GOA-WO-ML: Enhancing Internet of Things Security with Gannet Optimization and Walrus Optimizer-Based Machine Learning

Jing GUO*, Wen CHEN, Xu ZHANG
School of Big Data, Chongqing College of Mobile Communication, Chongqing City, 401520, China
Chongqing Key Laboratory of Public Big Data Security Technology, Chongqing City, 401520, China

***Abstract*—The rapid development of the Internet of Things (IoT)-based Wireless Sensor Networks (WSNs) has fueled security challenges, necessitating efficient intrusion detection approaches. The computationally intensive nature and the high-dimension data preclude the direct employment of machine learning-based Intrusion Detection Systems (IDSs). This study introduces GOA-WO-ML, a robust IDS system that integrates the Gannet Optimization Algorithm (GOA) and Walrus Optimizer (WO) for feature selection and parameter tuning in machine learning algorithms. The system is tested on the NSL-KDD dataset, indicating better cyberattack detection performance. The experimental findings suggest that GOA-WO-ML improves intrusion detection accuracy, decreases false positives, and has low computational overhead compared to traditional methods. By adopting bio-inspired methods, the proposed system successfully counteracts security issues in IoT-WSNs through efficient surveillance. Future research directions include considering deep learning improvements and real-time deployment methods in dynamic environments for further intrusion detection performance.**

Keywords*—Internet of things; intrusion detection; machine learning; optimization

I. INTRODUCTION

The Internet of Things (IoT) and Wireless Sensor Networks (WSNs) have revolutionized many industries by allowing for easy communications and real-time data acquisition [1]. The networks comprise many connected devices used to monitor, control, and automate processes. The IoT-enabled WSNs are used in health, agriculture, smart cities, and environmental surveillance [2]. However, increased connectivity and the use of radio frequency communications make them susceptible to a broad variety of cyber-attacks, including unauthorized access, data manipulation, and denial-of-service attacks [3].

Intrusion Detection Systems (IDSs) are vital for protecting IoT-WSNs by detecting malicious behaviors in the network and initiating proper countermeasures. With growing IoT networks, the role of effective intrusion detection mechanisms gains significance [4]. A good IDS must recognize various real-time attacks while maintaining network security and firmness. Classical IDSs mostly depend upon preconfigured signatures or simple anomaly detection approaches. Such systems are mainly unable to keep pace with changing threats in dynamic and resource-limited environments such as IoT-WSNs [5].

The main problem in designing an IDS for IoT-WSNs is dealing with the high dimensionality of data in these networks [6]. The extensive network traffic data with high dimensionality demands sophisticated feature selection and optimization techniques. Moreover, conventional IDS methods are burdened by excessive false alarms, delayed detection, and ineffective parameter adjustment. ML has demonstrated the potential to overcome these challenges, but the workload remains a problem without optimization. Also, traditional optimization methods are not effective in dealing with the high-dimensional, noisy nature of data in IoT-WSNs, resulting in suboptimal outcomes.

In response to these challenges, we introduce a new hybrid IDS, GOA-WO-ML, integrating the Gannet Optimization Algorithm (GOA) and the Walrus Optimizer (WO) algorithms for effective feature selection and parameter adjustment in Machine Learning (ML) models. GOA is a bio-mimetic search algorithm based on the predation mechanism of the gannet [7], whereas the WO has mimicked the behavior of walruses [8]. Machine learning has demonstrated significant utility across diverse domains, including healthcare, agriculture, and economics, by uncovering patterns in high-dimensional data [9]. These capabilities are now being increasingly adopted to secure IoT environments through intelligent intrusion detection frameworks.

This system, leveraging the potential of GOA and WO, improves intrusion detection accuracy and minimizes false positives and computation complexities. We couple these optimization algorithms with a Support Vector Machine (SVM) classifier, guaranteeing top performance even in large, high-dimensional datasets such as NSL-KDD.

The remaining content of this paper is presented as follows. Section II reviews related research on intrusion detection in IoT-WSNs. Section III discusses the details of the proposed GOA-WO-ML system. The results section is given in Section IV and discussions comparing the efficiency of GOA-WO-ML with conventional IDS approaches are given in Section V. Section VI offers an in-depth discussion of the findings. The conclusion of this paper and research directions are given in Section VI.

II. RELATED WORK

Zhao, et al. [10] proposed a Network Intrusion Detection (NID) system for IoT based on a Lightweight Deep Neural Network (LDNN). They used dimension reduction with Principal Component Analysis (PCA) to overcome raw traffic

features limitations in high dimensions. The LNN contains a compressible and expanded design, residual inverse architecture, and shuffled channels for low-complexity feature extraction. They also proposed a novel NID loss function for imbalanced sample distribution multiclassification.

Gangula and V [11] proposed a network intrusion detection system utilizing the Improved Flower Pollination Algorithm (IFPA) and ensemble classification. Using a scaling factor for improved convergence, they employed IFPA to choose the best features from the NSL-KDD and UNSW-NB15 datasets. The features identified were passed through an ensemble classifier, where multiple models, random forest, decision trees, and SVM were combined.

Asgharzadeh, et al. [12] suggested a deep learning-based intrusion detection scheme for IoT devices through automatic feature extraction with Convolutional Neural Networks (CNNs). They employed a hybrid model, IoTFECNN, based on combining deep learning with a Binary Multi-objective Improved Capuchin Search Algorithm (BMECapSA) for feature selection.

Recent studies have explored the integration of vision transformers with convolutional architectures for more accurate and scalable classification tasks. For example, a fuzzy hybrid stacked ensemble combining ViTs and CNNs was proposed to detect defects in metal surfaces, demonstrating the effectiveness of hybrid deep learning approaches in real-time industrial environments [13].

Hanafi, et al. [14] developed a new intrusion detection system by utilizing an optimized Binary Golden Jackal Optimization (BGJO) algorithm, along with LSTM networks. The OBL was used to optimize the IBGJO for optimal feature selection and prevent local optima. The BGJO-LSTM model resulted in an accuracy of up to 98.2% for CICIDS2017 and NSL-KDD datasets. The results were more accurate than those of BGJO-LSTM and in contrast with other SVM-based methods.

Yang, et al. [15] developed a Lightweight Convolutional Neural Network (LSCNN) for detecting intrusions in the IoT, targeting high-dimensional data. They proposed a Data Purification Algorithm (DPA) to transform unstructured data into images, eliminate duplicate data, and enhance the performance of CNN. LSCNN, drawing from separate convolution, was more efficient in terms of time consumption and detecting intrusions.

Makhadmeh, et al. [16] introduced a novel network IDS, MPAC, derived from the Marine Predators Algorithm (MPA) augmented by a crossover operator. MPAC emphasizes effective feature selection by optimizing the most valuable features for NIDS. They showed MPAC's performance is better than that of alternative methods, with high accuracy in various datasets. The system reported strong results, performing better in detecting network attacks than various current models.

Shi, et al. [17] presented an ensemble system of intrusion detection for the security of IoTs based on an Enhanced Artificial Hummingbird Algorithm (EAHA). The system employed the binary version of EAHA (BEAHA) in feature selection and ensemble classifier design for intrusions in a network. The accuracy of their model, when validated through use with CSE-CIC-IDS2018, CIC-IDS2017, and NSL-KDD benchmark datasets performed well while reducing feature dimensionality by at least 69%, demonstrating efficiency as well as competitiveness.

Asif [18] proposed the OSEN-IoT, a stacked ensemble network for IoT, by utilizing multiple convolutional neural networks (DenseNet121, MobileNetV2, and ResNet50V2) in a stacking manner. The system is augmented by a channel and a spatial attention mechanism, and it was optimized by a genetic algorithm. OSEN-IoT achieved better performance with accuracy rates of 99.71% in Edge-IIoTset, 99.15% in UNSW-NB15, and other data sets, surpassing current deep learning approaches in cyber-threat detection.

TABLE I. RECENT INTRUSION DETECTION SYSTEMS FOR IOT NETWORKS

Reference	Key techniques	Datasets	Accuracy	Shortcoming
[10]	Lightweight deep neural network and PCA for feature reduction	UNSW-NB15 and Bot-IoT	96.15 and 86.11	The model is computationally intensive and may overfit on small datasets.
[11]	Enhanced flower pollination algorithm and ensemble classifier	UNSW-NB15 and NSL-KDD	99.32% and 99.67%	The ensemble approach is complex and may struggle with feature selection flexibility.
[12]	Convolutional neural networks and enhanced capuchin search algorithm	TON-IoT and NSL-KDD	99.99% and 99.85%	The model consumes high resources and has slow training times.
[14]	Binary golden jackal optimization and LSTM	NSL-KDD and CICIDS2017	98.21% and 99.25	Sensitive to class imbalances and requires intricate parameter tuning.
[15]	Lightweight CNN and data purification algorithm	AWID and NSL-KDD	91.7 % and 85.13%	May not generalize well to unseen attack types and is computationally complex.
[16]	Marine predators algorithm with crossover operator	NSL-KDD, UNSW-NB15, Bot-IoT2018, and CICIDS2017	99.58%, 98.98%, 99.98%, 97.67%	The method may converge to local optima and demands extensive computational resources.
[17]	Artificial hummingbird algorithm and ensemble classifier	NSL-KDD, CIC-IDS2017, and CSE-CIC-IDS2018	99.74%, 99.59%, and 98.51%	The method can suffer from overfitting on large datasets and has slower convergence.
[18]	Stacked ensemble CNN and genetic algorithm	Edge-IIoTset, UNSW-NB15, and IoT_Malware	99.71%, 99.15%, and 96.17%	The model is computationally expensive and may not scale efficiently for larger datasets.

The current approaches are primarily based on traditional ML methods or sophisticated deep learning approaches, not optimizing feature selection and model complexity for IoT-WSNs. Based on the observations in Table I, most methods aim to improve performance with large-sized models, which are inappropriate for the resource-limited environment of IoT devices. Moreover, most traditional methods do not handle high-dimensional, noisy data, resulting in suboptimum performance.

There is also a demand for efficient real-time detection methods, maintaining a trade-off between accuracy, efficiency, and computational overhead.

Our GOA-WO-ML framework fills all these gaps by integrating the GOA and WO, which are superior in exploration and exploitation for feature selection and parameter adjustment. Our hybrid solution balances high detection accuracy with low computational complexity, better fitting the environment of the IoT. Our system is capable of performing well under real-time conditions, thereby helping develop intrusion detection methods for IoT-enabled WSNs.

III. Proposed Intrusion Detection System

Hybrid metaheuristic algorithms have proven effective in addressing complex optimization problems in smart grids and energy systems [19]. Inspired by such advances, the present study proposes a new, automated GOA-WO-ML technique involving the combination of GOA with ML to ensure effective intrusion detection in the domain of IoT-integrated WSNs. The GOA-WO-ML methodology uses nature-based optimization techniques, GOA, and ML to design a robust and reliable IDS for IoT-WSNs.

This study plans to secure the WSN-IoT systems by utilizing the GOA-WO-ML methodology to detect intrusions promptly. The mechanism streamlines the intrusion detection process while ensuring efficient detection and dependability, thereby maintaining data and device integrity in the network. The method generally contributes to detecting intrusions, effectively solving the security issues introduced by the ubiquitous and widely dispersed characteristics of IoT and WSN equipment.

GOA-WO-ML enables accurate differentiation between multiple cyberattacks, thereby enhancing WSN-IoT security. The process comprises four major steps: data scaling, selecting GOA features, classification with ML, and parameter tuning utilizing WO. Fig. 1 illustrates the framework for detecting threats and upgrading security in the WSN-IoT environment.

The initial step in GOA-WO-ML is data standardization, in which the values are equivalent to a predefined range. This standardization helps keep the weighted summation of inputs within the limits in model initialization. In some cases, it can lead to inefficient training and slower convergence when data is not scaled appropriately. On the other hand, data scaling decreases dimensionality, thus enabling faster processing. Eq. (1) is employed, in which data is scaled by mapping it between zero and one.

$$Z_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Where Z_{norm} represents the scaled data, x is the original value, x_{max} is the maximum value, and x_{min} is the minimum value used for scaling.

GOA is used to select the most applicable features in the intrusion detection mechanism for IoT-enabled WSNs. Its motivation comes from the diving nature of gannets, which plunge from a great height to catch prey in the water. This nature-inspired behavior is mimicked for searching and exploring feature space in optimal feature selection. The process starts with creating a first population of solutions, in which every solution contains a set of feature subsets in the search space. The solutions are initialized at random. The locations of the individuals in the population are stored in a matrix, as in Eq. (2).

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,Dim1} & x_{1,Dim-1} \\ x_{2,1} & \cdots & x_{2,Dim-1} & x_{2,Dim} \\ \vdots & \ddots & \vdots & \vdots \\ x_{N,1} & \cdots & x_{N,Dim-1} & x_{N,Dim} \end{bmatrix} \quad (2)$$

Where x_i denotes the position of the i^{th} individual, and each element $x_{i,j}$ in the matrix X is calculated using Eq. (3).

$$x_{i,j} = r_1 \times \left(UB_j - LB_j\right) + LB \quad (3)$$

Where UB_j and LB_j are the upper and lower bounds for the j^{th} dimension of the problem, respectively, N represents the number of individuals in the population, Dim refers to the dimensional size of the problem, and r_1 is a random number between 0 and 1.

A memory matrix MX is also introduced, in which the best positions of the individuals during the optimization process are stored. The memory matrix plays a central role in retaining the optimum solutions in memory for future iterations. The memory matrix is refreshed after each evolution step by calculating the fitness value of each individual. For each candidate solution, whenever it is better, it replaces the incumbent solution in the memory matrix with its position. Otherwise, the solution from the current matrix is retained.

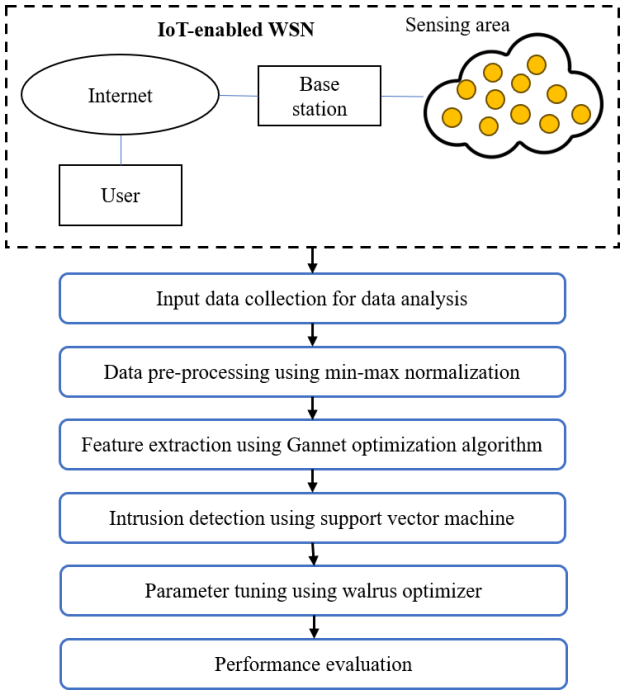

Fig. 1. Workflow of the GOA-WO-ML methodology.

The second step is exploring the feature space by the two dive strategies motivated by gannets: the U-shaped dive, also known as the plunge dive, and the V-shaped dive. As shown in Fig. 2, these strategies lead the search process, in which the U-shaped dive corresponds to long-range, deep exploration, while

the V-shaped dive represents a concentrated, shallow search. The U-shaped dive is controlled by Eq. (4) and Eq. (5).

$$\alpha = 2.\cos(2.\pi.r_2).t \quad (4)$$

$$b = 2.\sqrt{2.\pi.r_3}.t \quad (5)$$

Where r_2 and r_3 are random numbers between 0 and 1, and t is the iteration count. These calculations help determine the trajectory of the dive, simulating the search behavior of gannets. For the V-shaped dive, the update is more localized, given in Eq. (6).

$$V(x) = \begin{cases} -\frac{1}{\pi}.x + 1, & if\ x \in (0, \pi) \\ \frac{1}{\pi}.x - 1, & if\ x \in (\pi, 2\pi) \end{cases} \quad (6)$$

In this case, the algorithm uses the V-shaped dive to narrow down the search space, focusing more precisely on potential solutions. Once the exploration phase identifies promising solutions, the exploitation phase begins. Here, the position of each individual is refined based on the best-performing solution so far, represented as X_{Best}, and the average position of all individuals in the population, X_m. The update is performed using Eq. (7).

$$MX_i(t+1) = \begin{cases} X_i(t) + \beta 1 + \beta 2, & q \geq 0.5 \\ X_i(t) + \upsilon 1 + \upsilon 2, & q < 0.5 \end{cases} \quad (7)$$

Where q is a random number used to choose between the two dive strategies. The variables $\beta 1$ and $\upsilon 2$ are calculated using Eq. (8) and Eq. (9).

$$\beta 2 = A.(X_i(t) - X_r) \quad (8)$$

$$\upsilon 2 = B.(X_i(t) - X_m) \quad (9)$$

Where A and B are scaling factors defined by Eq. (8) and Eq. (9).

$$A = (2.r_4 - 1).\alpha \quad (8)$$

$$B = (2.r_5 - 1).b \quad (9)$$

These position updates guide the algorithm towards more optimal feature sets by refining the current solutions. The parameter values r_4 and r_5 are random values between 0 and 1, ensuring a stochastic search.

In the exploitation phase, the GOA adjusts the position of each individual solution based on the best-performing solution found so far (X_{Best}) and the average position of all individuals. The position update is based on a random selection of exploration or exploitation strategies using a random number q. The update rule is as follows:

$$MX_i(t+1) = \begin{cases} X_i(t) + \delta.\left(X_i(t) - X_{Best}(t)\right) + \\ X_i(t), \quad if\ Capturability \geq c \\ X_B(t) - \left(X_i(t) - X_{Best}(t)\right). \\ P.t, \quad if\ Capturability < c \end{cases} \quad (10)$$

Where P is the Levy flight function and δ is computed based on capturability, calculated using Eq. (11).

$$\delta = Capturability.|X_i(t) - X_{Best}(t)| \quad (11)$$

The Levy flight is used to refine the search and help escape local optima by introducing randomness into the search process, calculated as follows:

$$P = Levy(Dim) = 0.01 \times \frac{\mu.\sigma}{|\upsilon|^{\frac{1}{\beta}}} \quad (12)$$

Where μ and σ are random values between 0 and 1, and β = 1.5 β=1.5 is a pre-determined constant. Additionally, the scaling factor σ is defined using Eq. (13).

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (13)$$

The capturability metric determines whether the algorithm can effectively catch the optimal feature subset, calculated as follows:

$$Capturability = \frac{1}{R.t_2} \quad (14)$$

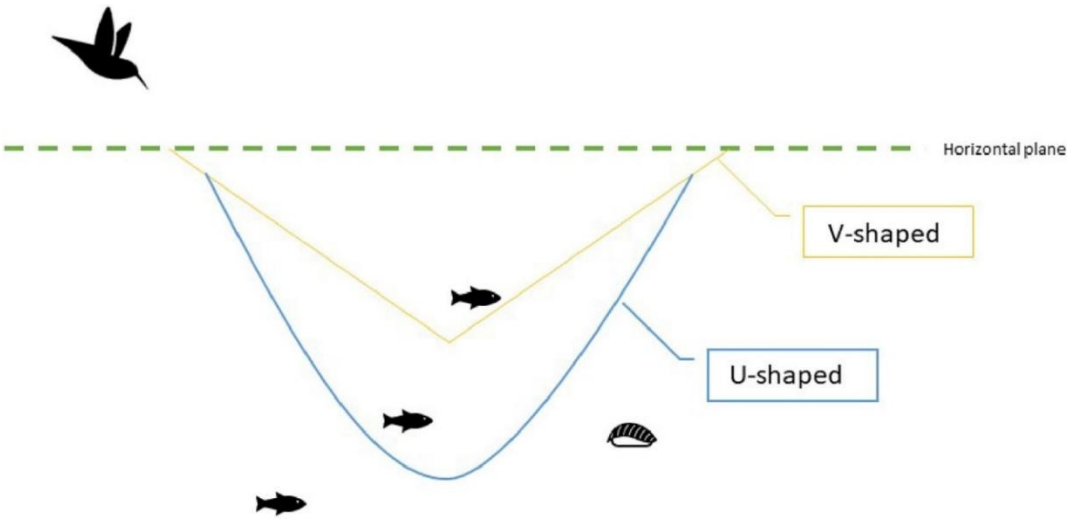

Fig. 2. Exploration strategies in GOA-WO-ML.

Where R represents the energy required to catch the optimal feature subset, and t_2 adjusts based on the time spent during the optimization process. This ensures that the algorithm refines its search for the most relevant features as the optimization progresses, ultimately converging on an optimal feature subset.

Algorithm 1 presents the pseudocode of the GOA. The GOA keeps refining the solutions until the stopping conditions are achieved, usually when the solution converges or the maximum iterations are reached. The optimal solution for intrusion detection in IoT-WSNs is represented by the feature subset thus selected. Overall, GOA-based feature selection judiciously balances exploration with exploitation to identify the best features for intrusion detection. Modeling the foraging behavior of the gannet, the algorithm successfully narrows the feature set while making detection more accurate, keeping costs low.

Algorithm 1 Pseudocode of GOA

Inputs: population size (N), dimensionality of the problem (Dim), and the maximum iteration count ($T_{\max\_iter}$).

Outputs: The optimal solution (location of the gannet) and its associated fitness score.

Random initialization:

Initialize the population of solutions X randomly. Each solution X_i is assigned values within the bounds of the problem as specified in Eq. 2.

Memory matrix setup:

Create an auxiliary matrix MX to store the best solutions encountered during the optimization process.

Fitness evaluation:

Compute the fitness values for all solutions in the population.

Optimization loop:

Repeat the following steps until the maximum number of iterations is reached:

Decision between exploration and exploitation:

A random number $rand$ is generated.

If $rand$ > 0.5, proceed with the exploration strategy; otherwise, exploit the best solutions.

Exploration phase:

For each solution in the memory matrix MX:

If $q \geq 0.5$, update the position using Eq. 7a to explore the feature space.

If $q < 0.5$, update the position using Eq. 7b for a different exploration pattern.

Exploitation phase:

For each solution in the memory matrix MX:

If the capturability value exceeds a threshold, update the solution using Eq. 10a to exploit the current best solutions.

Otherwise, use Eq. 10b to perform a less aggressive exploitation.

Memory update:

After calculating the fitness of all solutions in MX, compare them with the corresponding solutions in X.

If a solution in MX outperforms its counterpart in X, replace the corresponding solution in X with the one from MX.

Termination:

End the optimization process when the stopping condition is fulfilled.

Intrusion detection in WSN-IoT systems takes advantage of the SVM classifier and parameter optimization through optimization algorithms. The GOA is employed for feature selection optimization and bias, while the WO is used to fine-tune the SVM parameters to enhance classifying performance.

To begin, we assume an input vector $x = [x_1, x_2, \dots, x_n]$, where the network contains M neurons in the hidden layer. The weighted sum of inputs for each neuron in the hidden layer is calculated using Eq. (15).

$$z_i = \sum_{j=1}^{M} w_{ij}.x_j + b_i \quad (15)$$

Where w_{ij} represents the weight from the input x_j to the neuron in hidden layer i, and b_i denotes the hidden neuron's bias.

A non-linear activation function is applied to the weighted sum of each hidden neuron, with the tanh function being one possible example:

$$h_i = activation(z_i) \quad (16)$$

For each output neuron k, a weighted sum of the hidden layer outputs is computed:

$$y_k = \sum_{i=1}^{M} v_{ki}.h_i + c_k \quad (17)$$

Where v_{ki} is the weight from the i^{th} hidden neuron to the k^{th} output neuron, and c_k is the bias term for the k^{th} output neuron.

The final output is obtained by applying the activation function to the weighted sum of the outputs from each output neuron:

$$y_k = activation(y_k) \quad (18)$$

GOA is used for feature selection and bias terms in SVM optimization. GOA replicates the predatory nature of gannets and is employed to determine the most informative features from a big dataset, thereby decreasing feature space while ensuring better computational efficiency. GOA conducts exploration and exploitation in feature space by applying a blend of random search methods and iterative refining. The optimization allows the most valuable features to be selected for training SVM, thereby improving the accuracy and efficiency of the model.

Subsequent SVM parameter tuning is performed by utilizing the WO. The WO performs excellently in global optimization problems by iteratively traversing the solution space, searching for close-to-optimum SVM parameter configurations. The algorithm fine-tunes the SVM's parameters, including the penalty term C and the kernel coefficients, in a search for the highest attainable classification accuracy.

The integrated system is a good and effective alternative for detecting intrusion in IoT-based WSNs by utilizing GOA for feature selection and WO for parameter tuning. The mechanism proposed by the authors maintains computations while ensuring good performance in terms of classification.

The weight terms w_{ij} and v_{ki}, as well as the biases b_i and c_k, are updated using gradient descent. The GOA helps identify the most relevant features from the input data, while WO ensures that the SVM's hyperparameters are optimized for the best performance. This combined approach leads to a more efficient and accurate intrusion detection model capable of identifying potential threats in IoT-WSNs with minimal computational overhead.

IV. RESULTS

This section introduces the performance evaluation of GOA-WO-ML in terms of intrusion detection when used in the NSL-KDD dataset, which has 149,000 records belonging to two classes detailed in Table II. The GOA-WO-ML model was executed by using the Scikit-Learn program on a computer with primary hyperparameters specified as follows: learning rate of 0.01, batch size of 32, dropout rate of 0.2, Tanh activation function, and number of epochs of 60.

TABLE II. DISTRIBUTION OF SAMPLES IN THE NSL-KDD DATASET

Category	Type of attack	Sample distribution
Attack traffic	Remote to Local (R2L)	Training: 1,240
		Test: 3,023
	Probing Attack (PA)	Training: 2,636
		Test: 5,883
	User to Root (U2R)	Training: 53
		Test: 227
	Denial of Service (DoS)	Training: 45,967
		Test: 11,963
Normal traffic	Normal	Training: 67,344
		Test: 10,664
Total samples		149,000
		Training: 117,240
		Test: 31,760

During evaluation of intrusion detection systems such as GOA-WO-ML, the model's performance should be measured in terms of several metrics to validate its performance across various detection dimensions. The metrics used are Area Under Curve (AUC), F1-score, specificity, sensitivity, and accuracy, each of which can offer distinct insights into model performance.

Accuracy is defined as the number of correct predictions (both true positives and true negatives) divided by the total number of predictions. It is a broad criterion for the performance of the model, indicating the frequency with which the model classifies the samples appropriately.

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN} \tag{19}$$

Where TP stands for true positives (correctly identified attacks), TN refers to true negatives (correctly identified normal samples), FP denotes false positives (normal samples incorrectly identified as attacks), and FN outlines false negatives (attacks incorrectly identified as normal).

Sensitivity, also known as recall or true positive rate, calculates the model's performance in identifying the attack samples (the true positives) against all the actual positives. It is a crucial metric in intrusion detection since a highly sensitive model guarantees a substantial proportion of the attacks are detected, avoiding attacks that might go unnoticed.

$$Sensitivity=\frac{TP}{TP+FN} \tag{20}$$

Specificity is the complement of sensitivity and measures how well the model can accurately label normal traffic (true negatives) from all real negative samples. This metric is beneficial in cases where false positives are costly, as it reduces the number of standard samples flagged incorrectly as attacks.

$$Specificity=\frac{TN}{TN+FP} \tag{21}$$

F1-score is the harmonic mean of recall and precision. It balances the trade-off between recall and precision, giving a single value for a model's performance when it considers false positives and false negatives. The F1-score is particularly useful when working with imbalanced sets, as it considers both recall and precision, not giving preference to either.

$$F1-score=2\times\frac{Precision\times Sensitivity}{Precision+Sensitivity} \tag{22}$$

$$Precision=\frac{TP}{TP+FP} \tag{23}$$

$$Sensitivity=\frac{TP}{TP+FN} \tag{24}$$

AUC refers to the area under the receiver operating characteristic curve. It is a metric of the overall performance of a binary classifier, indicating how well the model separates the classes. The AUC value is between 0 and 1, with a higher value near 1 representing good model performance. The higher the AUC, the better the model can separate normal from attacking traffic.

$$AUC=\int_0^1 Sensitivity(1-Specificity)dx \tag{25}$$

The metrics for evaluation of the GOA-WO-ML are given in Table III, with 80% of the data used for training, and Table IV shows metrics with 20% testing. The metrics show GOA-WO-ML's success at detecting threats in WSN-IoT environments. Fig. 3 demonstrates the performance of GOA-WO-ML trained with 80% of the dataset against the rest, i.e., 20%.

TABLE III. PERFORMANCE EVALUATION OF THE GOA-WO-ML METHOD WITH 80% TRAINING DATA

Attack types	AUC	F1-score	Specificity	Sensitivity	Accuracy
PA	98.35	95.82	99.63	97.76	99.33
U2R	97.72	94.34	99.77	96.82	99.21
R2L	98.98	96.61	99.56	98.88	99.52
DoS	99.47	98.24	99.83	99.05	99.45
Normal	99.76	99.49	99.86	99.73	99.67
Average	98.85	96.9	99.73	98.44	99.43

TABLE IV. PERFORMANCE EVALUATION OF THE GOA-WO-ML METHOD WITH 20% TESTING DATA

Attack types	AUC	F1-score	Specificity	Sensitivity	Accuracy
PA	98.17	94.69	99.73	97.12	99.24
U2R	97.65	93.61	99.58	98.26	99.61
R2L	98.39	92.49	99.72	97.33	99.26
DoS	99.42	96.07	99.35	98.41	99.47
Normal	99.48	99.93	99.95	99.87	99.34
Average	98.62	95.35	99.66	98.19	99.38

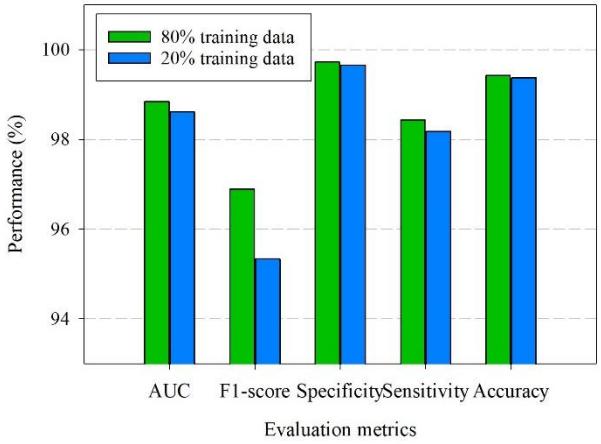

Fig. 3. Comparison of evaluation metrics for 80% training and 20% testing data sets.

Table V presents the performance metrics from 90% of the data used in training, while Table VI contains the metrics from the data set used in testing. The results further prove the reliability of the GOA-WO-ML model in real-life intrusion detection. The results of the GOA-WO-ML model trained with a 90% dataset are presented in Fig. 4 when tested with a dataset of 10%.

Fig. 5 depicts the GOA-WO-ML model's training and validation accuracy. Both values rise with time, implying that the model improves over time. The rising training accuracy suggests that the model is learning well from training data, while the increasing validation accuracy indicates its superior capability of generalizing well to unseen data, also showing the model's strength.

TABLE V. PERFORMANCE EVALUATION OF THE GOA-WO-ML METHOD WITH 90% TRAINING DATA

Attack types	AUC	F1-score	Specificity	Sensitivity	Accuracy
PA	98.93	96.22	99.97	98.41	99.71
U2R	98.52	94.83	99.88	97.65	99.88
R2L	99.27	96.51	99.79	98.98	99.53
DoS	99.97	98.73	99.98	99.18	99.59
Normal	99.91	99.87	99.89	99.72	99.75
Average	99.32	97.23	99.9	98.78	99.69

TABLE VI. PERFORMANCE EVALUATION OF THE GOA-WO-ML METHOD WITH 10% TESTING DATA

Attack types	AUC	F1-score	Specificity	Sensitivity	Accuracy
PA	98.89	95.62	99.79	97.33	99.42
U2R	98.73	94.66	99.74	98.61	99.35
R2L	98.86	95.89	99.79	97.87	99.71
DoS	99.93	96.83	99.56	98.52	99.63
Normal	99.75	99.95	99.91	99.89	99.58
Average	99.23	96.59	99.75	98.44	99.53

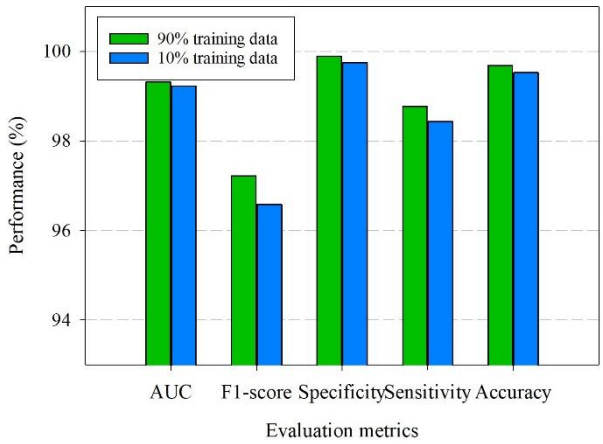

Fig. 4. Comparison of evaluation metrics for 90% training and 10% testing data sets.

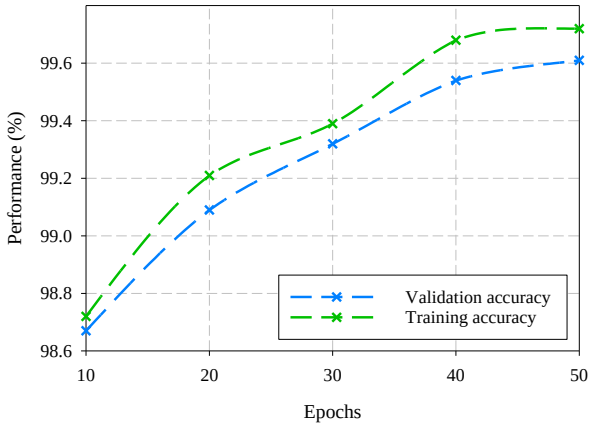

Fig. 5. Training and validation accuracy of the GOA-WO-ML model over epochs.

Fig. 6 measures the GOA-WO-ML model's training loss and validation loss. The declining trends in both parameters show the model's performance to minimize training loss and validation loss, ensuring effective learning and good generalization, essential in real-time intrusion detection. In Table VII and Fig. 7, the performance of GOA-WO-ML in terms of classification is compared with several other models. The performance results from the experiments reveal that GOA-WO-ML performs better in accuracy than the rest of the algorithms.

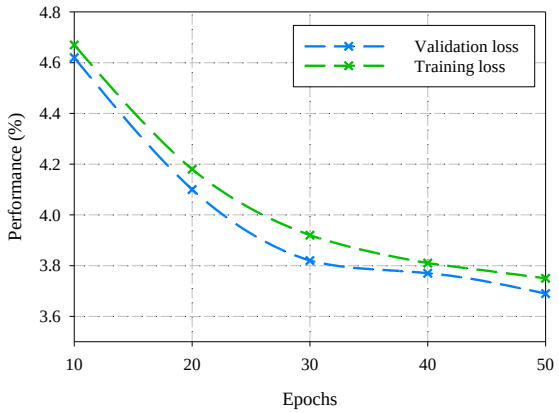

Fig. 6. Training and validation loss of the GOA-WO-ML model over epochs.

TABLE VII. PERFORMANCE EVALUATION OF GOA-WO-ML OVER OTHER METHODS WITH 80% TRAINING AND 10% TESTING DATA

Methods	AUC	F1-score	Specificity	Sensitivity	Accuracy
KNN-PSO	96.51	95.43	98.38	95.37	96.43
XGBoost	95.68	94.56	97.87	94.88	95.35
ALO	93.27	92.41	93.25	92.89	93.44
Random forest	94.97	93.79	95.66	93.44	94.95
LightGBM	94.67	93.47	99.21	93.07	94.65
GOA-WO-ML	98.85	96.9	99.73	98.44	99.43

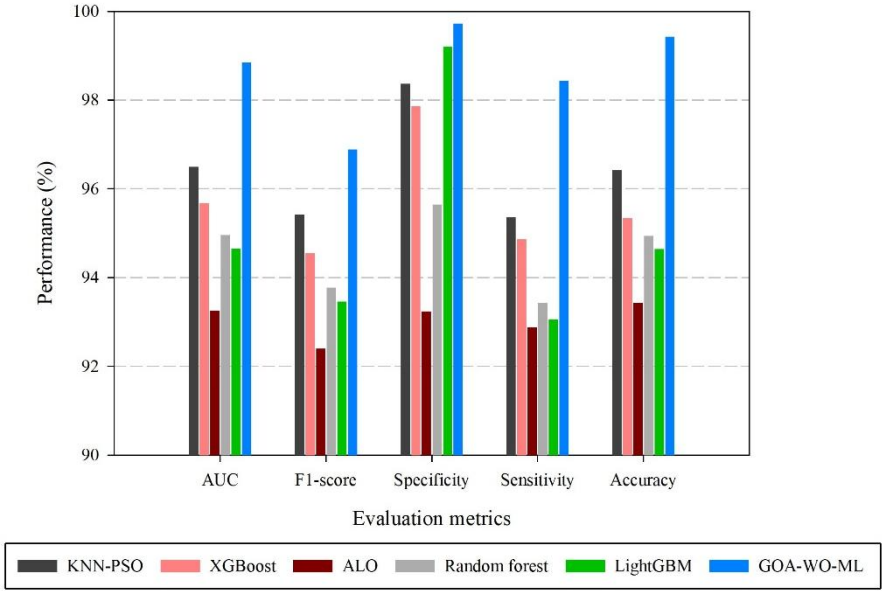

Fig. 7. Performance evaluation of GOA-WO-ML over other methods.

TABLE VIII. COMPUTATION TIME OF GOA-WO-ML OVER OTHER METHODS

Methods	Time (Sec)
KNN-PSO	13.82
XGBoost	10.31
ALO	14.62
Random forest	12.34
LightGBM	15.41
GOA-WO-ML	7.25

Finally, Table VIII and Fig. 8 compare GOA-WO-ML's computational time with other algorithms. Experimental data shows GOA-WO-ML has a computation time of 7.25 seconds. This is evidence of the GOA-WO-ML technique's efficiency, qualifying it for real-time recognition of intrusions in WSN-IoT systems. The results confirm that combining GOA for feature selection with WO for parameter adjustment can significantly enhance the accuracy and efficiency of the intrusion detection system. GOA-WO-ML is a highly reliable and robust framework for detecting intrusions in the WSN-IoT system.

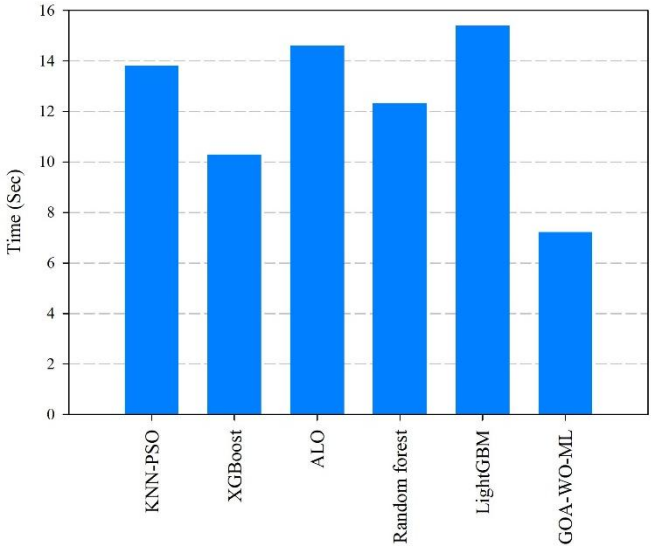

Fig. 8. Computation time of GOA-WO-ML over other methods.

V. DISCUSSION

The experimental outcomes of the GOA-WO-ML model underscore its effectiveness in addressing key challenges associated with intrusion detection in IoT-enabled Wireless Sensor Networks. The observed performance, across various training/testing splits, confirms the robustness of the hybrid feature selection and parameter optimization strategy. Notably, the model demonstrates high classification accuracy, low false positive rates, and minimized computational time, which are critical in constrained IoT environments.

The integration of the GOA and WO proved particularly beneficial in handling high-dimensional network traffic data. GOA's ability to efficiently explore the feature space reduced the dimensionality of input data, while WO effectively tuned the SVM classifier's hyperparameters to maximize detection performance. This hybrid approach not only enhanced accuracy but also contributed to a significant reduction in training loss and validation loss.

Compared to prior studies, GOA-WO-ML achieved competitive or superior results with less computational overhead. For instance, while ensemble models like OSEN-IoT or deep learning frameworks such as IoTFECNN demonstrated high accuracy, they often incurred high computational costs or suffered from overfitting on small datasets. In contrast, GOA-WO-ML balances detection performance and computational efficiency, making it more suitable for real-time deployment in lightweight IoT devices.

However, it is important to acknowledge the limitation related to the dataset used in this study. The NSL-KDD dataset, although widely used and improved over the original KDD'99, was collected over a decade ago and lacks representation of modern, sophisticated attack vectors. Its continued use is justified in part by its structured format, benchmark status, and extensive prior utilization that facilitates comparative evaluation. Nevertheless, the evolving nature of cyber threats, especially in IoT contexts, necessitates validation on more recent and complex datasets to ensure broader applicability.

To enhance the relevance and applicability of future research, we recommend extending the evaluation of GOA-WO-ML using contemporary datasets such as CSE-CIC-IDS2018, CIC-DDoS2019, and Edge-IIoTset. These datasets incorporate diverse and modern intrusion types, emulate realistic IoT scenarios, and reflect the dynamic behaviors of current network environments. Incorporating them would enable further validation of GOA-WO-ML's scalability, adaptability, and resilience against emerging threats.

VI. CONCLUSION

In this paper, we introduced the GOA-WO-ML method to detect intrusions in WSN-IoT systems, integrating feature selection through GOA with WO for parameter tuning in an ML environment. The combination of the optimization methods with a ML classifier, in this case SVM, improved the detection efficacy while reducing the cost of computations. The GOA-WO-ML performance was explored in the NSL-KDD dataset, with extensive experiments across varied training and testing splits. The experiments confirmed the better accuracy, sensitivity, specificity, F1-score, and AUC performance of the model when it outperformed traditional approaches in terms of accuracy, sensitivity, specificity, F1-score, and AUC, thus proving effectual in identifying intrusions in IoT-enabled wireless sensor network.

Along with performance, GOA-WO-ML was highly computationally efficient, running the task in appreciably lower computational time than alternative procedures at a processing time of 7.25 seconds. This renders GOA-WO-ML a high-performance solution and a scalable, efficient, real-time intrusion detection procedure. Future research can involve expanding the introduced method for more sophisticated attack cases, performing performance testing with real-time IoT data, and studying the integration of deep learning methods to further improve detection accuracy. Real-time deployment and online learning can also be investigated to adapt to changing network environments and new attack trends. GOA-WO-ML has a solid basis for accurate vulnerability identification in WSN-IoT architectures, qualifying it as a plausible solution for ensuring the security of IoT-based systems and applications.

FUNDING

This work was supported by Chongqing Municipal Education Commission Science and Technology Research Program (Youth Project): "Research on the Spiking Neural P Systems for Arithmetic Operations with Signed Bits" (No. KJQN202402401).

REFERENCES

[1] H. Zhang and M. Li, "Towards an intelligent and automatic irrigation system based on internet of things with authentication feature in VANET," Journal of Information Security and Applications, vol. 88, p. 103927, 2025.

[2] K. Sharma and S. K. Shivandu, "Integrating artificial intelligence and Internet of Things (IoT) for enhanced crop monitoring and management in precision agriculture," Sensors International, p. 100292, 2024.

[3] E. Rivandi and R. Jamili Oskouie, "A Novel Approach for Developing Intrusion Detection Systems in Mobile Social Networks," Available at SSRN 5174811, 2024, doi: https://dx.doi.org/10.2139/ssrn.5174811.

[4] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9326-9337, 2019.

[5] R. Saadouni, C. Gherbi, Z. Aliouat, Y. Harbi, and A. Khacha, "Intrusion detection systems for IoT based on bio-inspired and machine learning techniques: a systematic review of the literature," Cluster Computing, vol. 27, no. 7, pp. 8655-8681, 2024.

[6] Q. A. Al - Haija and A. Droos, "A comprehensive survey on deep learning - based intrusion detection systems in Internet of Things (IoT)," Expert Systems, vol. 42, no. 2, p. e13726, 2025.

[7] J.-S. Pan, L.-G. Zhang, R.-B. Wang, V. Snášel, and S.-C. Chu, "Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems," Mathematics and Computers in Simulation, vol. 202, pp. 343-373, 2022.

[8] M. Han, Z. Du, K. F. Yuen, H. Zhu, Y. Li, and Q. Yuan, "Walrus optimizer: A novel nature-inspired metaheuristic algorithm," Expert Systems with Applications, vol. 239, p. 122413, 2024.

[9] M. B. Bagherabad, E. Rivandi, and M. J. Mehr, "Machine Learning for Analyzing Effects of Various Factors on Business Economic," Authorea Preprints, 2025, doi: https://doi.org/10.36227/techrxiv.174429010.09842200/v1.

[10] R. Zhao et al., "A novel intrusion detection method based on lightweight neural network for internet of things," IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9960-9972, 2021.

[11] R. Gangula and M. M. V, "Network intrusion detection system for Internet of Things based on enhanced flower pollination algorithm and ensemble

classifier," Concurrency and Computation: Practice and Experience, vol. 34, no. 21, p. e7103, 2022.

[12] H. Asgharzadeh, A. Ghaffari, M. Masdari, and F. S. Gharehchopogh, "Anomaly-based intrusion detection system in the Internet of Things using a convolutional neural network and multi-objective enhanced Capuchin Search Algorithm," Journal of Parallel and Distributed Computing, vol. 175, pp. 1-21, 2023.

[13] A. Hosseinzadeh, M. Shahin, M. Maghanaki, H. Mehrzadi, and F. F. Chen, "Minimizing wastevia novel fuzzy hybrid stacked ensembleof vision transformers and CNNs to detect defects in metal surfaces," The International Journal of Advanced Manufacturing Technology, pp. 1-26, 2024, doi: 10.1007/s00170-024-14741-y.

[14] A. V. Hanafi, A. Ghaffari, H. Rezaei, A. Valipour, and B. Arasteh, "Intrusion detection in Internet of things using improved binary golden jackal optimization algorithm and LSTM," Cluster Computing, vol. 27, no. 3, pp. 2673-2690, 2024.

[15] T. Yang, J. Chen, H. Deng, and B. He, "A lightweight intrusion detection algorithm for IoT based on data purification and a separable convolution improved CNN," Knowledge-Based Systems, vol. 304, p. 112473, 2024.

[16] S. N. Makhadmeh, S. Fraihat, M. Awad, Y. Sanjalawe, M. A. Al-Betar, and M. A. Awadallah, "A crossover-integrated Marine Predator Algorithm for feature selection in intrusion detection systems within IoT environments," Internet of Things, p. 101536, 2025.

[17] L. Shi, Q. Yang, L. Gao, and H. Ge, "An ensemble system for machine learning IoT intrusion detection based on enhanced artificial hummingbird algorithm," The Journal of Supercomputing, vol. 81, no. 1, p. 110, 2025.

[18] S. Asif, "OSEN-IoT: An optimized stack ensemble network with genetic algorithm for robust intrusion detection in heterogeneous IoT networks," Expert Systems with Applications, p. 127183, 2025.

[19] M. Ahmadi et al., "Optimal allocation of EVs parking lots and DG in micro grid using two - stage GA - PSO," The Journal of Engineering, vol. 2023, no. 2, p. e12237, 2023.