International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

December 2024 | Issue 61 | Volume 16 | Number 4 | Pages 147-161

THE MOST HARNESSING OPTIMIZATION ALGORITHMS COMBINED WITH MACHINE LEARNING TO ENHANCE INTRUSION DETECTION SYSTEM: A COMPREHENSIVE REVIEW

H. Khoulimi O. Benammar

Applied Mathematics and Computing Laboratory, Higher Normal School, Hassan II University, Casablanca, Morocco
hind-khoulimi-etu@etu.univh2c.ma, othman.benammmar@univh2c.ma

Abstract- As cyber threats become increasingly pervasive across networks and IT systems, enhancing Intrusion Detection Systems (IDS) with advanced and precise tools is essential to safeguard cybersecurity. This survey explores recent advancements in Artificial Intelligence (AI) particularly in Machine Learning (ML) and Deep Learning (DL) for optimizing IDS performance. We systematically review a range of optimization algorithms (OAs) applied to IDS, including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), examining their roles in improving threat detection accuracy, reducing false alarm rates, and boosting IDS robustness. Furthermore, we analyze hybrid models that combine ML/DL with OAs to achieve high detection accuracy and efficiency, summarizing key metrics and datasets used for performance evaluation. Our findings outline the strengths and weaknesses of each approach, with a focus on practical implications for real-world cybersecurity. Finally, we address current limitations of hybrid IDS models in cybersecurity and identify potential research directions to enhance their adaptability and scalability in response to evolving threats.

Keywords: Intrusion Detection System, Optimization Algorithms, Artificial Intelligence, Machine Learning, Cyber Security, Deep Learning,

1. INTRODUCTION

Cybersecurity involves a set of practices and technologies designed to protect data, systems, and networks from unauthorized access and potential threats, thereby ensuring that information is kept confidential, intact, and available. The importance of cybersecurity has grown exponentially with the proliferation of the Internet, which has connected people through various devices in their daily lives. Currently, approximately 12.3 billion connected devices including autonomous vehicles, smartphones, and smart home systems generate vast quantities of data. This study focuses on the optimization algorithms employed to enhance cybersecurity, especially in the scope of Intrusion Detection Systems (IDS) capable of autonomously responding to cyber threats. A review of recent literature reveals that optimization algorithms (OAs) have become increasingly vital in the realm of IT security.

To mitigate cybersecurity threats, several mechanisms are employed, including:

- Encryption: This process encodes sensitive data to protect it from unauthorized access.
- Access Control: Implementation of access control lists ensures guarantee access to certain resources is limited to authorized users.
- User Education and Training: Enhancing awareness can significantly reduce human error, such as the use of weak passwords.
- Antivirus Software: These programs detect, prevent, and remove malicious data from systems.
- Firewalls: Acting as barriers between internal and external networks, firewalls monitor and regulate inbound and outbound traffic based on established security protocols, thus blocking unauthorized access while permitting legitimate traffic.
- Intrusion Detection Systems (IDS): These systems are intended to observe network traffic, identify any suspicious activity, and trigger alerts when potential malicious behavior is detected.

Traditional IDS typically rely on predefined rules and signatures, which can limit their effectiveness in identifying novel attacks. The incorporation of AI, particularly ML, and DL, into IDS can significantly improve their effectiveness by enabling the identification of unknown threats through advanced pattern recognition in data. The increasing sophistication of cyberattacks and the high dimensionality of data present significant challenges to conventional security mechanisms, necessitating innovative approaches. In this context, optimization algorithms have emerged as a critical component of cybersecurity strategies aimed at improving anomaly detection efficiency. This article provides a comprehensive and contemporary overview of the hybridization between optimization algorithms and ML/DL techniques as applied to IDS, aiming to enhance data security in distributed networks while minimizing false positives and negatives.

Registered: 240801 | Article Info: 1909 | Accepted: 241109
Reserved: 241005 | Available: 241231

This focus aligns with the fundamental objectives of maintaining confidentiality, integrity, and availability. Our research is centered on keywords related to IDS, ML and DL, optimization algorithms, and hybridization. We restricted our review to articles and surveys published between 2019 and 2024, carefully analyzing abstracts and titles to ensure relevance to the study's objectives. We focus on reviewing the most relevant optimization algorithms applied to cybersecurity, particularly intrusion detection systems (IDS) capable of responding autonomously to cyber threats over the past five years.

The following sections of the paper are organized as such: Section 2 elaborates on the different concepts that evolved in this article. Section 3 conducts an in-depth analysis of optimization algorithms used in cyber security. Section 4 discusses the datasets employed to assess the models within the field of cybersecurity. A classification of the selected scientific articles on hybridizing AI with OA based on the proposed solution is detailed in Section 5. Sections 6 and 7 cover discussions, research challenges, and future directions. Lastly, Section 8 provides the conclusion of the article.

2. BACKGROUND

The application of optimization algorithms finds significant utility in cybersecurity, particularly in the development and enhancement of IDS.

2.1. Intrusion Detection System

A system or equipment that aims to protect either an application, a machine and even a network from any breach that violates its security policy. If malicious activity is detected, the system will trigger an alarm. Generally, there are two types of detection as shown in Figure 1.

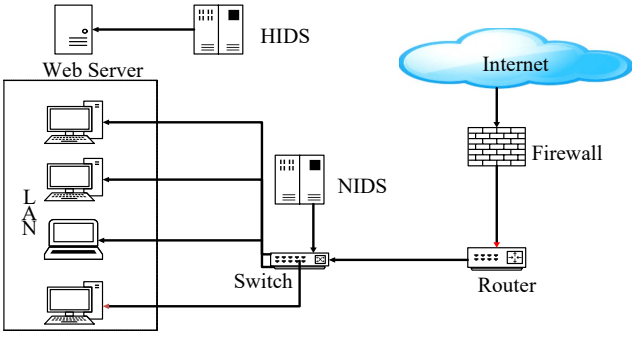

Figure 1. IDS Types

- Network Intrusion Detection System (NIDS): Following the firewall, this system monitors and analyzes both incoming and outgoing network traffic.
- Intrusion Detection System (HIDS): a system installed for each machine, it has a limited vision, and it just collects and analyzes the data of system files and log files of a single machine.

• There are two main detection methods used by the Intrusion Detection System:

- Signature-based Detection: Following the firewall, this system monitors and analyzes both incoming and outgoing network traffic.
- Anomaly-based Detection: IDS is based on the comparison of malicious traffic detected by an already predefined model over a while based on Machine Learning methods. So, any deviation from this behavior is considered an attack. This type of detection suffers from false positive detection, the traffic unknown by the system is classified as an attack.

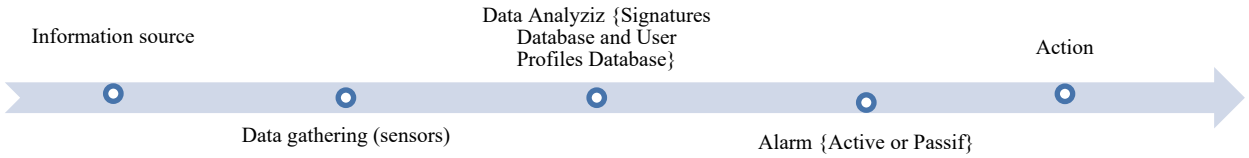

Figure 2. General IDS process

HIDS and NIDS both utilize collectors to gather data and events. HIDS collects information directly from machines, while NIDS gathers data from various network sensors. The collected data is then analyzed using an engine that relies on either a signature database or user profiles, depending on whether the detection method is misuse detection or anomaly detection. An alarm is triggered once a potential threat is identified, as illustrated in Figure 2. There are two types of IDS responses:

1) Passive response: The IDS notifies the network administrator when an attack is detected.
2) Active Response: The IDS is not limited to sending alerts; it identifies malicious activity, attempts to block it, and then sends a report to the network administrator. This system is known as an Intrusion Prevention System (IPS). The IPS can remove packets identified as malicious and block malicious traffic from circulating on the network.

There are many classification metrics used for IDS including:

- Confusion matrix: the confusion matrix provides which classes are being predicted correctly, which incorrectly, and what type of errors are being made. The confusion matrix consists of 4 values (TP, FP, TN, and FN).
- Accuracy is a metric used to assess the performance of a classification model across two or more classes. It measures the rate of correct predictions for all individuals as Equation (1) [1].

$$\frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

- Precision: It corresponds to the correct prediction rate among the positive predictions as Equation (2).

$$\frac{TP}{TP+FP} \quad (2)$$

- Recall: It corresponds to the rate of positive individuals detected by the model as Equation (3).

$$\frac{TP}{TP+FN} \tag{3}$$

- F-Measure: integrates precision and recall into a single metric by calculating their harmonic mean, effectively balancing false positives and false negatives as Equation (4).

$$2\times\frac{precision\times recall}{precision+recall} \tag{4}$$

- False Alarm Rate: refers to the frequency at which an intrusion detection or prevention system incorrectly identifies benign activity as malicious.
- ROC Curves: allows describing the performance of a model through two indicators: sensitivity and specificity.

2.2. Artificial Intelligence

AI is used to make intelligent decisions to eliminate certain human tasks through these two types: ML and DL algorithms, and it has the potential to surpass human abilities [2].
- ML: is a subfield of AI that simulates intelligent behavior in computers by giving the machine the ability to learn from data Without requiring explicit programming. Each technology is a subset of the one preceding it. This means that all ML algorithms are included under the broader category of AI.
- DL: Deep Learning is a more advanced branch of ML. Inspired by how human brains work. Especially, deep artificial neural networks, it means that DL teaches to do as the same way that humans naturally do, which means learning by example. The main difference between DL and ML is that the DL model does not need to be provided with features in classification tasks.

ML and DL techniques are based generally on four stages [3]:
1. Data collection and preprocessing:
- Data collection: gather large volumes of network traffic data, including normal and malicious traffic from different sources such as network logs, and system logs.
- Preprocessing: involves cleaning and normalizing the data to eliminate noise and irrelevant information [4].
2. Feature Engineering:
- Feature extraction: identify essential attributes from the raw data.
- Feature selection: use statistical methods or optimization algorithms to identify the most pertinent features, thereby reducing the data's dimensionality and improving model performance.
3. Model Training

A step that involves extracting valuable information derived from the data processed during the preprocessing phase, to gain insights and learn how to classify or predict new data. This is achieved by splitting the dataset into training and testing sets and utilizing ML techniques. to develop a mode, we distinguish three of learning methods:
- Supervised learning: train models using labeled datasets where the outcomes (normal or malicious) are known.
- Unsupervised Learning: use unlabeled data to find hidden patterns and anomalies.
- Semi-Supervised Learning: Combine limited quantities of labeled data with large amounts of unlabeled data to improve learning accuracy.
4. Model Evaluation:

Evaluating the model's performance involves assessing it using either cross-validation (holdout) or k-fold cross-validation methods. This evaluation is carried out utilizing metrics like accuracy, recall, precision, and detection time, among others. This process is essential for understanding the effectiveness of the model, as outlined in the discussion on IDS performance evaluation.

3. OPTIMIZATION ALGORITHMS

An optimization algorithm is a computational method that iteratively seeks the optimal solution to a problem within a defined search space, often adjusting candidate solutions based on specific criteria. These include both classical and stochastic techniques, the latter further divided into heuristic and metaheuristic approaches [5], as shown in Figure 4 such as evolutionary algorithms inspired by natural processes [6]. Metaheuristics involve two stages: exploration (a global search for promising areas) and exploitation (refining solutions in those areas) [7], balancing both is essential for efficiency [8]. In cybersecurity, these algorithms address challenges like network defense and vulnerability assessment, enhancing system resilience and resource allocation. This article focuses on Swarm Intelligence Algorithms, which use simple agents that locally interact and share information to optimize search processes. Hundreds of such algorithms have been developed for tasks like feature selection and extraction in Intrusion Detection Systems (IDS). Figure 3 represents the general steps in applying an optimization algorithm within an IDS framework which following these steps:

- Begin the optimization process.
- Create an initial population of candidate solutions (swarm, particles; chromosomes, bees, among others).
- Evaluate the fitness of each candidate solution based on a predefined fitness function (e.g., detection accuracy, false alarm rate).
- Depending on the algorithm, select, update, or replace solutions based on their fitness.
- Evaluate if the stopping criterion has been met (e.g., maximum iterations, convergence threshold).
- If YES, proceed to output the best solution.
- If NO, continue to refine the population.
- Return the optimized solution, which can then be applied to the IDS for improved performance.

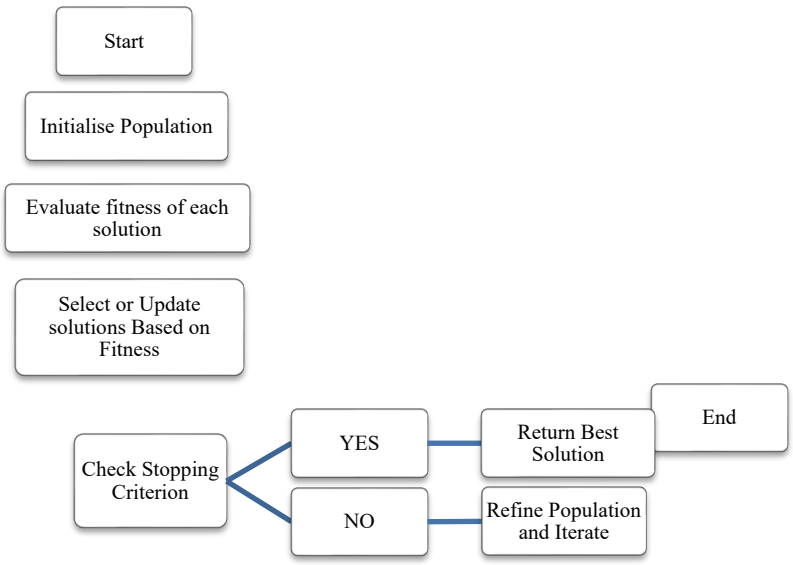

Figure 3. The general steps in applying an optimization algorithm within an IDS

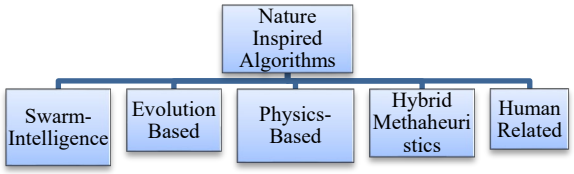

Figure 4. Classification of nature-inspired algorithms

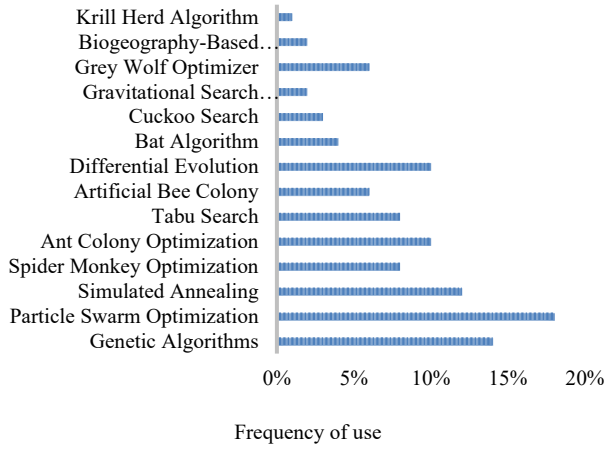

Figure 5. Frequency of Use of OA in cyber security

3.1. Genetic Algorithms (GA)

GA presented in the 1960s, imitates the principles of natural selection and evolutionary genetics, they have since found widespread application in various fields for addressing optimization and search challenges. The primary components of genetic algorithms include the following: Initially, a set of potential solutions is formed through either random generation or a heuristic technique Then, an evaluation function evaluates the performance of each one of the populations via the allocation of a fitness score that reflects how effectively the individual addresses the optimization problem. The fitness function guides the selection process by indicating which individuals are more likely to survive and reproduce. Following selection, crossover occurs, where genetic information from two parent individuals is combined to create offspring. This process involves swapping or merging segments of the parent chromosomes to generate new solutions. Crossover helps to explore the search space and potentially produce better solutions. Mutation introduces random alterations to the offspring chromosomes, helping to maintain genetic diversity within the population.

3.2. Particle Swarm Optimization (PSO)

PSO has evolved into a powerful optimization algorithm, drawing inspiration from the collective behaviors of organisms in nature, observed in bird flocking and fish schooling, where individuals synchronize their movements to achieve shared goals. At the beginning of the optimization process, particles are randomly initialized within the problem space generated by randomly distributing 1 and 0. Each particle changes its position iteratively shared within the swarm (referred to as global best). This adjustment is performed by updating the velocity of the particle and the position of every particle. The current position i and its velocity are expressed in Equations (6) and (7) [9]:

$$x_k = \{x_{k1}, x_{k2}, x_{k3}, \cdots, x_{kd}\} \tag{6}$$

$$v_k = \{v_{k1}, v_{k2}, v_{k3}, \cdots, v_{kd}\} \tag{7}$$

The velocity of the particle and the position k are calculated by Equation (8).

$$\begin{aligned} v_{kd}^{t+1} &= w \times v_{kd}^{t} + c_1 \times r_1 \times \left(p_{kd} - x_{kd}^{t}\right) + c_2 \times r_{2k} \times \left(p_{gd} - x_{kd}^{t}\right) \\ x_{kd}^{t+1} &= v_{kd}^{t} + v_{kd}^{t+1} \end{aligned} \tag{8}$$

The movement of each particle is influenced by two main factors:

- Personal Best (*pBest*): This represents the best solution that a particle has encountered thus far in its search history.
- Global Best (*gBest*): This signifies the suitable solution discovered by any of them within the entire population.

3.3. Simulated Annealing (SA)

SA, first has been widely employed since then to solve a range of combinatorial optimization problems. SA achieves a stable, low-energy crystalline structure. The SA algorithm follows these steps:

- Initialization: the process begins with an initial solution S=S_0 for the optimization problem. This starting solution can be created either randomly or using a heuristic approach.
- Temperature Schedule: Define a temperature reduction function using an alpha parameter. Generally, three main types of temperature reduction rules are commonly used:

• Linear Reduction Rule: $t = t - \alpha$

• Geometric Reduction Rule: $t = t \times \alpha$

• Slow-Decrease Rule: $t = \frac{t}{1+\beta t}$

Each reduction rule decreases the temperature at varying rates, and each method is more effective for optimizing different types of models. For the third rule, an arbitrary constant β is used at the initial temperature, which then undergoes n iterations before the temperature is reduced according to a factor α. This process continues until termination conditions are met. Termination conditions might include reaching a specific temperature, achieving a certain performance threshold with the given parameters, or other criteria. The relationship between time and temperature, as well as the speed of temperature reduction, as well as the entire procedure, is referred to as the annealing schedule.

- Neighbor Generation: Given the proximity of the N(s) solutions, one of these neighboring solutions is selected. The neighborhood of a solution consists of all solutions that are close to it.
- Acceptance Criterion: The neighboring solution an objective function is used to assess it, and a decision is made on whether to accept it as the updated solution.
- Termination: Upon reaching the maximum iterations, falling below a minimum temperature, or finding an acceptable solution.

3.4. Ant Colony Optimization (ACO)

The key concept behind ACO is the emulation of the pheromone-based communication among ants in real ant colonies. Ants communicate with each other through the deposition and detection of chemical pheromones on the paths they traverse. This communication mechanism enables them to collaboratively discover the most efficient route between their nest and a food source. Here's a general overview of how the Cuckoo Search algorithm works:

• Construction of Solutions: Initially, a population of artificial ants is distributed across the problem space. Each ant builds a candidate solution by iteratively choosing edges (or components) according to probabilistic rules.

• Pheromone Update: After constructing a solution, each ant deposits pheromone on the edges it traverses.

• Solution Evaluation: Once all ants have constructed their solutions. Each solution's quality is evaluated through the use of an objective function.

• Global Pheromone Update: After evaluating the solutions, the additional pheromone is deposited globally on the edges according to the effectiveness of the solutions. This global update of pheromones strengthens the routes corresponding to superior solutions and promotes the investigation of potentially fruitful areas within the solution space.

• Iteration: The process of constructing solutions, updating pheromones, and evaluating solutions is repeated until reaching the termination criteria.

• Ant Exploration: Ants use a combination of pheromone paths and heuristic data to direct their exploration of the solution space. This balance between the exploitation of known good solutions and the exploration of new areas allows ACO to effectively search for optimal solutions.

• Termination: Reaching the termination criteria.

3.5. Tabu Search (TS)

TS is a metaheuristic OA designed to address combinatorial and discrete optimization challenges. It was presented by Fred Glover in the late 1980s. TS is inspired by the notion of "tabu" in decision-making, where certain moves or solutions are temporarily prohibited to prevent the algorithm from revisiting previously visited states or getting stuck in local optima.

The core idea behind Tabu Search is to iteratively explore the vicinity of the current solution while maintaining a memory of recent moves to guide the search process. This memory, known as the "tabu list," contains information about moves that are temporarily forbidden or penalized to promote exploration of various areas within the search space. Here's how the Tabu Search algorithm typically operates:

• Initialization: They begin with a starting point for the optimization problem.

• Neighborhood Exploration: At each iteration, they generate a set of candidate solutions by making small modifications to the current solution. These modifications depend on the specific problem being solved and can include operations such as swapping, insertion, deletion, or other local moves.

• Tabu List Management: They evaluate each candidate solution and update the tabu list accordingly. The tabu list contains information about recent moves that are prohibited for a specific number of iterations or until specific conditions are fulfilled. This prevents the algorithm from revisiting the same solutions repeatedly and encourages the investigation of new areas within the search space.

• Aspiration Criteria: Occasionally, a candidate solution may be allowed even if it violates a tabu condition if it offers a significant improvement over the current solution. This is known as an aspiration criterion and helps the algorithm escape local optima.

• Solution Selection: They choose the best candidate solution from the neighborhood based on an objective function or evaluation criterion. This solution becomes the new current solution for the next iteration.

• Termination: Reaching the termination criteria.

3.6. Artificial Bee Colony (ABC)

The ABC algorithm emulates the food-foraging behavior of honeybees, where they seek out optimal food sources by utilizing familiar ones and exploring new potential sources. The ABC algorithm comprises three primary components: employed bees, onlooker bees, and scout bees, each playing a distinct role in the search process.

- Employed Bees: Employed bees concentrate on utilizing familiar food sources. Within the optimization context, they represent candidate solutions, with each employed bee associated with a specific solution within the search space. Employed bees iteratively enhance their solutions by investigating the vicinity of their current solutions.
- Onlooker Bees: Onlooker bees watch the waggle dance performed by employed bees to determine which food sources to utilize based on the information they receive.
- Scout Bees: Scout bees play a crucial role in discovering new food sources when the solutions from employed bees become stagnant or less effective. In the ABC algorithm, scout bees randomly search through the solution space to identify new candidate solutions.

The ABC algorithm functions through the following iterative process:

- Initialization: Initialize the population of employed bees using random solutions.
- Employed Bee Phase: Employed bees enhance their solutions by iteratively exploring the neighborhood of their current solutions.
- Onlooker Bee Phase: Onlooker bees choose food sources using a probabilistic approach that considers the quality of the employed solutions bees' solutions and update their solutions accordingly.
- Scout Bee Phase: Scout bees replace solutions that have not been improved for several iterations.
- Termination: Reaching termination criteria.

3.7. Differential Evolution (DE)

DE effective for solving continuous and differentiable optimization problems.

DE operates following these steps:

- Initialization: Randomly initialize a group of possible solutions within the boundaries of the search space.
- Mutation: Generate new candidate solutions (mutants) by perturbing the current solutions in the population. This is typically achieved by creating a mutant vector for each solution in the population using the formula: $mutant = target + F \times (rand_1 - rand_2)$ where the target is the current solution being mutated, $rand_1$ and $rand_2$ are solutions randomly selected from the population (distinct from the target), and F is a scaling factor controlling the magnitude of the perturbation.
- Crossover: Generate trial solutions by combining the mutant vectors with the original solutions in the population. This step ensures that useful information from the original solutions is preserved while incorporating new information from the mutants. Various crossover strategies, such as binomial or exponential crossover, can be used.
- Selection: They select the next generation of solutions according to their fitness values. They replace the original solutions in the population.
- Termination: Reaching termination criteria.

3.8. Grey Wolf Optimizer (GWO)

GWO is a metaheuristic algorithm introduced in 2014, is inspired by the hunting strategies of grey wolves. The algorithm iteratively refines these solutions by emulating the wolves' cooperative and competitive hunting process.

The GWO operates following these steps:

- Initialization: Randomly initialize the position of the (α, β, δ, Ω) wolves within the search space.
- Hunting Process: At each iteration, update the locations of the (α, β, δ, Ω) wolves inspired by their hunting behavior:

- Alpha Wolf (α): Reflects the optimal solution located up to this point. It guides the hunting process and coordinates the movements of other wolves.
- Beta Wolf (β): Represents the second-best solution. It supports the alpha wolf and assists in exploring promising regions within the search space.
- Delta Wolf (δ): Represents the third-best solution which explores the space independently and tries to improve upon the solutions found by the alpha and beta wolves.
- Omega Wolf (ω): Represents the worst solution which explores the space extensively.

- Update Positions: They modify the locations of the (α, β, δ, Ω) wolves according to their hunting behavior. This typically involves adjusting the positions using mathematical equations that simulate the movement and cooperation among the wolves.
- Evaluation: They evaluate the fitness of each wolf (solution).
- Update Pack Hierarchy: Update the hierarchy of the pack in terms of the fitness values of the wolves. The (α, β, δ, Ω) wolves are reallocated according to their fitness, ensuring that the alpha wolf continues to lead the pack.
- Termination: Reaching termination criteria.

3.9. Bat Algorithm (BA)

BA imitates the behavior of bats. In nature, bats use ultrasonic pulses for navigation and prey detection, and this natural process is emulated in the algorithm. The Bat Algorithmics the hunting behavior of bats by employing echolocation-inspired mechanisms for investigating the search space and finding optimal solutions to optimization problems. Bats use echolocation to locate prey, determine their distance, and adjust their flight path accordingly.

- The BA works following these steps:

- Initialization: Initializing a population of bats within the search space.
- Emission of Ultrasonic Pulses: At each iteration, each bat emits ultrasonic pulses (i.e., echolocation) to investigate the search space. The frequency and loudness of the pulses rely on the quality of the bat's current solution and its proximity to the optimal solution identified so far.
- Movement towards Prey: Bats modify their positions based on the information gathered from the emitted pulses. Bats with higher loudness (indicating higher fitness) move

to the best solution discovered so far, while bats that emit lower loudness engage in a more thorough exploration of the search space.

- Frequency Adjustment: Bats adjust their pulse frequencies dynamically.
- Local Search: By applying deterministic or stochastic operators to refine the positions of the bats locally.
- Update Best Solution: Refresh best solution found by bat fitness.
- Termination: Reaching termination criteria

3.10. Cuckoo Search (CS)

CS is inspired from the brood parasitism practices of certain cuckoo bird species, which lay their eggs in the nests, relying on these hosts to incubate and raise their young.

Below is a general outline of how the Cuckoo Search algorithm operates:

- Initialization: Start by randomly positioning a population of nests.
- Egg Laying and Abandonment: At each iteration, cuckoos lay eggs (i.e., create new solutions) probabilistically in nests based on their quality. Cuckoos lay eggs in better nests with a higher probability, mimicking the principle of elitism. However, some cuckoos may also lay eggs in random nests to introduce diversity and exploration. Additionally, a fraction of nests is abandoned with a certain probability, simulating the rejection of less promising solutions.
- Egg Replacement: After laying eggs, cuckoos evaluate the quality of the nests (solutions) and replace the eggs in nests with higher-quality eggs.
- Local Search: Applying deterministic or stochastic operators to refine the positions of the nests locally.
- Update Best Solution: Updating optimal solution based on nest fitness.
- Termination: Repeat the egg laying and replacement until reaching termination criteria.

4. DATASETS

The datasets used in the literature are of two types: current and simulated. Current means that data is collected from physical objects. However, simulating means that the researchers created the database for experiments [10].

DARPA 98 dataset: This dataset is based on communications between source and destination IP addresses between 214000 nodes. It contains 38 attack types divided into four primary categories: Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probing.

- KDDCuP 99 dataset: It is a new version of the DARPA 98, with records categorized into 41 features. It suffers from redundant and duplicate records and does not accurately reflect real network traffic. Despite these issues, it remains widely used for experimenting with new intrusion detection systems.

- NSL-KDD dataset: This dataset was developed in 2009 to address issues present in the KDD Cup dataset, including redundant and duplicated entries in both the training and testing datasets. Its goal is to remove duplicate record features and reduce the data size, thereby improving the accuracy of models.

- KYOTO dataset: Researchers at Kyoto University have implemented a range of systems, including honeypots and darknet sensors, across five networks both within and outside the university. The dataset is derived from actual network traffic., labeled as normal, known attacks, and unknown attacks. It contains 14 statistical features extracted from this data.

UNSW-NB 15 dataset: This data set has been produced by the Australian Centre for Cyber Security (ACCS) and includes two million records and features multiple types of attacks, such as Backdoors, DoS, Worms, Port Scans, and more.

- DEFCON dataset: We make a distinction between the DEFCON-8 data set, as well as the DEFCON-10 dataset, which was established in 2002. The DEFCON 10 data set includes attacks such as port scans and sweeps, malicious packets, exploits to gain administrative privileges, and FTP attacks carried out through the Telnet protocol. This dataset is utilized for assessing alert correlation techniques.

- CAIDA dataset: The CAIDA (Cooperative Association for Internet Data Analysis) dataset refers to various datasets provided by CAIDA, a collaborative organization that conducts research on Internet traffic, topology, routing, security, and related aspects. CAIDA datasets are widely used in network research and include information. It comprises three distinct types of datasets: CAIDA OC 48, CAIDA DDoS attack dataset, and CAIDA internet trace 2016.

- TWENTE dataset: Produced by Twente University in 2009, this dataset includes data collected from a honeypot network with the help of NetFlow, with services such as OpenSSH, the Apache web server, and ProFTP are configured to use auth/ident on port 113. It features labeled data and provides a more realistic traffic profile, although it is limited by its volume and the diversity of attacks. Additionally includes certain alert traffic that is unknown and uncorrelated.

- AWID dataset: Aegean Wi-Fi Intrusion Dataset: is a dataset that is publicly accessible. It was created based on a small network environment with 11 clients, approximately 37 million packets were captured based on their packet format. From each packet, 156 features were extracted. They execute 16 specific attacks targeting the network to extract malicious network traffic. AWID is a labeled data set. It is separated into a training dataset and a testing dataset.

- ADFA 2013 dataset: Collected in 2013, the researchers set up Apache, MySQL, and Tikiwiki to deliver web services, a database server, remote access, and an FTP server for the development of the ADFA dataset. It comprises normal data for training and validation purposes.

- BOT-IoT dataset: Developed within a realistic network setting at UNSW Canberra's Cyber Range Lab, this dataset integrates both normal and botnet traffic. It includes various types of attacks, including DDoS, DoS, OS vulnerabilities, service scanning, keylogging, and data exfiltration.

Table 1. Pros and Cons among the most prevalent nature-inspired optimization techniques

OA	Pros.	Cons.	Specific Application	Performance Outcome
GA	• Effective for problems with large search spaces • Adept at solving problems with multiple parameters • Generation and identification of acceptable solutions to complex optimization problems	• Slow convergence • Computational complexity • Choosing a good evaluation function is also critical. It must take into account the right parameters of the problem. • Exhibit relatively weak local search capabilities and are less effective in exploitation	• Network anomaly detection • Feature selection in IDS	• Improved detection accuracy • Reduced overfitting
PSO	• Characterized by its simplicity and efficiency • No explicit knowledge of the objective function or its derivatives is required. Instead, it leverages the principles of self-organization and social learning to navigate complex search spaces efficiently. • Easy to implement • Prized for its speed of convergence	• It can sometimes get stuck in optimal spaces.	• IoT • Smart grid IDS • Real-time threat detection	• Increased detection • Accuracy • Low false positive rates
SA	• Easy to implement and use. • Provide effective solutions for a diverse array of problems. • SA can escape local optima and find global optima with a certain probability	• Slow for complex problems • Execution can take a long time if the annealing program is very long. • There are many adjustable parameters in this algorithm.	• IDS parameter optimization, • Anomaly detection	• Stable parameter tuning • Moderate detection improvement
SMO	• It is known for the balance it strikes between exploring and exploiting as well as its ability to adapt to different types of optimization problems.	• High complexity • Few applications	• IDS for hierarchical networks, resource allocation	• Enhanced detection accuracy • Effective feature selection
ACO	• They are particularly well-suited for problems with large solution spaces and complex constraints, where traditional optimization techniques may struggle	• Slow convergence • Dependence on the quantity of pheromones	• Routing-based IDS in IoT • Anomaly detection in resource-constrained systems	• Improved anomaly detection • Good adaptability for IoT
TS	• Investigate complex solution spaces and discover near-optimal solutions for various optimization problems, including combinatorial optimization, scheduling, and routing. • Its simplicity is a key advantage, as it requires only a few control parameters. This reduces the need for extensive parameter tuning, which is often a challenge in applying metaheuristics.	• May not be suitable for large-scale distribution network planning, especially when an appropriate initial solution is difficult to estimate. Additionally, there is no guarantee of achieving a global optimum	• Hybrid IDS optimization • Dynamic intrusion response	• Enhanced adaptability • Stable detection accuracy
ABC	• It has a reputation for simplicity, efficiency, reliability., and capacity to discover high-quality solutions, particularly in problems with large solution spaces.	• May converge slowly for some problems • Sensitive to population parameters	• Lightweight IDS for IoT • Feature selection	• Efficient resource use • Increased detection rates
DE	• Simplicity, robustness, and efficiency. It is particularly well-suited for optimization of Nonlinear and nonconvex objective function problems • Simple and efficient algorithm for continuous problems • Often used for optimizing the weights of neural networks.	• May require extensive testing to determine correct parameters. • Sensitive to the choice of mutation and crossing parameters	• IDS parameter tuning • Anomaly-based IDS in complex networks	• High detection accuracy • Reduced false positives
GWO	The algorithm is straightforward to implement • It is adaptable to various problems and scalable to different sizes • It requires very few parameters to be adjusted • The algorithm provides rapid convergence and a strong level of exploitation	• It is prone to becoming trapped in local optima • There is minimal information exchange among search agents • The algorithm often suffers from reduced diversity, leading to premature convergence	• Feature selection for IDS • Multi-modal IDS optimization	• Improved feature selection • Stable anomaly detection
BA	• It effectively balances exploration and exploitation within the search space, while also being simple and efficient.	• Premature convergence which requires fine tuning to avoid it	• IDS for critical infrastructure • Anomaly detection in mixed datasets	• Moderate accuracy improvement, efficient resource usage
CS	• Compared to many other search techniques, it uses fewer control parameters	• Can be slow for large research spaces	• Hybrid IDS in multi-layered networks, adaptive anomaly detection	• Enhanced accuracy • Adaptability in layered security setups

5. LITERATURE REVIEW

AI and OA are applied by many researchers on their IDS models to improve their accuracies. This section will demonstrate many kinds of research that illustrate the hybridization of AI and OA:

- Shokoohsaljooghi and Mirvaziri presented an advanced IDS founded leverages the combination of PSO and neural networks algorithms applied on KDDCUP99, NSL-KDD, and CIDD datasets which a preprocessing is previously done on these datasets [11].
- A.H.M. Ghanem and Jantan introduced a novel method for implementing intrusion detection using the ABC algorithm in conjunction with Monarch Butterfly Optimization (MBO) for preselecting the suitable bias and weight for ANN to enhance the precision of classification to distinguish between malicious and non-malicious. They used three databases to evaluate the model: KDD CuP 99, also ISCX 2012, and UNSWNB-15 reached an accuracy of (87.62%, 100%, 95.72% respectively), and a False Alarm Rate of (0.2309, 0, 0.507, respectively) and a Detection Rate of (97.39%, 100%, 96.41%, respectively) [12].
- Moghanian, et al. introduced an IDS based on ANN as a learning technique and a Grasshopper Optimization Algorithm (GOA) as an OA to reduce the error rate in intrusion detection within the neural network by selecting suitable weight and bias. They used two famous databases KDD CuP 99 and UNSWNB-15 and reached an accuracy of (95.15% and 98.88%) a specificity of (93.17% and 98.09%) and a sensitivity of (89.25% and 98.14%) respectively [13].
- Kanna and Santhi introduced a robust hybrid IDS model designed with a MapReduce framework that incorporates an architecture combining Black Widow Optimization with Convolutional and Long Short-Term Memory (BWO-CONV-LSTM) networks. The model utilizes Convolutional and LSTM neural networks, with hyperparameters optimized by the Black Widow Optimization (BWO) algorithm to create an efficient architecture. The NSL-KDD, ISCX-IDS, UNSWNB-15, and CSE-CIC-IDS 2018 datasets were employed to evaluate the model's performance. model. They reach an accuracy of (98.67%, 97.003%, 98.66%, 98.25% respectively) [14].
- Imran, et al. Proposed a novel method for anomaly detection using ANN optimized cuckoo search algorithm, the model was tested on the NSL KDD dataset and reached an accuracy of 99.8% [15].
- Kadry, et al. Present a model for accurately detecting attacks using a Quantum Neural Network (QNN) combined with a Whale Optimization Algorithm (WOA)-based IDS framework. This approach offers a robust solution for real-time intrusion detection. Additionally, the research demonstrates that this model enhances secure data storage and addresses security concerns effectively [16].
- Salih, et al. proposed a Feature Selection based on multi-phase particle swarm Optimization to handle missing values and remove redundant and irrelevant attributes [17].
- Davahli, et al. Proposed the GA-GWO framework, which hybridizes a combination of GA and GWO used in IoT intrusion detection systems (IoTIDS) to reduce the dimensionality of large-scale wireless network traffic by strategically selecting the most relevant features. The model is tested on AWID (Agean Wifi Intrusion Dataset) and reached an accuracy of 99.10%, also a Detection Rate of 99.32%, a Precision of 96.03%, an F1-score of 97.64%, a False Positive Rate of 0.69% [18].
- Elsedimy, et al. Presented a novel intrusion detection model that combines the Quantum Support Vector Machine (QSVM) with the Improved Grey Wolf Optimizer (IGWO) algorithm utilized to improve detection accuracy and minimize false positive alarms in Host Intrusion Detection Systems (HIDS) [19].
- Tajari introduced an innovative feature selection technique for intrusion detection that incorporates the Gravity Search Algorithm (GSA), PSO, and Biogeography-Based Optimization (BBO) techniques, leveraging the strengths of these methods to enhance performance in classification tasks. The model was tested on UNSW-NB 15 and reached an Accuracy of 99.8%, F1-score of 99.1%, Specificity of 99.6%, Sensitivity of 99.8%, and Precision of 99.3% [20].

6. DISCUSSION

This section of the article aims not only to evaluate the effectiveness of optimization algorithms (OAs), but also to discuss their practical applicability and potential challenges encountered during their implementation. A hybrid methodology combining GA and PSO leverages the extensive search capabilities of GA and the swift convergence of PSO. Similarly, integrating ant colony optimization (ACO) with differential evolution (DE) achieves equilibrium between exploration and exploitation, providing a deeper investigation of the solution space. The synergy between these algorithms significantly enhances intrusion detection systems (IDS), improving detection accuracy, reducing false positives, and optimizing resource allocation.

A PSO-based optimization algorithm was integrated with ML algorithms to improve anomaly detection in smart grids. PSO was used to optimize feature selection and parameter tuning, enhancing the model's detection accuracy and reducing false alarm rates. This approach increased detection rates by approximately compared to traditional methods, with PSO enabling efficient processing of high-dimensional data typical of smart grid networks. The deployment also demonstrated robustness against large-scale Distributed Denial of Service (DDoS) attacks, illustrating PSO's practical value in safeguarding critical infrastructure.

GA was applied to optimize the configuration of an anomaly-based IDS in an enterprise setting, specifically to tune the hyperparameters of a neural network model and select the most relevant features from network data. By employing GA, the IDS reduced false positives and improved detection accuracy. GA's iterative nature allowed for continuous optimization, ensuring that the IDS adapted effectively to the evolving network environment, which is particularly beneficial in complex corporate settings.

ACO was used to develop a routing-based IDS in an IoT environment, optimizing paths based on data flow patterns to detect and mitigate anomalous behavior. ACO also assisted in feature selection, reducing computational load on the IoT devices while maintaining detection effectiveness. ACO achieved a balance between computational efficiency and accuracy in anomaly detection rates. The algorithm's focus on path optimization aligned well with the low-power requirements of IoT devices, making it feasible for large-scale deployment in resource-constrained settings like smart cities.

Optimizing deep neural network (DNN) hyperparameters using PSO can reduce training time and improve intrusion detection accuracy. ACO can also be used to optimize IDS detection for emerging threats. Methods like simulated annealing (SA) and cuckoo search (CS) are effective in avoiding local minima, which is advantageous in finding optimal solutions within complex loss landscapes. SA, in particular, improves the robustness of machine learning (ML) models for classification by efficiently exploring the search space.

However, optimizing hyperparameters in deep learning (DL) models using GA can be computationally expensive, especially for large datasets. Some optimization algorithms, such as PSO and the bat algorithm (BA), may suffer from issues like premature convergence or instability, which can affect the reliability of security models.

GA and PSO are effective in finding global optimal solutions, making them useful for anomaly detection and hyperparameter optimization in ML/DL models. Meanwhile, ACO and the ABC algorithms are employed for resource optimization, relevant for network security management and intrusion detection. Algorithms like DE and GWO enhance the convergence and accuracy of classification and regression models used in threat detection, while SA and tabu search (TS) are employed for local search, making them effective in finding optimal configurations for security systems.

As shown in Table 2, PSO achieves the highest accuracy (99.66%), followed by GA. BA and CS are flexible and adaptive, capable of adjusting to different types of data and threats in dynamic security environments. However, these algorithms, including GA and PSO, can be computationally expensive and resource-intensive. Some algorithms, such as PSO and DE, may get stuck in local optima, requiring hybrid strategies to improve overall convergence. Others, like GWO and BA, may need specific adaptations to handle large-scale data.

Integrating these OAs with complex ML/DL models introduces additional complexity, requiring advanced expertise in both optimization and machine learning. Although such integration leads to significant improvements in performance and efficiency, it also poses challenges in terms of complexity, computational cost, and scalability, as Figure 6, a comparative evaluation of different optimization algorithms is provided in Table 3.

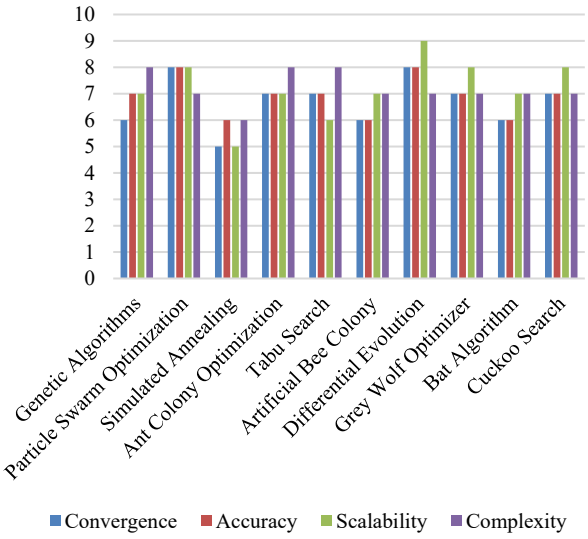

Figure 6. Comparison of Hybridized Optimization Algorithms with AI in IDS for cybersecurity

7. RESEARCH CHALLENGES AND FUTURE DIRECTIONS

In the dynamically developing field of cybersecurity, the integration of optimization algorithms presents both significant challenges and exciting opportunities for advancement. As cyber threats grow more sophisticated, algorithms are urgently needed to efficiently analyze vast datasets, detect anomalies, and react to attacks in real-time. Future investigations should prioritize enhancing the flexibility of these algorithms to cope with emerging threats, optimizing the equilibrium between security and system performance, and ensuring the resilience of security frameworks against diverse attack vectors. Additionally, exploring the application of machine learning in conjunction with optimization techniques can lead to more proactive and predictive cybersecurity measures. By addressing these challenges, we can pave the way for innovative solutions that strengthen our defenses in an increasingly interconnected digital environment:

- Deploying IDS for identifying suspicious traffic and various types of new attacks in the network is an added value, but unfortunately to test the efficiency of the IDS and validate it, we need a valid cyber security dataset that contains an actual attack is an important challenge, since there is the oldest dataset. So, the challenge is developing a new dataset containing the new network traffic to train a model with high accuracy.
- The KDD Cup 99 and NSL-KDD datasets are frequently used benchmarks in IDS research due to their comprehensive coverage of various network intrusions and ease of access. The KDD Cup 99 dataset, widely adopted for evaluating IDS models, provides a broad array of attack types, including DoS, R2L, and U2R attacks, making it suitable for baseline IDS performance comparisons. However, it suffers from significant limitations, such as outdated attack signatures, redundant records, and an imbalance in attack representation, which may not accurately represent modern network threats.

Table 2. Accuracy comparison of recent Intrusion Detection System

Related Work	Classifier	Optimization Algorithms	Dataset	Evaluation Metric
[9]	Decision tree	Correlation Coefficient Algorithm and Cuttlefish Algorithm	• KDD CuP 99	• Accuracy: 95.03% • Detection rate: 95.23% • False positive rate: 1.65%
[12]	ANN	Artificial Bee Colony (ABC) and Monarch Butterfly Optimization (MBO)	• KDD CuP 99 • ISCX 2012 • UNSW-NB 15	KDD Cup 99: • Accuracy: 87.62% • FAR: 0.2309 • DR: 97.39% ISCX 2012: • Accuracy: 100% • FAR: 0 • DR: 100% UNSW-NB 15: • Accuracy: 95.72% • FAR: 0.0507 • DR: 96.41%
[13]	ANN	Grasshopper Optimization Algorithm (GOA)	• KDD CuP 99 • UNSW-NB 15	KDD CuP 99: • Accuracy: 95.15% • Specificity: 93.17% • Sensitivity: 89.25% UNSW-NB 15: • Accuracy: 98.88% • Specificity: 98.09% • Sensitivity: 98.14%
[21]	Convolutional and Long Short Term Memory (CON-LSTM)	Black Widow Optimized (BWO)	• NSL –KDD • ISCX-IDS • UNSW-NB 15 • CSE-CIC-IDS 2018	• Accuracy: • NSL –KDD: 98.67% • ISCX-IDS: 97.003% • UNSW-NB 15: 98.66% • CSE-CIC-IDS 2018: 98.25%
[22]	Deep Neural Network (DNN)	Auto Encoder (AE)	• NSL KDD • CICIDS 2017	NSL KDD: • Accuracy: 98.43% • Precision: 99.22% • Recall: 97.12% • F1-score: 98.57% CICIDS 2017: • Accuracy: 98.92% • Precision: 97.45% • Recall: 98.97% • F1-score: 98.35%
[23]	Evolutionary Neural Network (ENN)	Mutation Cuckoo Fuzzy (MCF) → is a modified Cucko Search Algorithm (CSA)	NSL KDD	• Accuracy: 98.81% • Detection Rate: 97.25% • False Alarm Rate: 0.022
[18]	Support Vector Machine (SVM)	Genetic Algorithm (GA) and Grey Wolf Optimizer → GA-GWO	AWID (Agean Wifi Intrusion Dataset)	• Accuracy: 99.10% • Detection Rate: 99.32% • Precision: 96.03% • F1-score: 97.64% • FPR: 0.69%
[24]	Random Forest	PSO and Gray Wolf Optimization (GWO)	NSL KDD	• Accuracy: 99.86% • Precision: 99.94% • Recall: 99.94% • F1-Measure: 99.86%
[19]	Quantum Support Vector Machine (QSVM)	Improved Grey Wolf Optimizer (IGWO)	Bot-IoT dataset	• Accuracy: 99.11% • Precision: 99.45% • Recall: 99.34% • F1-score: 97.48%
[25]	Stacked ensemble Learning algorithm	Improved Grey Wolf Optimization (IGWO)	• Cooja Simulated Datasets • NSL KDD • UNSW-NB 15 • CICIDS2017 • MQTTSet	Cooja Simulated: Accuracy: 99.44%, Precision: 98.41%, Recall: 98.06%, F-score: 98.24%, Error Rate: 0.56% NSL KDD: Accuracy: 99.60%, Precision: 98.41%, Recall: 97.15%, F-score: 97.61%, Error rate: 0.40 UNSW-NB 15: Accuracy: 94.64%, Precision: 71.11%, Recall: 72.41%, F-score: 71.76%, Error rate: 5.36 CICIDS 2017: Accuracy: 99.90%, Precision: 98.69%, Recall: 91.59%, F-score: 95.01%, Error rate: 0.10 MQTTSet: Accuracy: 99.95%, Precision: 99.73%%, Recall: 98.75%, F-score: 99.24%, Error rate: 0.05
[26]	Enhanced Multiclass Support Vector Machine (EMSVM)	Orthogonal Learning Chaotic Grey Wolf (OLCGW)	• CIC-DDoS 2019 • TON-IoT	CIC-DDoS 2019 (Binary Classification): Precision: 92.65%, Recall: 96.95%, F1-score: 94.75% TON-IoT (Binary Classification): Precision: 96.24%, Recall: 98.71%, F1-score: 97.46%
[15]	ANN	Cuckoo Search Algorithm	NSL KDD	• Accuracy: 99.8%
[27]	Decision Three	Cuckoo Search	• NSL KDD • KDD CuP 99 • Bot net ISCX 2017	NSL KDD: • Accuracy: 99.60% KDD CuP 99: • Accuracy: 99.94% Bot net ISCX 2017: • Accuracy: 99.98%
[28]	Support Vector Machine (SVM)	• Hyper Clique-Improved Binary • Gravitational Search Algorithm (HC-IBGSA)	• NSL KDD • UNSW-NB 15	NSL KDD: • Detection rate: 98.72% • False Alarm rate: 1.27 UNSW-NB 15: • Detection rate: 98.47% • False Alarm rate: 2.18
[20]	K-means	Gravity Search Algorithm (GSA) and Particle Swarm Optimization (PSO) and Biogeography Based Optimization (BBO)	• UNSW-NB 15	• Accuracy: 99.8% • F1-score: 99.1% • Specificity: 99.6% • Sensitivity: 99.8% • Precision: 99.3%
[29]	fuzzy C-means clustering algorithm (FCM)	Krill Herd Algorithm (KHA)	KDD CuP	• Detection Rate: 90.73% • False Alarm Rate: 0.0312%
[30]	Extreme Learning Machine (ELM)	Krill Herd Algorithm (KHA)	NSL KDD	• Accuracy: 87% • False Alarm Rate: 2% • Detection Time: 1.5 (sec)

Table 3. Comparative analysis of different optimization algorithms

	Convergence	Accuracy	Scalability	Complexity	Real-Time Application	Adaptability to Emerging Threats
GA	Fast but can get stuck in optima premises	Moderate to high, strongly depends on the chosen parameters.	Good for medium size problems	High temporal complexity, requires frequent evaluation of solutions	Iterative processing can introduce latency, impacting real-time detection	Moderate; adaptable through parameter tuning but requires frequent retraining
PSO	High temporal complexity, requires frequent evaluation of solutions	High with well-adjusted parameters	Good, capable of dealing with large problems	Moderate complexity, simple to implement	Fast convergence benefits real-time applications but risks premature convergence	Limited adaptability; may require frequent updates to handle novel attacks
SA	Slow, but able to escape the local optima	Good for medium size problems	Moderate, less effective for very large problems	High temporal complexity, simple to implement	Less suitable for real-time applications due to cooling process delays	Limited; not inherently adaptive, requiring tuning for evolving threats
ACO	Moderate, balance between exploration and exploitation	High for combinatorial problems	Good for medium size problems	High complexity, requires a large number of iterations	Not ideal for real-time due to iterative nature and high processing demands	Moderate adaptability with adjustments, though limited to simpler scenarios
TS	Fast, effective to avoid local optima	High for well-defined problems	Moderate to good	High complexity, requires effective management of the tabu list	Prone to delays due to memory-based search process	Low; struggles with changing data or attack patterns due to static search
ABC	Fast, effective for initial exploration	Moderate, upgradable with hybrid variants	Good	Moderate complexity, easy to implement	Limited real-time applicability due to repetitive evaluations	Limited adaptability; benefits from tuning but sensitive to parameter changes
DE	Fast and robust	High, especially for continuous problems	Great	Moderate complexity requires good parameterization	Iterative process impacts real-time suitability	Moderate; adaptable with parameter adjustments but sensitive to new threats
GWO	Fast, with a good balance between exploration and exploitation	High	Great	Moderate, recent, and growing complexity in terms of popularity	Limited real-time use as convergence speed is algorithm-dependent	Moderate adaptability; may adjust to changes with hyperparameter tuning
BA	Fast, effective to escape the local optima	Moderate to high	Moderate to good	Moderate complexity	Limited in real-time applications due to iteration requirements	Low adaptability; requires significant adjustments for new threats
CS	Fast, with good exploration ability	High for continuous problems	Great	Moderate complexity, based on a simple method	Slower real-time response due to iterative nature	Limited adaptability; may require tuning or hybrid models for evolving threats

Similarly, the NSL-KDD dataset improves upon KDD Cup 99 by reducing redundant entries and balancing classes, thereby improving evaluation fairness. Despite these enhancements, NSL-KDD remains limited in its applicability to real-world scenarios, as it does not reflect contemporary attack vectors or the complex, evolving nature of cyber threats in modern networks.

- To better align IDS evaluations with present-day cybersecurity challenges, it is essential to consider modern datasets that encompass more recent and sophisticated attacks, as well as diverse traffic patterns. Publicly available alternatives, such as the CICIDS2017, UNSW-NB15, and CSE-CIC-IDS2018 datasets, provide more comprehensive coverage of recent attack methods, including botnets, infiltration, and web-based attacks, and include benign traffic across multiple protocols, making them more representative of today's network environments. Integrating these modern datasets into IDS evaluation practices could significantly enhance the relevance and robustness of IDS models, ensuring they are better suited to detect and respond to current cyber threats in real-world deployment.
- The AI models that rely on noisy data or social media data which suffer of missing values, outlier data, interoperability and data standardization which impact the accuracy of model. So, to create efficient AI models, a good preprocessing of data is required.
- The less training data to train AI models impacts the model's efficiency, which can suffer from overfitting or underfitting.
- AI models take a long time to train something which increases latency and reduces model efficiency and accuracy.
- Unlike machine ML, DL algorithms need less elaborate feature extraction.
- Attackers can inject malicious data to perturb the training process of ML or DL algorithms which affect decision-making.
- Premature convergence can lead to a less accurate final solution which often results in reduced accuracy.
- Detection models applied to the raw data set without pretreatment led to inaccurate detection. which proves that pre-treatment is an important step before model training.
- Several of the bio-inspired algorithms above still have high computational time.
- The models based on GWO are usually trapped in the local optima because of their limited diversity and tendency for premature convergence. This has led to the disregard of some informative features and, therefore, to inaccurate detection.
- A slow rate of convergence results in a longer construction time for the intrusion detection model by classifiers.

- The selection of inappropriate parameters in certain OA can lead to suboptimal performance.
- Selection of inappropriate parameters related to some of the OA, and reduction in the number of characteristics may result in a reduction in the accuracy of the IDS.
- The complexity of the IDS development and deployment process is increased by integrating OA with AI.
- The optimization and training of AI models can be resource-intensive and can require significant computing power and storage.
- High-quality, labeled datasets are essential for training accurate ML and DL models
- Training DL models requires significant computational power and storage.
- Creating a dataset that accurately represents the real environment and includes a diverse range of new attack types.
- Developing new models based on recent DL algorithms that trained on recent datasets based on new features.
- Given the complexity of models trained on large datasets, it is advisable to utilize a cloud platform for computational resources or a high-performance GPU platform.
- Employ advanced techniques for feature extraction aimed at enhancing the dataset, streamlining the training process, and reducing model complexity to achieve the best accuracy.
- Combining Genetic Algorithms (GA) with Grey Wolf Optimizer (GWO) into a GA-GWO hybrid addresses the challenges of limited diversity and early convergence related to GWO while enhancing GA's convergence speed. This approach helps avoid local optima and facilitates finding the global optimum.

8. CONCLUSION

This paper underscores the potential of integrating optimization algorithms (OA) with Machine Learning (ML) and Deep Learning (DL) techniques to enhance Intrusion Detection Systems (IDS) in cybersecurity, leading to advantages such as improved model accuracy, optimized hyperparameters, and increased detection efficiency. Initially, an extensive review of IDS concepts and classification schemes is presented, followed by a summary of popular optimization algorithms in cybersecurity, particularly for IDS applications. The comparative analysis highlights the strengths and limitations of each algorithm, considering factors like convergence, accuracy, scalability, and complexity.

The study reveals that the current trend of integrating DL and ML with OAs like Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) yields promising improvements in IDS performance, such as higher detection rates and reduced false alarms. Approximately 80% of reviewed methodologies utilized DL techniques, generally outperforming purely ML-based approaches in IDS accuracy. Notably, most studies employ legacy datasets like KDD Cup'99 and NSL-KDD for testing; however, these datasets lack relevance for detecting modern network attacks, limiting real-world applicability. Future research should consider evaluating IDS models on updated and diverse datasets that reflect contemporary cyber threats, enhancing model reliability in practical scenarios.

This review suggests that the selection of an OA should be guided by the specific requirements of the cybersecurity application, as each algorithm brings unique strengths suited to different types of IDS challenges. Although challenges remain such as maintaining scalability and relevance to current threats the integration of optimization algorithms with ML and DL techniques holds significant promise for real-world IDS deployments. Future research should focus on developing adaptable hybrid models that can leverage real-time data, address evolving threats, and balance detection accuracy with computational efficiency.

REFERENCES

[1] E. Sandhya, Annapurani Kumarappan, "Enhancing the Performance of an Intrusion Detection System Using Spider Monkey Optimization in IoT", IJIES, Vol. 14, No. 6, pp. 30-39, December 2021.

[2] S. Kumar, W.M. Lim, U. Sivarajah, J. Kaur, "Artificial Intelligence and Blockchain Integration in Business: Trends from a Bibliometric-Content Analysis", Inf. Syst. Front, Vol. 25, No. 2, pp. 871-896, Avril 2023.

[3] S. Moghanian, F. Bagheri Saravi, G. Javidi, E.O. Sheybani, "GOAMLP: Network Intrusion Detection with Multilayer Perceptron and Grasshopper Optimization Algorithm", IEEE Access, Vol. 8, pp. 215202-215213, November 2020.

[4] M.E. Mohadab, B. Bouikhalene, S. Safi, "Fake News Detection: A Data Mining Perspective", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 59, Vol. 16, No. 2, pp. 383-389, June 2024.

[5] M.O. Okwu, L.K. Tartibu, "Introduction to Optimization", Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory, and Applications, M.O. Okwu, et al., (Eds.), Cham: Springer International Publishing, pp. 1-4, 2021.

[6] O. Niyomubyeyi, T.E. Sicuaio, J.I. Diaz Gonzalez, P. Pilesjo, A. Mansourian, "A Comparative Study of Four Metaheuristic Algorithms, Amosa, Moabc, Mspso, and Nsga-II for Evacuation Planning", Algorithms, Vol. 13, No. 1, Art. January 2020.

[7] A.M. Ahmed, T.A. Rashid, S.A.M. Saeed, "Cat Swarm Optimization Algorithm: A Survey and Performance Evaluation", Computational Intelligence and Neuroscience, Vol. 2020, No. 1, pp. 485-489, 2020.

[8] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, "Slime Mould Algorithm: A New Method for Stochastic Optimization", Future Generation Computer Systems, Vol. 111, pp. 300-323, October 2020.

[9] A.G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review", Arch Computat Methods Eng, Vol. 29, No. 5, pp. 2531-2561, 2022.

[10] M.A. Umer, K.N. Junejo, M.T. Jilani, A.P. Mathur, "Machine Learning for Intrusion Detection in

Industrial Control Systems: Applications, Challenges, and Recommendations", International Journal of Critical Infrastructure Protection, Vol. 38, p. 100516, September 2022.
[11] A. Shokooh Saljooghi, H. Mirvaziri, "Performance Improvement of Intrusion Detection System Using Neural Networks and Particle Swarm Optimization Algorithms", Int. J. Inf. Tecnol., Vol. 12, No. 3, pp. 849-860, September 2020.
[12] W.A.H.M. Ghanem, A. Jantan, "Training a Neural Network for Cyberattack Classification Applications Using Hybridization of an Artificial Bee Colony and Monarch Butterfly Optimization", Neural Process Lett, Vol. 51, No. 1, pp. 905-946, Febuary 2020.
[13] S. Moghanian, F.B. Saravi, G. Javidi, E.O. Sheybani, "GOAMLP: Network Intrusion Detection with Multilayer Perceptron and Grasshopper Optimization Algorithm", IEEE Access, Vol. 8, pp. 215202-215213, 2020.
[14] P.R. Kanna, P. Santhi, "Hybrid Intrusion Detection Using MapReduce based Black Widow Optimized Convolutional Long Short-Term Memory Neural Networks", Expert Systems with Applications, Vol. 194, p. 116545, May 2022.
[15] M. Imran, S. Khan, H. Hlavacs, F.A. Khan, S. Anwar, "Intrusion Detection in Networks Using Cuckoo Search Optimization", Soft Comput, Vol. 26, No. 20, pp. 10651-10663, October 2022.
[16] H. Kadry, A. Farouk, E.A. Zanaty, O. Reyad, "Intrusion Detection Model Using Optimized Quantum Neural Network and Elliptical Curve Cryptography for Data Security", Alexandria Engineering Journal, Vol. 71, pp. 491-500, May 2023.
[17] M.A. Salih, M.M. Hamad, W.M. Jasim, "Optimization Feature Selection Techniques for Big Data Using Multi Phase Particle Swarm Optimization Algorithm", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issu 56, Vol. 15, No. 3, pp. 188-196, September 2023.
[18] A. Davahli, M. Shamsi, G. Abaei, "Hybridizing Genetic Algorithm and Grey Wolf Optimizer to Advance an Intelligent and Lightweight Intrusion Detection System for IoT Wireless Networks", J Ambient Intell Human Comput, Vol. 11, No. 11, pp. 5581-5609, November 2020.
[19] E.I. Elsedimy, H. Elhadidy, S.M.M. Abohashish, "A Novel Intrusion Detection System Based on a Hybrid Quantum Support Vector Machine and Improved Grey Wolf Optimizer", Cluster Comput, Vol. 27, No. 7, pp. 9917-9935, October 2024.
[20] T. Siahmarzkooh, Aliakbar, "An Improved K-Means Clustering Feature Selection and Biogeography Based Optimization for Intrusion Detection", International Journal of Web Research, Vol. 6, No. 2, pp. 57-66, December 2023.
[21] P.R. Kanna, P. Santhi, "Hybrid Intrusion Detection Using MapReduce Based Black Widow Optimized Convolutional Long Short-Term Memory Neural Networks", Expert Systems with Applications, Vol. 194, p. 116545, May 2022.
[22] A. Bhardwaj, V. Mangat, R. Ving, "Hyperband Tuned Deep Neural Network with Well Posed Stacked Sparse AutoEncoder for Detection of DDoS Attacks in Cloud", IEEE Access, Vol. 8, pp. 181916-181929, 2020.
[23] S. Sarvari, N.F. Mohd Sani, Z. Mohd Hanapi, M.T. Abdullah, "An Efficient Anomaly Intrusion Detection Method with Feature Selection and Evolutionary Neural Network", IEEE Access, Vol. 8, pp. 70651-70663, 2020.
[24] E. Krishna, T. Arunkumar, "Hybrid Particle Swarm and Gray Wolf Optimization Algorithm for IoT Intrusion Detection System", IJIES, Vol. 14, No. 4, pp. 66-76, 2021.
[25] J. Manokaran, G. Vairavel, "IGWO-SoE: Improved Grey Wolf Optimization Based Stack of Ensemble Learning Algorithm for Anomaly Detection in Internet of Things Edge Computing", IEEE Access, Vol. 11, pp. 106934-106953, 2023.
[26] K. Shaik, N. Thumboor, S. Veluru, N. Bommagani, D. Sudarsa, G. Muppagowni, "Enhanced SVM Model with Orthogonal Learning Chaotic Grey Wolf Optimization for Cybersecurity Intrusion Detection in Agriculture 4.0", International Journal of Safety and Security Engineering, Vol. 13, No. 3, pp. 509-517, June 2023.
[27] I. Syarif, R.F. Afandi, F. Astika Saputra, "Feature Selection Algorithm for Intrusion Detection Using Cuckoo Search Algorithm", in 2020 International Electronics Symposium (IES), pp. 430-435, September 2020.
[28] M.R. Gauthama Raman, S. Nivethitha, J. Sahruday, M. Tina, S. Thirumaran, K. Kannan, V.S. Shankar, "An Efficient Intrusion Detection Technique Based on Support Vector Machine and Improved Binary Gravitational Search Algorithm", Artif Intell Rev, Vol. 53, No. 5, pp. 3255-3286, June 2020.
[29] C. Rui, Z. Fengbin, X. Liang, "Anomaly Detection Algorithm Based on FCM with Improved Krill Herd", J. Phys.: Conf. Ser., Vol. 1187, No. 4, p. 042028, Avril 2019.
[30] P. Kaliraj, B. Subramani, "Intrusion Detection Using Krill Herd Optimization Based Weighted Extreme Learning Machine", JAIT, Vol. 15, No. 1, pp. 147-154, 2024.

BIOGRAPHIES

Name: **Hind**
Surname: **Khoulimi**
Birthday: 09.05.1990
Birthplace: Casablanca, Morocco
Bachelor: Professional Degree in System, Database and Network, Faculty of Science, Hassan II University, Casablanca, Morocco, 2012
Master: Logistics, Faculty of Science, Hassan II University, Casablanca, Morocco, 2014
Doctorate: Student, Cyber Security, Applied Mathematics and Computing Laboratory, Faculty of Science, Hassan II University, Casablanca, Morocco, Since 2019
The Last Scientific Position: Lecturer, Primary School Teacher, Casablanca¸ Morocco, Since 2016
Research Interests: Application of Artificial Intelligence on Intrusion Detection Systems
Scientific Publications: 1 Paper, 1 Book Chapter

Name: **Othman**
Surame: **Benammar**
Birthday: 08.07.1982
Birthplace: Casablanca, Morocco
Bachelor: Computer Engineering, Faculty of Science, Hassan II University, Casablanca, Morocco, 2006
Master: Computer and Internet Engineering, Faculty of Science, Hassan II University, Casablanca, Morocco, 2008
Doctorate: Computer Science, Information Systems Security, Faculty of Science, Hassan II University, Casablanca, Morocco, 2016
The Last Scientific Position: Prof., Mathematics and Computer Science, Hassan II University, Casablanca, Morocco, Since 2018
Research Interests: Information Systems Security
Scientific Publications: 10 Papers
Scientific Memberships: Applied Mathematics and Computing Laboratory