International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2022/214503 Volume 9, Issue 4, July – August (2022)

REVIEW ARTICLE

Investigations on Bio-Inspired Algorithm for Network Intrusion Detection – A Review

Jeyavim Sherin R C
School of Computer Science and Engineering, VIT University, Chennai, Tamil Nadu, India
jeyavimsherin.rc2021@vitstudent.ac.in

Parkavi K
School of Computer Science and Engineering, VIT University, Chennai, Tamil Nadu, India
parkavi.k@vit.ac.in

Received: 16 June 2022 / Revised: 31 July 2022 / Accepted: 08 August 2022 / Published: 30 August 2022

Abstract – A network is a collection of interconnected devices that can share information and resources, exchange files, and enable electronic communications. IDS is an important part of Network Security to secure a network. An Intrusion Detection System (IDS) is a fundamental building block in network security. A wide variety of techniques have been proposed and implemented to improve the performance and accuracy of intrusion detection models. It is used by many MNC companies such as Wipro, TCS, and L&T and they are having their IDS in the organization system. CNN (Convolutional Neural Network, Machine Learning, Data mining, Deep learning models such as SVM (Support Vector Machine) are performing well above the benchmark to prevent the systems from all kinds of attacks. Recently, bio-inspired optimization algorithms are metaheuristics that mimic the nature of solving optimization problems. Bio-inspired algorithms are gaining the moment that brings a revolution in computer science. This paper investigates the feature selection techniques of bio-inspired algorithms-driven Intrusion Detection Systems. This paper categorises these SI approaches based on their applicability in improving various aspects of an intrusion detection process. Furthermore, the paper discusses the capabilities and characteristics of various datasets used in experimentation. The main goal is to assist researchers in evaluating the capabilities and limitations of SI algorithms in identifying security threats and challenges in designing and implementing an IDS for the detection of cyber-attacks across multiple domains. The survey identifies existing issues and provides recommendations for how to effectively address them.

Index Terms – IDS, Deep Learning, Optimization, Classification, Feature Selection, Bio-Inspired Algorithm.

1. INTRODUCTION

In the last decade, ensuring the security of individual and corporate digital devices is a difficult task, predominantly in the age of IoT, the quantity of connected devices is expanding at exponential rates. In cyber-physical system (CPS) world, adversaries use sophisticated algorithms that affect the triad: Confidentiality (C), Integrity (I), and Availability (A) (CIA), It is critical that robust, automated detection and classification of malicious sophisticated intruders of CPS's information security. An IDS primarily analyses the network traffic of inbound and outbound to detect malicious activity and take corrective action against it.

The increase of IoT in CPSs has transformed the perspective on communication. The IoT reduces human intervention by enabling the CPSs to handle, store and forward the required data. IoT devices have resource constrain on their storage, memory and power that have proportionally increased the attacks.

Intrusion problems are inevitable in an ever-expanding network. Attackers can easily attack the system, but set up an intrusion detection system to address cyber security challenges and use an optimized model to quickly detect attacks [1] [2]. Traditional intrusion detection and prevention systems like firewalls, mechanisms to the authority of access, and encrypted their own limitations to completely protect any networks and systems from increasingly complicated and intricate attacks like DoS. Moreover, most of the systems built based on the basis of techniques from high FP and FN detection rates and it lacks adapting the continuously changing attack behaviors [3].

A decade of adapting several ML and DL models have adopted to the different challenges of IDS with the positive trend of bring improves detection rates and adaptability. These methods are frequently employed to keep attack information bases current and thorough.

1.1. Overview of Intrusion Detection System (IDS)

IDS is a crucial component of cyber-security. It is the network that ensures missile safety and prevents enemy missiles from being fired. Attacks could lead to data breaches or system outages. Malicious behaviours, activities, and violations of network host security policies are common in IDSs [2]. IDS is

REVIEW ARTICLE

implemented as a software implementation or specific machine that constantly monitors network traffic to detect malicious material or activity [4]. Many organizations use IDS.

Unauthorized access to a system is known as an intruder (Hackers). Intruders could be external to the network or legitimate network users. Attempting to break into or misuse the system is known as an intrusion. We should have IDS to prevent all of these scenarios.

Hackers and intruders have many successes in trying to destroy well-known corporate networks and services. In recent years, computer attacks have increased from 2009 to 2022 in Figure 1.

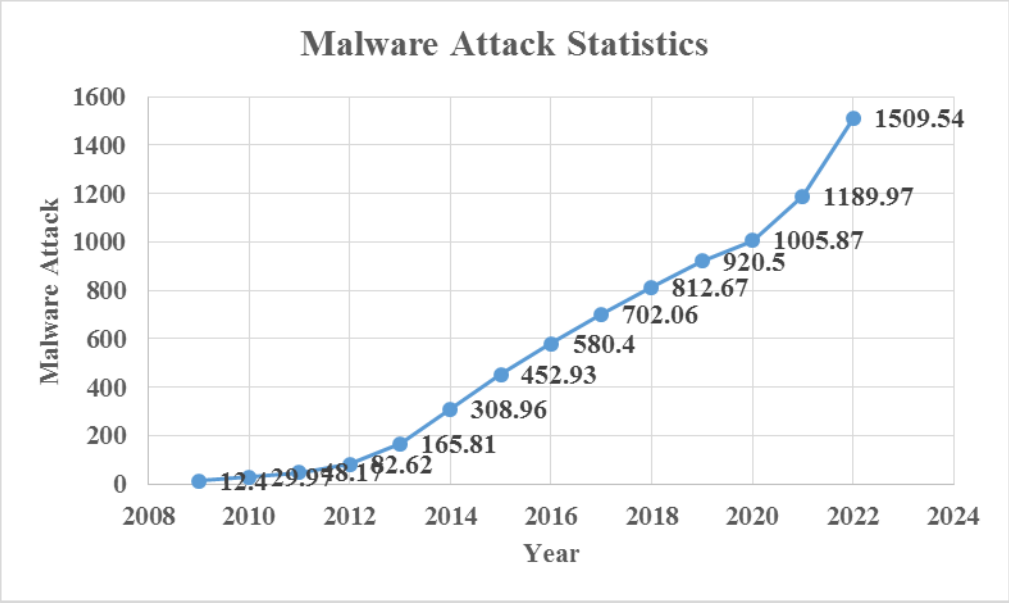

Figure 1 Growth Rate of Malware Attack

1.2. Types of IDS

1. Network-Based Intrusion Detection Systems (NIDS)
2. Host-Based Intrusion Detection Systems (HIDS)

NIDS is basically present in the hardware systems it has communication networks like switches, routers and gateways etc. The NIDS analyse data packets to find any such patterns it treats as a threat for various actions like notifying the administrators or block the source IP addresses [2] [4]. In a communication network, the HIDS are software components loaded on host-based machines. A HIDS is similar to NIDS, it is the potential in monitoring the computing system internals and analysing the network packets.

IDS detection has four categories.

• **Signature-Based Detection:** It matches pre-installed attack signatures to detect intrusions in networks. This approach detects known assaults simply and effectively, have a high rate of detection accuracy and low false positive incidences.

• **Anomaly-Based Detection:** It classifies intruder behaviour when it detects a divergence from regular network behaviour. This strategy can be used to track down unknown attackers. It generates a high number of FPs.

• **Specification-Based Detection:** It keeps track of how network traffic behaves in particular situations. When a deviation from the norms is observed, it is presumed to be intruder behaviour.

• **Hybrid-Based Detection:** It has the combined benefits of anomaly-based and signature-based detection.

2. NATURE-BASED TECNOLOGY

It presents tremendously diverse, dynamic, robust, sophisticated, and fascinating phenomena, nature provides an immense and great source of originality for addressing complex and challenging complications of computer science [5]. Nature-inspired algorithms are meta heuristics to quickly solve optimization issues that resemble that of nature, ushering in a new age in computation [6]. Bio-inspired algorithms will usher in a new era in computer science. This needs collaboration among academics from diverse domains, such as computer science, ecology, AI, environmental science and biology, to acquire a larger and more in-depth view and analysis of each micro-level step/interaction, resulting in

REVIEW ARTICLE

considerably more significant and exceptional outcomes. Environmental-based technologies are one of the most effective optimization algorithms with the ability to make a significant difference.

Traditional problem-solving methodologies include exact approaches (logical, mathematical programming) and heuristics. When addressing difficult and complex optimization problems, the heuristic technique appears to be superior, particularly when standard methods fail. Many biological processes can be viewed as restricted optimization strategies. BIAs are heuristics that mimic/imitate nature's method. They make a lot of random decisions, so they're a subset of randomized algorithms [6]. The process design depends on the problem, fitness function evaluation, and creation of solutions.

Numerous papers have lately been published on the success of bio-inspired strategies for tackling tough issues in different computer science fields, and increasing trends in the literature on bio-inspired attempts to address a wide range of problems. The two most popular and successful BIA classes or directions are EA and SBA, both of which are influenced by natural evolution and animal collective behaviour. This will be enhanced further so that the algorithms can be categorized according to the area of natural inspiration, providing a more thorough perspective of the subject [7]. The purpose is to review BIAs, as well as information on taxonomy and application areas. Figure 2 depicts the BIO algorithms defined by the area of inventiveness. The bio inspired algorithm is classified into three types of inspiration: evolution, swarm based, and ecology.

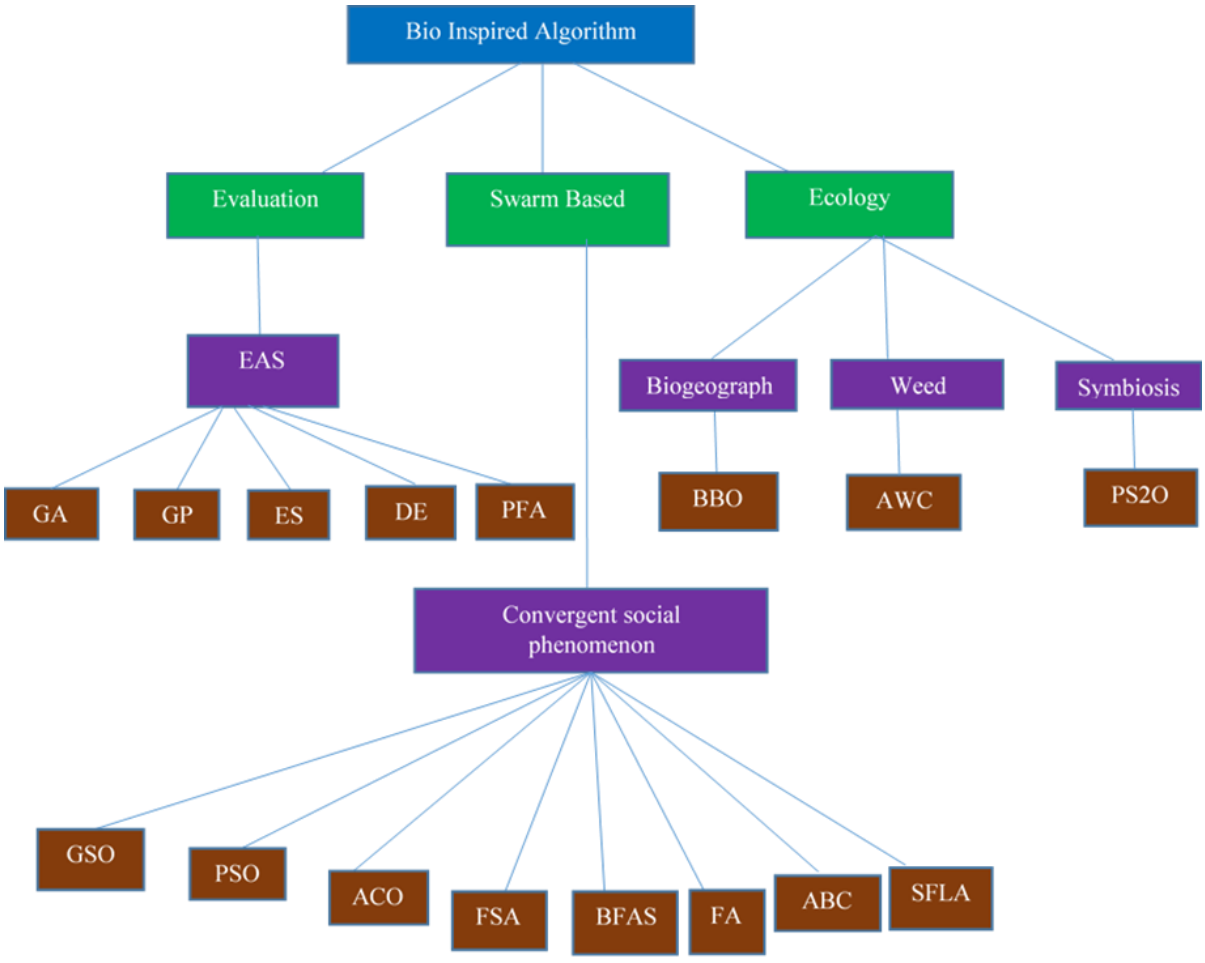

Figure 2 Inspiration Based Bio Inspired Optimization Algorithms

2.1. Evolutionary Algorithms (EA)

Evolutionary computation (EC) is an intellectual method that incorporates iterative progress to profit from many group phenomena in adaptable occupants of issue solvers, such as evolution, growth rate, selection, reproduction, and survival as seen in an inhabitant. EAs are well-known algorithms [3]. Built of this algorithm is on the basis on natural evolution and formation of all living beings that exist in the world, as well as adopting different strategies used to interact. EAs are

REVIEW ARTICLE

designed based on this powerful philosophy to address challenging issues.

EAs include the genetic algorithm (GA), genetic programming (GP), evolutionary strategy (ES), differentiated evolution, and the recently developed Paddy Field Algorithm. Many features are shared by the EA family. All of them are survival-based population-based stochastic search methods. Each algorithm starts iteratively first with a feasible solution population and it matures from generation after generation until the best-discovered solution. Within the population of solutions, fitness-based choosing occurs in consecutive iterations of the EA algorithm [6]. The learning is prioritised based on better solutions that are carried to the next generation. Figure 3 shows the flow diagram of EA.

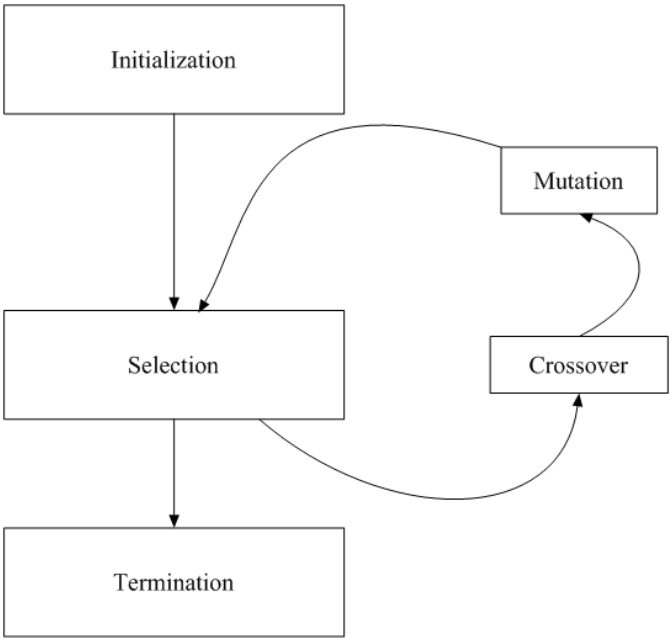

Figure 3 Flow Diagram of EA

2.2. Swarm Based Algorithm (SBA)

Swarm intelligence is another name for collective intelligence. Scientists were studying the behaviour of social insects because of its ability to solve complex challenges like determining the quickest path between the nests and source of food. Despite their lack of individual education, these insects work miracles as a swarm by communicating with one another with their environment. Numerical optimization technique has been utilised to simulate the behaviour of diverse swarms used in hunting or mating [6]. This section summarises and lists the stages involved in the operation of eight different swarm intelligence-based algorithms. An ant colony optimizer, a particle swarm optimizer, an artificial bee colony algorithm, a glow worm algorithm, a firefly algorithm, a cuckoo search strategy, a bat algorithm, and a hunting search algorithm are some of the tactics that were used [8] [9]. The performance of all eight methods is compared as they solve two optimization tasks from the literature. The swarm intelligence-based are simple algorithms and reliable ways of quickly determining the optimum answer to optimization issues without a lot of math.

SI is targeted to resolve complex issues using computational intelligence methods. Swarm Intelligence is the collective study of how individuals in a population interchange with one another on a local level. Nature is a great source of inspiration when we focus on biological systems. There is the nonexistence of a centralised control structure to predict individual agent behaviour rather the agents follow simple rules. The irregular iteration of a certain extent between the agencies results in the "smart" behaviour that individual agents [10].

The population-based technique of Particle Swarm Based Optimization was inspired by bird flocks and fish schools. PSO is a computational technique that is quite similar to evolutionary computation. The system gives random solutions in a population to choose from, and the best one is found after generations of updates [11].

PSO [12] is a type of evolutionary computation that is the same as of genetic algorithm (GA). Iteratively updating generations to identify optimal solutions, the system starts with a permutation population of solutions. The PSO, unlike the GA, lacks evolution [13] operators like crossover and modification. Instead, PSO relies on societal and individual behavioural characteristics as well as particle velocity and position. Particles, or prospective solutions, move through the issue search space, following the most promising portion at any given time. The PSO algorithm is widely used in different applications from attack detection, classification, and clustering.

The birds are represented as particles which are persistently searching for the optimised global known position (G_{best}). The best position of the local particle (P_{best}) is used as a guide to finding the G_{best}. If a particle finds another best position, its P_{best} position is updated. As a result of each particle's collaboration mechanism, the Gbest value is obtained by optimising the value. In a D-dimensional target space with N uniformly distributed particles. The position and velocity of the ith particle are denoted by Xi and Vi, respectively as in equation 1.

Velocity $V_i^{k+1} = \omega V_i^k + c_1.rand().(P_{best} - X_i^k) + c_2.rand().(G_{best} - X_i^k)$ (1)

Position, as in equation 2. $X_i^{k+1} = X_i^k + V_i^k$ (2)

Where $c_1$and $c_2$represent the coefficients of social and cognitive acceleration, and ω is the weight of inertia. It's also appealing because there are only a few frameworks to

REVIEW ARTICLE

configure. One version works well in a wide range of implementations with small adjustments. Particle swarm optimization has been utilised in a number of techniques for a variety of issues as well as for a variety of applications that focus on specific needs [14]. An IDS must be carefully designed in order to overcome PSO limitations such as delayed integration and aggregation to local minima or sub-optimal values.

Steps:

1: Calculate the total number of fragments and their initial weights.

2: Start the swarm

3: Created as a particle, the seed vector

4: Generate the remaining particles at random.

5: Generate initial velocities at random.

6: while (iterations lesser than maximum iterations) and (Improvement greater than minimum improvement)

7: Determine the strength of individual particles.

8: Improve the global best fitness of each particle.

9: Do this for each particle

10: Determine the location and the most fit particle.

11: Find the greatest fitness in the globe and your location.

12: Change the particle's speed.

13: Set the position of the particle.

14: Find the world's top fitness destination.

Figure 4 shows the flow diagram of PSO.

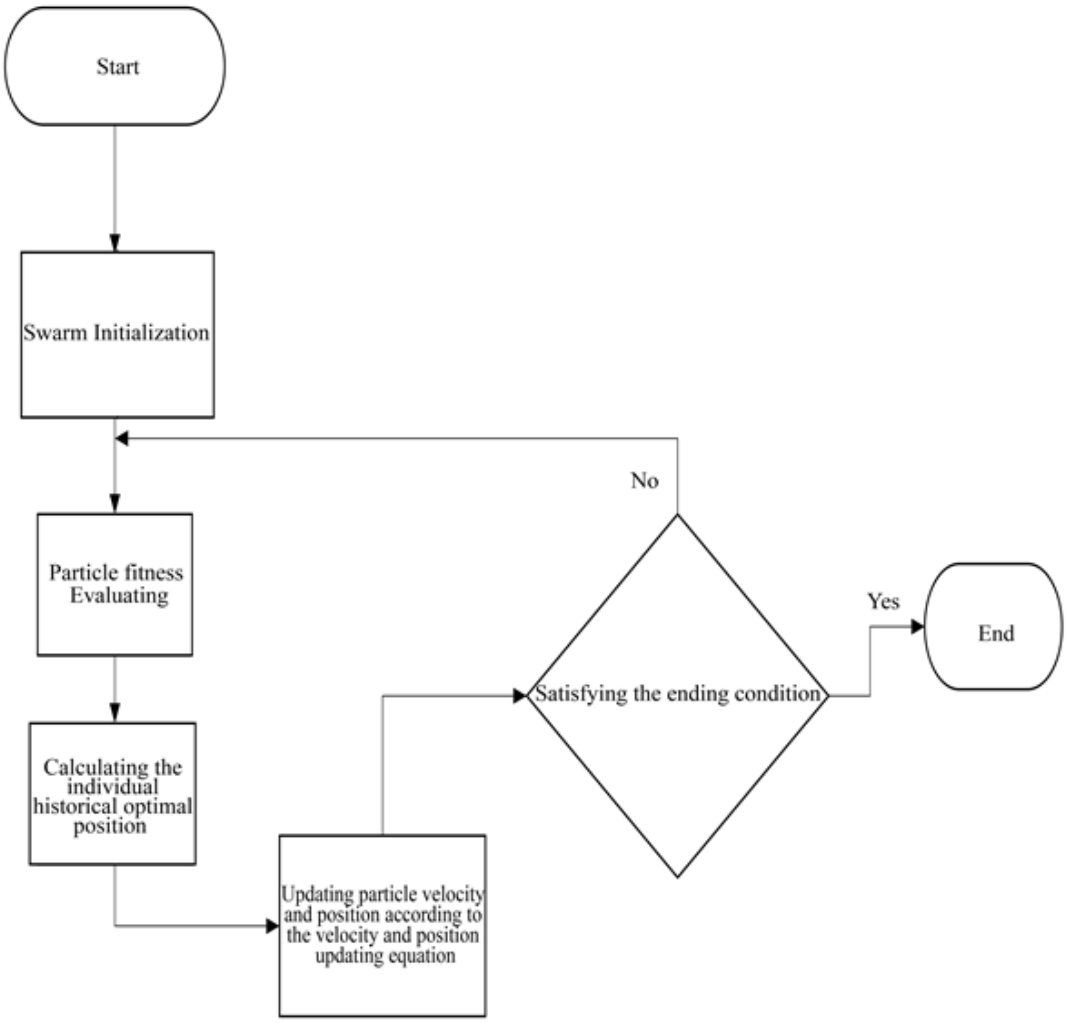

Figure 4 Flow Diagram of PSO

REVIEW ARTICLE

In this article, we list the top vulnerabilities in IoT [1] [15] that allow hackers to hack IoT systems.

- Interfaces lacking security
- A lack of up-to-date security mechanisms
- Components with obsolete configurations
- Lacking privacy protection process
- In communication there is no encryption method
- Lack of security Environment

2.3. Artificial Bee Colony (ABC)

Karaboga created the ABC optimization technique in 2005. The honeybee's communal behavior is commonly used in numerical problem optimization [16]. ABC's operational model is made up of three essential factors. First, is the food source quality in this case nectar, which is determined by a variety of profitable factors like nest distance, quantity in richness, and removal effort. The foragers are next on the list also called as worker bees, who are tasked with bringing details of the place, and food sources availability to the nest. Behaviour of a swarm of bees in solving uni-dimensional, multi-dimensional, and other dimensional problems. The problems in while classifying the attacks, choosing the features and finding out the attacks [17]. ABC optimization includes the following phases in Figure 5.

1. Phase of Initial population
2. Phase of Working Bees
3. Phase of Observer Bees
4. Phase of Lookout Bees

The ABC algorithm is also useful to solve combinative inflation issues. However, it was initially proposed for the numerical optimization.

Figure 5 Flow Diagram of ABC Optimization Algorithm

REVIEW ARTICLE

2.4. Bat Algorithm (BA)

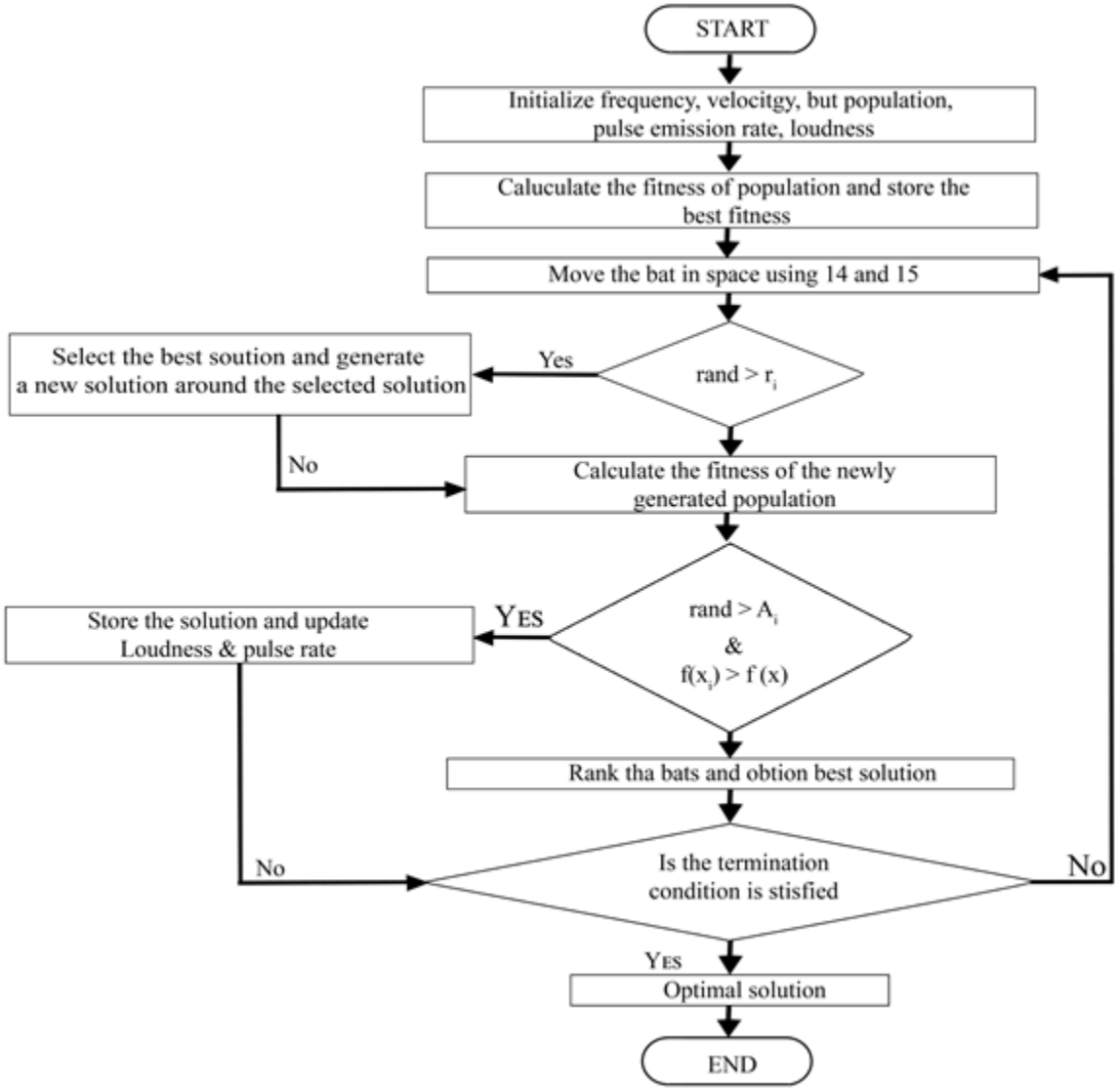

Figure 6 Flow Diagram of Bat Algorithm

BA algorithm uses echolocation behaviour of bat to perform global optimization. Several bat-inspired algorithms can be proposed [18]. The idealized rules are:

1. All bats sense the distance using echolocation, and they "determine" the difference between their prey and its background barriers in some unknown way.

2. To find prey, bats randomly fly with velocity (V_i) at position (x_i) with a fixed frequency, varying wavelength, and loudness (A_0). Depends mainly on the proximity of their target, bats can adjust the emitted pulses wavelength also their rate of pulse emission r [0,1].

3. Although the loudness can vary in different ways, we assume it ranges from a large (positive) A_0 to a small constant value A_{min}.

Major aspects of the BA model, including initialization, velocity, frequency, and position of the BA, as well as loudness and pulse rate, are determined by the reflections. The movement of the bat in the probe space was discovered during the setup phase. The bat fitness determines the distinction of food. The initial bat population is given by in equation 3.

$$X_{i,j=}X_{min,j} + rand * (X_{min,j} - X_{max,j}) \quad (3)$$

where i = 1, 2, 3 . . . n and j = 1, 2, 3. . . d. Moreover, and Dimension j's lower boundaries and upper boundaries are

REVIEW ARTICLE

represented by $X_{min,j}$ and $X_{max,j}$, respectively. In this process, the frequency controls the velocity and step size to generate new solutions, which in turn updates the velocity and location. The bat's frequency f_i, movement X_i, and velocity V_i are given by equation 4, 5 and 6.

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (4)$$

$$V_i^t = V_i^{t-1} + (X_i^t - X_*)f_i \quad (5)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (6)$$

Figure 6 shows the flow diagram of Bat Algorithm.

2.5. Cuckoo Search Algorithm (CSA)

The algorithm's primary learning comes from breeding and egg-laying behavior. This method uses adults and their egg-laying behaviour on other birds' nests. The host birds will not destroy the cuckoo egg, and eventually it will hatch into a mature cuckoo. Bird populations migrate and are influenced by environmental factors to converge and find the best place to reproduce and breed [19]. For the objective function, this is the ideal place. Cuckoo behaviour is the basis for CSA's unique continuous metaheuristic. Like the other population-based optimization algorithms, it starts with a cuckoo population set [20]. In CSA, a random set of likely solutions that are thought to represent the habitat is generated. The cuckoo's movements are modelled using Levy flights, which are provided by a stochastic mathematical model. Three idealized rules underpin the CS algorithm:

- A cuckoo lays one egg at a time and places it in nest that is chosen randomly.
- The best choice of nests with the highest quality eggs is passed to the next generation as a solution.
- A host has a chance of finding an alien egg with a probability of pa $\varepsilon[0,1]$ and there is an infinite number of possible host nests. In this situation, the host bird has two options: either dump the egg or leave the nest and construct a new nest somewhere else (Yang 2009).

A Lèvy flight is executed by generating the new solutions x(t+1) for a cuckoo as in equation 7.

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus \text{Lèvy}(\lambda) \quad (7)$$

Where $\alpha > 0$ is the step size that should be related to the scale of the problem of interest, and $\alpha = 1$ can be used in most cases. This entry-wise product is comparable to those used in PSO, but because the random walk via Lévy flight takes many longer steps, it is ultimately more effective at exploring the search space. A random walk is essentially what the Lévy flight gives, with the random step duration coming from a Lévy distribution.

2.6. Flamingo Search Algorithm (FSA)

The migratory birds, flamingos are gregarious. Algae, shrimp, worms, and insect larvae are their primary sources of nutrition. They typically wander around with their bent beaks contacting the water's surface while bending their necks down and turning their heads over as they feed [21]. Foraging and migration are their two primary behavioural characteristics: The model is primary optimization by using the following ways,

1. Communicate by calls for their location as well as the food availability.
2. The flamingo population no longer knows where the maximum amount of food availability in a search area. Instead, we will do our best to replace each flamingo's environment (which suffers from foraging and locomotion behaviour) and discover a food environment that is plenty of already familiar food in the region of search [21]. Flamingo's behaviour of comply into the concept of swarm intelligence algorithm optimization. In short, find the most effective answer in the world in the positive explore capacity [22]. While the search agent is doing the search, it is not possible to understand what the current international maximum response to the search area is. The search agent is a flamingo within the FSA, which discovers search areas and improves through statistics that change between all guidelines from neighborhood movements and glued guidelines that are the biggest answer in the long run.
3. The flamingos has behaviour this is foraging behaviour and migration behaviour.

The equation for updating flamingo foraging behaviour is as follows (equation 8):

$$x_{ij}^{(t+1)} = (x_{ij}^t + \varepsilon_1 \times xb_j^t + G2 \times |G1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t|)/K \quad (8)$$

In (8), $x_{ij}^{(t+1)}$ indicates where the ith flamingo is in the population's jth dimension in the (t+1)th iteration, x_{ij}^t denotes the position of the ith flamingo in the jth dimension in the t iteration of the flamingo population. $G1 = N(0,1)$ and $G2 = N(0,1)$ are arbitrary numbers. ε_1 and ε_2 are randomized by -1 or 1.

2.6.1. Migratory Behaviour

The population moves to a different region where food is plentiful when there is a food shortage in the current foraging area. The formula for flamingo population migration is given below, assuming that the position of the food-rich area in the jth dimension is xbj as in equation 9.

$$x_{ij}^{t+1} = x_{ij}^t + \omega \times (xb_j^t - x_{ij}^t) \quad (9)$$

REVIEW ARTICLE

In equation 9, $x_{ij}{}^{t+1}$ indicates where the ith flamingo is in the population's jth dimension in the iteration starting at time t+1, $\omega = N(0, n)$ is an n-dimensional Gaussian random number that is used to increase the search space and replicate the randomness of individual flamingo behaviour throughout a particular migration phase in Figure 7.

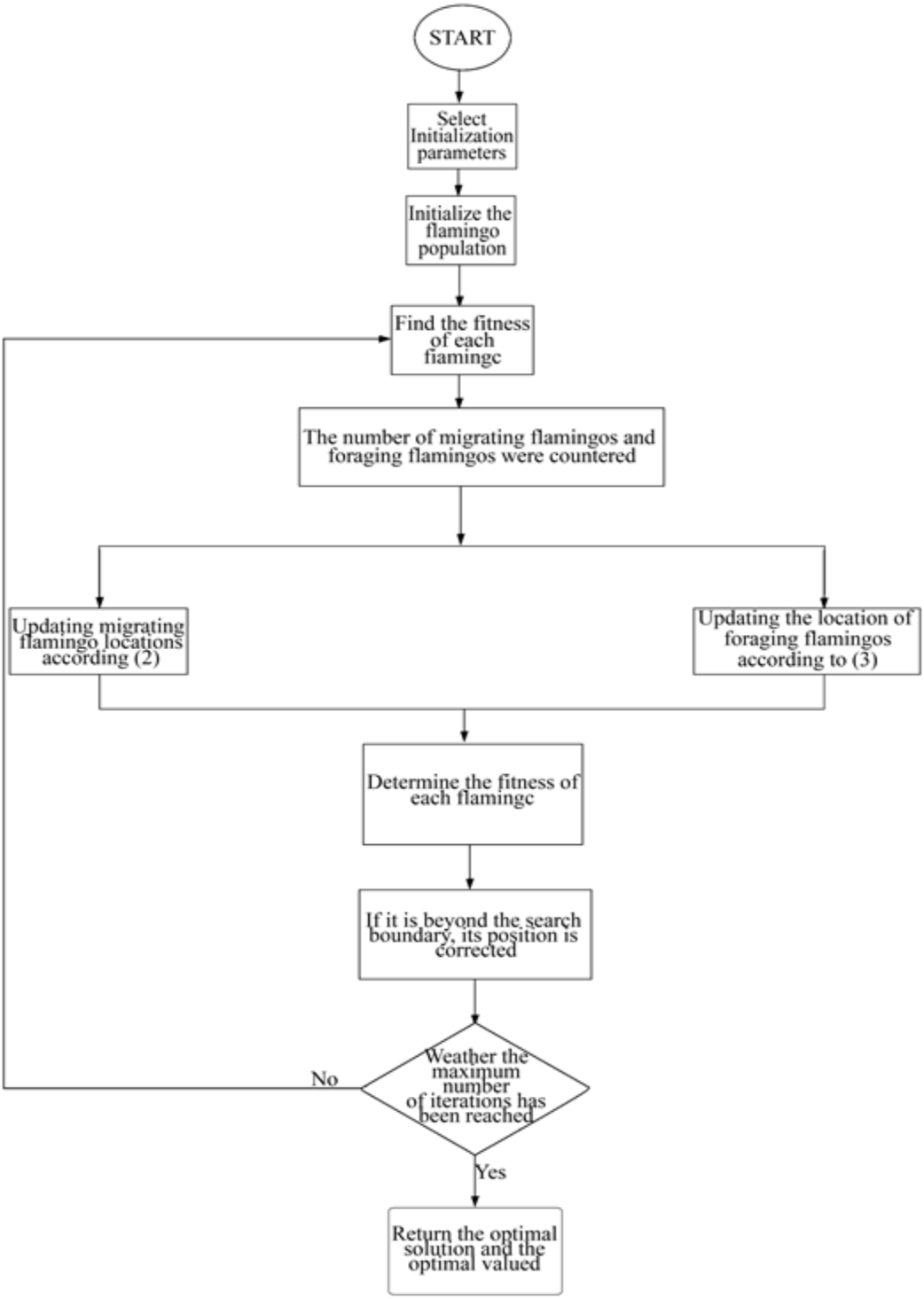

Figure 7 Diagram of Flamingo Search Algorithm

REVIEW ARTICLE

2.7. SFLA (Shuffled Frog-Leaping Algorithm)

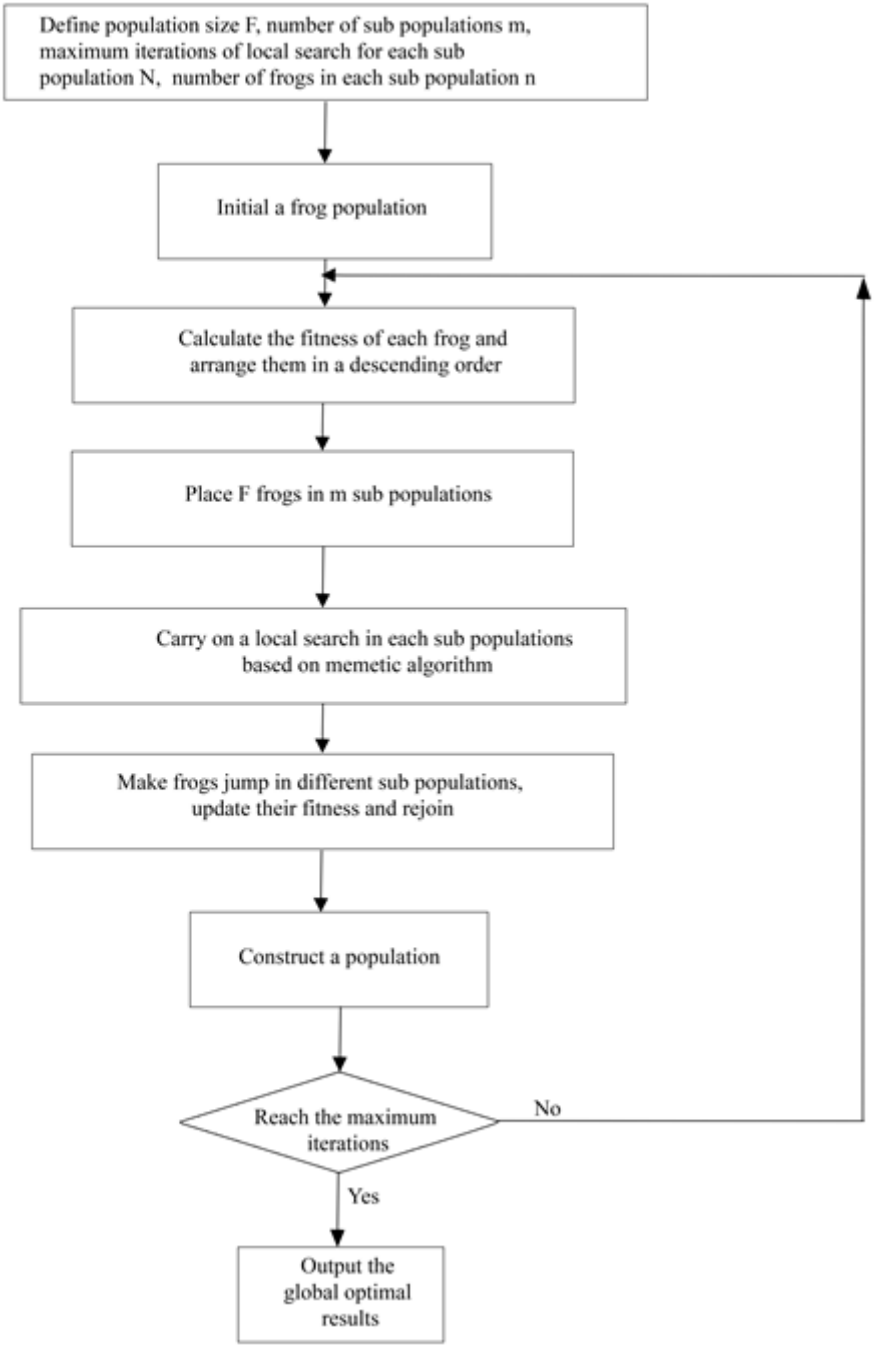

Figure 8 Flow Diagram of SFLA Algorithm

SFLA is one of the most innovative optimization algorithms inspired by the natural social behaviour of frogs, and it is classified as a behavioural algorithm or a memetic algorithm. The frog jump optimization algorithms are also known as the frog jump algorithm, frog leaping algorithm, and SFLA algorithm. The SFLA algorithm is a metaheuristic memetics-based algorithm. The population-based memetic algorithm is employed to resolve challenging and significant optimization issues. The main idea behind this algorithm is to improve the aqueous presentation of the search intensification process by

REVIEW ARTICLE

using a local search method within the structure of the genetic algorithm. The memetic algorithm encrypts the sum of the initial answers before calculating the utility of each response based on a fitness function and generating new solutions.

The SFLA algorithm is based on how frogs find food. This algorithm employs the nomometric method to search locally among frog subgroups. The frog hybrid jump algorithm employs the hybrid strategy and allows for message exchange in local search. The advantages of the Nomometric algorithm and particle group optimization are combined in this algorithm. Words are exchanged in the frog hybrid jump algorithm not only in local search but also in global search. Thus, this algorithm effectively combines local and global searches. The frog hybrid jump algorithm is simple to implement and highly searchable. Many nonlinear, undetectable, and multi-state problems can be solved using the frog hybrid jump algorithm.

Figure 8 shows the flow diagram of SFLA Algorithm.

The following steps summarise the SFLA algorithm's meta-exploration strategy, which is divided into two main stages: global exploration and local examination.

Step 1: Initialize

Select M, N, where M represents the number of memeplexes and n represents the number of frogs in each memeplex, and $f = m * n$ represents the total population size of the pond.

Step 2: Create a virtual population.

From the available space, select F virtual frogs U (1), U (2), ... U(F). Determine the competency value of f(i) for each U(i).

$U(i) = (Ui1, Ui2, \ldots Uid)$. Also, d is the number of decision variables.

Step 3: Sorting and grading the frogs

Store the frogs in the array X = {U(i), f(i) & i=1..., F} in descending order according to their merits. Take note of the best Px frog in the population. (U = Px(1)).

Step 4: Split the frogs up into memeplexes.

Divide array X in Y that each has n frogs.

Step 5: Evolution of memetics within each memeplex

Each Yk memeplex contains (k = 1, 2, 3… m)

Step 6: Connect memeplexes.

After a certain number of memetic evolutions have occurred in each memeplex, place the memeplexes (Y1,..., Ym) in X, so that the relation X = Y(k), k = 1, 2,..., m is established. Then, update the population's best position (Px).

Step 7: Convergence Analysis

If the convergence conditions are met, stop. If not, proceed to the fourth step of the global search.

2.8. Glowworm Optimization Algorithm (GSO)

The GSO algorithm simulates glowworm's (GS) aggregation and luminosity to perform optimization. Individual GS are drawn together and aggregation occurs primarily due to their own brightness (fluorescein value). The following are the steps:

Step 1: Set the glowworm position $Z = \{z1, z2, \ldots, zn\}$, as well as the moving step length, fluorescein starting value l_0, GS number n, as well as other relevant parameters.

Step 2: Determine the GS fitness based on the objective function, i.e., fluorescein value li(t) as in equation 10.

$$l_i(t) = (1 - p)l_i(t - 1) + \gamma J(x_i(t)) \quad (10)$$

$J(x_i(t))$ That stands for the conversion position to fluorescein values, p for fluorescein velocity, and γ for fluorescein enhancement ratio.

Step 3: Probability Determination (equation 11).

$$P_{ij(t)} = \frac{l_j(t) - l_i(t)}{\sum_{k \epsilon N_{i(t)}} l_k(t) - l_i(t)} \quad (11)$$

$N_{i(t)}$ It denotes the set of glow-worms.

Step 4: Each glow worms position to be updated in equation 12

$$x_i(t + 1) = x_i(t) + s \times \frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \quad (12)$$

Step 5: Glow worms radius to be updated in equation 13

$$r_d i(t + 1) = min\{r_s, max\{0, r_d i(t) + \beta(n_t - |N_i(t)|)\}\} \quad (13)$$

$r_d i(t)$ Denotes the perceptual realm of glow worm.

Step 6: End

2.9. Firefly Algorithm (FA)

Firefly Algorithm (FA) was proposed by yang in 2007, it is based on the firefly's behaviours and flashing behaviours patterns. The FA basically have idealised three rules:

1. Fireflies are naturally unisexual, firefly attracted to one another irrespective of gender.
2. As the distance between two objects increases, both the attractiveness and the brightness diminish. Thus, if there are two flashing fireflies, the less brilliant one will go in the direction of the more brilliant one. It will move randomly if there isn't another firefly that is brighter than it.

REVIEW ARTICLE

3. The brightness of a firefly is determined by the geography of the objective function.

The attraction of a firefly I to another, more attractive (brighter) firefly j is determined by in equation 14.

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma r^2}{}_{ij}(x_j(t) - x_i(t) + \alpha\varepsilon_i{}^t \quad (14)$$

Where β_0 is the attractiveness at r = 0, and the attraction is the reason for the second term, randomization is the third concept, $\varepsilon_i{}^t$ is a vector of random numbers drawn at time t from a Gaussian or unchanging distribution. If β_0= 0, it is simply a random walk. In addition, the randomization $\varepsilon_i{}^t$ it extends readily to other distributions, such as Lévy flights. The Algorithms are shown in Table 1.

Table 1 Nature-Inspired Optimization Algorithms

Year	Algorithms	Advantages	Disadvantages
2021	Horse herd optimization algorithm (HOA)	HOA useful for the feature selection for high dimensional dataset [23].	Do not fit in the herd because they are too low in rankings.
2020	Chimp Optimization Algorithm (COA)	The main purpose of this algorithm is to maximize production efficiency while minimising production costs [24].	Slow convergence rate.
2020	Mayfly Optimization Algorithm (MOA)	Major uses in Swarm intelligence and evolutionary algorithms [25].	Sensitivity to the problem initial guess.
2020	Coronavirus Optimization Algorithm (COA)	In this optimization algorithm is to selecting features without affecting the system performance [16].	It can fall into sub-optimal solutions.
2020	Black Widow Optimization Algorithm (BWO)	When compared to other optimized algorithms it can be achieving fitness value and early detection [14].	Group signature-based approaches is avoiding
2020	Water strider algorithm (WSA)	Can find the optimal answers with a high accuracy [26].	Limited to adapt in different requirements
2017	Rain-fall Optimization Algorithm (ROA)	To solve the drilling problem and it is able to find the answers very quick & accurately [27].	The algorithm is not applicable for different scenarios
2016	Dragonfly algorithm (DFA)	Able to optimize in different real-world problems [28].	Abundant of search area and disruption of random flights.
2015	Ant Lion Optimizer (ALO)	It's simple, scalable, and adaptable, and it'll boost the intrusion detection system's accuracy and real-time performance efficiency [29].	The optimal solution arrived with less accuracy
2015	Water Wave Optimization (WAO)	This algorithm uses refraction, propagation and breaking a high-dimensional space of the problem. It is an evolutionary algorithm [26].	Low calculation Accuracy
2010	Bat Algorithm (BA)	Simplicity & flexibility [27].	• Convergence rate is slow. • Optimization accuracy is not high.

REVIEW ARTICLE

2007	Firefly Algorithm (FA)	Problem-solving for continuous-discrete optimization [9].	• Slow convergence speed. • Computational complexity
2009	Glowworm Swarm Optimization (GSO)	Capability for resolving multimodal problems [8].	• Low calculation Accuracy. • Weakness to locate the global optimum solutions.
2009	Cuckoo Search Algorithm (CSA)	Easier application and fewer tuning parameters [19].	• Initialization • Parameter • Boundary Issues
2009	Bee Colony Optimization (BCO)	Simplicity and proper exploration ability [17].	The action course is stochastic; single agent behaviour is random and noisy.
2008	Fast Bacterial Swarming Algorithm (FSA)	Bacterium with best fitness value [28].	Limited adaptation to wide application
2007	Imperialistic Competitive Algorithm (ICA)	A nonlinear equation is solved with the imperialist competitive algorithm (ICA) [7].	Cannot solve linear problems
2008	Roach Infestation Optimization (RIO)	Improves algorithm effectiveness in finding global optima [14].	It requires many iterations to solve complex problems
2006	Cat Swarm Optimization (CSO)	To solve the multiple conflicting objects [30].	Low convergence speed
2003	Queen-Bee Evolution (QBE)	Enhance local search while retaining the global capabilities of GA [31].	• Inadequate use of secondary data. • When sequential processing is used, it is slow.
2001	Harmony Search Algorithm (HAS)	• Easy implementation. • Quick convergence [32].	• Less adjustable parameters.
1995	Particle Swarm Optimization (PSO)	• Easily Implemented. • Computation efficiency is higher when compared to mathematical algorithms [10].	• Low convergence rate. • Low-quality solution
1992	Genetic Programming (GP)	• Data and text classification. • Ensuring network security. • Supporting other machine	• Computational cost is high • Solution is not guaranteed to coverage

REVIEW ARTICLE

		learning methods [33].	to the global maxima/minima.
1992	Ant Colony Optimization (ACO)	• Solving computational problems by probabilistic approach [17].	• Theoretical analysis is challenging. • Sequences of random decisions. • Probability distribution changes by iteration
1989	Tabu Search (TS)	• Mathematical Optimization [34]. • Optimize a multi-parameter model that can yield exceptional results [35].	• The number of iterations can be extremely high. • It has a wide range of tuneable parameters.
1975	Genetic Algorithms (GA)	• Easy to understand • Supports multi-objective optimization. • Work well on mixed discrete/continuous problem [36].	• GA performance remained an art form. • GA is computationally expensive as well as time-consuming. • It can be difficult to represent an operation.

3. FEATURE SELETION USING BIO INSPIRED ALGORITHMS

Choosing a feature is a difficult task. We provide high quality solutions within an acceptable period.

3.1. Features Selection of Genetic Algorithm (GA)

An evolutionary method of search that also address optimization problems based on natural selection. These can be solved by random generation to form a population. The genetic algorithm uses the fitness function to evaluate the population [33]. It is measured using accuracy, root mean square error (RMSE), F-Measure, and area under the curve (AUC). In a series of reproductive operations Fitter individuals were selected, that includes crossovers and mutations [13]. Repeated process until it meets the termination criteria leading to a generations of formation.

3.2. Corona Virus Herd Immunity Optimizer (CHIO)

Based on the human response to Coronaviruses and environmental factors, CHIO is an extremely powerful optimization method. Consequently, optimization is generated using the input of a sensitive, infected, and immunized population [16]. The following six steps comprise this algorithm:

First, Set the parameters for CHIO and define your optimization problem with Equation: min f(x), where x∈(lb,ub), f(x) denotes immunity rate whereas x_n is a gene defined as (x1,x2,...,xn). The lower bound is referred to as lb, and the upper bound is referred to as ub. CHIO parameters have the number of infected cases (C0), (Max – ltr). HIS is denoted as population size, n is denoted as the problem dimension, and the Maximum infected cases age is (Max_{Age}).

Second, A heuristic two-dimensional matrix is [16] generated to represent the Herd Immunity Population. This matrix has the same number of objects as the HIS.

$$\text{HIP} = \begin{bmatrix} x_1^1 & x_2^1 & x_n^1 \\ x_1^2 & x_2^2 & x_n^2 \\ x_1^{HIS} & x_2^{HIS} & x_n^{HIS} \end{bmatrix} \quad (15)$$

The HIP matrix of object is calculated by $x_i^j = lb_i + (ub_i - lb_i) \times U(0,1)$, Equation (15) will used to calculate the immunity rate. The HIS length (S vector) status is started with infected cases, which seem to be 0 and 1, in both.

Third, the evolution is the most important step in CHIO [16] improvement. Taking BR_r into account, the social distance causes alterations in genotype (x_i^j). Equation depicts every possible state as in equation 16.

$$x_i^j(t+1) \leftarrow \begin{cases} C\left(x_i^j(t)\right) & r < \frac{1}{3} \times BR_r \\ N\left(x_i^j(t)\right) & r < \frac{2}{3} \times BR_r \\ R\left(x_i^j(t)\right) & r < BR_r \end{cases} \quad (16)$$

REVIEW ARTICLE

r is a number between 0 and 1 that is chosen at random. $C\left(x_i^j(t)\right)$ is a term that refers to an infected situation and is described as equation 17.

$$C\left(x_i^j(t)\right) = x_i^j(t) + r \times (x_i^j(t) - x_j^c(t)) \quad (17)$$

$x_j^c(t)$ = random number that can be selected from any case of infection (equation 18).

$$N\left(x_i^j(t)\right) = x_i^j(t) + r \times (x_i^j(t) - x_j^m(t) \quad (18)$$

$x_j^m(t)$ = random number which is chosen from any vulnerable case in equation 19.

$$R\left(x_i^j(t)\right) = x_i^j(t) + r \times (x_i^j(t) - x_j^v(t)) \quad (19)$$

The best immune [16] case is used to distribute $x_j^c(t)$.

Forth, At this point, replace the current case $(x^j(t))$ with$(x^j(t+1))$. Shows in equation 20.

$$f((x^j(t+1)) < f(x^j(t)) \quad (20)$$

If $S_j = 1$, the age vector will be expanded. The following are the vector S_j update equation 21:

$$S_j \leftarrow \begin{cases} f(x^j(t+1)) < \dfrac{f(x)^j(t+1)}{\Delta f(x)} \\ f(x^j(t+1)) > \dfrac{f(x)^j(t+1)}{\Delta f(x)} \end{cases}$$
$$S_j = 0 \text{ is Corona } (x^j(t+1)) \quad (21)$$

$\Delta f(x)$ is denoted as the mean of population immune rate. When any of the infected case's values are assigned to the new case, Corona $(x^j(t+1))$ returns 1 in binary form.

Fifth, If the current level of infection continues, there will be cases of mortality. $S_j = 1$ The case is declared to be dead if it cannot be improved after a given number of cycles specified by the Max-Age component. Furthermore, in order to avoid the local optimum and diversify population levels, A_j and S_j will be equal to 0.

Sixth, To achieve the algorithm stopping criterion, i.e., the maximum allowable updated version, the algorithm steps will be repeated from 3 to 6 to reach the algorithm stopping criterion.

3.3. Grey Wolf Optimization

The hunting strategy of grey wolves (Canis lupus) inspired the meta-heuristic algorithms, proposed in 2014. The hunting strategy involves searching, surrounding, and attacking their prey with a well-established leadership hierarchy. To solve optimization problems, the hierarchy is followed as alpha, beta, delta, and omega with four types of wolves.

A wolf pack typically has five to twelve individuals, and the leadership is comprised of an alpha couple, who are not essentially the robust but possess basic skills to lead a pack [22]. The majority of policy decisions, and the pack's other wolves express their approval by holding their tail feathers weaker than usual.

Democratic decision-making is only observed in rare instances. Alpha wolves are followed by beta wolves, who serve as advisors to the alpha and assist in decision-making. Omega wolves, who are at the bottom of the hierarchy, are susceptible to all other dominant wolves. Democratic decision-making behaviour has been observed on occasion [23].

The alpha wolves' heirs, the beta wolves, serve as counsellors and aid the alpha in making judgments. Omega wolves are at the bottom of the food chain and are preyed upon by all other dominant wolves. They aren't a necessary member of the pack, but their absence causes internal strife and issues. Omegas serve to keep the pack's dominance structure intact by satiating the entire pack [22].

Delta wolves cycle through alphas and betas, but omegas are their domain. Scouts patrol the pack's boundaries and alert the pack if there is a threat; sentinels serve as guards; Hunters assist in hunting and supplying food for the pack, as do experienced elves who have previously served as alpha or beta.

Caregivers who keep a watch on the pack's weak and injured members fall into this category as well. In order to tackle a variety of optimization issues, this hunting and leadership hierarchy has been mathematically described.

4. FEATURE SELETION

The selection of key attributes is the initiative for IDS. Feature selection defined as carefully selecting subset of features originally present and matches the specific criteria in Figure 9. It is critical for top-dimensional data [9] [37] [38] [39]. The subset of features can be selected by the following step lists:

1. Create subset of features in the given dataset.

2. Subset to be Evaluated using the provided criteria.

3. Determine whether the objectives were met. This may be the precise functional boundary that leads to better solutions or satisfactory results based on the error rate.

4. The goal has been met, validate the results and close the loop.

REVIEW ARTICLE

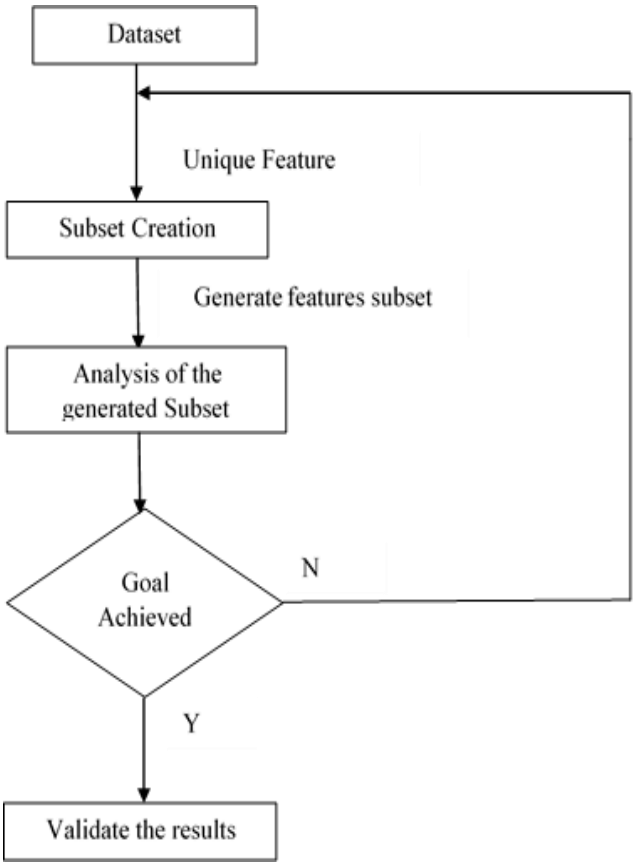

Figure 9 Feature Selection Flow Diagram

5. INTRUSION DETECTION DATASETS

The evaluation datasets are essential in validating any IDS approaches (Table 2). It helps to assess the suggested method's and ability to find out intrusive behaviour [40].

Table 2 Type of Dataset used in the Existing Research Work and the Features

Data Set	Reference	Advantages	Features
KDDCUP99	[38]	To construct a network intrusion detection system, DARPA created approximately 4,900,000 single connection vectors in 1999.	There are 41 features classified as either normal or attacked.
NSL-KDD	[41]	• In a newly revised version of the KDD '99 data set, there are roughly 1,074,992 single connection vectors. • It removes irrelevant records from the training set, preventing classifiers from being biassed toward more frequently occurring records. • The train and test data sets have a logical number of records, making it much easier to conduct the work on the entire data set without selecting irregular short fragments. The results of various work evaluations will be accurate.	41 features have been classified as either normal or attacked.
UNSW-NB15	[42]	Published in the year 2015. Which is having 9 attack types	45 Features
BoT-IoT	[43]	It is having 73 million big data.	10 Features
CICIDS2017	[44]	Monitoring and maintaining network security it consists of labeled network flows	More than 80 Features
CSE-CICIDS2018	[45]	• Brute-force Denial of Service attacks • SSH brute-force • Infiltration • Heartbleed vulnerability • Web-based attacks • Botnet	80 Features

REVIEW ARTICLE

5.1. KDDCUP99 Dataset

DARPA's IDS evaluation programme used the KDD CUP99 dataset. This dataset covers network traffic packets totaling seven gigabytes. This method can store approximately contains 5m records [46]. Each feature vector is either an attack or a normal.

5.2. NSL-KDD Dataset

The NSL-KDD dataset (Table 3) used to perform a benchmark test on an IDS was created over the course of several investigations [47]. KDDTrain+ contains 22 attack types as well as standard packet data types, whereas KDDTest+ contains 37 attack types divided into four attacks. NSL-KDD+* is an NSLKDD dataset without any new types of attacks (by eliminating 17 new types of attacks). It contains 41 features as well as one class label.

5.3. UNSW-NB15 Dataset Features

The UNSW-NB 15 dataset (Table 4) contains raw network packets that were generated by the UNSW Canberra Cyber Range Lab's IXIA PerfectStorm tool to generate a hybrid of true modern routine operations and artificial modern attack behaviours [42]. To generate a total of 49 features with the class label, the Argus and Bro-IDS tools are used, and twelve algorithms are developed.

5.4. BoT-IoT Dataset

It was created at the UNSW Canberra Cyber Range Laboratory. The feature of BoT-IoT dataset is shown in Table 6.

5.5. CICIDS2017 Dataset Features

Table 7 depicts the features of CICIDS2017 dataset.

5.6. CSE-CICIDS2018 Dataset Features

There are several attacks in the CSE-CIC-IDS2018 (Table 8) which have been described below [46] [47] [48]

- Infiltration
- Brute Force Attacks
- SQL Injection
- DDoS
- DoS
- Heartbleed (Heartleech)

The pie chart shows (Figure 10) datasets usage in percentage [52].

6. EVALUATION OF BIO-INSPIRED ALGORITHM

Classified the various feature of intrusion detection into four groups in order to guide new researchers in identifying research gaps and moving forward with their research work [3]. Multi-objective ID models are Classification, feature selection, weight selection/optimization among them [53] [54]. Furthermore, Table 9 gives the details of particular bio-inspired algorithm and previously used other models and what are the performance metrics are implemented.

6.1. Issues to be Addressed to Pursue New Research Directions

The review identifies existing challenges and carefully addresses them, and potentially arriving at solutions. Furthermore, the considers new directions in research and subsequent measures in building a useful IDS model [52]. Listed below are the major challenges identified in literature.

Limitation 1: The classifier's performance is improved by combining ML techniques with different techniques like swarm intelligence. However, it is difficult to select appropriate algorithmic parameters in parameter optimization or feature selection,

Limitation 2: The quality of feature used in detection and classification has a larger impact on the performance of the classifier of intrusion detection. This requires longer time to investigate the features that impacts the performance of IDS.

Limitation 3: The enormous quantity of data in a real-world network environment demands an efficient IDS proficient of handling of huge volumes of data.

Limitation 4: The choice of dataset in intrusion detection is critical in evaluating the performance. In most of the research focused on the datasets KDD Cup 99 and DARPA. But it lacks real time network attack scenarios, there is a strong need for a dataset that are more realistic and diverse that represent scenarios of intrusion detection in a real-world [56].

Limitation 5: The network environment is constantly changing and dynamic. This demands more adaptable IDS for new attacks in the ever-changing network situations with the purpose of maintaining maximum detection accuracy.

With considering the above challenge, the survey performed in the research on the current advancement highlights the strength of constraints which demands the research community to further investigations.

7. EVALAUATION METRICS

To analyse the success of the EA algorithm and ML models, evaluation metrics are used:

- True Positive - A true positive outcome is one in which the model accurately estimates the positive class. It is denoted by TP [38].

REVIEW ARTICLE

Table 3 Features of KDD CUP99 Dataset

No	Name	Description
1	Duration	Duration of connection in seconds
2	Protocol type	Connection protocol (tcp, udp,icmp)
3	Service	Dst port mapped to service (e.g. http, ftp,..)
4	Flag	Normal or error status flag of connection
5	Src bytes	Number of data bytes from src to dst
6	Dst bytes	Bytes from dst to src
7	Land	1 if connection is from/to the same host/ port;else 0
8	Wrong fragment	Number of 'wrong fragments' (values 0,1,3)
9	Urgent	Number of urgent packets
10	Hot	Number of 'hot' indicators (bro - ids feature)
11	Num failed logins	Number of failed login attempts
12	Logged in	1 if successfully logged in; else 0
13	Num compromised	Number of failed login attempts
14	Root shell	1 if root shell is obtained; else 0
15	Su attempted	1 if 'Su root' command attempted; else 0
16	Num root	Number of 'root' accesses
17	Num file creations	Number of file creation operations
18	Num shells	Number of shell prompts
19	Num access files	Number of operations on access control files
20	Num outbound cmds	Number of outbound commands in an ftp session
21	Is hot login	1 if login belongs to 'hot' list (e.g. root, adm); else 0
22	Is guest login	1 if login is 'guest' login (e.g. guest, anonymous); else 0
23	Count	Number of connections to same host as current connection in past two seconds
24	Srv	Number of connections to same service as current connection in past two seconds
25	Serror rate	% of connections that have 'SYN' errors
26	Srv rate	% of connections that have 'SYN' errors
27	Rerror rate	% of connections that have 'REJ' errors
28	Srv rerror rate	% of connections that have 'REJ' errors
29	Same srv rate	% of connections to the same service
30	Diff host rate	% of connections to different services
31	Srv diff host rate	% of connections to differernt hosts
32	Dst hot count	Count of connections having same dst host
33	Dst host_srv_count	Count of connections having same dst host and using same service
34	Dst_host_same_srv_rate	% of connections having same dst host and using same service
35	Dst host diff srv rate	% of different services on current host
36	Dst host same src _post rate	% of connections to current host having same src port
37	Dst_host_srv_diff_host_rate	% of connections to same service coming from diff. hosts
38	Dst host serror _rate	% of connections to current host that have an S0 error
39	Dst_host_srv_serror_rate	% of connections to current host and specified service that have an S0 error
40	Dst_host_rerror_rate	% of connections to current host that have an RST error
41	Dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an RST error

REVIEW ARTICLE

Table 4 Features of NSL-KDD Dataset

No	Data Features	Description
1	Duration	Duration of connection in seconds
2	Protocol type	Connection protocol (tcp, udp,icmp)
3	Service	Dst port mapped to service (e.g. http, ftp,..)
4	Flag	Normal or error status flag of connection
5	Scr bytes	Number of data bytes from src to dst
6	Dst bytes	Bytes from dst to src
7	Land	1 if connection is from/to the same host/ port;else 0
8	Wrong fragment	Number of 'wrong fragments' (values 0,1,3)
9	Urgent	Number of urgent packets
10	Hot	Number of 'hot' indicators (bro - ids feature)
11	Num failed logins	Number of failed login attempts
12	Logged in	1 if successfully logged in; else 0
13	Num compromised	Number of failed login attempts
14	Root shell	1 if root shell is obtained; else 0
15	Su attempted	1 if 'Su root' command attempted; else 0
16	Num root	Number of 'root' accesses
17	Num file creations	Number of file creation operations
18	Num shells	Number of shell prompts
19	Num access files	Number of operations on access control Files
20	Num outbound cmds	Number of outbound commands in an ftp session
21	Is host login	1 if login belongs to 'hot' list (e.g. root, adm); else 0
22	Is guest login	1 if login is 'guest' login (e.g. guest, anonymous); else 0
23	Count	Number of connections to same host as current connection in past two seconds
24	Srv_count	Number of connections to same service as current connection in past two seconds
25	Serror rate	% of connections that have 'SYN' errors
26	Srv rate	% of connections that have 'SYN' errors
27	Rerror rate	% of connections that have 'REJ' errors
28	Srv rerror rate	% of connections that have 'REJ' errors
29	Same srv rate	% of connections to the same service
30	Diff host rate	% of connections to different services
31	Srv diff host rate	% of connections to differernt hosts
32	Dst hot count	Count of connections having same dst host
33	Dst host_srv_count	Count of connections having same dst host and using same service
34	Dst_host_same_srv_rate	% of connections having same dst host and using same service
35	Dst host diff srv rate	% of different services on current host
36	Dst host same src _post rate	% of connections to current host having same src port
37	Dst_host_srv_diff_host_rate	% of connections to same service coming from diff. hosts

REVIEW ARTICLE

38	Dst host serror _rate	% of connections to current host that have an S0 error
39	Dst_host_srv_serror_rate	% of connections to current host and specified service that have an S0 error
40	Dst_host_rerror_rate	% of connections to current host that have an RST error
41	Dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an RST error

Table 5 Features of UNSW-NB15 Dataset

No	Feature Name	No	Feature Name	No	Feature Name
1	Id	16	Dloss	31	Response_ body_ len
2	dur	17	Sinpkt	32	Ct_srv_src
3	Proto	18	Dinpkt	33	Ct_state_ttl
4	Service	19	Sjit	34	Ct_dst_ltm
5	State	20	Djit	35	Ct_src_dport_ltm
6	Spkts	21	Swin	36	Ct_dst_sport_ltm
7	dpkts	22	Stepb	37	Ct_dst_src_ltm
8	Sbytes	23	Dtepb	38	Is_ftp_login
9	dbytes	24	Dwin	39	Ct_ftp_cmd
10	rate	25	Tcprtt	40	Ct_flw_http_mthd
11	Sttl	26	Synack	41	Ct_src_ltm
12	Dtll	27	Ackdat	42	Ct_srv_dst
13	Sload	28	Smean	43	Is_sm_ips_ports
14	Dload	29	Dmean	44	Attack _cat
15	Sloss	30	Trans depth	45	label

Table 6 Features of BoT-IoT Dataset

No	Feature	Description
1	pkSeqID	Row Identifier
2	Seq	Argus sequence number
3	Mean	Average duration of aggregated records
4	Stddev	Standard deviation of aggregated records
5	Min	Minimum duration of aggregated records
6	Max	Maximum duration of aggregated records
7	Srate	Source-to-destination packets per second
8	Drate	Destination-to-source packets per second
9	NINConn PSrcIP	Total Number of packets Per source IP
10	NINCConn PDstIP	Total number of packets Per Destination IP

REVIEW ARTICLE

Table 7 Features of CICIDS2017 Dataset

No	Feature	No	Feature	No	Feature	No	Feature
1	Flow ID	22	Flow packets/s	43	Fwd packets/s	64	Fwd Avg packets/Bulk
2	Source IP	23	Flow IAT Mean	44	Bwd packets/s	65	Fwd Avg Bulk Rate
3	Source Port	24	Flow IAT Std	45	Min packet Length	66	Bwd Avg Bytes/Bulk
4	Destination IP	25	Flow IAT Max	46	Max packet Length	67	Bwd Avg packets/Bulk
5	Destination Port	26	Flow IAT Min	47	Packet Length Mean	68	Bwd Avg Bulk Rate
6	Protocol	27	Fwd IAT Total	48	Packet Length Std	69	Subflow Fwd packets
7	Timestamp	28	Fwd IAT Mean	49	Packet Length Variance	70	Subflow fwd Bytes
8	Flow Duration	29	Fwd IAT Std	50	FIN Flag Count	71	Subflow Bwd packets
9	Total Fwd Packets	30	Fwd IAT Max	51	SYN Flag count	72	Subflow Bwd Bytes
10	Total Back ward packets	31	Fwd IAT Min	52	RST Flag Count	73	Init_Win_bytes_forward
11	Total Length of Fwd Packets	32	Bwd IAT Total	53	PSH Flag Count	74	Init_Win_bytes_backward
12	Total Length of BWD Packets	33	Bwd IAT Mean	54	ACK Flag Count	75	Act_data_pkt_fwd
13	Fwd packet Length Max	34	Bwd IAT Std	55	URG Glag Count	76	Min_seg_size_forward
14	Fwd Packet Length Max	35	Bwd IAT Max	56	CWE Flag Count	77	Active Mean
15	Fwd packet Length Mean	36	Bwd IAT Min	57	ECE Flag count	78	Active Std
16	Fwd Packet Length Std	37	Fwd PSH Flags	58	Down/up Ratio	79	Active Max
17	Bwd Packet Length Max	38	Bwd PSH Flags	59	Average Packet Size	80	Active Min
18	Bwd packet Length Min	39	Fwd URG Flags	60	Avg Fwd Segment Size	81	Idle Mean
19	Bwd packet Length Mean	40	Bwd URG Flags	61	Avg Bwd Segment Size	82	Active Std
20	Bwd packet Length Std	41	Fwd Header Length	62	Fwd Header Length	83	Idle Max
21	Flow Bytes/s	42	Bwd Header Length	63	Fwd Avg Bytes/Bulk	84	Idle Min

REVIEW ARTICLE

Table 8 Features of CSE-CICIDS2018 Dataset

Attributes	Description
1-4	Basic features of network connections
5-16	Features of network packets
17-22	Features of network flows
23-45	Statistics of network flows
46-63	Content – related traffic features
64-67	Features of network subflows
68-79	General purpose traffic features
80-83	Basic features of network connections

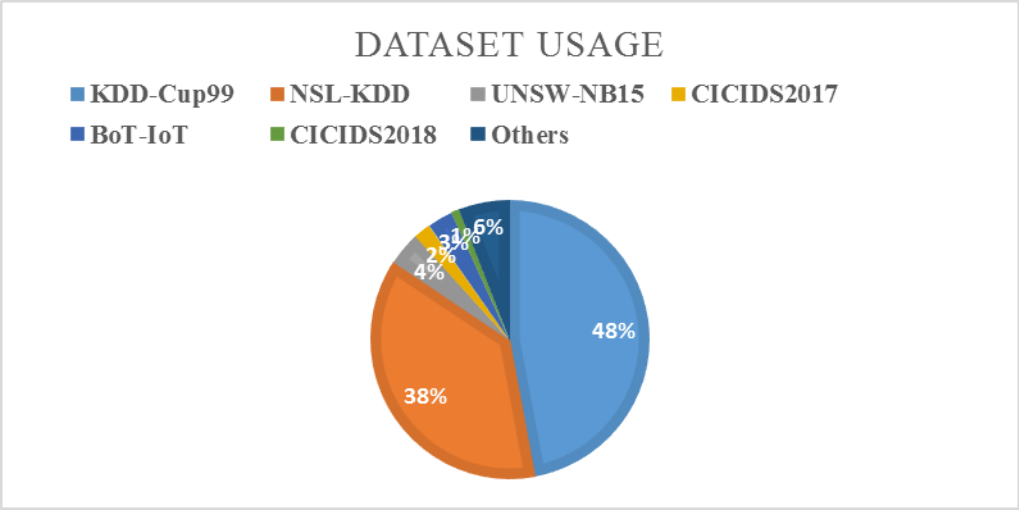

Figure 10 Swarm Inspired Intrusion Detection Models Evaluation using Datasets

- True Negative - A true negative outcome is one in which the model accurately estimates the negative class. It is denoted by TN.
- False Positive - FP denotes the number of erroneous attack tests.
- False Negative - The number of benign samples misclassified is denoted by FN.

The total number of samples is divided by the number of samples that have been correctly categorised. It's referred to as precision (ACC) [42], shows in equation 22.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (22)$$

The harmonic mean of precision and recall is the F1 score [40], shows in equation 23.

$$F1 = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (23)$$

The total number of assault samples is divided by the number of attack samples that were successfully classified. It is known as the detection rate (DR) as in equation 24.

$$DR = \frac{TP}{TP+FN} \quad (24)$$

FAR - The false alarm rate is determined by dividing the number of attack samples that were wrongly classified by the total number of benign samples [4], shows in equation 25.

$$FAR = \frac{FP}{FP+TN} \quad (25)$$

Intrusion Detection Comparison different Feature Learning method's in correct rate.

REVIEW ARTICLE

Figure 11 shows the study analysis of DBN-based feature learning method beats the standard feature of the learning process for training IDS was performed on four different set of data. The DBN-based feature learning method outperformed the PCA method by 11.58 percent and the gain ratio method by 12.91 percent for the large data set S4 [11] [45]. DBN-based feature learning algorithms, as a result, are better suited for high-dimensional feature learning.

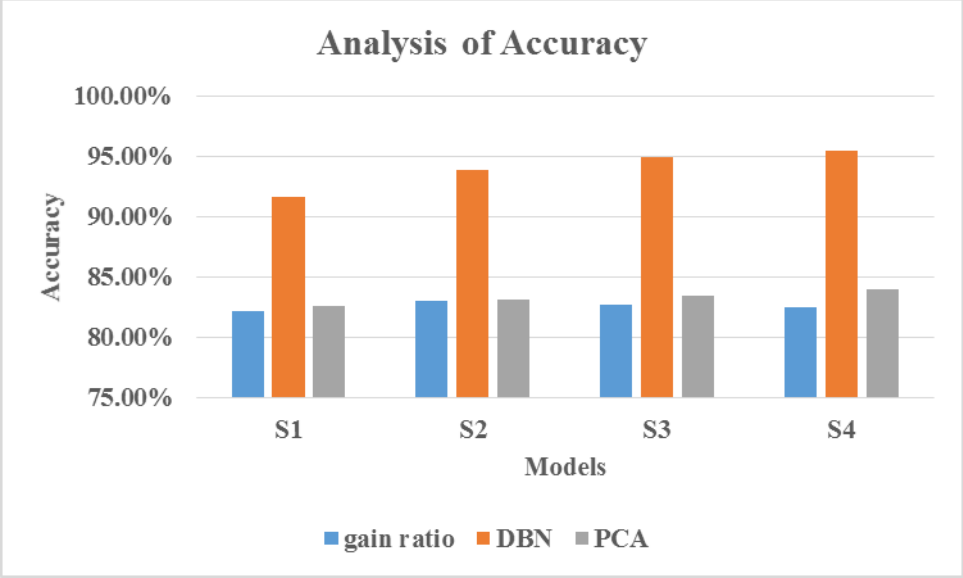

Figure 11 Analysis of the Detection Accuracy of Feature Learning Models

8. CONCLUSIONS

This article helps to define the challenging problem and illustrates existing SI-based programmes to improve it. Presents a systematic and robust approach to find, evaluate, and classify the present state of SI methods in four aspects. Intervention classification, machine learning-based classification model strength optimization, feature extraction, and multi-objective models that integrate optimization and feature extraction. As technology evolves, traditional measures like firewalls and authentication systems fail to provide complete protection against advanced attackers. The gap identified in the traditional methods can be addressed with the IDS and IPS required to find out in a timely and effective manner about any threats and it gives potential solutions to the intrusion detection problem.

The terms frequently used in swarm intelligence are life, mind and intelligence. In general the term swarm is represent insects social behavior, the authors attempted point to cognitive behaviour of any operating colony. Any living social colony engages in swarm colony-like behaviour on average. A socio-psychological simulation programme is successfully used in the commercial field. Particle swarm optimization is being worked on by researchers around the globe. This method is extremely adaptable and strong of the swarm intelligence is still in its early stages. Metaheuristic hybridization opens up a slew of new possibilities for us.

REFERENCES

[1] Y. K. Saheed and M. O. Arowolo, "Efficient Cyber Attack Detection on the Internet of Medical Things-Smart Environment Based on Deep Recurrent Neural Network and Machine Learning Algorithms," IEEE Access, vol. 9, pp. 161546–161554, 2021, doi: 10.1109/ACCESS.2021.3128837.

[2] M. T. Ali A. Ghorbani, Wei Lu, "Network Intrusion Detection and Prevention Advances in Information Security," Inf. Syst., p. 223, 2010.

[3] B. A. Tama and S. Lim, "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation," Comput. Sci. Rev., vol. 39, p. 100357, 2021, doi: 10.1016/j.cosrev.2020.100357.

[4] W. Peng, X. Kong, G. Peng, X. Li, and Z. Wang, "Network intrusion detection based on deep learning," Proc. - 2019 Int. Conf. Commun. Inf. Syst. Comput. Eng. CISCE 2019, pp. 431–435, 2019, doi: 10.1109/CISCE.2019.00102.

[5] M. S. Husain, "Nature Inspired Approach for Intrusion Detection Systems," Des. Anal. Secur. Protoc. Commun., pp. 171–182, 2020, doi: 10.1002/9781119555759.ch8.

[6] S. Roy, S. Biswas, and S. Sinha Chaudhuri, "Nature-Inspired Swarm Intelligence and Its Applications," Int. J. Mod. Educ. Comput. Sci., vol. 6, no. 12, pp. 55–65, 2014, doi: 10.5815/ijmecs.2014.12.08.

[7] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," 2007 IEEE Congr. Evol. Comput. CEC 2007, pp. 4661–4667, 2007, doi: 10.1109/CEC.2007.4425083.

[8] Z. Li and X. Huang, "Glowworm Swarm Optimization and Its Application to Blind Signal Separation," Math. Probl. Eng., vol. 2016, 2016, doi: 10.1155/2016/5481602.

[9] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," Comput. Secur., vol. 81, pp. 148–155, 2019, doi: 10.1016/j.cose.2018.11.005.

[10] O. Almomani, "SS symmetry Detection System Based on PSO , GWO , FFA and," A Featur. Sel. Model Netw. Intrusion Detect. Syst. Based

REVIEW ARTICLE

PSO, GWO, FFA GA Algorithms, vol. 33, no. 32, pp. 1–22, 2020.
[11] Y. Luo, "Research on Network Security Intrusion Detection System Based on Machine Learning," Int. J. Netw. Secur., vol. 23, no. 3, pp. 490–495, 2021, doi: 10.6633/IJNS.202105.
[12] M. S. Abbasi, H. Al-Sahaf, M. Mansoori, and I. Welch, "Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection," Appl. Soft Comput., vol. 121, p. 108744, 2022, doi: 10.1016/j.asoc.2022.108744.
[13] A. Ayough, M. Zandieh, and H. Farsijani, "GA and ICA approaches to job rotation scheduling problem: Considering employee's boredom," Int. J. Adv. Manuf. Technol., vol. 60, no. 5–8, pp. 651–666, 2012, doi: 10.1007/s00170-011-3641-7.
[14] P. Mukilan and W. Semunigus, "Human object detection: An enhanced black widow optimization algorithm with deep convolution neural network," Neural Comput. Appl., vol. 33, no. 22, pp. 15831–15842, 2021, doi: 10.1007/s00521-021-06203-3.
[15] R. Malik, Y. Singh, Z. A. Sheikh, P. Anand, P. K. Singh, and T. C. Workneh, "An Improved Deep Belief Network IDS on IoT-Based Network for Traffic Systems," J. Adv. Transp., vol. 2022, 2022, doi: 10.1155/2022/7892130.
[16] A. S. Mahboob, H. S. Shahhoseini, M. R. Ostadi Moghaddam, and S. Yousefi, "A coronavirus herd immunity optimizer for intrusion detection system," 2021 29th Iran. Conf. Electr. Eng. ICEE 2021, pp. 579–585, 2021, doi: 10.1109/ICEE52715.2021.9544165.
[17] A. Thakkar and R. Lohiya, "Role of swarm and evolutionary algorithms for intrusion detection system: A survey," Swarm Evol. Comput., vol. 53, no. December 2019, p. 100631, 2020, doi: 10.1016/j.swevo.2019.100631.
[18] A. C. Enache and V. Sgârciu, "Enhanced intrusion detection system based on bat algorithm-support Vector Machine," SECRYPT 2014 - Proc. 11th Int. Conf. Secur. Cryptogr. Part ICETE 2014 - 11th Int. Jt. Conf. E-bus. Telecommun., pp. 184–189, 2014, doi: 10.5220/0005015501840189.
[19] X. S. Yang and S. Deb, "Engineering optimisation by cuckoo search," Int. J. Math. Model. Numer. Optim., vol. 1, no. 4, pp. 330–343, 2010, doi: 10.1504/IJMMNO.2010.035430.
[20] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, "Cuckoo Search Optimization- A Review," Mater. Today Proc., vol. 4, no. 8, pp. 7262–7269, 2017, doi: 10.1016/j.matpr.2017.07.055.
[21] W. Zhiheng and L. Jianhua, "Flamingo Search Algorithm: A New Swarm Intelligence Optimization Algorithm," IEEE Access, vol. 9, pp. 88564–88582, 2021, doi: 10.1109/ACCESS.2021.3090512.
[22] M. H. Nasir, S. A. Khan, M. M. Khan, and M. Fatima, "Swarm Intelligence inspired Intrusion Detection Systems — A systematic literature review," Comput. Networks, vol. 205, no. January, p. 108708, 2022, doi: 10.1016/j.comnet.2021.108708.
[23] A. Hosseinalipour and R. Ghanbarzadeh, "A novel approach for spam detection using horse herd optimization algorithm," Neural Comput. Appl., vol. 0123456789, 2022, doi: 10.1007/s00521-022-07148-x.
[24] S. Khosravi and A. Chalechale, "Chimp Optimization Algorithm to Optimize a Convolutional Neural Network for Recognizing Persian/Arabic Handwritten Words," Math. Probl. Eng., vol. 2022, no. Dl, 2022, doi: 10.1155/2022/4894922.
[25] M. Amudha, R. Manickam, and R. Gayathri, "A Study on Climate Change with Mayfly Algorithm Optimization," Recent trends Manag. Commer., vol. 2, no. 3, pp. 2–8, 2021, doi: 10.46632/rmc/2/3/5.
[26] Y. J. Zheng, "Water wave optimization: A new nature-inspired metaheuristic," Comput. Oper. Res., vol. 55, pp. 1–11, 2015, doi: 10.1016/j.cor.2014.10.008.
[27] P. Kanchan, "Rainfall Analysis and Forecasting Using Deep Learning Technique," J. Informatics Electr. Electron. Eng., vol. 2, no. 2, pp. 1–11, 2021, doi: 10.54060/jieee/002.02.015.
[28] M. Alshinwan et al., "Dragonfly algorithm: a comprehensive survey of its results, variants, and applications," Multimed. Tools Appl., no. February, 2021, doi: 10.1007/s11042-020-10255-3.
[29] J. Pierezan et al., "Multiobjective Ant Lion Approaches Applied to Electromagnetic Device Optimization," Technologies, vol. 9, no. 2, p. 35, 2021, doi: 10.3390/technologies9020035.
[30] S. Idris, O. Oyefolahan Ishaq, and N. Ndunagu Juliana, "Intrusion Detection System Based on Support Vector Machine Optimised with Cat Swarm Optimization Algorithm," 2019 2nd Int. Conf. IEEE Niger. Comput. Chapter, Niger. 2019, 2019, doi: 10.1109/NigeriaComputConf45974.2019.8949676.
[31] C. Kiran Kumar and M. Govindarajan, "An efficient rapid intrusion detection method for detecting intrusions in networks," Int. J. Sci. Technol. Res., vol. 9, no. 2, pp. 5991–5997, 2020.
[32] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," Simulation, vol. 76, no. 2, pp. 60–68, 2001, doi: 10.1177/003754970107600201.
[33] M. Sazzadul Hoque, "An Implementation of Intrusion Detection System Using Genetic Algorithm," Int. J. Netw. Secur. Its Appl., vol. 4, no. 2, pp. 109–120, 2012, doi: 10.5121/ijnsa.2012.4208.
[34] M. Romero Montoya et al., "Solution Search for the Capacitated P-Median Problem using Tabu Search," Int. J. Comb. Optim. Probl. Informatics, vol. 10, no. 2, pp. 17–25, 2019, [Online]. Available: www.editada.org
[35] K. Dnrxu and O. Dú, "Tabu- * HQHWLF $ OJRULWK P," IEEE Int. Conf. Comput. Sci. Eng., pp. 215–220, 2017.
[36] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, 2019, doi: 10.1186/s42400-019-0038-7.
[37] M. A. Siddiqi and W. Pak, "Optimizing filter-based feature selection method flow for intrusion detection system," Electron., vol. 9, no. 12, pp. 1–18, 2020, doi: 10.3390/electronics9122114.
[38] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," Expert Syst. Appl., vol. 148, 2020, doi: 10.1016/j.eswa.2020.113249.
[39] T. S. Naseri and F. S. Gharehchopogh, "A Feature Selection Based on the Farmland Fertility Algorithm for Improved Intrusion Detection Systems," J. Netw. Syst. Manag., vol. 30, no. 3, 2022, doi: 10.1007/s10922-022-09653-9.
[40] C. Zhang, F. Ruan, L. Yin, X. Chen, L. Zhai, and F. Liu, "A Deep Learning Approach for Network Intrusion Detection Based on NSL-KDD Dataset," Proc. Int. Conf. Anti-Counterfeiting, Secur. Identification, ASID, vol. 2019-Octob, pp. 41–45, 2019, doi: 10.1109/ICASID.2019.8925239.
[41] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in Network Intrusion Detection Systems," Expert Syst. Appl., vol. 186, no. May, p. 115782, 2021, doi: 10.1016/j.eswa.2021.115782.
[42] A. E. Ibor, O. B. Okunoye, F. A. Oladeji, and K. A. Abdulsalam, "Novel Hybrid Model for Intrusion Prediction on Cyber Physical Systems' Communication Networks based on Bio-inspired Deep Neural Network Structure," J. Inf. Secur. Appl., vol. 65, no. January, p. 103107, 2022, doi: 10.1016/j.jisa.2021.103107.
[43] M. D. Moizuddin and M. V. Jose, "A bio-inspired hybrid deep learning model for network intrusion detection," Knowledge-Based Syst., vol. 238, p. 107894, 2022, doi: 10.1016/j.knosys.2021.107894.
[44] M. Choraś and M. Pawlicki, "Intrusion detection approach based on optimised artificial neural network," Neurocomputing, vol. 452, pp. 705–715, 2021, doi: 10.1016/j.neucom.2020.07.138.
[45] M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher, and M. Portmann, "Feature Extraction for Machine Learning-based Intrusion Detection in IoT Networks," 2021, [Online]. Available: http://arxiv.org/abs/2108.12722
[46] E. ul H. Qazi, M. Imran, N. Haider, M. Shoaib, and I. Razzak, "An intelligent and efficient network intrusion detection system using deep learning," Comput. Electr. Eng., vol. 99, no. February 2021, p. 107764, 2022, doi: 10.1016/j.compeleceng.2022.107764.
[47] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," J. Inf. Secur. Appl., vol. 58, no. March, p. 102804, 2021, doi: 10.1016/j.jisa.2021.102804.
[48] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj,

REVIEW ARTICLE

"Anomaly-based intrusion detection system for IoT networks through deep learning model," Comput. Electr. Eng., vol. 99, no. February, p. 107810, 2022, doi: 10.1016/j.compeleceng.2022.107810.

[49] M. Hammad, N. Hewahi, and W. Elmedany, "MMM-RF: A novel high accuracy multinomial mixture model for network intrusion detection systems," Comput. Secur., vol. 120, 2022, doi: 10.1016/j.cose.2022.102777.

[50] N. Gupta, V. Jindal, and P. Bedi, "CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems," Comput. Secur., vol. 112, p. 102499, 2022, doi: 10.1016/j.cose.2021.102499.

[51] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[52] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic," Appl. Sci., vol. 11, no. 17, 2021, doi: 10.3390/app11177868.

[53] J. Zhang, H. Ishibuchi, and L. He, "A classification-assisted environmental selection strategy for multiobjective optimization," Swarm Evol. Comput., vol. 71, no. September 2021, p. 101074, 2022, doi: 10.1016/j.swevo.2022.101074.

[54] W. Tang, X. M. Yang, X. Xie, L. M. Peng, C. H. Youn, and Y. Cao, "Avidity-model based clonal selection algorithm for network intrusion detection," IEEE Int. Work. Qual. Serv. IWQoS, 2010, doi: 10.1109/IWQoS.2010.5542731.

[55] H. Bangui and B. Buhnova, "Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms," Comput. Electr. Eng., vol. 100, no. March, p. 107901, 2022, doi: 10.1016/j.compeleceng.2022.107901.

[56] F. Hosseinpour, K. A. Bakar, A. H. Hardoroudi, and A. F. Dareshur, "Design of a new distributed model for intrusion detection system based on artificial immune system," Proc. - 6th Intl. Conf. Adv. Inf. Manag. Serv. IMS2010, with ICMIA2010 - 2nd Int. Conf. Data Min. Intell. Inf. Technol. Appl., pp. 378–383, 2010.

Authors

Jeyavim Sherin R C received the B.E degree in Computer Science & Engineering from C.S.I Institute of Technology, Thovalai, Tamilnadu in 2007. She pursued M.E in Computer Science & Engineering from Karpagam University, Coimbatore, Tamilnadu in 2011. She is currently pursuing PhD in School of Computer Science & Engineering from VIT University, Chennai. Her areas of interest Network security and IoT.

Dr. Parkavi is currently working as an Assistant Professor-Senior grade in VIT University Chennai. She has completed her B.E Information Technology in 2004 and completed her M.E in 2008 and Ph.D in 2011 at College of Engineering Guindy, Anna University. She has 16+ years of teaching and research experiences. Dr. Parkavi held various academic and research positions. Her research interests include wireless networks, computer vision, soft computing, cyber security, IoT, Machine Learning and Deep learning. She has published more SCI and Scopus indexed journals and she is the reviewer of many international journals.

How to cite this article:

Jeyavim Sherin R C, Parkavi K, "Investigations on Bio-Inspired Algorithm for Network Intrusion Detection – A Review", International Journal of Computer Networks and Applications (IJCNA), 9(4), PP: 399-423, 2022, DOI: 10.22247/ijcna/2022/214503 .