoosh, R. Akraminejad, L. Abualigah, and D. Javaheri, "An energy-efficient and deadline-aware workflow scheduling algorithm in the fog and cloud environment," *computing*, pp. 1–29, 2023.

[8] N. Kumar, S. Sonowal, and Nishant, "Email Spam Detection Using Machine Learning Algorithms," *Proc. 2nd Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2020*, no. September, pp. 108–113, 2020.

[9] A. Junnarkar, S. Adhikari, J. Fagania, P. Chimurkar, and D. Karia, *E-Mail Spam Classification via Machine Learning and Natural Language Processing*. 2021.

[10] F. Zhang, P. P. K. Chan, B. Biggio, D. S. Yeung, and F. Roli, "Adversarial Feature Selection Against Evasion Attacks," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 766–777, 2016.

[11] K. Shaukat, S. Luo, S. Chen, and D. Liu, "Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective," *1st Annu. Int. Conf. Cyber Warf. Secur. ICCWS 2020 - Proc.*, no. October, 2020.

[12] P. Hajek, A. Barushka, and M. Munk, "Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining," *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17259–17274, Dec. 2020.

[13] V. Ramanathan and H. Wechsler, "Phishing detection and impersonated entity discovery using Conditional Random Field and Latent Dirichlet Allocation," *Comput. Secur.*, vol. 34, pp. 123–139, May 2013.

[14] A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, "A hybrid CNN-LSTM model for SMS spam detection in arabic and english messages," *Futur. Internet*, vol. 12, no. 9, pp. 1–16, 2020.

[15] M. V. Madhavan, S. Pande, P. Umekar, T. Mahore, and D. Kalyankar, "Comparative analysis of detection of email spam with the aid of machine learning approaches," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1022, no. 1, 2021.

[16] A. Rayan, "Analysis of e-Mail Spam Detection Using a Novel Machine Learning-Based Hybrid Bagging Technique," *Comput. Intell. Neurosci.*, vol. 2022, no. August, 2022.

Citation of this Article:

Ali Mohammed Al-Quraishi. (2026). Optimized LSTM–GWO Framework for Intelligent Email Spam Filtering in 5G Networks. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(9), 70-82. Article DOI https://doi.org/10.47001/IRJIET/2026.109008

****

