SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Improving Botnet Attack Identification in Iot Systems Through Gwo and Ofo Methods: An Innovative Strategy

[1]Pradeep Kumar Mishra, [2]Asheesh Tiwari

[1]Department of Computer Engineering and Applications, GLA University, Mathura, Uttar Pradesh, India.
E-mail: pradeep.rid@gmail.com
[2]Department of Computer Engineering and Applications, GLA University, Mathura, Uttar Pradesh, India.
E-mail: asheesh.tiwari@gla.ac.in

ABSTRACT
To protect computer networks from malicious threats it is necessary to have intelligent and adaptive Intrusion Detection Systems. To address this challenge, this article presents a new enhancement to the traditional Grey wolf optimization (GWO) algorithm for improving defensive capability.We propose Optical Flow Optimization (OFO) to intelligently initialize the population, which lays a solid foundation for global search and achieves faster and more effective threat detection. This optimized initialization is integrated with a Support Vector Machine (SVM) classifier in a supervised machine learning framework to form a powerful GWO-OFO-SVM hybrid. The proposed approach significantly improves the convergence speed, accuracy of the solution and resilience against local optima which are the main drawbacks of traditional GWO. Extensive performance validation evaluations are carried out on Mirai botnet attacks and compared with other state-of-the-art metaheuristic algorithms such as Salp Swarm-Ant Lion Hybrid Optimization (SSA-ALO), GWO and One-Class SVM (GWO-OCSVM), and Bi-directional LSTM with Recurrent Neural Networks for Botnet detection (BLSTM-RNN).Experimental results on the N-BaIoT dataset show the superiority of GWO-OFO-SVM with outstanding metrics, 97.30% accuracy, 97.09% recall, 97.08% precision, and F1-score of 97.45%. The results show the model's ability to build reliable, high-performing intrusion detection systems in practical network settings.

Keywords: Intrusion Detection, Grey Wolf Optimizer (GWO), Optical Flow Optimization (OFO), Support Vector Machine (SVM), N-BaIoT Dataset, Metaheuristic Algorithms.

1. Introduction

The growing interconnection of the global internet infrastructure means that security professionals discover new vulnerabilities each months, consequently hackers may penetrate these systems and carry out destructive tasks [1]. Recognizing many forms of network assaults, particularly unforeseen ones, poses a significant challenge. Cybercriminals can gain access to databases that store sensitive information, such as financial and medical records; therefore, cybersecurity specialists and developers must innovate new Intrusion Detection Systems (IDSs) to thwart these attacks. IDSs examine packets captured from the network to aid in identifying threats. Denial-of-service (DoS) attacks are among the most typical risks as they prevent authorized users denying people to access network resources by generating unwanted traffic. Malware refers to harmful software that exploits vulnerabilities in computer networks to fulfill its objectives. The growth of information and communication technologies (ICTs) has significantly benefited internet users. At the same time, information security has become an escalating concern due to the widespread occurrence of risks of network penetration, including cross-site scripting, probing and DDoS attacks. In the domain of cybersecurity, systems for detecting network intrusions (NIDSs) serve as essential security measures that detect and prevent malicious attacks. A prevalent challenge in classification is network intrusion detection, which continuously monitors network activities and assesses when to trigger an alert to the network [3]. IDS improves the performance of classifiers in recognizing invasive behavior. Using comprehensive descriptions such as rules or signatures, traditional intrusion detection systems (IDSs) monitored traffic and frequently detected false positives and negatives; this led to false alarms. Systems for intrusion detection, or IDSs, may be founded on networks, hosts, anomalies, signatures, or hybrid IDSs, such as HIDSs (hybrid IDSs) or NIDSs (network IDSs) [5]. NIDS and HIDS features are combined in HIDSs, which are extremely dependable security frameworks [6]. AIDS has been employed in a number of works to increase false positive detections and improve accuracy [7–11].

Numerous relevant studies in the literature have developed automated models for detecting network intrusions using machine learning techniques and support vector machines [17], such as random forests, decision trees, etc. Models for machine learning work well enough with modest amounts of data [18-20]. However, the network generates enormous amounts of data in real time, and these machine learning models are unable to learn big amounts of data alone. This work's innovation is that it builds the functional

model with the combination of GWO-OFO-SVM algorithms. Traditional IDS [12] generally relies on signature based techniques which are inadequate for unknown and zero-day attacks. Machine learning and optimization techniques are extensively used to improve the detection ability of intrusion detection systems. Recent advances in metaheuristic algorithms have shown promising results in optimizing intrusion detection models [13]. Two such algorithms are the Grey Wolf Optimizer (GWO) [4] and the Optical Flow Optimization (OFO) [5] which have shown excellent performance in solving complex optimization problems. A popular machine-learning method for intrusion detection is the Support Vector Machine (SVM), which can handle high-dimensional data. [2]. Although [3] described, optimizing SVM parameters is still challenging.

This article proposes a novel hybrid approach, GWO-OFO-SVM, which combines GWO and OFO with SVM to improve intrusion detection accuracy. The way gray wolves hunt served as the model for the GWO meta-heuristic algorithm [9, 15]. In the iterative GWO process, the followers (ω) are led by the leader wolves (α, β, and δ), who are always focused on approaching the prey. The position of the prey or the present an all-encompassing answer to global exploration have an impact on the leader during the prey search [16]. The followers, on the other hand, do thorough local exploration. By doing this, the likelihood of slipping into local optimum solutions is significantly decreased and mining and exploration capacities are balanced. A growing number of academics have focused on the GWO algorithm's improvement and application fields due to its straightforward structure, low number of control settings, as well as excellent convergence precision [21].

2. RESEARCH MOTIVATION AND CONTRIBUTIONS

The main motivation and contribution of this study is to develop an effective and efficient intrusion detection system that utilizes a hybrid metaheuristic approach. The specific contributions are:

- To assess how well the suggested GWO-OFO-SVM model performs using the Mirai botnet attack using the N-BaIoT dataset and discussed other competitive algorithms.
- For intrusion detection, this work suggests a unique mix of GWO, OFO, and SVM.
- The suggested method builds an efficient IDS by utilizing the advantages of each strategy and increases the accuracy of feature extraction by optimizing the OFO parameters using GWO.
- The proposed model increases the accuracy by 1-4% when compared with XGBoost and SVM alone.
- Furthermore, outperforms various metaheuristic methods such as Salp Swarm and Ant Lion Hybrid Optimization (SSA-ALO) model, Grey Wolf Optimization and One-Class Support Vector Machine (GWO-OCSVM), and Botnet Long Short Term Memory and Recurrent Neural Network (BLSTM-RNN).

2.1 Organization of the paper

The remaining portion of this article is organized as follows: Section 3 explains the relevant literature reviews; Section 4 provides the fundamentals and technical background; Section 5 discusses the methodology and proposed model GWO-OFO-SVM in detail; Section 6 displays the tests and findings; and Section 7 presents the paper's conclusion.

3. Literature Review

Because of the straightforward layout of the GWO algorithm, control parameters and a high degree of precision in convergence, an increasing number of scholars are concentrating on its improvement and application areas [17]. Few researchers have integrated GWO with other meta-heuristics to attain a fair balance between exploration and exploitation. In a recent work article [18] resolve the 3-constraints engineering problem and PV model parameter extraction, the collaboration-based hybrid GWO-SCA optimization approach cHGWOSCA was developed. The outcomes of the trial indicate that this tactic works effectively. The article [23] proposed a hybrid WOAGWO algorithm that may be utilized to tackle engineering issues like pressure vessel design by fusing the advantages of the GWO algorithm with the whale optimization algorithm (WOA) [19]. The article [20] tackled the optimal reactive power scheduling for the entire grid issues using a hybrid GWO-PSO technique. The outcomes of the simulation verify that the hybrid GWO-PSO approach leads to a noticeable enhancement in grid performance on a broad scale. In order to estimate the parameters of FM acoustic waves and solve the production flow diagram and the issues with leather layering, the article [21] suggested three distinct hybrid algorithms for combining PSO and GWO. The correctness of these procedures was confirmed by the matching findings.

Many academics are increasingly focusing on using GWO to tackle various real-world problems due to advancements in intelligent optimization methods and machine learning. The research [28] proposed MG-GWO, a GWO-based approximation technique for PV solar cells, to take out the features of A single-diode model of a photovoltaic cell. According to the simulation findings, the MG-GWO algorithm performs better than the conventional GWO method in terms of convergence speed and robustness [29]. An emotion identification system based on a single-channel ECG utilizing a GWO support vector machine (X-GWO-SVMs) was developed in a related work [30] that is both dependable and effective. The X-GWO-SVM technique is significantly more efficient than the current deep neural network-based solutions, as shown by the comparison trials [22]. The structural issue of coupled beams is outperformed by the combination

of extreme machine learning (ELM) and GWO. The GWO-ELM hybrid model has been found to perform better than the individual ELM and GWO models [31]. The author has recently developed a grey wolf algorithm for estimating distribution in relation to the low-carbon site selection path problem [26], particularly within the framework of carbon pricing.

Combining probabilistic model learning with the gray wolf method has been shown to improve optimization in the site selection route problem, the emerging model may successfully reduce overall expenses and carbon emissions once carbon pricing is implemented [32]. Reverse mixed-frequency data sampling (R-MIDAS) was employed in the study [33] in conjunction with the GWO algorithm to develop a novel model for estimating the returns of 27 industry stock indexes and investment options. In conclusion, the GWO method can balance exploration and mining, increase local search optimization accuracy [33, 34], prevent the creation of nearby extreme points in little areas [35], and speed up global search convergence through linear convergence. However, GWO is not very effective for complex multibarrier problems, particularly multiple-constraint engineering optimization challenges or problems with more local extreme value regions.

It is necessary to make improvements and adjustments based on various application backgrounds and problem types to adopt a harmony between local and global search, as well as between global search speed and reversibility, and fineness and perturbation in local search [23]. In order to develop a technique that can be used in every situation, an increasing number of academics have concentrated on developing new algorithms and improvement techniques [24]. The Internet of Things (IoT) ecosystem [26], which is made up of linked devices with low power consumption, inexpensive CPUs, and memory sources that can only carry out a few specific tasks, was covered in this course of action article [34]. In order to replicate the Mirai assault on Internet of Things devices [27], they looked at the correlation between samples of varying sizes from the Kitsune dataset [25]. The accuracy of the Mirai attack's training and retraining phases was also assessed. To reduce training time on low-resource devices, several strategies are tested in lower sample numbers. In order to prevent overfitting categorization techniques throughout the learning phase, cross-validation was employed. In a related paper [36], N-BaIoT, a network-based strategy for the Internet of Things, detects anomalies using deep learning methods. In particular, a deep autoencoder (one for each device) is trained to comprehend the IoT device's normal activities following statistical features that collect behavioral snapshots of benign IoT data is obtained [28]. Autoencoders tried to compress the snapshots, they explained. It is a clear sign that the observed behavior is unusual when an autoencoder is unable to reconstruct a snapshot (the IoT device has been hacked and is showing an unknown behavior). Deep autoencoders have the benefit of being able to recognize intricate patterns, such as those of different device functions. As a result, there are very few false alerts from the anomaly detector.

Only a small number of researchers demonstrated greater accuracy using GWO. Additionally, the results indicate that the LSTM-RNN [29] can effectively learn all attack types present in the training data when utilizing the GWO classifier. This classifier was able to accurately identify a significant percentage of assaults in the test data. Consequently, compared to the results obtained from using the entire feature set, the combination of LSTM-RNN with GWO can reduce the rate of misclassification. In this study, we computed various performance metrics, including the confusion matrix, recall, F-score, precision, training and testing accuracy, and true positive rate (TPR and FPR). Additionally, we compared the performance of SVM and random forests with that of LSTM-RNN enhanced by the GWO algorithm. The findings highlight the effectiveness of LSTM-RNN in relation to both SVM and random forests, as well as its performance when optimised with GWO. Overall, the results demonstrate the efficacy of LSTM-RNN in both binary and multiclass classification tasks.

4. BASICS AND TECHNICAL BACKGROUND

4.1. Dataset

Features taken from network traffic data are included in the N-BaIoT dataset, whose IoT networks use to detect intrusions [40]. This dataset can be used to identify IoT botnet attacks because it contains information from nine commercial IoT gadgets that have been infected with the malware Mirai and BASHLITE. Features including packet size, protocol, timestamp, and other network traffic characteristics frequently utilized in intrusion detection are probably included in the dataset. The potential of the N-BaIoT dataset for machine learning-based intrusion detection has been demonstrated by research that has used deep learning techniques to detect distributed denial of service (DDoS) and denial of service (DoS) assaults. The N-BaIoT dataset is a useful tool for researchers and developers working on IoT security, particularly in the area of network-based detection of botnet attacks, as its focus is on this type of attack. As discussed in Fig. 1A and Fig. 1B.

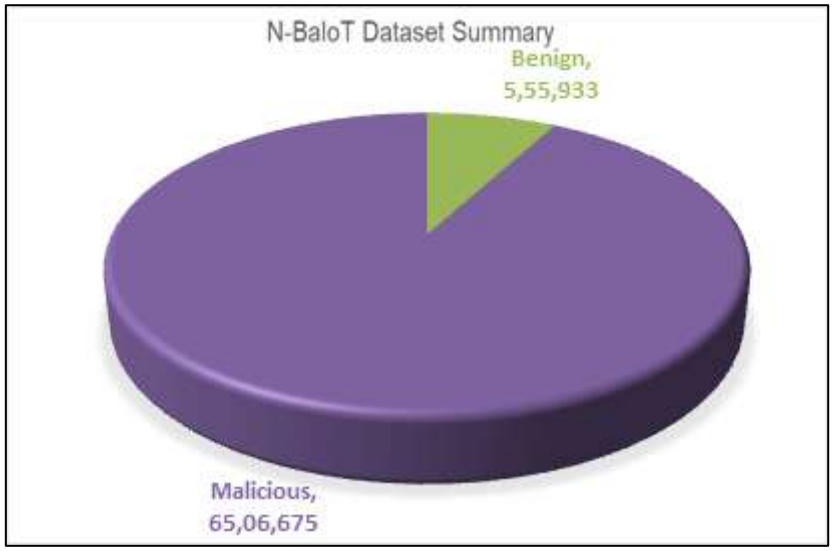


Figure 1A. N-BaIoT Dataset Summary

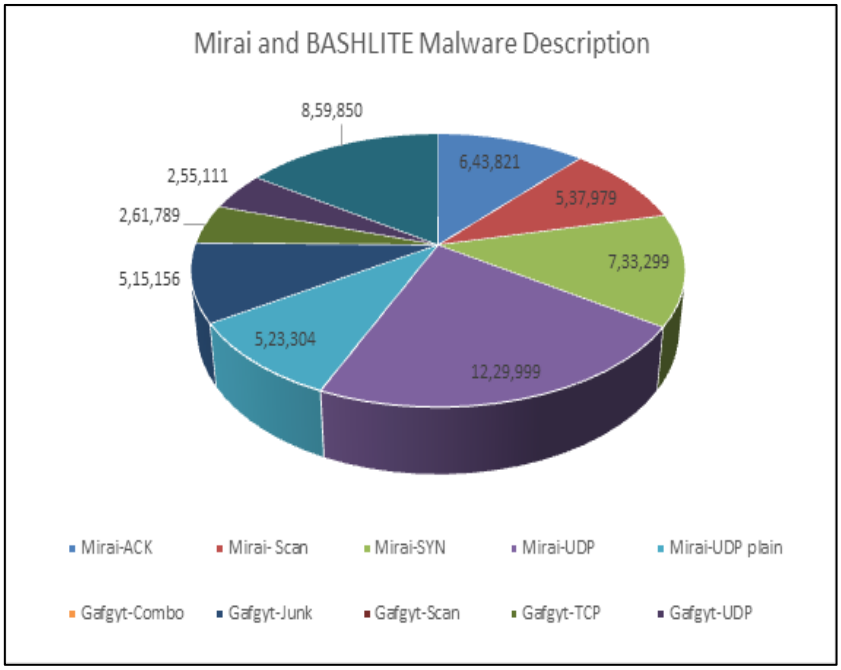


Figure 1B. Description of Mirai and BASHLITE malware

Using default credentials, the Mirai assault is a widespread assault on smart gadgets that causes denial of service. To identify cyberattacks using network traffic, it is required to distinguish between attack and routine network packets. We trained samples of various sizes using our model, GWO-OFO-SVM. Mirai is a kind of malware that uses IoT devices, including IP cameras or smart home appliances, to target the vulnerable Telnet service and collect botnets for DDoS assaults [37]. It has infected billions of Internets of Things devices that were produced without taking fundamental security precautions into account. The main goal of Mirai was to confiscate manufactured devices that had default passwords and usernames. DDoS assaults are then launched against compromised devices. The whole system described in Fig 2.

An extraordinary DDoS assault on the "Kreb on Security" blog happened just one month after the Mirai botnet was initially identified by a malware "white hat" research organization in August 2016 [3]. Then, in October 2016, Dyn, the DNS provider for major corporations like Netflix and Twitter, was the target of a massive DDoS assault launched by Mirai. Additionally, Liberia's whole network was taken offline using Mirai. [37]. The Mirai attack quickly spread due to vulnerabilities in IoT device authorization; attackers began creating other variants and sharing the Mirai source code. Several of these included versions have been dubbed botnets, including Satori [38], Okiru [39], and OMG [39]. The ARC processors, the second most common CPU core, are the target of the first botnet, Ikiru [42]. A significant quantity of bitcoin is occupied by the Satori botnet, which searches open Ethereum mining sites [43]. Lastly, the OMG botnet protects attackers' identities by converting compromised IoT devices into proxy servers [40].

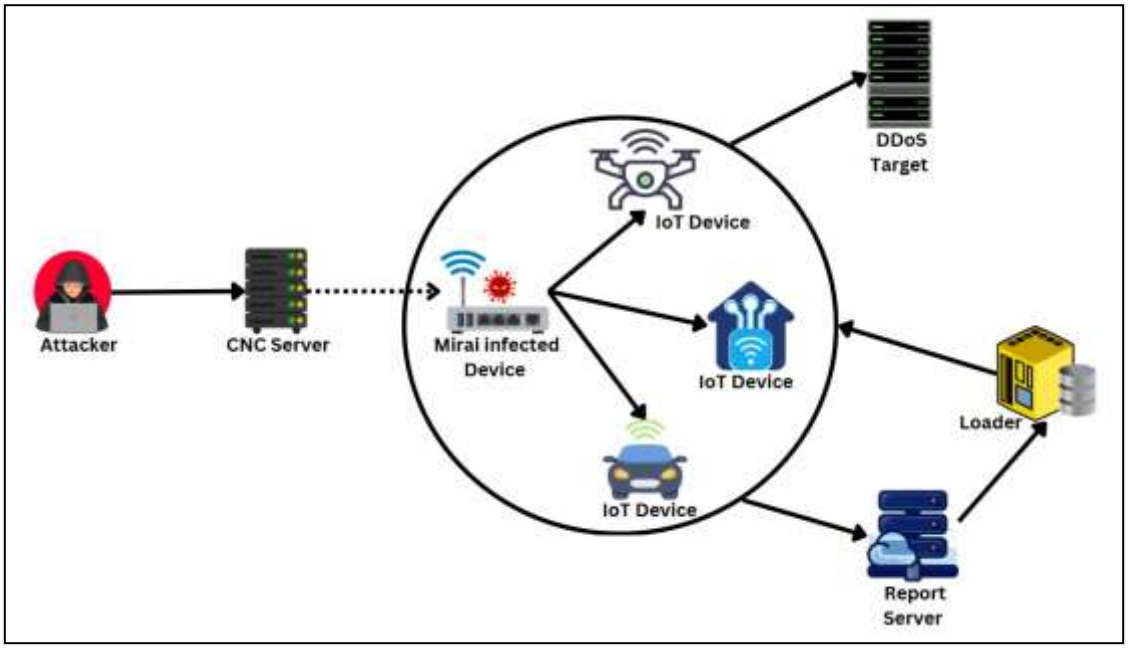


Figure 2. Traditional Architecture of Intrusion Detection System for IoT

4.2. Grey Wolf Optimization

According to a strict social order, gray wolves live in packs. The α-wolf, the pack leader, is in charge of deciding where to hunt, distribute food, and rest [12]. As members of the secondary level, the β-wolves primarily support the α-wolf in making decisions. The δ-wolves are sentry and scouting personnel of the tertiary level. At the bottom of the hierarchy, the ω wolves are mostly in charge of preserving harmony among the group [38]. This illustrates the gray wolf population's hierarchy as well as predation mechanism in Fig. 3. and Fig. 4.

The position of the i-th gray wolf may be described as in equation 1, if the size of the gray wolf population is N and the size of the GWO's solution space used to solve the optimization issue is d.

$W_i = \{W_i^1, W_i^2, W_i^3, \ldots,, W_i^d\}, \quad i = 1, 2, 3, \ldots\ldots N$ (1)

The gray wolf population's ideal, suboptimal, and third optimum solutions are denoted by the symbols α, β, and δ, respectively, while the other options are denoted by ω. ω continuously modifies the position dependent on the locations of α, β, and δ in order to determine the ideal solution or position. Using equation 2, the gray wolves' whereabouts are calculated.

$W(t+1) = Wp(t) - A \cdot |C \cdot Wp(t) - W(t)|$ (2)

$A = 2a \cdot r_1 - a$ (3)

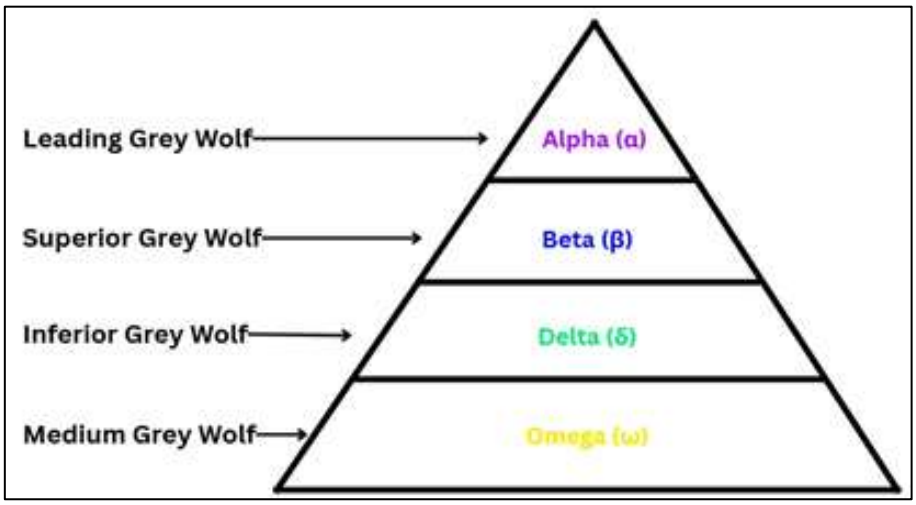


Figure 3: Grey Wolf hierarchy as per their position

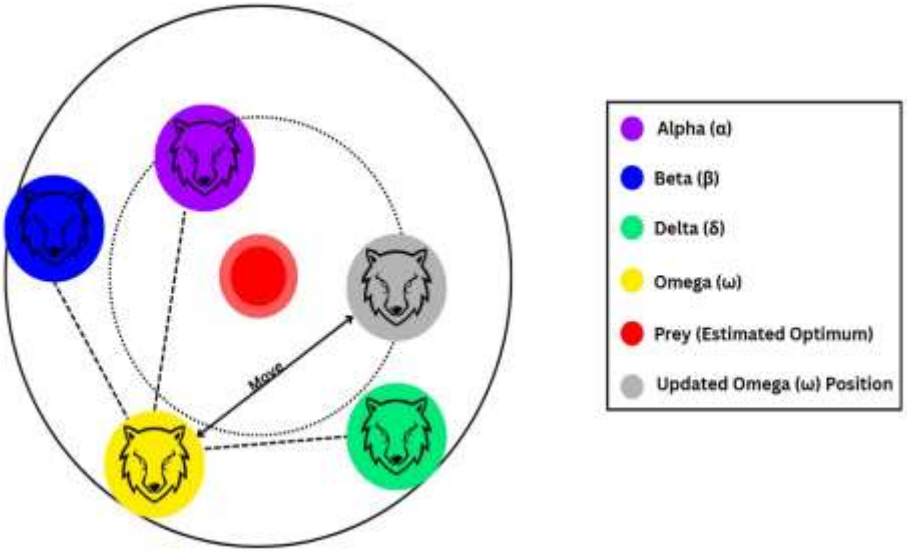


Figure 4: Optimal position and behaviour of Grey Wolf with Prey

$C = 2r_2$ (4)

Where t is the iteration count, Wp (t), the convergence factor is denoted by A, and the swing factor by C. is the prey's position, and W (t) is the gray wolf's position following t iterations. The distance control parameter, a, has a linear drop between two and zero as the number of iterations increases and the best solution approaches, while r_1 and r_2 are taken as random variables between [0, 1]. The following is how this is expressed mathematically:

$a(t) = 2(1-t/T_{max})$ (5)

Where, T_{max} denotes the maximum number of iterations as described in equations 3, 4, 5. The α-wolf leads the β- and δ-wolves in pursuing the prey once it has been surrounded. The prey's escape during this process results in a shift in the gray wolf population's location. As a result, the gray wolf population updates its positions using the locations of the wolves α, β, and δ, or Wα, Wβ, and Wδ, respectively. The following is a mathematical expression for the update process is discussed in equation 6.

$$\left.\begin{aligned} W_1(t+1) &= W_\alpha(t) - A_1 \cdot IC_1 \cdot W_\alpha(t) - W(t)I \\ W_2(t+1) &= W_\beta(t) - A_2 \cdot IC_2 \cdot W_\beta(t) - W(t)I \\ W_3(t+1) &= W_\delta(t) - A_3 \cdot IC_3 \cdot W_\delta(t) - W(t)I \\ W(t+1) &= (W_1 + W_2 + W_3)/3 \end{aligned}\right\} \quad (6)$$

Where W (t + 1) is the gray wolf's most recent updated position. The symbols Wα, Wβ, and Wδ stand for the locations of α, β, and δ. The prey positions indicated by W_1, W_2, and W_3 are calculated from the positions of α, β, and δ, respectively. As Algorithm 1. displays the pseudo-code.

4.3. Support Vector Machine

A repeatable hyperplane between classes is defined via the Support Vector Machine (SVM) approach [34]. The main target of learning techniques is to truthfully categorize each sample based on its attributes. This leads to the classifier not generalizing well and remembering training data instead of learning models, particularly for training data that is overly fitted. By analyzing every sample inside each class in the training set and dividing and giving them a surface that optimizes the space between them, the SVM method increases the capacity for generalization [12]. The SVM decision-making process is trained by maximizing the difference between the two class tags' support vectors. SVM does not necessarily need to be linear, even though it is most frequently preferred that way. The complexity of linear SVM problems varies based on how many features are utilized. With two features, the hyperplane is simply a line; with three features, it becomes a 2-dimensional plane [41]. If we assume the features possess linear separability, we can create a flat hyperplane referred to as a linear classifier, as illustrated in Fig. 5 on the graph of the features that separates the two labels of the corresponding class.

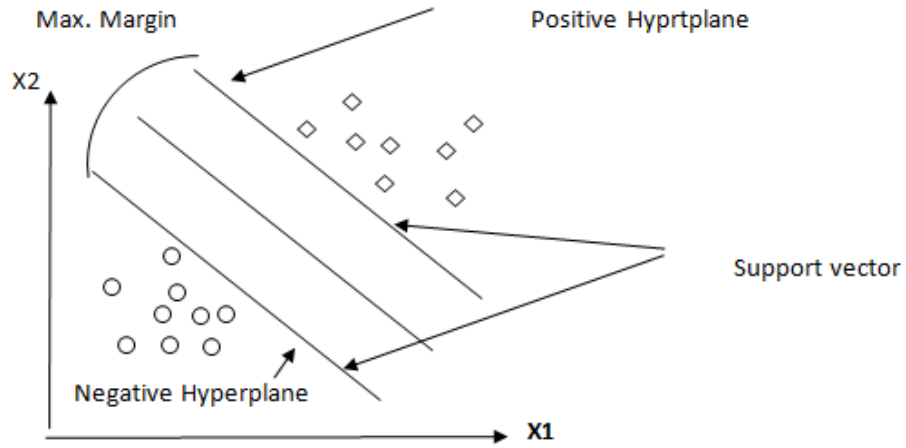


Figure 5. Decision Hyperplane with Support Vectors

5. THE PROPOSED IGWO ALGORITHM

The paper suggests optimizing the initial population using the optical flow. The optimization method aims to lay the foundation for global search. To enhence the effectiveness of the conventional GWO algorithm, Support Vector Machine (SVM) is combined with the Grey Wolf Optimizer (GWO) and Optical Flow Optimization (OFO). The conventional GWO technique can rapidly reach the global optimum [38], but when faced with high-dimensional or complex optimization problems, it is prone to slipping into local minima. This decline in performance is brought on by the original GWO algorithm's limited capacity for exploration. We can improve the algorithm's exploration capability and, consequently, its optimization capability by making the initial population optimal, changing the parameters, and altering the search method. In conclusion, these tactics improve the algorithm's performance and increase its level of competition.

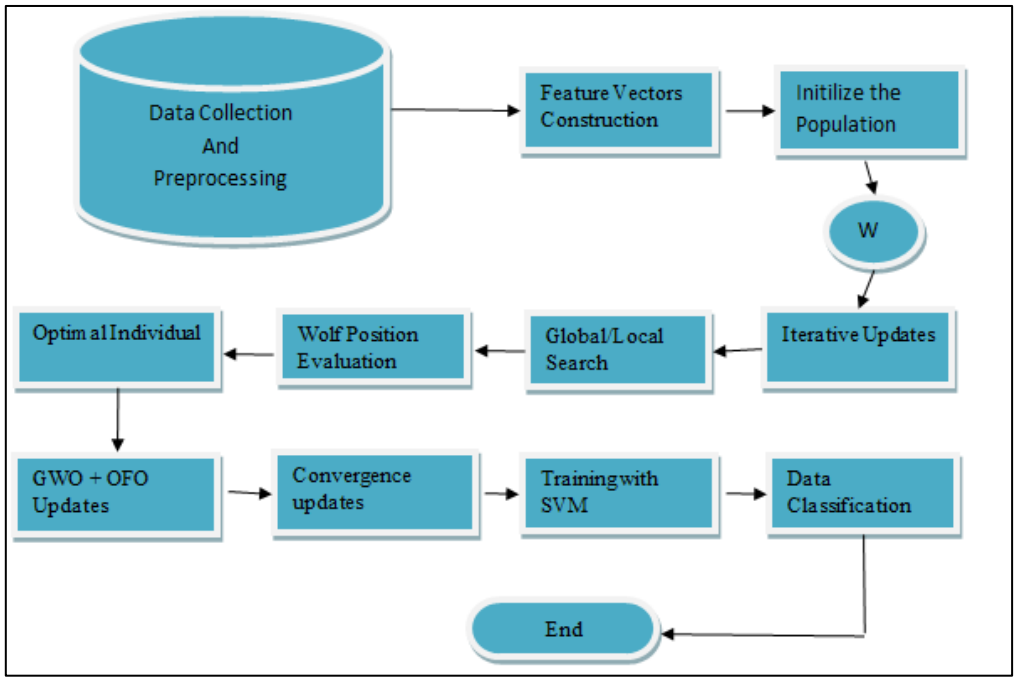


Figure 6. The Flow Chart of Proposed GWO-OFO-SVM Model

5.1 Optical Flow Optimization

Following repetition, the GWO algorithm causes the population's variety to decline later on, and the population of gray wolf concentrates close to the ideal individual's location. The difficulty of escaping when the ideal individual falls into local minima is the root cause of the phenomena of premature algorithm termination and a reduction in the search for the ideal point's accuracy. To address this issue, we provide the optical flow optimization technique, which broadens the search scope and conducts a two-way search inside the search area to identify the optimal answer while taking into account both the real and anticipated solutions using an objective function in equation 7.

The formulation of the optimization problem is:

$$\left.\begin{aligned} &\text{Minimize } J = \sum[L(y_{true}, y_{pred}) + \mu * R(u)] \\ &\text{Subject to: } \nabla u * \nabla L(y_{true}, y_{pred}) = 0 \end{aligned}\right\} \quad (7)$$

Where u is the optical flow field, J is the objective function, L is taking as the loss function, y_true and y_pred are the true labels (normal or anomalous), and is the gradient of the optical flow field. The optical flow field's u_x and u_y are the horizontal and vertical component is the regularization parameter, and R (u) is the regularization term. Regularization is a machine learning approach that adds a penalty term to the loss function in order to prevent overfitting. The regularization term and regularization parameter are key components of this technique.

The regularization term is a mathematical expression that is added to the loss function to penalize large weights or complex models. The model is presented through tipical flow chart in Fig. 6.

This article used the regularization parameter μ, given in equation 8, which controls the strength of the regularization term. A larger value means a stronger penalty for large weights or complex models. To adjust and control the regularization parameter, we used the following strategy:

$$\mu(t+1) = \mu(t) * \left(1 + \cos\left(\pi * \frac{t}{T}\right)\right) \quad (8)$$

Where the control parameter at iteration t, t is the current iteration and T is the total number of iterations, and is the mathematical constant. It has been observed that it avoids local optima and provides a fast convergence rate. It is robust to noisy objective functions and easy to implement. Algorithm 1. discussed the pseudocode.

Algorithm 1: *SVM Hyperparameter Optimization using GWO and OFO*

1: Input: Training data (Xtrain, ytrain), Test data (Xtest, ytest)
2: Log-scale bounds: C ∈[10−2 , 103], γ ∈[10−5 , 101]
3: GWO parameters: epochs, population size
4: OFO parameters: step size δ, max iterations
5: Regularization coefficient μ = 0.01
6: function ObjectiveFunction(x)
7: Let x = [log 10(C), log 10(γ)]
8: Convert to real scale: C = 10x1, γ = 10x2
9: Train SVM with C, γ using (Xtrain, ytrain)
10: Predict on Xtest and compute accuracy A
11: Compute regularization term: R = C + γ
12: Compute objective: J = (1 − A) + μ · R
13: return J
14: end function
15: Step 1: Grey Wolf Optimizer (GWO)
16: Initialize a population of wolves in log scale
17: for each epoch do
18: Update wolves' positions based on leadership hierarchy
19: Evaluate fitness using ObjectiveFunction
20: end for
21: Let x ∗ be the best solution from GWO
22: Step 2: Optical Flow Optimization (OFO)
23: Initialize: xbest ← x ∗, Jbest ← ObjectiveFunction(xbest)
24: for iteration = 1 to max iter do
25: Generate perturbed solution: xnew = xbest + δ · (rand − 0.5)
26: Clip xnew to bounds
27: Jnew ← ObjectiveFunction (xnew)
28: if Jnew < Jbest then
29: xbest ← xnew, Jbest ← Jnew
30: end if
31: end for
32: Step 3: Final SVM Training and Evaluation
33: Convert xbest to real scale: Copt, γopt
34: Train final SVM with Copt, γopt
35: Predict on Xtest and compute final accuracy and classification report
36: Output: Optimized SVM model with Copt, γopt, and evaluation metrics

5.2 Methodology

In the proposed model, the N-BaIoT dataset is utilized, focusing exclusively on Mirai botnet attack traffic. The collected dataset is sent for dataset preprocessing block and processed. The preprocessing handled

missing values, removing inputted records with null entries, addressed infinite values, and replaced or normalized infinite values to ensure numerical stability. The normalized data is separated into testing and training sets. The extracted samples are distributed into classes as in Table 1.

Table 1. Description of labels and samples

Label	Sample
Benign	10000
UDP	10000
SCAN	10000
SYN	10000
ACK	10000
UDP PLAIN	10000

The suggested model is linked to elements that are used to strengthen and expedite the network. Dropout work is used in the model to make the overall interaction effective and skilled. The dropout serves to both avoid overfitting of the training set and indiscriminately increase the power of the model. Standardization is applied to normalize features to a standard scale (mean = 0 and variance = 1). Principal component analysis (PCA) has been used in the field of intrusion detection and is widely recognized as a crucial tool in data reduction and feature extraction. PCA generates a collection of orthonormal principle components and minimizes the number of dimensions needed to categorize fresh data. By focusing on the directions in the feature space where the variance is highest, it lowers the dimensionality of the data. A feature's eigenvalue determines how much of the total variance it accounts for [11]. To decrease the dimensions (D) of the N-BaIoT dataset, we use PCA. To identify the k characteristics needed for classification, we employ the crucial eigenvalue test and the screeplot test [15]. According to the critical eigenvalue test, any principal components whose eigenvalues are greater than the threshold D of 0.9 should be chosen [11], where D is the original dataset's dimension. It is believed that there is noise or redundancy in the remaining d-k features. We choose 15 characteristics for the N-BaIoT dataset, resulting in an 85% reduction in the input data dimension. As explained in figure 7.

Table 2. The initial parameter of the classification models

Algorithm	**Parameter**	**Value**	**Description**
GWO	epoch	50	Number of Optimization Iteration
	Pop_size	20	Number of candidate solutions(wolves) in the pack
	Obj_func	Objective_function	The fitness function function to be minimized(1-accuracy+0.001*regularization)
	bounds	C:[-2,3](log), gamma:[-5,1](log)	The log-scale search space for SVM hyperparameters.
	minmax	"min"	The optimization goal is to minimize the opbjective function.
OFO	best-solution	Gwo.solve(…)	The starting solution for refinement, taken from GWO's output.
User-defind	Step-size	0.1 (default)	Size of the random local search step in each refinement iteration.
User-defind	Max-iter	10 (default)	Number of refinement iterationsfor OFO

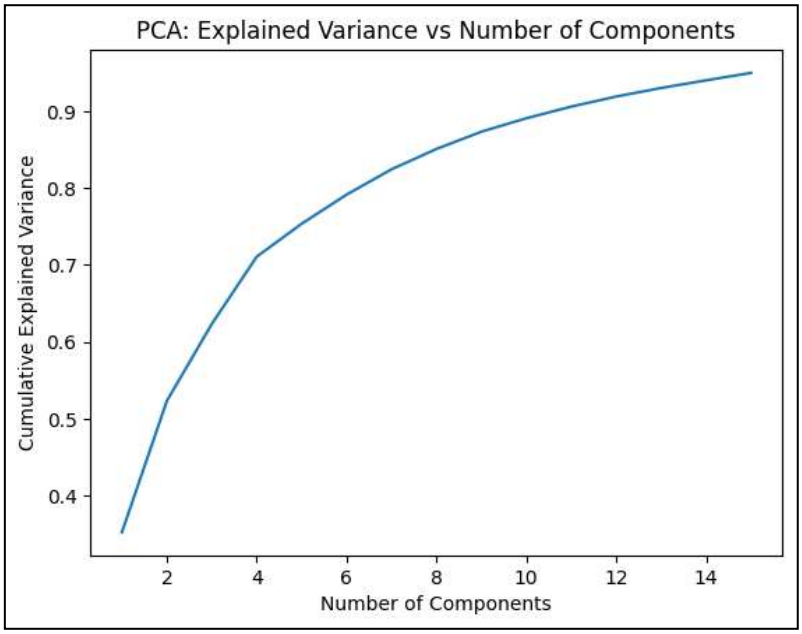


Figure 7. Variance with number of components

6 Experimental Results and Discussion

The tests carried out to assess the efficacy of the suggested GWO-OFO-SVM model in contrast to the current intrusion detection systems (IDSs) are described in depth in this section. A 64-bit desktop Windows operating system was used for the investigations. The machine has 16 GB of RAM and an Intel i7-3540 CPU operating at 3.0 GHz. The suggested model has several parameters that must be initialized prior to optimization. Grid search methods were used to train GWO-OFO-SVM. The model's starting parameters are displayed in Table 2.

Table 4 summarizes the GWO-OFO-SVM model's performance using the N-BaIoT dataset. It is evident that the suggested model continuously exhibits the greatest level of accuracy across a broad range of dataset sizes. Support Vector Machine (SVM) and XGBoost attain a minimum of 93.23% and 96.27%, respectively, for low test accuracy. As explained in Fig. 8.

Using the N-BaIoT dataset, table 4 summarizes the way of the suggested model GWO-OFO-SVM. It's evident that the suggested model continuously exhibits the greatest degree of precision throughout the broad range of dataset sizes. Low test accuracy is attained by Support Vector Machine (SVM) and XGBoost, which reach at least 93.23% and 96.27%, respectively, as shown in Tables 3 and 4. While XGBoost outperforms the other models with a maximum test accuracy of 97.28%, XGBoost shows improvement in these two models with a test accuracy of 96% as discussed in table 5. The lowest test accuracy, 93.23%, is attained by SVM for the same dataset size as shown in Fig. 9. However, SVM obtains the greatest training time of 18.79 seconds, XGBoost reaches 3.15 seconds, and the suggested model covers the moderate training time of 16.19 seconds as discussed in Fig. 10. However, when looking at F1-scores, SVM and XGBoost provide the lowest value of 93% and 96% respectively, although the recommended strategy is more effective and produces the greatest outcome, obtaining the highest F1-score of 97.28%. The suggested model and the competing algorithms SVM and XGBoost against the N-BaIoT dataset distinguish the training and assessment data for every class, as described by the confusion matrix in Fig .8(A) to Fig. 8(C). It highlights how well the suggested model's confusion matrix performs in comparison to all competing models as discussed in Fig. 8.

A comparison of a few new algorithms has been discussed in table 7, which shows the superiority of the proposed model, GWO-OFO-SVM algorithms, over others. To optimize IoT traffic and get more accuracy, article [37] used the SVM-PSO technique and achieved 96.2% accuracy. In another approach, an article [38] suggested a novel model, namely GWO-OCSVM, to use the N-BaIoT data to forecast IoT network assaults achieved 96.80% accuracy. In the article [39], the SSA-ALO technique is used over the N-BaIoT dataset, achieving 97.1% accuracy. Similarly, in the article [40], the BLSTM-RNN technique is used over the N-BaIoT dataset, achieving 96.70% accuracy. As explained in Fig. 11.

Table 3: Classification Report for proposed GWO-OFO-SVM model

Feature	Precision	Recall	F1- rating	Support
ACK	0.93	0.91	0.92	1999
Benign	1.00	1.00	1.00	2004
SCAN	1.00	1.00	1.00	1946
SYN	1.00	1.00	1.00	2068

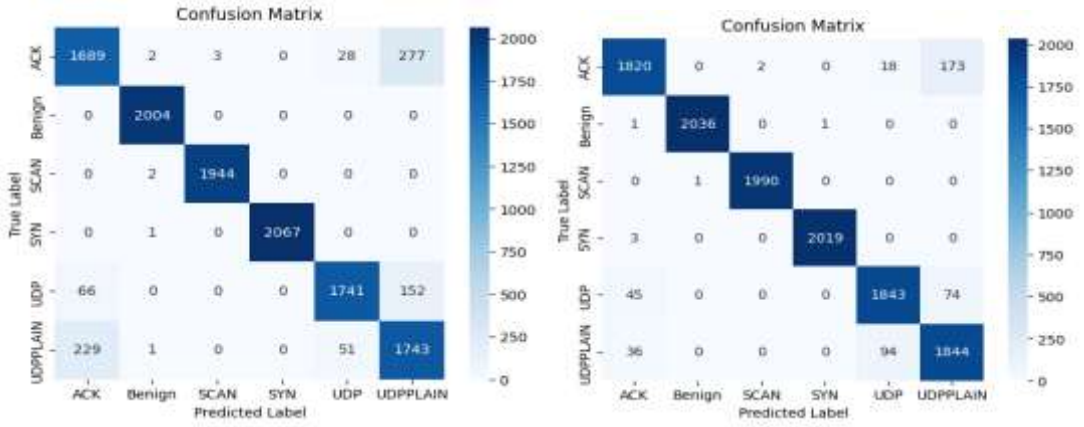
UDP	0.98	0.99	0.98	1959
UDP PLAIN	0.93	0.94	0.93	2024
Accuracy			0.97	12000
Macro Avg.	0.97	0.97	0.97	12000
Weighted Avg.	0.97	0.97	0.97	12000

Table 4: Classification Report for SVM

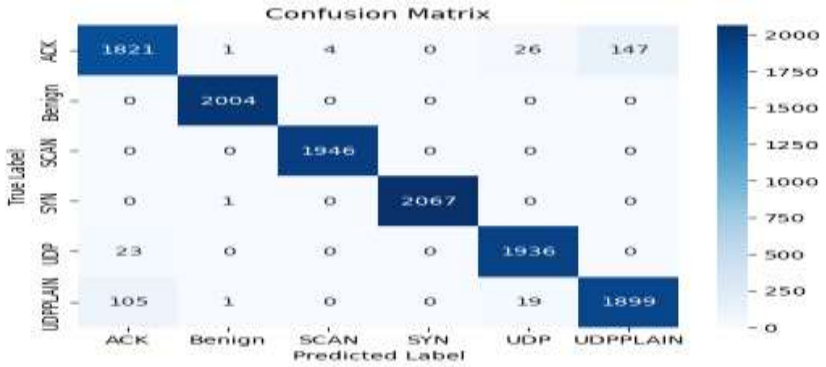
Feature	Precision	Recall	F1- score	Support
ACK	0.85	0.84	0.85	1999
Benign	1.00	1.00	1.00	2004
SCAN	1.00	1.00	1.00	1946
SYN	1.00	1.00	1.00	2068
UDP	0.96	0.89	0.92	1959
UDP PLAIN	0.80	0.86	0.83	2024
Accuracy			0.93	12000
Macro Avg	0.93	0.93	0.93	12000
Weighted Avg	0.93	0.93	0.93	12000

Table 5: Classification report XGBoost

Feature	Precision	Recall	F1- score	Support
ACK	0.96	0.90	0.93	2013
Benign	1.00	1.00	1.00	2038
SCAN	1.00	1.00	1.00	1991
SYN	1.00	1.00	1.00	2022
UDP	0.94	0.94	0.94	1962
UDP PLAIN	0.88	0.93	0.91	1974
Accuracy			0.96	12000
Macro Avg.	0.96	0.96	0.96	12000
Weighted Avg.	0.96	0.96	0.96	12000



(A) Confusion matrix of SVM **(B) Confusion matrix of XGBoost**



(C) Confusion matrix of proposed model

Figure 8. Quantitative result analysis displayed through confusion matrices in (A), (B) and proposed model in (C)

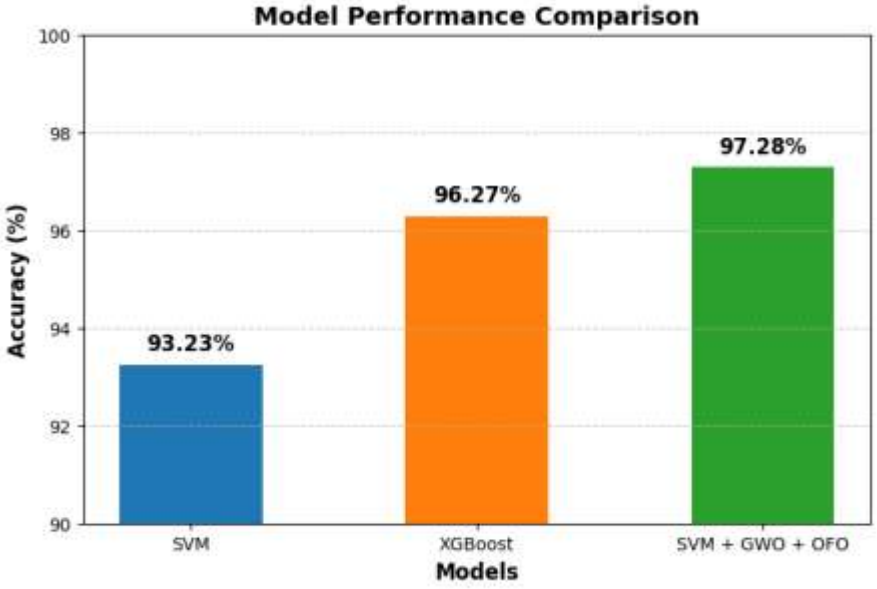


Figure 9. The performance comparison of SVM and XGBoost with proposed GWO-OFO-SVM algorithm using N-BaIoT dataset

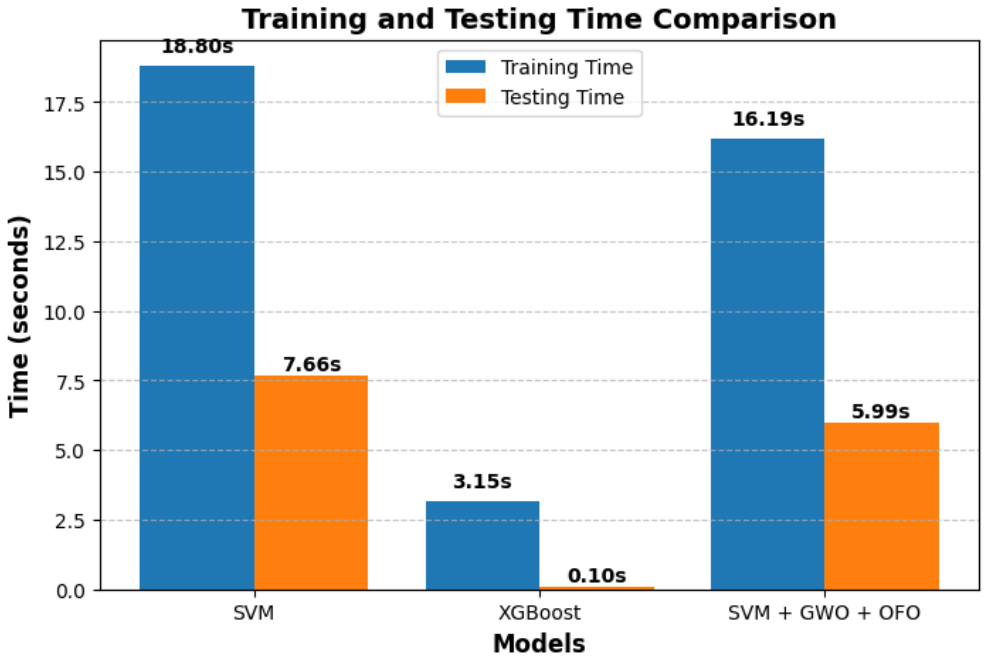


Figure 10. Comparison of training and testing time of SVM and XGBoost algorithms with proposed GWO-OFO-SVM algorithm

Table 6. Comparing the result with earlier studies

Related Literature	Dataset	Classification Model	F1-score	Accuracy (%)
[42]	N-BaIoT	GWO-OCSVM	95.89	96.80
[43]	N-BaIoT	SSA-ALO	--	97.1
[44]	N-BaIoT	BLSTM-RNN	96.75	96.70
Proposed Model	N-BaIoT	GWO-OFO-SVM	97.45	97.30

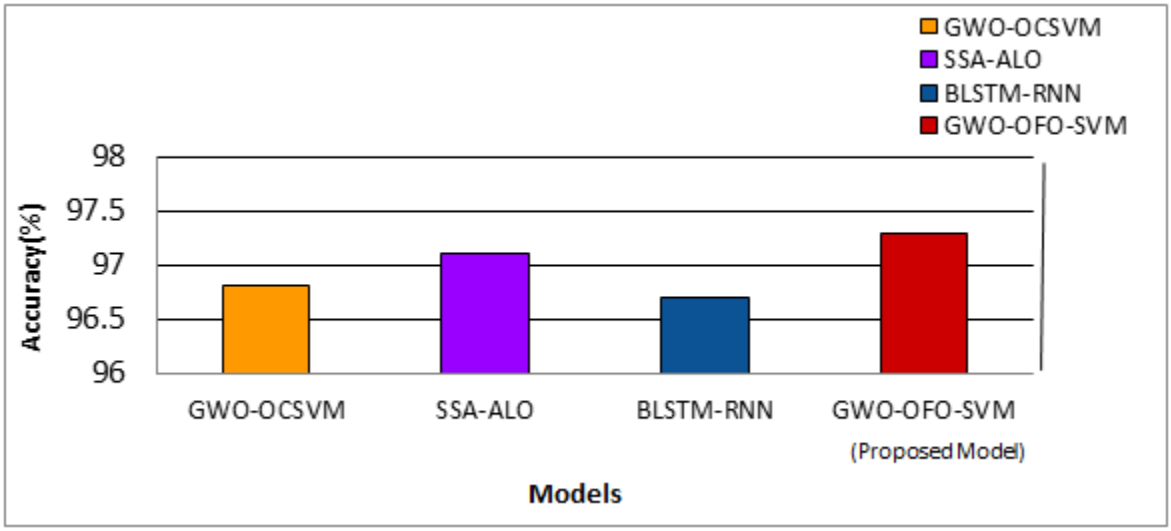


Figure 11. Accuracy comparison with existing algorithms GWO-OCSVM, SSA-ALO, BLSTM- RNN and proposed GWO-OFO-SVM algorithm using N-BaIoT dataset

7 Conclusion

This research suggests a new ML-based security technique to solve the vulnerabilities of the IoT systems. The proposed approach has been designed to address the increasing number of security issues related to the Internet of Things. This study focused on vulnerabilities, intelligent solutions and cutting edge security measures in IoT based smart systems using machine learning (ML) as a major technology to enhance IoT security. The paper demonstrated a security model with ML to automatically meet the increasing demands for high security in the IoT sector and showed the merits and demerits of applying ML in IoT environment. This study uses the N-BaIoT dataset for the assessment of the intrusion detection system. The proposed IDS model has been applied in the study on a dataset containing more than nine different types of attacks. In addition, the proposed model was evaluated in a real smart building environment, and the N-BaIoT dataset was used to evaluate the intrusion detection mechanism. The suggested approach is more accurate and faster than current metaheuristic algorithms, which makes it a practical choice for enhancing the security of IoT systems. The combination of Optical Flow Optimization (OFO) and Grey Wolf Optimizer (GWO) with Support Vector Machine (SVM) and adversarial learning is an effective method to locate Mirai botnet. The resilience of the Mirai botnet attack protection is coupled with the GWO-OFO-SVM algorithm. Our integrated model consistently attains remarkable accuracy and practical efficacy of 97.3% in solving the dynamic problems of Mirai botnet operations over many datasets.

DECLARATIONS

Ethical Acceptance

Inapplicable.

Competing Interests

The corresponding author certifies that there is no conflict of interest on behalf of all authors.

Authors' contributions

The authors confirm contribution to the paper as follows: study conception and design: Conceptualization; Pradeep Mishra and Asheesh Tiwari; Methodology, Pradeep Mishra; data collection: Pradeep Mishra; analysis and interpretation of results: Pradeep Mishra and Asheesh Tiwari; draft manuscript preparation: Pradeep Mishra and asheesh Tiwari. All authors reviewed the results and approved the final version of the manuscript.

Funding

There has been no external financing.

Availability of data and materials

On the request of the corresponding author, the data for this paper will be shared.

Referrences

[1] L. Li, Y. Yu, S. Bai, Y. Hou, and X. Chen, “An effective two-stepintrusion detection approach based on binary classificationand k-NN,” IEEE Access, vol. 6, pp. 12060–12073, 2018.

[2] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detectionmethod integrating anomaly detection with misuse detection,"Expert Systems with Applications, vol. 41, no. 4, pp. 1690–1700,2014.
[3] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selectionanalysis and building hybrid efficient model," Journal of Com-putational Science, vol. 25, pp. 152–160, 2018.
[4] W. Zhong and F. Gu, "A multi-level deep learning system formalware detection," Expert Systems with Applications,vol. 133, pp. 151–162, 2019.
[5] D. Papamartzivanos, F. G. Mármol, and G. Kambourakis,"Introducing deep learning self-adaptive misuse networkintrusion detection systems," IEEE Access, vol. 7, pp. 13546–13560, 2019.
[6] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learningapproach to network intrusion detection," IEEE transactionson emerging topics in computational intelligence, vol. 2, no. 1,pp. 41–50, 2018.
[7]. Muaadh A., Shukor R., Maheyzah Md S., Ibtehal N., Fuad A. G.,Faisal S., and Maged N.: Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning: A Systematic Literature Review, MDPI, Applied Science, 2021
[8]. Davahli, A., Shamsi, M., Abaei, G.: A lightweight Anomaly detection model using SVM for WSNs in IoTthrough a hybrid feature selection algorithm based on GA and GWO''. J Comput Secur 7, 63–79 (2020)
[9]. Li, X., Xu, M., Vijayakumar, P., Kumar, N., Liu, X.: Detection of low-frequency and multi-stage attacks in industrial internet of things. IEEE Trans Vehicle Technol 69, 8820–8831 (2020)
[10]. Kim, S., Hwang, C., Lee, T.: Anomaly based unknown intrusion detection in endpoint environments. Electronics. 9, 1022–1041 (2020)
[11] Artificial neural network–Wikipedia. Available online: https://en.wikipedia.org/wiki/Artificial_neural_network (accessed on 29 April 2020).
[12] Peddabachigari, S., Abraham, A., & Thomas, J. (2004). Intrusion detection systems using decision trees and support vector machines. *International Journal of Applied Science and Computations, USA*, *11*(3), 118-134.
[13]. Rai, K.; Devi, M.S.; Guleria, A. Decision Tree Based Algorithm for Intrusion Detection. *Int. J. Adv. Netw. Appl.* **2016**, *7*, 2828–2834.
[14]. Ingre, B.; Yadav, A.; Soni, A. K Decision Tree-Based Intrusion Detection System for NSL-KDD Dataset. In Proceedings of the International Conference on Information and Communication Technology for Intelligent Systems, Ahmedabad, India, 25–26 March 2017, doi:10.1007/978-3-319-63645-0_23.
[15] Farnaaz, N.; Jabbar, M.A. Random Forest Modeling for Network Intrusion Detection System. *Procedia Comput. Sci.* **2016**, *89*, 213-217.
[16]. Alom, M.Z.; Taha, T.M. Network intrusion detection for cybersecurity using unsupervised deep learning approaches. In Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 27–30 June 2017, doi:10.1109/NAECON.2017.8268746.
[17]. Yuan, Y.; Huo, L.; Hogrefe, D. Two Layers Multi-class Detection Method for Network Intrusion Detection System. In Proceedings of the IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece , 3–6 July 2017**,** doi:10.1109/ISCC.2017.8024620.
[18]. Gurav, R.; Junnarkar, A.A. Classifying Attacks in NIDS Using Naïve- Bayes and MLP. *Int. J. Sci. Eng. Technol. Res. (IJSETR)* **2015**, *4*, 2440–2443.
[19]. Tangi, S.D.; Ingale, M.D. A Survey: Importance of ANN-based NIDS in Detection of DoS Attacks. *Int. J. Comput. Appl.* **2013**, *83*, doi:10.5120/14494-2876.
[20]. Szegedy, C.; Toshev, A.; Erhan, D. Deep Neural Networks for Object Detection. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2; Curran Associates Inc.: Red Hook NY USA, 2013, pp. 2553–2561.
[21]. Wang, M.; Huang, Q.; Zhang, J.; Li, Z.; Pu, H.; Lei, J.; Wang, L. Deep Learning Approaches for Voice Activity Detection. In Proceedings of the International Conference on Cyber Security Intelligence and Analytics, Shenyang, China, 21–22 February 2019; pp. 816–826.
[22]. Tang, T.A.; Mhamdi, L.; McLernon, D.; Zaidi, S.A.R.; Ghogho, M. Deep Learning Approach for Network Intrusion Detection in Software-Defined Networking. In Proceedings of the 2016 International Conference
[23]. Arora, K.; Chauhan, R. Improvement in the Performance of Deep Neural Network Model using learning rate. In Proceedings of the Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017, doi:10.1109/IPACT.2017.8245184.
[24]. Tang, T.A.; Mhamdi, L.; McLernon, D.; Zaidi SA, R.; Ghogho, M. Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks. In Proceedings of the 4th IEEE International Conference on Network Softwarization (NetSoft), Montreal, QC, Canada, 25–29 June 2018, doi:10.1109/NETSOFT.2018.8460090.

[25]. Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access* **2017**, *5*, 21954–21961, doi:10.1109/ACCESS.2017.2762418.
[26]. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Applying convolutional neural network for network intrusion detection. In Proceedings of the International Conference on Advances in Computing, Communications, and Informatics (ICACCI), Udupi, India, 13–16 September 2017, doi:10.1109/ICACCI.2017.8126009.
[27]. Zhao, G.; Zhang, C.; Zheng, L. Intrusion Detection Using Deep Belief Network and Probabilistic Neural Network. In Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC*)*, Guangzhou, China, 21–24 July 2017, doi:10.1109/CSE-EUC.2017.119.
[28]. Kim, J.; Kim, J.; Thu HL, T.; Kim, H. Long Short-Term Memory Recurrent Neural Network Classifier for Intrusion Detection. In Proceedings of the International Conference on Platform Technology and Service (PlatCon), Jeju, Korea, 15–17 February 2016, doi:10.1109/PlatCon.2016.7456805.
[29]. Staudemeyer, R.C. Applying long short-term memory recurrent neural networks to intrusion detection. *S. Afr. Comput. J.* **2015**, *56*, 136–154, doi:10.18489/sacj.v56i1.248.
[30]. Meng, F.; Fu, Y.; Lou, F.; Chen, Z. An Effective Network Attack Detection Method Based on Kernel PCA and LSTM-RNN. In Proceedings of the International Conference on Computer Systems, Electronics, and Control (ICCSEC), Dalian, China, 25–27 December 2017, doi:10.1109/ICCSEC.2017.8447022.
[31]. Staudemeyer, R.C.; Omlin, C.W. Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*; Association for Computing Machinery: New York, NY, USA, 2013; pp. 218–224, doi:10.1145/2513456.2513490.
[32]. Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. A Deep Learning Approach for Network Intrusion Detection System. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BIONETICS), New York, USA, May 2016, pp. 21–26, doi:10.4108/eai.3-12-2015.2262516.
[33]. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets. *ArXiv* **2018**, arXiv:1806.03517.
[34]. Artificial neural network–Wikipedia. Available online: https://en.wikipedia.org/wiki/Artificial_neural_network (accessed on 29 April 2020).
[35] GÜVEN, E. Y. (2023). Mirai Botnet Attack Detection in Low-Scale Network Traffic. *Intelligent Automation & Soft Computing*, *37*(1).
[36] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, *17*(3), 12-22.
[37] Alsalman, D. (2024). A comparative study of anomaly detection techniques for IoT security using adaptive machine learning for IoT threats. *IEEE Access*, *12*, 14719-14730.
[38] Al Shorman, A., Faris, H., & Aljarah, I. (2020). Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection. *Journal of Ambient Intelligence and Humanized Computing*, *11*(7), 2809-2825.
[39] Abu Khurma, R., Almomani, I., & Aljarah, I. (2021). IoT botnet detection using salp swarm and ant lion hybrid optimization model. *Symmetry*, *13*(8), 1377.
[40] Kim, J., Won, H., Shim, M., Hong, S., & Choi, E. (2020). Feature analysis of IoT botnet attacks based on RNN and LSTM. *International Journal of Engineering Trends and Technology*, *68*, 43-47.
[41] Hadi, A. A. (2024). Intrusion Detection in IOT Networks using Machine Learning Techniques. *Journal of Electronics, Computer Networking and Applied Mathematics*, *4*(02), 1-18.
[42] Gewida, M., & Qu, Y. (2025). Enhancing IoT Security: Predicting Password Vulnerability and Providing Dynamic Recommendations using Machine Learning and Large Language Models. *European Journal of Electrical Engineering and Computer Science*, *9*(1), 8-16.