https://dx.doi.org/10.4314/dujopas.v11i1c.9
ISSN (Print): 2476-8316
ISSN (Online): 2635-3490
**Dutse Journal of Pure and Applied Sciences (DUJOPAS),** *Vol. 11 No. 1c March 2025*

# Enhancing Intrusion Detection in IoT Platforms Using a Novel Hybrid Gorilla Troops and Bird Swarm Optimization Algorithm

**Aisha Muhammad Sambo, Mustapha Aminu Bagiwa, Yusuf Sahabi Ali, Amina Hassan Abubakar**

Department of Computer Science
Ahmadu Bello University, Zaria

Email: aishamsambo94@gmail.com

## Abstract
*In the era of advancing technology, the proliferation of the Internet of Things (IoT) has become pervasive, influencing various facets of contemporary life. Intrusion Detection Systems (IDS) stand as a crucial guardian of these interconnected networks. Feature Selection emerges as a pivotal element in the design of effective IDS, aiming to discern the optimal subset of features for accurate attack classification within an extensive feature set. This paper introduces an approach that enhances a Hybrid Gorilla Troops Optimizer (GTO) algorithm and Bird Swarm Algorithm (BSA) with Step size parameters. The aim of adding the step size is to have controlled movement and adaptively explore and exploit the search space, thereby enhancing the performance of the hybrid algorithm. The hybridization leverages the strengths of both algorithms in identifying the optimal feature subset. The resulting Improved Hybrid GTO with BSA Algorithm (IGTO-BSA), utilizes metaheuristic techniques to boost feature selection, striking a balance between exploration and exploitation, foster faster convergence and deliver superior solutions within reasonable time. The effectiveness of IGTO-BSA is assessed using three diverse IoT datasets: NSL-KDD, CICIDS-2017, and UNSW-NB15. Performance evaluation is conducted using five metrics: accuracy, sensitivity, specificity, computational time and the number of selected features. Comparative analysis with an existing technique from the literature establishes the efficacy of the proposed approach.*

## INTRODUCTION
In the realm of advancing technology, the concept of the Internet of Things (IoT) has permeated various aspects of contemporary life (Heidari & Jabraeil Jamali, 2023). The term "Internet of Things" has historical roots, with Nikola Tesla envisioning a world transformed into a massive brain through wireless application (Ben-daya & Bahroun, 2020). The evolution of IoT has been shaped by emerging concepts like cloud computing, information-centric networking, big data, social networking, and constant technological influences (Xu *et al*., 2023). However, as IoT applications proliferate, security becomes a paramount concern, especially in the realm of cloud computing (Xu *et al*., 2023). The synergy between IoT systems and cloud computing relies heavily on effective data storage and analysis, making security vulnerabilities a focal point (Sarker *et al*., 2023).

In this landscape, Intrusion Detection Systems (IDS) emerge as a crucial line of defense against cyber-attacks in IoT systems*(Sarker et al*., 2023). IDS, categorized into anomaly detection and

**Author for Correspondence*

misuse detection, plays a pivotal role in identifying harmful actions and distinguishing between genuine and malicious instances (Stea, 2022; Perwej *et al*., 2019; Hoz, 2013).Despite their effectiveness, IDS systems have inherent flaws, such as high False Positive (FP) rates and difficulty distinguishing unknown attacks (Heidari & Jabraeil Jamali, 2023). Researchers have turned to machine learning techniques to address these challenges, yet achieving the desired level of accuracy remains an ongoing endeavor (Mishra *et al*., 2019).

One notable challenge in machine learning techniques is the complexity introduced by a wide range of attack types and network traffic attributes (Heidari & Jabraeil Jamali, 2023; Xu *et al*., 2023). Feature selection, the process of choosing the optimal subset of features, becomes critical in addressing this complexity (Isuwa *et al*., 2022). Traditional approaches face feasibility issues given the vast amount of data generated today, leading to the consideration of metaheuristic algorithms (Isuwa *et al*., 2023).

Metaheuristic algorithms, renowned for their dynamic search behavior and global search capabilities, have emerged as effective solutions for feature selection problems. Algorithms like Genetic Algorithm (GA) (Ali & Saeed, 2023), Particle Swarm Optimization (PSO) (Hassan, Mohammed, *et al*., 2023), Grey Wolf Optimizer (GWO)(El-Kenawy & Eid, 2020), and Gorilla Troops Optimizer (GTO) (Abdoullazadeh *et al*., 2021)have been proposed to tackle these challenges, with hybrid approaches demonstrating superior performance .

The GTO algorithm, inspired by the social structure of gorilla groups, simulates the decision-making processes of a silverback gorilla group's leader(Mostafa *et al*., 2023). Divided into initialization, global exploration, and local exploitation phases, GTO represents a unique approach to optimization problems (Abdoullazadeh *et al*., 2021).

In parallel, the Bird Swarm algorithm (BSA) (Miramontes & Melin, 2023), rooted in swarm intelligence and inspired by bird behaviors, has shown significant prowess in solving optimization problems(Miramontes & Melin, 2023). With behaviors encompassing foraging, vigilance, and flight, the BSA offers a distinctive approach to addressing complex challenges (Meng *et al*., 2016).

Kennedy and Eberhart (1995) introduced Particle Swarm Optimization (PSO), a pioneering nature-inspired optimization algorithm that simulates the social behaviors of birds flocking or fish schooling. PSO leverages a population of candidate solutions, referred to as particles, which explore the search space by adjusting their positions based on their own experience and the experience of neighboring particles. Each particle updates its velocity and position by considering its personal best position and the global best position found by the swarm. This iterative process enables particles to converge towards optimal solutions over successive generations. PSO has been widely acclaimed for its simplicity, ease of implementation, and robust performance across various optimization problems. Kennedy and Eberhart demonstrated the effectiveness of PSO through numerous benchmark tests, showing that it consistently finds high-quality solutions with relatively fast convergence times.

Meng *et al*. (2016) proposed a novel approach known as the Bird Swarm Algorithm (BSA). This algorithm emulates the foraging and vigilance behaviors of bird swarms to optimize complex problems. BSA divides the swarm into three groups based on different behaviors: producers, scroungers, and vigilantes. Producers search for food resources, scroungers follow the producers to exploit these resources, and vigilantes keep an eye on potential threats. The algorithm leverages these dynamics to balance exploration and exploitation during the optimization process. Meng et al. demonstrated the efficacy of BSA through extensive experiments on benchmark functions, showcasing its superior performance in terms of convergence speed and solution accuracy when compared to other state-of-the-art algorithms such as PSO and GA.

Gauthama *et al*. (2017) proposed the utilization of a genetic algorithm for parameter optimization and feature selection within a hypergraph-based intrusion detection system. The study leveraged the hyper-clique property of hypergraphs to generate initial populations, expediting the search for optimal solutions and mitigating the risk of being trapped in local optima. The HyperGraph Genetic Algorithm (HG-GA) employed a weighted objective function to determine the optimal number of features while simultaneously maximizing the detection rate and minimizing the false alarm rate. On the NSL-KDD dataset, HG-GA demonstrated a detection rate of 95.32% and a false alarm rate of 3.17%. It was identified as scalable, adaptive, robust, and applicable to a wide range of problems, although it was acknowledged that GA might encounter local optimal solutions.

Saremi *et al*. (2017) introduced the Grasshopper Optimization Algorithm (GOA), a novel nature-inspired metaheuristic algorithm designed to solve complex optimization problems. GOA simulates the swarming behavior of grasshoppers in nature, particularly their movement and social interactions. The algorithm models the long-range and short-range attraction forces between grasshoppers to balance exploration and exploitation during the search process. By doing so, GOA efficiently navigates the search space to identify optimal solutions. Saremi et al. conducted extensive experiments on a variety of benchmark optimization problems to validate the performance of GOA. The results demonstrated that GOA outperforms several existing optimization algorithms, including PSO and GA, in terms of convergence speed and solution accuracy. Specifically, GOA achieved significant improvements in accuracy, with performance metrics indicating superior ability to find optimal or near-optimal solutions across diverse problem domains. The robustness and efficiency of GOA make it a valuable tool for tackling a wide range of optimization challenges.

Salp Swarm Algorithm (SSA), introduced by Mirjalili *et al*. (2017), a novel nature-inspired optimization algorithm that simulates the swarming behavior of salps in the ocean. SSA is inspired by the unique chain-like formation of salps, which helps them efficiently navigate and forage in the ocean depths. In SSA, the population of salps is divided into leaders and followers, where the leaders guide the swarm towards optimal solutions and the followers adjust their positions based on the leaders and their immediate neighbors. This dynamic interaction between leaders and followers allows SSA to effectively balance exploration and exploitation throughout the optimization process. Mirjalili *et al*. validated SSA through extensive experimentation on a wide range of benchmark optimization problems, demonstrating its superior performance in terms of convergence speed and solution accuracy compared to other state-of-the-art algorithms like PSO and Genetic Algorithm(GA). The results showed that SSA achieved significant improvements in accuracy and robustness, making it a powerful tool for solving complex optimization challenges across various domains.

M. Swarna *et al*., (2020) initiated the preprocessing stage by employing one-hot encoding to transform the data. Subsequently, the transformed dataset underwent a hybrid approach involving the Principal Component Analysis (PCA) and Grey Wolf Optimization (GWO) algorithms. PCA was utilized for dimensionality reduction, effectively reducing the dataset's dimension. Following PCA, GWO was applied to transform the coordinates, ensuring the preservation of dimension variety. The feature values were then converted to numerical representations. Upon subjecting the preprocessed dataset to various classifiers, a notable improvement of approximately 15% in accuracy was observed, accompanied by reduced training time.

In a different approach, Ghosh and Das (2021) proposed a novel hybrid Cuckoo Search (CS) - PSO algorithm for rapid and effective attack classification. The model, evaluated on the NSL-KDD dataset, involved preprocessing and normalizing the data to reduce training time. The algorithm computed the objective function value for each data point, assigning pbest and gbest values to track individual and group best results at each stage. The CS-PSO algorithm effectively addressed the exploration-exploitation tradeoff, utilizing the Levy Flight technique during the exploration phase to jump into random positions and achieve better global optimized values.

Abdollahzadeh *et al*. (2021) introduced the Artificial Gorilla Troops Optimizer (AGTO), a novel nature-inspired metaheuristic algorithm designed to tackle global optimization problems. AGTO mimics the social hierarchy and collective behaviors of gorilla troops in the wild. The algorithm models the foraging strategies, leadership dynamics, and territorial disputes observed in gorilla groups to enhance exploration and exploitation capabilities. By simulating these interactions, AGTO effectively navigates the search space to identify optimal solutions. The authors validated AGTO through rigorous testing on a variety of benchmark optimization problems, demonstrating its competitive performance against established algorithms like Particle Swarm Optimization (PSO) and Differential Evolution (DE). The results indicated that AGTO excels in convergence speed and solution quality, making it a valuable addition to the suite of metaheuristic optimization tools. Abdollahzadeh et al. highlighted AGTO's potential applications in diverse fields, emphasizing its robustness and efficiency in solving complex optimization challenges.

In a separate study, Kareem *et al*. (2022) proposed a hybrid GTO/BSA algorithm. Recognizing that GTO, like many other meta-heuristic algorithms, is susceptible to local optima due to an imbalance between exploration and exploitation, the authors addressed this challenge. Four strategies were implemented to enhance GTO's local and global searching capabilities: a control randomization parameter, an advanced non-linear transfer function for balancing exploration and exploitation, and a novel local updating position strategy based on the BSA algorithm. Fine-tuning randomization parameters played a crucial role in ensuring correct search algorithm behavior and achieving a balanced exploration-exploitation dynamic. The control randomization parameter generated variable numbers between positive and negative values, facilitating a comprehensive exploration of the search space and preventing stagnation in sub-local optimal solutions. The transfer function, vital in transitioning the search algorithm between exploration and exploitation phases, replaced the linear transfer function present in GTO, as it often failed to strike a balance.

Despite the efficiency of integrating gorilla troop optimization with bird swarm algorithm, it still suffers from computational complexity, computational overhead and the potential of falling in the trap of local optima. to overcome these limitations, this study proposes the enhancement of GTO-BSA whose aim is to reduce computational complexity and avoid falling into the trap of local optima and therefore achieve superior solutions in optimizing intrusion detection in IoT platforms.

### A. Gorilla Troops Optimization (GTO)

GTO draws inspiration from the collective lifestyle and social intelligence observed in gorillas. Within a gorilla troop, adult male silverback gorillas coexist with multiple adult female gorillas. The Silverback, distinguished by unique hair on its back during puberty and typically around the age of 12, assumes the leadership role within the troop. The Silverback is responsible for decision-making on behalf of the group, resolving conflicts, and determining group movements (Xiao *et al*., 2022). In general, both male and female gorillas may migrate from one birth group to another. In the event of the silverback's demise, males may engage in fierce battles for group dominance and mating rights with adult females. Leveraging insights from these observed gorilla troop behaviors, a mathematical model is formulated for GTO. The GTO model comprises three key phases: initialization, global exploration, and local exploitation (Xiao et al., 2022).

#### a) Initialization phase

Assuming there are $n$ gorillas in a d-dimensional space, the position of the $i^{th}$ gorilla in the space can be defined as $X_i = (x_{i,1}, x_{i,2}, x_{i,3} \ldots x_{i,d})$, $i = 1, \ldots, N$. Therefore, the initialization of the gorilla process can be defined as shown in equation 1.

$$X_{n\times D} = Rand(N, D) \times (ub - lb) + lb \quad (1)$$

where $ub$ and $lb$ are the upper and lower boundaries respectively, and $Rand\ (N, D)$ denotes the matrix with $N$ rows and $D$ columns. Each element is a random number between 0 and 1.

#### b) Exploration phase

Upon departure from their native troop, gorillas navigate through diverse natural landscapes. In the context of the GTO algorithm, all gorillas are regarded as candidate solutions, with the optimal solution identified in each optimization iteration termed as the "silverback." The position update of a gorilla during the exploration stage is governed by three distinct strategies: migration toward unfamiliar positions, relocation to familiar locales, and movement toward other groups. The formulation of this concept is encapsulated in Equation 2.

$$GX(t+1) = \begin{cases} (ub - lb) \times r2 + lb & r1 < P \\ (r_3 - C) \times X_a(t) + L \times Z \times X(t) & r1 \geq 0.5 \\ X(t) - L \times \left(L \times \left(X(t) - X_B(t)\right) + r_4 \times \left(X(t) - X_B(t)\right)\right), & r1 < 0.5 \end{cases} \quad (2)$$

where $t$ specifies the current iteration, $X(t)$ specifies the current position vector of individual gorillas. $GX(t+1)$ refers to the candidate position of search agents in the next iteration. Also, $r_1, r_2, r_3$, and $r_4$are all random numbers in the range 0 and 1. $X_a(t)$and $X_b(t)$are two randomly selected gorilla positions in the current population.P is a constant and $Z$ specifies a row vector in the problem dimension with values that are randomly generated in the interval [-C, C] where $C$ is calculated according to equation.3.

$$C = (\cos(2 \times r_5) + 1) \times \left(1 - \frac{t}{Maxiter}\right) \quad (3)$$

where $cos(.)$ represents the cosine function, $r_5$ is a random number in the range of 0 to 1, and $Max - iter$ indicates the maximum iterations. $L$ can be computed as in equation 4

$$L = C \times l \qquad (4)$$

where $l$ is a random number between [-1 and 1].

#### c) Exploitation phase

The mathematical expression for the modeling of the exploitation phase of the GTO is shown in equation 5.

$$GX(t+1) = L \times M \times (X(t) - X_{silverback}) + X(t) \qquad (5)$$

Where $L$ is evaluated using equation 4 and $X_{silverback}$ represents the best solution obtained so far and $X(t)$ specifies the current position vector. $M$ is be calculated using equation 6.

$$M = \left(\left|\sum_{i=1}^{N} X_i(t)/N\right|^{2^L}\right)^{\frac{1}{2^L}} \quad (6)$$

where $N$ specifies the population size, and $X_i(t)$ denotes each position vector of the gorilla in the current iteration. If $C < W$, it means that the latter mechanism is chosen and the location of gorillas can be updated using equations 7 to. 9.

$$GX(t+1) = X_{silverback} - (X_{silverback} \times Q - X(t) \times Q) \times A \quad (7)$$

$$Q = 2 \times r_6 -1 \quad (8a)$$

$$A = \varphi \times E \quad (8b)$$

$$E = \begin{cases} N_1, r_7 \geq 0.5 \\ N_2, r_7 < 0.5 \end{cases} \quad (9)$$

In equation 7, $X(t)$ specifies the current position and $Q$ stands for the impact force, which is calculated using equation 8a. $r_6$ is a random value in the range of 0 and 1. Moreover, the coefficient $A$ is used to mimic the violence intensity in the competition is evaluated by equation 8b. $\varphi$ specifies a constant and the values of $E$ are assigned using equation 9. $r_7$ is also a random number in the range [0, 1]. If $r_7 \geq 0.5$, $E$ would be defined as a 1-D array of normal distribution and $D$ is the spatial dimension. If $r_7 < 0.5$, $E$ will be equal to a stochastic number that conforms to the normal distribution. At the end of the exploitation process, the fitness value of the newly generated candidate $GX(t+1)$solution is also calculated. If $F(GX) < F(X)$, the solution $GX$will be preserved and participate in subsequent optimization, the optimal solution within all individuals is specified by the silverback.

**B. Bird Swarm Optimization Algorithm (BSA)**

The BSA algorithm draws inspiration from the social dynamics observed in bird swarms, incorporating key elements such as feeding, flying, and vigilance compartments (Miramontes & Melin, 2023). This algorithm is intricately governed by predefined rules that dictate its behavior, and mathematical equations within the BSA framework are systematically formulated to align with these rules (Miramontes & Melin, 2023).

i. **Rule 1:** Each bird exhibits dynamic behavior by probabilistically transitioning between vigilance and foraging states. This decision-making process is characterized as a stochastic decision.

ii. **Rule 2:** During the foraging state, each bird adeptly records and updates its individual best experiences related to food patches, as well as the collective best experiences of the swarm. This acquired knowledge is promptly disseminated across the entire swarm, facilitating informed decision-making in the food search. Social information is efficiently shared among swarm members.

iii. **Rule 3:** Birds strive to converge towards the central area of the swarm while maintaining vigilance, a behavior influenced by the interference resulting from swarm competition. Birds with higher resource reserves tend to position themselves closer to the swarm's center compared to those with lower reserves.

iv. **Rule 4:** Birds engage in frequent relocation, transitioning between different locations. During these movements, birds alternate between producing and scrounging behaviors. The bird with the highest reserves assumes the role of a producer, while the one with the lowest reserves adopts the role of a scrounger. Intermediate reserve levels prompt random decisions for birds to choose between being a producer or a scrounger.

v. **Rule 5:** Producers actively seek food, while scroungers randomly follow a producer in their collective quest for food (Rule 5).

### a) Foraging behavior

Individual bird groups engage in food-seeking activities influenced by both their individual experiences and the collective knowledge amassed from other birds within the swarm. This behavior can be mathematically modeled using Equation 10

$$X_{i,j}^{t+1} = X_{i,j}^{t} + (P_{i,j} - X_{i,j}^{t}) \times C \times rand(0,1) + (g_j - X_{i,j}^{t}) \times S \times rand\,(0,1) \qquad (10)$$

where $j$ specifies the set of uniformly distributed numbers within the range 0 and 1. $S$ specifies the social accelerated coefficients and $C$ specifies cognitive accelerated coefficients. While $P_{i,j}$ is the previous location of the $i^{th}$ bird and $g_j$ the previous best location of the swarm.

### b) Vigilance behavior

Each bird attempts to proceed to the middle of the swarm to engage in surveillance behavior. This can be modeled as in equation 11 to 12.

$$X_{i,j}^{t+1} = X_{i,j}^{t} + A1\left(mean_j - X_{i,j}^{t}\right) \times rand(0,1) + A2\left(P_{k,j} - x_{i,j}^{t}\right) \times rand\,(-1,1) \qquad (11)$$

$$A1 = a1 \times \exp\left(\frac{-PFit_i}{sumFit + \varepsilon}\right) \times N \qquad (12a)$$

$$A2 = a2 \times \exp\left(\left(\frac{pFit_i - pFit_k}{|pFit_k - pFit_i| + \varepsilon}\right)\frac{NpFit_k}{sumFit + \varepsilon}\right) \qquad (12b)$$

where $k$ specifies a positive number between 1 and $N$ chosen at random. The best fitness value at the $i^{th}$ position is $pFit_i$ and the total of the swarm's best objective value is sumFit. $\varepsilon$ is used to avoid the zero division error. $A1$ and $A2$ represent the positive constant values (0,2).

### c) Flight behavior

In reaction to potential predation threats, birds can migrate to diverse locations for foraging and other activities (Miramontes & Melin, 2023). Upon reaching a new location, birds continue their quest for food. Producers, who actively seek and find food, are then followed by scroungers who consume the located food (Miramontes & Melin, 2023). The behaviors of both producers and scroungers are formalized in Equations 13 and 14.

$$x_{i,j}^{t+1} = x_{i,j}^{t} + randn(0,1) \times x_{i,j}^{t} \qquad (13)$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \left(x_{k,j}^{t} - x_{i,j}^{t}\right) \times FL \times rand\,(0,1) \qquad (14)$$

## MATERIALS AND METHODS

This section delineates the research methodology employed. It discusses the details of the proposed and that of Kareem *et al.*(2022), IGTO-BSA, the datasets used, and the experimental design.

## EXPERIMENTAL SETUP

All experiments were carried out on a Windows 10 operating system with Intel(R) Core (TM) i7-6700HQ CPU (2.60 GHz; 2.59 GHz with 64 GB. The MATLAB R2021b programming language environment was used.

## Dataset Description

The suggested model underwent testing with three datasets: NSL-KDD, CICIDS-2017, and UNSWNB-15 Dataset. These datasets are commonly utilized by researchers to assess the effectiveness of their proposed systems.

### a). The NSL-KDD dataset

The NSL-KDD dataset was introduced as a remedy to address inherent challenges in the KDDCUP'99 dataset (Al-Khassawneh, 2023). In comparison to the original KDD dataset, NSL-KDD offers several advantages: it eliminates redundant data from the training set, mitigating classifier bias toward more prevalent records (Al-Khassawneh, 2023). Comprising 41 features and 5 classes (normal and four forms of attack: DoS, Probe, R2L, and U2R), the NSL-KDD dataset provides a refined and improved data structure for research purposes.

### b). CICIDS-2017 dataset

The CICIDS-2017 dataset, unveiled by the Canadian Institute for Cybersecurity (CIC) encompasses prevalent benign activities and contemporary cyber threats (Krsteski *et al*., 2023). Positioned as one of the latest intrusion-detection datasets, it incorporates up-to-date attack scenarios. Comprising a total of 2,830,743 records distributed across eight files, each entry within this dataset is characterized by 78 distinct features along with their corresponding labels (Krsteski *et al*., 2023).

### c). UNSW-NB15 dataset

The UNSW-NB15 dataset, developed at UNSW Canberra was created using IXIA perfect storm to generate a comprehensive mix of benign and attack traffic (Sallam *et al*., 2023). This effort resulted in a 100GB dataset represented as PCAP files, featuring a substantial number of novel generated features (Sallam *et al*., 2023). The primary objective of this dataset is to serve as a resource for the creation and validation of intrusion detection systems. The dataset encompasses nine distinct attack types, namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and worms (Fathima *et al*., 2023). Comprising a total of 2,540,044 records organized into four CSV files, a subset of this dataset has been designated for training and testing purposes. Specifically, the training set encompasses 175,341 records, while the testing set incorporates 82,332 records, encompassing all attack types and standard records (Fathima *et al*., 2023).

## Evaluation Metrics

To assess the efficacy of the proposed IGTO-BSA approach, various evaluation metrics have been applied as shown in equation 16 to 20. The evaluation encompasses the use of a confusion matrix to measure the accuracy, specificity, and sensitivity of the classifier in both the existing method and our proposed method, along with other pertinent evaluation methods.

i. Accuracy = $\frac{TP+TN}{TP+FP+FN+TN}$ (15)

ii. Sensitivity= $\frac{TP}{TP+TN}$ (16)

iii. Specificity= $\frac{TN}{TN+FP}$ (17)

where $TP$ = True Positive, $TN$ = True Negative, $FP$ = False Positive, $FN$ = False Negative

iv. Average accuracy ($AVG_{acc}$): This measure is employed to compute the accuracy of data classification.

v. Average number of features ($AVG_{|fsBest|}$): This metric is employed to assess the capability of a method to reduce the number of features across multiple runs.

vi. Standard Deviation (STD): This metric is used to evaluate each method's quality and analyze the data obtained in multiple runs.

$$Standarddeviation = \sqrt{\frac{1}{P}\sum(B_i - Mean)^2} \quad (18)$$

vii. Average computation time: This metric is employed to determine the average duration taken by the algorithm to complete one iteration of a task.

$$AVG_{Time} = \frac{1}{Nr}\sum_{K=1}^{Nr} Time_{Best}^{k} \quad (19)$$

**Proposed Model**

i. **Position Update During Exploration:** The initial exploration strategy in the original GTO utilized equation (2). In our proposed IGTO-BSA, we enhanced the last strategy by incorporating a step size parameter into the update mechanism. The modified equation (20) in the proposed approach calculates the updated position for an individual at index $i$ within the gorilla population. This equation plays a pivotal role in refining the current solution by guiding it towards the solution of the best-performing neighbor. The adjustment's magnitude is now determined by the parameter 'alpha,' acting as a controlling factor that influences the extent of the individual's movement towards the optimal neighbor's solution.

$$GX(t+1) = X(t) - \propto \times \left(L \times (L \times \left(X(t) - X_B(t)\right) + r_4 \times (X(t) - X_B(t)))\right) \quad (20)$$

ii. **Position Update During Exploitation:** Equation 22 computes the updated position for the individual at index i within the bird swarm population. Equation (11) was modified to equation (21)by introducing the step sizes 'beta' and 'gamma'. This modification aims to enhance the algorithm's performance by providing more control over the individuals' movement and introducing additional variability into the update process. The parameter 'beta' functions as a controlling factor, influencing the magnitude of the adjustment and directing the individual towards the solution of the best-performing neighbor. On the other hand, 'gamma' is introduced with the purpose of enhancing diversity in the update process.

$$X_{i,j}^{t+1} = X_{i,j}^{t} + \beta \times A1\left(mean_j - X_{i,j}^{t}\right) \times rand(0,1) + \gamma \times A2\left(P_{k,j} - x_{i,j}^{t}\right) \times rand(-1,1) \quad (21)$$

The algorithm for the proposed model, the flowchart and the parameter settings are shown in Algorithm 1, Figure 1 and table 1.

**Algorithm 1: Improved GTO-BSA (IGTO-BSA)**

1. *Initialize the population size N and the maximum number of iterations Maxiter*
2. *Determine suitable parameters for gorilla and bird*
3. *Initialize the random gorilla and bird population* $X_i(1,2,\dots,N)$
4. *Calculate the fitness value of all gorilla individuals*
5. *While* $t < Maxiter$ *do*
6.     *Update the parameter C according to equation 3*
7.     *Update the parameter L according to equation 4*
8.     *For each* $X_i$ *do*
9.         *Update position of gorilla according to equation 20*
10.     *End for*
11.     *Evaluate the fitness values of all gorillas*
12.     *Save optimal position as a silverback* ($X_{silverback}$)
13.     *For each gorilla* $X_i$ *do*

14. *If C≥ W then*
15. *Update the position of the current gorilla according to equation 7*
16. *Else update the position of the birds according to equation 21*
17. *Update the fitness values of all gorillas*
18. *Update the global best solution ,X-silverback.*
19. *t = t+1*
20. *End While*
21. *Output the global best solution*

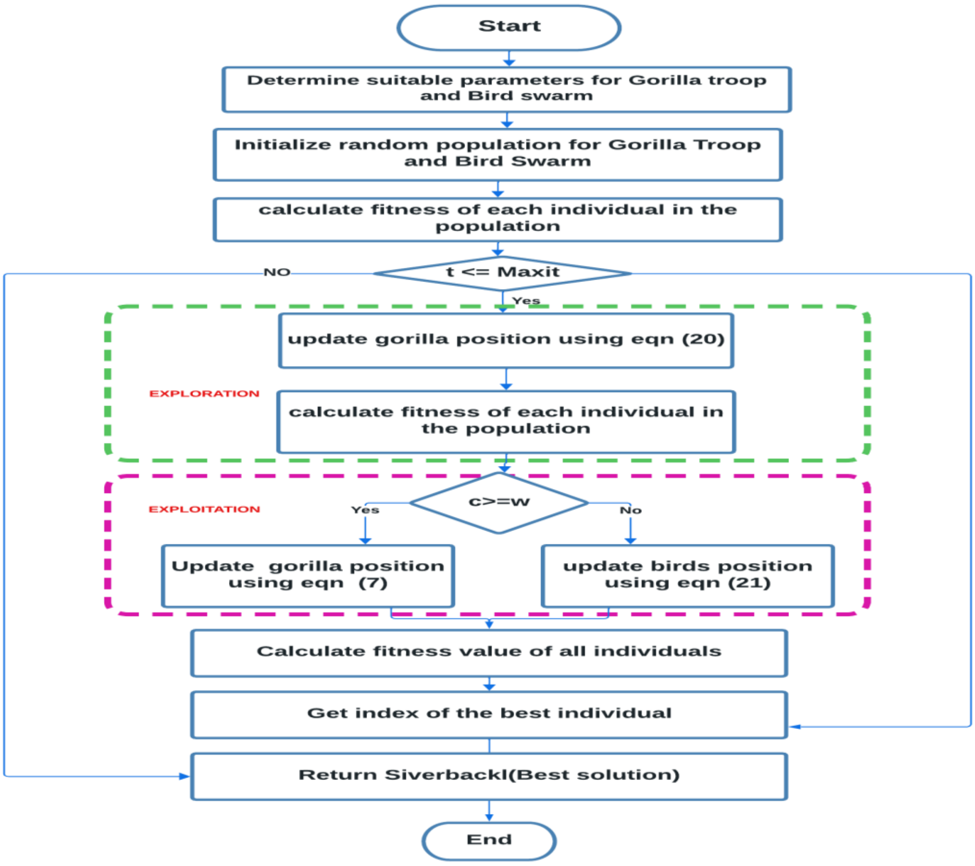



Figure 1: Flowchart of the proposed IGTO-BSA algorithm

Table 1: Parameter Settings

| Parameter | Value |
|---|---|
| Number of iterations | 100 |
| Population size | 30 |
| Search domain | [0,1] |
| Number of runs | 25 |
| $\beta$ | 0.01 |
| $\alpha$ | 0.99 |
| Alpha | 0.1 |
| Beta | 0.2 |

## RESULTS AND DISCUSSION

Analyzing Table 2 with a focus on the NSL-KDD dataset reveals that the proposed method, which incorporates step size parameter, exhibits superior performance compared to GTO-BSA across four key metrics: accuracy, specificity, computational time, and the count of selected features.

Table 2: Performance comparison between GTO-BSA and the proposed method using the NSL-KDD datasets

| Dataset | Models | | Performance Measures | | | | |
|---|---|---|---|---|---|---|---|
| | | | Accuracy (%) | Sensitivity (%) | Specificity (%) | Computational Time | No. of features |
| NSL-KDD | Proposed Method IGTO-BSA | Avg. | <u>**0.9845**</u> | 0.8229 | <u>**0.9900**</u> | <u>**211.78**</u> | <u>**13.0000**</u> |
| | | STD | ±0.0026 | ±0.1673 | ±0.0500 | ±6.3532 | - |
| | Existing System GTO-BSA | Avg. | 0.955964 | <u>**0.914219**</u> | 0.97365 | 10,205.83 | 14.75 |
| | | STD | ±0.000777 | ±0.006015 | ±0.001684 | ±1531.406 | - |

### A. Performance comparison between GTO-BSA and the proposed method using the UNSW-NB15 dataset

Examining Table 3, particularly in the context of the UNSW-NB15 dataset, it is evident that the proposed method, which integrates step size parameters, outperforms GTO-BSA across all five metrics: accuracy, sensitivity, specificity, computational time, and the count of selected features.

Table 3: Performance comparison between GTO-BSA and the proposed method using the UNSW-NB15 datasets

| Dataset | Models | | Performance Measures | | | | |
|---|---|---|---|---|---|---|---|
| | | | Accuracy (%) | Sensitivity (%) | Specificity (%) | Computational Time | No. of features |
| UNSW-NB15 | Proposed Method IGTO-BSA | Avg. | <u>**0.8273**</u> | <u>**0.8894**</u> | <u>**0.8783**</u> | <u>**126.1023**</u> | <u>**10.0000**</u> |
| | | STD | ±0.0038 | ±0.1685 | ±0.1096 | ±3.7833 | - |
| | Existing System GTO-BSA | Avg. | 0.710138 | 0.815385 | 0.877049 | 161.2396 | 16.625 |
| | | STD | ±0.010759 | ±0.052656 | ±0.019192 | ±4.890585 | - |

### B. Performance comparison between GTO-BSA and the proposed method using the CICIDS-2017 dataset

From Table 4, considering the CICIDS-2017 dataset, it is noteworthy that the proposed method demonstrates superior performance over GTO-BSA specifically in terms of accuracy, sensitivity, and specificity. However, it is important to highlight that GTO-BSA exhibits better results in terms of computational cost and the number of selected features. In summary, while the proposed method excels in certain key metrics, GTO-BSA showcases advantages in computational efficiency and feature selection, indicating a nuanced comparative evaluation between the two approaches.

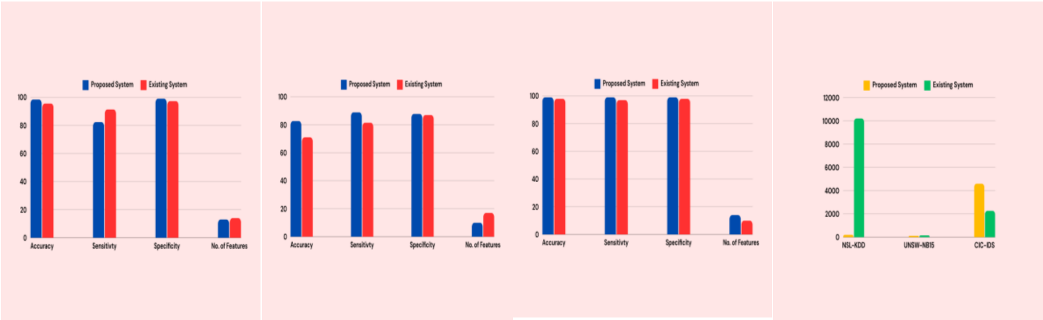
Table 4: Performance comparison between GTO-BSA and the proposed method using the CICIDS-2017 datasets

| Dataset | Models | | Accuracy (%) | Sensitivity (%) | Specificity (%) | Computational Time | No. of features |
|---|---|---|---|---|---|---|---|
| CICIDS-2017 | Proposed Method IGTO-BSA | Avg. | 0.9986 | 0.9994 | 0.9978 | 4604.313 | 14.0000 |
| | | STD | ±0.0006 | ±0.0003 | ±0.0011 | ±1889.0432 | - |
| | Existing System GTO-BSA | Avg. | 0.987915 | 0.972644 | 0.996798 | 2270.918 | 10 |
| | | STD | ±0.001997 | ±0.004299 | ±0.001977 | ±221.0268 | - |

The observed variations in performance between the proposed method and GTO-BSA on the CICIDS-2017 dataset underscore the relevance of the "No Free Lunch"(Montazeri et al., 2023) theory in optimization. This theory asserts that no universally superior optimization algorithm excels across all problem domains. The fact that the proposed method outperforms GTO-BSA in accuracy, sensitivity, and specificity while GTO-BSA demonstrates superiority in computational cost and the number of selected features aligns with the inherent trade-offs dictated by the "no-free lunch" principle. Essentially, the effectiveness of an optimization algorithm is contingent on the specific characteristics and nuances of the problem at hand. This acknowledgment emphasizes the importance of selecting or designing optimization methods tailored to the unique requirements of each specific task or dataset.

The charts below compare the performance metrics of the proposed Improved Hybrid GTO-BSA Algorithm (IGTO-BSA) against existing system(GTO-BSA). Chart (a) evaluates the NSL-KDD dataset, illustrating the superior accuracy, and specificity of IGTO-BSA, alongside a reduced number of selected features. Chart (b) shows the superiority of our proposed system in terms of higher accuracy, sensitvity, specificity and reduced number of features on UNSW-NB15 dataset. Chart (c) assesses the CICIDS-2017 dataset, again showcasing the proposed system's higher accuracy, sensitivity, and specificity. On the other hand the existing system shows superiority with less seclected number of features. Chart (d) examines computational time across the three datasets, revealing IGTO-BSA's enhanced efficiency, especially noticeable in the NSL-KDD and UNSW-NB15 datasets. These visualizations underscore the improved efficacy and efficiency of the IGTO-BSA method in intrusion detection for IoT environment.

**NSL-KDD** **UNSW-NB15** **CICIDS-2017** **COMPUTATIONAL TIME**



**(a)** **(b)** **(c)** **(d)**

## CONCLUSION

The research concludes with the successful development and evaluation of the IGTO-BSA feature selection algorithm, specifically tailored for Intrusion Detection Systems in the context of Internet of Things security. By integrating elements from Gorilla Troop Optimization (GTO) and Bird Swarm Algorithm (BSA), and incorporating step size parameters, the proposed algorithm exhibits superior performance over the baseline GTO-BSA.

Upon conducting experiments using benchmark datasets such as NSL-KDD, UNSW-NB15, and the CICIDS-2017, the findings reveal noteworthy results. In comparison to GTO-BSA, the IGTO-BSA algorithm consistently demonstrates improvements across key metrics, including accuracy, sensitivity, specificity, computational time, and the number of selected features. The step size parameters, encompassing alpha in GTO and beta and gamma in BSA, play a pivotal role in enhancing the algorithm's adaptability, enabling controlled adjustments and efficient exploration of the solution space.

Specifically, when analyzing the UNSW-NB15 dataset, the IGTO-BSA algorithm showcases significant enhancements, underlining its effectiveness in tackling feature selection challenges in IDS for IoT security. The step size parameters contribute not only to the algorithm's adaptability but also to its exploration efficiency, addressing issues related to premature convergence and local optima.

## REFERENCES


Abdollahzadeh, B., Soleimanian Gharehchopogh, F., & Mirjalili, S. (2021). Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *International Journal of Intelligent Systems, 36*(10), 5887–5958. https://doi.org/10.1002/int.22535

Al-Khassawneh, Y. A. (2023). An investigation of the Intrusion detection system for the NSL-KDD dataset using machine-learning algorithms. *IEEE International Conference on Electro Information Technology*, *2023-May*, 518–523. https://doi.org/10.1109/EIT57321.2023.10187360

Ali, W., & Saeed, F. (2023). Hybrid Filter and Genetic Algorithm-Based Feature Selection for Improving Cancer Classification in High-Dimensional Microarray Data. *Processes*, *11*(2). https://doi.org/10.3390/pr11020562

Ben-daya, M., & Bahroun, Z. (2020). *Internet of things and supply chain management : a literature review Internet of things and supply chain management : a literature review*. *November 2017*. https://doi.org/10.1080/00207543.2017.1402140

El-Kenawy, E.-S., & Eid, M. (2020). *Hybrid gray wolf and particle swarm optimization for feature* *selection*. *16*(3), 831–844. https://doi.org/10.24507/ijicic.16.03.831

Fathima, A., Khan, A., Uddin, M. F., Waris, M. M., Ahmad, S., Sanin, C., & Szczerbicki, E. (2023). Performance Evaluation and Comparative Analysis of Machine Learning Models on the UNSW-NB15 Dataset: A Contemporary Approach to Cyber Threat Detection. *Cybernetics and Systems*. https://doi.org/10.1080/01969722.2023.2296246

Gauthama Raman, M.R., Somu, N., Kirthivasan, K., Liscano, R., Shankar Sriram, V.S., (2017). An efficient intrusion detection system based on hypergraph - Genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowl.-Based Syst.* 134, 1–12. https://doi.org/10.1016/j.knosys.2017.07.005

Ghosh, T. K., & Das, S. (2021). A modified binary PSO algorithm for scheduling independent jobs in grid computing system. *International Journal of Next-Generation Computing, 7*(2), 144–154. https://doi.org/10.47164/ijngc.v7i2.211

Gupta, S., Pilani, S., Quamara, M., Energies, F. A., Commission, A. E., Chaudhary, P., & Aski, V. (2020). *Future IoT-enabled threats and vulnerabilities : State of the art , challenges , and future prospects*. *August*. https://doi.org/10.1002/dac.4443

Hassan, I. H., Mohammed, A., Ali, Y. S., Jeremiah, I., & Abdulraheem, S. A. (2023). Metaheuristic algorithms in text clustering. *Comprehensive Metaheuristics*, 131–152. https://doi.org/10.1016/B978-0-323-91781-0.00007-7

Heidari, A., & Jabraeil Jamali, M. A. (2023). Internet of Things intrusion detection systems: a comprehensive review and future directions. *Cluster Computing*, *26*(6), 3753–3780. https://doi.org/10.1007/S10586-022-03776-Z/METRICS

Hoz, E., Ortiz, A., Ortega, J., &, E. (2013)**.** Network anomaly classification by support vector classifiers ensemble and non-linear projection techniques. *In* J. Cabestany, I. Rojas, & G. Joya (Eds.), *Advances in Computational Intelligence. IWANN 2013. Lecture Notes in Computer Science* (Vol. 7902, pp. 103–104). Springer. https://doi.org/10.1007/978-3-642-40846-5_11

Isuwa, J., Abdullahi, M., & Abdulrahim, A. (2022). Hybrid particle swarm optimization with sequential one point flipping algorithm for feature selection. *July*, 1–18. https://doi.org/10.1002/cpe.7239

Isuwa, J., Abdullahi, M., Ali, Y. S., Kim, J., Hassan, I. H., & Buba, J. R. (2023). Optimizing Microarray Cancer Gene Selection using Swarm Intelligence : Recent Developments and An Exploratory Study Optimizing Microarray Cancer Gene Selection using Swarm Intelligence : Recent. *Egyptian Informatics Journal*.

Kareem, S. S., Mostafa, R. R., Hashim, F. A., & El-Bakry, H. M. (2022). An Effective Feature Selection Model Using Hybrid Metaheuristic Algorithms for IoT Intrusion Detection. *Sensors*, *22*(4), 1–23. https://doi.org/10.3390/s22041396

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, **4**, 1942–1948. IEEE. https://doi.org/10.1109/ICNN.1995.488968

Krsteski, S., Tashkovska, M., Sazdov, B., Radojichikj, L., Cholakoska, A., & Efnusheva, D. (2023). Intrusion Detection with Supervised and Unsupervised Learning Using PyCaret Over CICIDS 2017 Dataset. *Lecture Notes in Networks and Systems*, *724 LNNS*, 125–132. https://doi.org/10.1007/978-3-031-35314-7_12/COVER

Meng, X. B., Gao, X. Z., Lu, L., Liu, Y., & Zhang, H. (2016). A new bio-inspired optimization algorithm: Bird Swarm Algorithm. *Journal of Experimental & Theoretical Artificial Intelligence, 28*, 673–687. https://doi.org/10.1080/0952813X.2015.1042530

Miramontes, I., & Melin, P. (2023). A Comparative Study Between Bird Swarm Algorithm and Artificial Gorilla Troops Optimizer. *Studies in Computational Intelligence*, *1061*, 223–236. https://doi.org/10.1007/978-3-031-22042-5_13/COVER

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002

Montazeri, Z., Niknam, T., Aghaei, J., Malik, O. P., Dehghani, M., & Dhiman, G. (2023). Golf Optimization Algorithm: A New Game-Based Metaheuristic Algorithm and Its Application to Energy Commitment Problem Considering Resilience. *Biomimetics 2023, Vol. 8, Page 386*, *8*(5), 386. https://doi.org/10.3390/BIOMIMETICS8050386

Mishra, P., Varadharajan, V., Member, S., & Tupakula, U. (2019). A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection. *IEEE Communications Surveys & Tutorials*, *PP*(c), 1. https://doi.org/10.1109/COMST.2018.2847722

Mostafa, R. R., Gaheen, M. A., Abd ElAziz, M., Al-Betar, M. A., & Ewees, A. A. (2023). An improved gorilla troops optimizer for global optimization problems and feature selection. *Knowledge-Based Systems*, *269*, 110462. https://doi.org/10.1016/J.KNOSYS.2023.110462

Perwej, Y., Aboughaly, M. A., Ali, H., & Harb, M. (2019). An Extended Review on Internet of Things ( IoT ) and Its Promising Applications An Extended Review on Internet of Things ( IoT ) . *February*. https://doi.org/10.5120/cae2019652812

Sallam, Y. F., El-Nabi, S. A., El-Shafai, W., Ahmed, H. E. H., Saleeb, A., El-Bahnasawy, N. A., & El-Samie, F. E. A. (2023). Efficient implementation of image representation, visual geometry group with 19 layers and residual network with 152 layers for intrusion detection from UNSW-NB15 dataset. *Security and Privacy*, *6*(5), e300. https://doi.org/10.1002/SPY2.300

Stea, G. (2022). A survey of blockchain-based IoT eHealthcare : Applications , research issues , and challenges A Survey of Blockchain-Based IoT eHealthcare *: Applications , Research Issues , and Challenges*. *August*. https://doi.org/10.1016/j.iot.2022.100551

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper Optimisation Algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

Sarker, I. H., Khan, A. I., Abushark, Y. B., & Alsolami, F. (2023). Internet of Things (IoT) Security Intelligence: A Comprehensive Overview, Machine Learning Solutions and Research Directions. *Mobile Networks and Applications*, *28*(1), 296–312. https://doi.org/10.1007/S11036-022-01937-3/METRICS

Swarna Priya, R. M., Maddikunta, P. K. R., Parimala, M., Koppu, S., Gadekallu, T. R., Chowdhary, C. L., & Alazab, M. (2020). An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. *Computer Communications*, 160, 139–149. https://doi.org/10.1016/j.comcom.2020.06.011

Xiao, Y., Sun, X., Guo, Y., Li, S., Zhang, Y., & Wang, Y. (2022). An Improved Gorilla Troops Optimizer Based on Lens Opposition-Based Learning and Adaptive β-Hill Climbing for Global Optimization. *CMES - Computer Modeling in Engineering and Sciences*, *130*(3). https://doi.org/10.32604/cmes.2022.019198

Xu, H., Sun, Z., Cao, Y., & Bilal, H. (2023). A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Computing*, *27*(19), 14469–14481. https://doi.org/10.1007/S00500-023-09037-4/METRICS