Improving Insider Threat Detection from Irregular Login Patterns with Metaheuristic Optimised AdaBoost

Luka Jovanovic , Žaklina Spalević , Nebojsa Bacanin , Milan Simić * , Filip Marković

Posted Date: 24 December 2024

doi: 10.20944/preprints202409.1500.v2

Keywords: AdaBoost; Legal framework; Cyber security; Crayfish optimization algorithm; Insider threat; Metaheuristics

Article

Improving Insider Threat Detection from Irregular Login Patterns with Metaheuristic Optimised AdaBoost

Luka Jovanovic [1], Žaklina Spalević [1], Nebojsa Bacanin [1], Milan Simić [2,*] and Filip Marković [3]

[1] Singidunum University, Danijelova 32, 11000 Belgrade, Serbia
[2] RMIT University, 124 La Trobe Street, Melbourne VIC 3000, Australia
[3] University of Priština - Kosovska Mitrovica, Faculty of Technical Sciences, Knjaza Miloša 7, 38220 Kosovska Mitrovica, Serbia
* Correspondence: milan.simic@rmit.edu.au; Tel.: +61 3 992 56223

Abstract: Continuing process of changing cyber security landscape demands constant adaptability to maintain secure and safe systems' operation. With changing attack vectors constantly being developed, it is essential to act preemptively to maintain desired levels of security. A major challenge in modern systems, and a target for many malicious actors, is the human factor of the system. Insider threat is a critical cybersecurity concern. It is difficult to detect and identify it. Insider threat can cost companies millions of dollars in damage. We have investigated utilization of machine learning (ML) for insider threat detection, from irregular login behaviors of users. A specially tailored version of crayfish optimization algorithm (COA) is proposed and applied to hyperparameter tunning of AdaBoost models to ensure favorable performance. Simulations, conducted on publicly available data, with a majority of normal users' activities, showcases the versatility of models, optimized by the introduced modified algorithm. The best models are attaining accuracy of 94.6128%. Modified algorithm demonstrates adaptive convergence capable of avoiding local minima and finding more favourable solutions. The best models have also undergone Shapley additive explanation (SHAP) analysis to identify the key features and their contributions to model decisions.

Keywords: adaboost; legal framework; cyber security; crayfish optimization algorithm; insider threat; metaheuristics

1. Introduction

In modern, complex information systems, cyber security insider threat detection plays an increasingly important role [1]. Ransomware attacks and data breaches, often initialized by insiders [2], cost organizations millions of dollars. Many institutions were forced to shut the businesses down due to the loss of trust or finance. Because of the ever-evolving issues of cybersecurity, administrators and security specialists often struggle to keep up with new developments [3].

One promising approach, capable of adapting to the changing landscape of security in the digital realm, is the application of artificial intelligence (AI). AI algorithms have ability to discern patterns and effectively learn from observations of data. This allows the application of AI with minimal programming required. Additionally, algorithms learn from new data and are therefore capable of adapting to new developments [4].

There are several challenges to the proper application of AI in cyber security. The first one is data availability. Companies are often hesitant to make data concerning attacks publicly available and therefore real-world data is scarce. The second challenge is associated with parameter selections. Namely, algorithms are often designed with good general performance in mind, however, to be well suited to a specific task, parameter tuning is required to adjust the algorithm to the available data.

 doi:10.20944/preprints202409.1500.v2

This process can often be NP-hard due to the large search spaces when considering options for parameters.

A class of algorithms often selected by researchers to tackle hyperparameter tuning are metaheuristic algorithms [5]. These algorithms take a randomness-driven approach and often borrow inspiration from search strategies observed in nature to handle and guide optimizations toward an optimal solution. Algorithms could solve NP-hard problems with acceptable results, and within realistic time constraints, however, a true optimal solution is not guaranteed.

This report presents our investigation in the potential of AdaBoost classifier to handle detection of insider threats within an organization. A publicly available simulated cyber security dataset is used, and user login patterns are analyzed to detect malicious actions. Additionally, a modified version of the recently introduced crayfish optimization algorithm (COA) [6] is introduced specifically for the needs of this study.

1.1. Research Gap, Motivation and Scientific Contributions

There is a significant research gap on insider threat detection, leaving a crucial area of study underexplored. This research seeks to examine the potential of AI in preventing harm caused by insiders, with an emphasis on classifying user behavior. By filling this gap, the study aims to enhance methods for protecting organizations against the dynamic challenges of cybersecurity, particularly those related to insider threats.

Investigation is motivated by the need to address this gap by leveraging AI to develop advanced techniques for identifying and mitigating insider threats. The work specifically aims to propose a new framework based on the AdaBoost algorithm to enhance institutional cybersecurity. Furthermore, it introduces a modified version of the COA to overcome limitations in the original approach and evaluates several contemporary optimization algorithms to determine their effectiveness in optimizing AdaBoost for cybersecurity applications. By addressing these challenges, this research seeks to provide practical and innovative solutions for safeguarding organizations against evolving insider threats.

The main contributions of this work can be outlined as the following:

- Proposal for new insider threat detection framework, based on the AdaBoost algorithm to improve institution cyber security;
- Introduction of novel, modified version of the COA, designed to overcome some of the observed shortcomings of the original algorithm;
- Comparative analysis between several contemporary optimizers was conducted to determine the degree of effectiveness of the proposed approach evaluated on a publicly available dataset;
- Feature importance evaluations was also conducted using SHAP, on the best performing model, in order to improve future data collection efforts;

The remining part of this report is structured as following: Section 2 covers preceding research that inspired and motivated this investigation; Section 3 presents the new methodology and describes the proposed modified algorithm; Sections 4 and 5 describe experimental configuration and present the attained outcomes. Finally, Section 6 concludes the report, highlighting limitations, policy implications and potential future research.

2. Related Works

Proper management of cyberspace refers to the application of following principles: responsibility, transparency, rule of law, participation of the entire audience in cyberspace, institutional responsiveness, effectiveness of institutional and individual roles, as well as efficiency in operations. The basic problem in ensuring cyber security is the definition of legal norms and institutions that would monitor the flow of data and actions in cyberspace, as well as ensure the privacy rights of users. U.S. Congress still struggles to establish a system that provides essential privacy protection while retaining investigative capabilities [7]. Cybersecurity and privacy protection are subjects of intensive research [8], as well as governments' considerations around the world. In 2023 Australia has established its 2023-2030 Australian Cyber Security Strategy [9].

 doi:10.20944/preprints202409.1500.v2

The interconnected nature of cyberspace, "without borders", poses a real problem for the traditional framework of territorial application of laws [10]. Data and cyber activities are generated on servers that may fall under the jurisdiction of one state, while users or cyber victims may fall under the jurisdiction of another state or legal system [11].

It is often considered that laws applicable to offline activities should also apply to online activities, but clear characterization of such actions in practice is difficult to achieve. Cybersecurity raises complex legal questions primarily related to the right to privacy and freedom of expression. This complexity is further compounded by public-private collaboration and the related legal issues concerning responsibility and control. The issue of monitoring activities and data flows is complicated due to the diverse nature of actors involved in cyberspace. According to the broadest understanding, national oversight institutions oversee the work of various agencies or functional lines of administration. Consequently, state-level parliamentary committees may oversee the work of intelligence services, armed forces, or judicial bodies. On the other hand, public-private collaboration in the field of cybersecurity goes beyond the boundaries of individual agencies, leading to a collision of expert understandings of cyber activities and surveillance mandates. The consequence of this collision is the existence of many cases where surveillance is either inadequate or nonexistent. Regarding the overlap of responsibilities and control, the procedures of each government agency are linked in a chain of accountability from the first to the last.

In cyberspace, chains of command can be disrupted by the involvement of private actors and the establishment of public-private collaboration mechanisms. In practice, IT companies might be engaged with government agencies and work exclusively for the state. This relationship is often complex and obscured by numerous information asymmetries that reduce transparency and hinder the smooth and successful functioning of surveillance and control mechanisms [12].

The oversight boards in the government should control government agencies for which they are directly responsible. In this way, there might be an omission of private partners of these agencies from the oversight space, even in cases when they are directly funded or closely collaborate with these agencies. The technical specificity of characterizing cyberspace further complicates the traditional problems, faced by national parliamentarians, tasked with overseeing the security sector, leading to reduced effective accountability. Difficulties in reliably identifying perpetrators of cybercrimes can lead to hindered or even nonexistent accountability of the security sector to civilian authorities, contributing to a culture of impunity for these criminal acts. Thus, the judicial sector may grant special powers to law enforcement and intelligence agencies by issuing search warrants. This is particularly important in the context of communication interception. In practice, judicial oversight is often circumvented or restricted for reasons of national security preservation under emergency conditions.

The National Cyber Security Strategy of Sweden from 2016, which regulates issues from the legal regulation of ICT to the protection of critical infrastructure can be considered as a model of good legislation. However, it seems that there is not just one committee or subcommittee dealing solely with cybersecurity. Unlike most national cybersecurity strategies, the Swedish strategy includes strategic principles and an action plan that helps parliament hold both public and private actors accountable in the process of cyber security control. The principle of the rule of law is interpreted by international courts, such as the European Court of Human Rights (ECHR). This court has developed a rule-of-law test stating that "all restrictions on fundamental rights must be based on clear, precise, accessible, and predictable legal provisions and must pursue legitimate aims in a manner that is necessary and proportionate to the aim in question, and there must be an effective, preferably judicial, remedy". Consequently, authorities in states demand that private companies, who own social media platforms, ensure that their services do not harbor violent extremists and terrorists. To meet these demands, governments [13], and private companies holding social media, have developed specific terms and codes of conduct to control the content posted on these platforms, and generally, apply legal rules in the digital world. In this way, they have de facto established rules and norms on the Internet. However, these terms and rules are not the same on all platforms, creating ambiguity and legal uncertainties regarding the type of content prohibited on each platform.

 doi:10.20944/preprints202409.1500.v2

Hackers and various agencies routinely engage in eavesdropping on private conversations and intercept them at the "back door". In other words, when it comes to state security, there is no truly established need for the application of the rule of law, although the basic principles that could form the basis of such an important part of the universal fortress of human rights exist. With the increasing partnership between law enforcement agencies and intelligence and security services, this weakening of the rule of law threatens to spread and be transferred to the police and prosecutors. The lack of clear legal frameworks in this area, both domestically and internationally, poses an additional threat to the rule of law on the Internet and in the global digital environment [14].

Numerous existing approaches attempt to address cyber security, with traditional techniques like firewalls [15] and block lists proving useful over time [16]. However, rapid developments and the emergence of zero-day [17] vulnerabilities make it challenging for administrators to keep up with attackers. To adapt to the fast-paced information age, new techniques are imperative.

IoT networks are frequent targets for Distributed Denial-of-Service (DDoS) and DoS attacks [18], where relatively simple devices can disrupt operations on a massive scale and compromise information about their environment and users. Additionally, insider actors seeking revenge for perceived unfair treatment pose a significant threat vector [19].

A noticeable research gap exists in insider threat detection, creating a void in the field. This investigation aims to explore the potential of AI for preventing insiders from causing harm to organizations by focusing on classification of user behavior. By addressing this gap, the research contributes to advancing methods that can better safeguard against evolving cyber security challenges, particularly in the context of insider threats.

2.1. AdaBoost Classifier

AdaBoost [20] utilized an iterative approach in order to cast an approximation of the Bayes classifier. This is done by combining several weaker classifiers. From a starting point of an unweighted sample used to train the model, this approach builds a group of classifiers. If misclassification occurs, the weights of each classifier are reduced and if correct classification is made weights are incremented. Error of a weak classifier ε_t can be determined as given by Equation (1):

$$\varepsilon_t = \frac{\sum_{i=1}^{N} \omega_{i,t} I\left(h_t\left(x_i\right) \neq y_i\right)}{\sum_{i=1}^{N} \omega_{i,t}}, \tag{1}$$

where ε_t denotes the weighted error of the weak learner in the t-th iteration. Variable N represents the number of training instances. Term $\omega_{i,t}$ corresponds to the weight of the i-th instance in the t-th iteration. Expression $h_t(x_i)$ signifies the prediction made by the weak learner for the i-th instance in the t-th iteration. Variable y_i represents the true label of the i-th instance. Additionally, function I(·) is an indicator function that equals 1 if the condition within the parentheses is true, and 0 otherwise.

Further classifiers are built based on the attained weights. Weight adjustment process is repeated. Large groups of classifiers are usually assembled to create accurate classification. A score is given to each of these sub-models, and a linear model is constructed by their combination. Classifier weight in the ensemble can be determined according to Equation (2):

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \tag{2}$$

where weight α_t, assigned to each weak learner in the final ensemble, is calculated based on its performance. It depends on the weighted error ε_t and is used to determine the contribution of the weak learner to the final combined model. To update weights Equation (3) is used:

$$\omega_{i,t+1} = \omega_{i,t} \exp\left(-\alpha_t y_i h_t\left(x_i\right)\right), \tag{3}$$

 doi:10.20944/preprints202409.1500.v2

where $\omega_{i,t}$ represents the weight of the i-th instance in the t-th iteration, α_t denotes the weight of the weak learner in the t-th iteration, y_i stands for the true label of the i-th instance, and $h_t(x_i)$ signifies the prediction of the weak learner for the i-th instance in the t-th iteration.

AdaBoost algorithm is well suited for binary classification problems. It does struggle with multi-class classification. Since we are dealing with binary classification problems, AdaBoost algorithm is selected for optimization.

2.2. Metaheursitic Optimization

Hyperparameter selection can often be difficult in practice. There is currently no unified approach for selection. Researchers often resort to computationally expensive complete search techniques or a trial-and-error process. When dealing with a mixed set of parameters this challenge can quickly form a mixed NP-hard problem. Therefore, techniques capable of addressing this category of challenge are required.

Taking a heuristic approach is often preferable. Metaheuristic optimizers have demonstrated ability to handle NP-hard problems, often drawing inspiration from natural phenomena. Some notable examples include the genetic algorithm (GA) [21], particle swarm optimization (PSO) [22], firefly algorithm (FA) [23], sine cosine algorithm (SCA) [24], whale optimization algorithm (WOA) [25], reptile search algorithm (RSA) [26] and COLSHADE [27]. The driving reason for so many algorithms comes from the no free-lunch theorem of optimization (NFL) [28] that states that no single approach is perfectly suited to all challenges and across all metrics. Therefore, constant experimentation is needed to determine the most suitable optimizer for a given task.

Hybridization of existing algorithms is a popular approach to overcome some of the observed drawbacks of optimizers. Metaheuristics is successfully applied in several fields of optimization, including finance [29], medicine [30] [31], computer security [32], renewable power generation [33] and many others [34].

3. Methods

This section describes the base methods and algorithms that served as inspiration for our work. Following that, the potential for improvements is described alongside the modifications aimed at improving performance. Finally, the algorithm pseudocode is presented.

3.1. Original Crayfish Optimization Algorithm

The Crayfish optimization algorithm (COA) is a metaheuristic algorithm depicting the behavior of crayfish, a form of crustacean, in a natural settings [6]. These animals belong to the infraorder Astacidea family living in freshwater, such as lakes and rivers. They are omnivores, foraging the floor of the body of water for nutritious meals.

Algorithm emulates crayfish summer resort behavior which entails the crayfish searching for cool caves when the temperatures are high. This behavior acts as the algorithm's exploration stage. Next, these animals compete for the best shelter. Foraging, which happens when the temperatures allow, is also modeled. Competing and foraging are used as exploitation stages in COA.

As is the norm with swarm intelligence, the population of crayfish P is initialized in the beginning stage of the algorithm. To manage the stages of exploration and exploitation, temperature is represented by a random constant defined by the Equation (4):

$$temp = rand \times 15 + 20 \tag{4}$$

The summer retreat happens when the temperature is higher than 30°C, in which case the crayfish look for a cool shelter from the heat, such as caves. Temperatures between 15°C and 30°C are suitable for crayfish feeding, with 25°C being ideal. Since most reliable foraging behavior happens in the range of 20°C to 30°C, the model's temperature ranges from 20°C to 35°C. Mathematical representation of the crayfish feeding behavior is given by Equation (5):

 doi:10.20944/preprints202409.1500.v2

$$p = C_1 \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(temp - \mu)^2}{2\sigma^2} \right). \quad (5)$$

In this expression, μ marks the thriving temperature for crayfish, while C_1 and σ serve the purpose of controlling the food intake of crayfish at varying temperatures.

When *temp*>30, the exploring stage starts. Shelter position that crayfish take from the heat is modeled by Equation (6):

$$X_{shade} = X_G + \frac{X_L}{2} \quad (6)$$

where X_L marks the current colony optimal positioning, while X_G marks the best possible place gained, regarding the number of iterations.

Whether the crayfish competes for the shelter is randomly dictated by the variable *rand* value. When this value is less than 0.5, no competition between crayfish for the shelter occurs. Since there is no obstacle, the crayfish will enter the cave without issue, per the Equation (7) and Equation (8):

$$X_{t+1_{i,j}} = X_{t_{i,j}} + C_2 \times rand\left(X_{shade} - X_{t_{i,j}}\right) \quad (7)$$

$$C_2 = 2 - \frac{t}{T} \quad (8)$$

C_2 denotes a decreasing curve, T marks the topmost number of repetitions, and t marks live iteration, while t+1 depicts the repetition number for the next generation.

During the high temperatures, crayfish seeks shelter. Shelter, or cave, is a symbol of the best possible solution. In the summer resort stage, the crayfish heads towards the cave thus nearing the optimal solution. The closer to the cave they are, the better COA's potential for exploitation becomes, and the faster the algorithm converges.

When *rand*≥0.5, there is competition between crayfish for the shelter. This competition has played the role of the start of the exploitation stage. The conflict is represented by the Equation (9) and Equation (10):

$$X_{t+1_{i,j}} = X_{t_{i,j}} - X_{tz_j} + X_{shade} \quad (9)$$

$$z = round\left(rand\left(N - 1\right)\right) + 1 \quad (10)$$

where z marks the crayfish's random individual.

In the competition phase, crayfish fight with each other. Crayfish X_i adapts its position in relation to another crayfish's position X_z. This adaptation of positions expands the search range of COA, thereby boosting the algorithm's exploration capacity.

The crayfish feed in temperatures below or equal to 30°C. When such conditions are met, the crayfish moves towards the food. Food location, X_{food} and its quantity Q are defined by Equation (11) and Equation (12), respectively:

$$X_{food} = X_G \quad (11)$$

$$Q = C_3 \times rand \frac{fitness_i}{fitness_{food}} \quad (12)$$

In this context, C_3 stands for the food factor representing the biggest food source, with a constant value of 3. The *fitness* variable denotes the fitness value of the i-th crayfish, whereas $fitness_{food}$ indicates the fitness value linked to the food's location.

In the case when the food is too big, and Q>(C_3+1)/2, the process of tearing up the food is depicted in Equation (13):

 doi:10.20944/preprints202409.1500.v2

$$X_{food} = \exp(-Q)X_{food} \tag{13}$$

When the food is small enough, Q<(C3+1)/2, the crayfish will simply eat the food, as given by Equation (14):

$$X_{t+1_{i,j}} = X_{t_{i,j}} - X_{food}\,p + p \times randX_{t_{i,j}} \tag{14}$$

During the foraging phase, crayfish employs various feeding tactics depending on the size of their food denoted by Q, where the food location X_{food} signifies the ideal solution. They will move closer to the food of readily edible size. Conversely, when Q is excessively large, meaning a substantial disparity between the crayfish and the optimal solution, X_{food} will be decreased, thereby drawing it nearer to the meal.

3.2. Hybrid COA

Despite the admirable performance demonstrated by the COA, there is still plenty of room to explore potential improvements. Following that, we have introduced two new mechanisms into the original COA.

The initial modification incorporates quasi-reflective learning (QRL) [35] in the first T iterations. Following each iteration, the worst solutions are replaced by new solutions generated based on Equation (15):

$$A_z^{qr} = rand(\frac{lb_z + ub_z}{2}, a) \tag{15}$$

where *lb* and *ub* denote lower and upper bounds of the search space and *rand* denotes a random value within the given interval. The newly generated solution is not subjected to objective function evaluation thus the computational complexity of the modified algorithm is kept consistent with the original.

When examining optimization metaheuristics, it becomes crucial to find an equilibrium between exploration and exploitation. In order to enhance exploitation, a supplementary adjustment is incorporated, drawing inspiration from the widely recognized firefly algorithm (FA) [23]. The FA simulates the courtship behaviors of bioluminescent beetles through mathematical modeling, where individuals emitting brighter light attract those in their vicinity. The brightness of each agent is computed according to a problem-dependent objective function, outlined in Equation (16):

$$F_i = f(X_i) \tag{16}$$

Several environmental factors are also simulated to replicate real-world conditions such as light fading depending on the distance between agents, as well as the characteristics of the medium of propagation. The basic search mechanism of the FA is shown by Equation (17):

$$X_i(t+1) = X_i(t) + \beta e^{-\gamma r_{i,j}^2}\left(X_j(t) - X_i(t)\right) + \alpha\varepsilon_i(t) \tag{17}$$

Equation (17) is commonly swapped for Equation (18) to improve computational performance, where β_0 represents the attractiveness at r=0:

$$\beta(r) = \frac{\beta_0}{\left(1+\gamma \times r^2\right)} \tag{18}$$

where, $X_i(t)$ and $X_j(t)$ represent current positions of agents i and j, respectively, at a specific iteration t. Paramenter r_{ij} denotes the Cartesian distance between i and j agents during the corresponding iteration t. Parameter β is termed the agent attraction coefficient, and refers to the intensity of light, while γ denotes the light absorption coefficient, α controls the degree of randomness, and $\varepsilon_i(t)$ represents a stochastic vector.

Although the introduced search mechanism of the FA does enhance convergence, it is crucial to strike a balance throughout the optimization process. The firefly search mechanism becomes active

 doi:10.20944/preprints202409.1500.v2

only in the latter half of the optimization interactions. After each of these iterations, a pseudo-random value is generated within the range [0, 1] and compared to a threshold value ψ. If the generated value surpasses ψ, the firefly search is initiated; otherwise, a normal COA search is employed. The value of ψ is determined empirically to yield optimal results for the given problem, typically set at 0.6.

The described algorithm is dubbed the hybrid COA(HCOA). It is presented by the following Algorithm 1 pseudocode:

Algorithm 1 – HCOA Pseudocode

```
Set initial population size (N),
Set the maximum number of iterations (T),
Set coefficient ψ
while t < T do
    Evaluate agents using an objective function
    if t > T/2 then
        Update agent's locations using the appropriate COA
     search mechanisms
    else
        Generate a random value rand
        if rand > ψ then
            Update agent's locations using the appropriate
         COA search mechanisms
        else
            Update agent's locations using the Firefly search
         mechanisms
        end if
    end if

end while
```

In terms of time complexity, metaheuristic algorithms are usually compared based on the number of evaluations conducted during the optimization. As the modified version of the optimized algorithm does not introduce any additional evaluation, complexity is kept consistent with the baseline optimized. In terms of big-O notation, the COA has a complexity of $O(N \times D \times T)$. Time complexity of the QRL approach is $O(N \times T)$. Hence the introduced modifications do not increase computational complexity in comparison to the baseline algorithm.

4. Experimental Configuration

To facilitate experimentation, simulated dataset is utilized, provided by the Carnegie Mellon University CERT Division Software Engineering Institute [36] and it is publicly available [37]. While this dataset contains information on several malicious, insider users' threats, and their activities, our work focuses on logon activities and their relative period.

To reflect real-world scenarios dataset is heavily imbalanced with malicious actor activities being a minority. A total of 854661 samples represent normal users, with only 198 samples malicious actors. To facilitate model training, the majority class is downsampled to a 9:1 ratio of normal to malicious activities. During the testing, 1782 samples represented normal user activity and 198 malicious. Datasets are further split into training and testing with 70% allocated to train respective models and 30% withheld for evaluations. Dataset structure, as well as the structure of the training and testing portions, are shown in Figure 1.

Time of access features are considered, specifically, the day of access and time of the day. Interesting patterns can be observed in the behavior of malicious users when accessing the system.

 doi:10.20944/preprints202409.1500.v2

Insider threats often happen outside of the regular working hours. Additionally, insiders prefer accessing machines later in the week. The distribution of normal and malicious user access can be seen in Figure 2.

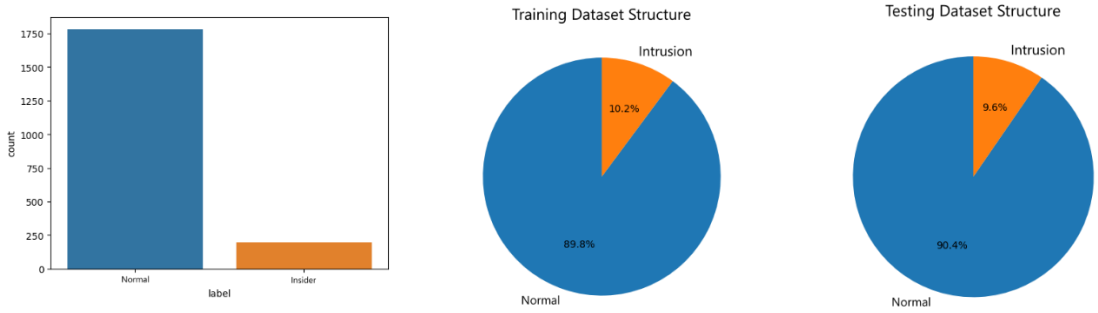


Figure 1. Dataset training and testing structure.

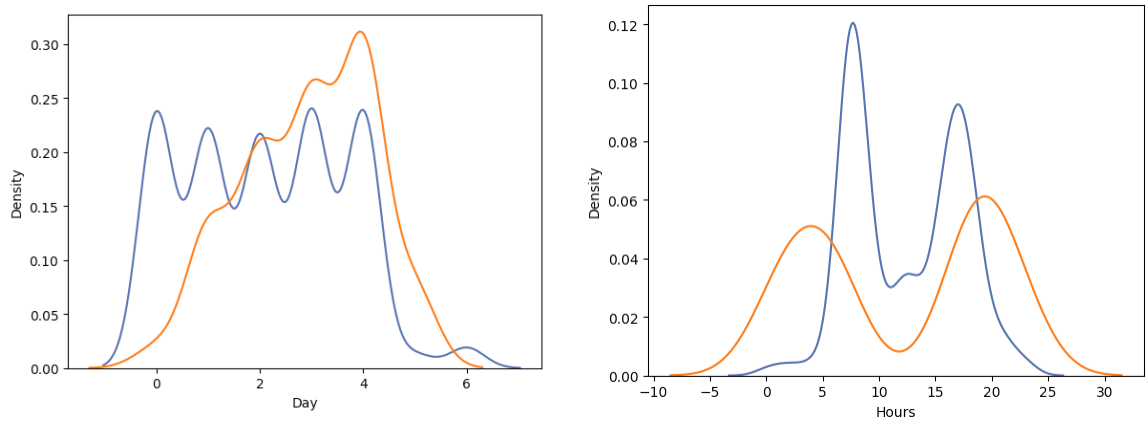


Figure 2. Normal, shown in blue, and insider thread user distributions shown in orange color.

Interesting patter differences in terms of days of activity can be observed. Normal users typically log in regularly, during workdays, in a consistent pattern, while users causing insider threat, attempt logins latter in the week, favoring Thursdays and Fridays. In terms of time of the day, insider threat actors seem to prefer to attempt logins prior to normal users (between 00:00 and 5:00) and after normal users in the evening (after 20:00).

Several contemporary optimizers were included in comparative analysis with our novel optimizer. They are: COA [6], GA [21], PSO [22], FA [23], SCA [24], WOA [25], RSA [26] and COLSHADE [27]. Algorithms were independently implemented in Python using standard machine-learning libraries provided by Sklearn. Additional supporting libraries utilized include Pandas and Numpy. Optimizers are implemented with parameters set to the values suggested in the original works.

Optimizes are tasked with selecting optimal control parameter values for the AdaBoost algorithms. These parameters and their respective ranges are the number of estimators [10, 50], depth [1, 10] and learning rate [0.1, 2]. A relatively modest number of estimators is used due to the heavy computational demands of the optimization. Each optimizer was allocated a population size of ten and allowed 15 iterations to improve attained solutions. Finally, the experimentation is repeated 30 times in independent executions to account for the randomness inherent in the application of metaheuristics.

Agent parameters, used in the population of each optimizer, are used to map Adaboost hyperparameters. Each agent consists of three parameters representing the number of estimators, depths and learning rate as shown in Figure 3.

 doi:10.20944/preprints202409.1500.v2

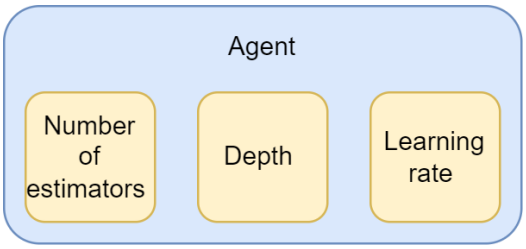


Figure 3. Agent parameter mapping visualization.

To guide the optimization Cohen's kappa metric is utilized due to its ability to evaluate classifications of imbalanced data well. Cohen's kappa score is determined by Equation (19):

$$k = \frac{c_0 - c_e}{1 - c_e} \tag{19}$$

Further metrics are tracked to ensure thorough comparisons. These include a set of standard classification metrics used to get a comprehensive overview of algorithm performances including accuracy shown by Equation (20), precision by Equation (21), recall Equation (22) and f1-score by Equation (23):

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{20}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{21}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{22}$$

$$\text{f1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{23}$$

Additionally, error rates are recorded for each algorithm determined by Equation (24):

$$\text{Error rate} = 1 - \text{Accuracy} \tag{24}$$

Error rate is the complement of accuracy and represents the proportion of incorrectly classified instances. It is the ratio of the number of misclassifications to the total number of instances. This is a convenient way to represent error rate in terms of more intuitive accuracy metric.

Flowchart of the proposed insider threat detection framework is provided in Figure 4. It demonstrates the role of optimizers, specifically the modified optimization technique, in improving insider threat detection in real world systems.

 doi:10.20944/preprints202409.1500.v2

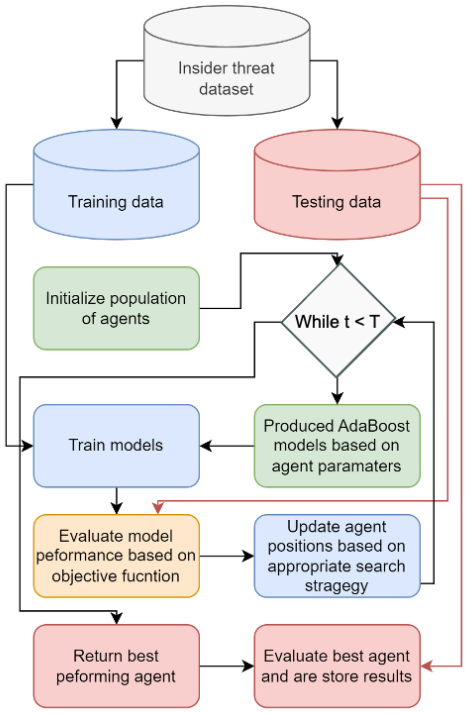


Figure 4. Proposed framework operational flowchart.

5. Simulation Outcomes

Simulation results in terms of objective and indicator functions are provided in Table 1 and Table 2 respectively. As evident, the introduced optimizer attained the best outcomes with the best scoring of 0.711939, mean scoring of 0.673919, and median executions scoring of 0.672113 in terms of objective function results. The best outcomes in terms of the worst-case performance are demonstrated by the FA attaining an objective function score of 0.630255.

These results are somewhat like the results in terms of an indicator function, with the introduced optimizer matching the best performance of 0.053872, sharing first place with the WOA, and attaining the best outcomes in mean and median executions scoring 0.057239 and 0.056397 respectively. The PSO attained the best results in the worst-case execution scoring 0.062290.

Algorithm stability comparisons can be viewed in Table 1 and Table 2, in terms of objective and indicator functions. In terms of stability, the PSO and SCA attained the highest rate of stability in comparison to other tested optimizes. However, they did not demonstrate the best performance. Visual comparisons are provided in Figure 5. While the introduced optimizer showcases a relatively low stability in terms of objective function it nonetheless demonstrates the best outcomes. Additionally, in terms of indicator function, the introduced optimizers show the highest outcomes, outperforming all other algorithms.

Table 1. Objective function outcomes for the best, worst, mean and median runs.

Method	Best	Worst	Mean	Median	Std	Var
AB-HCOA	**0.711939**	0.625510	**0.673919**	**0.672113**	0.024330	0.000592
AB-COA	0.671187	0.613295	0.637825	0.636679	0.018445	0.000340
AB-GA	0.677333	0.617112	0.645781	0.638252	0.019972	0.000399
AB-PSO	0.659553	0.614028	0.638885	0.638874	**0.012979**	**0.000168**

 doi:10.20944/preprints202409.1500.v2

AB-FA	0.672196	**0.630255**	0.644348	0.640337	0.013494	0.000182
AB-SCA	0.656084	0.613295	0.642558	0.647218	0.014036	0.000197
AB-WOA	0.694297	0.625510	0.647621	0.639584	0.022794	0.000520
AB-RSA	0.682312	0.625510	0.645486	0.645056	0.017144	0.000294
AB-COLSHADE	0.055556	0.063973	0.060816	0.061448	0.002719	7.40E-06

Table 2. Indicator function outcomes for the best, worst, mean and median runs.

Method	Best	Worst	Mean	Median	Std	Var
AB-HCOA	0.053872	0.063973	**0.057239**	**0.056397**	0.003260	1.06E-05
AB-COA	0.060606	0.063973	0.061658	0.061448	0.003663	1.34E-05
AB-GA	0.055556	0.069024	0.060816	0.060606	0.004075	1.66E-05
AB-PSO	0.057239	**0.062290**	0.060396	0.060606	0.002296	5.27E-06
AB-FA	0.055556	0.065657	0.062710	0.063131	0.004434	1.97E-05
AB-SCA	0.060606	0.063973	0.061027	0.061448	**0.002018**	**4.07E-06**
AB-WOA	0.053872	0.063973	0.059975	0.060606	0.003464	1.20E-05
AB-RSA	0.055556	0.063973	0.059975	0.060606	0.002516	6.33E-06
AB-COLSHADE	0.055556	0.063973	0.060816	0.061448	0.002719	7.40E-06

Convergence rate comparisons, in terms of objective, as well as indicator function are shown in the Table 2. A clear influence of the introduced modifications is evident in the modified version of the algorithm over the baseline.

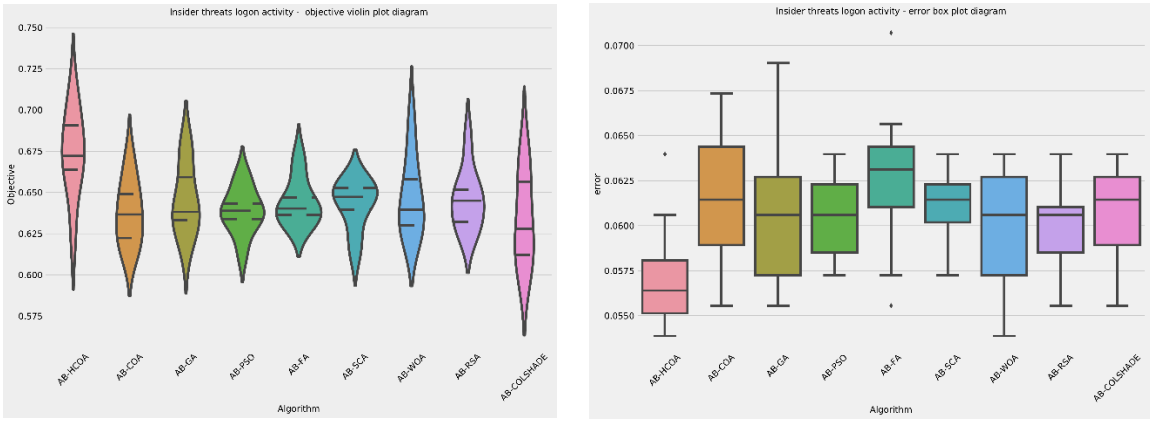


Figure 5. Objective and indicator function outcome distributions over 30 independent runs.

While many algorithms converge towards a local minimum, modified version locates the best solution towards the final executions, suggesting that the introduced alterations bring improvements.

Figure 6 shows the objective and indicator functions' outcome convergence plots.

 doi:10.20944/preprints202409.1500.v2

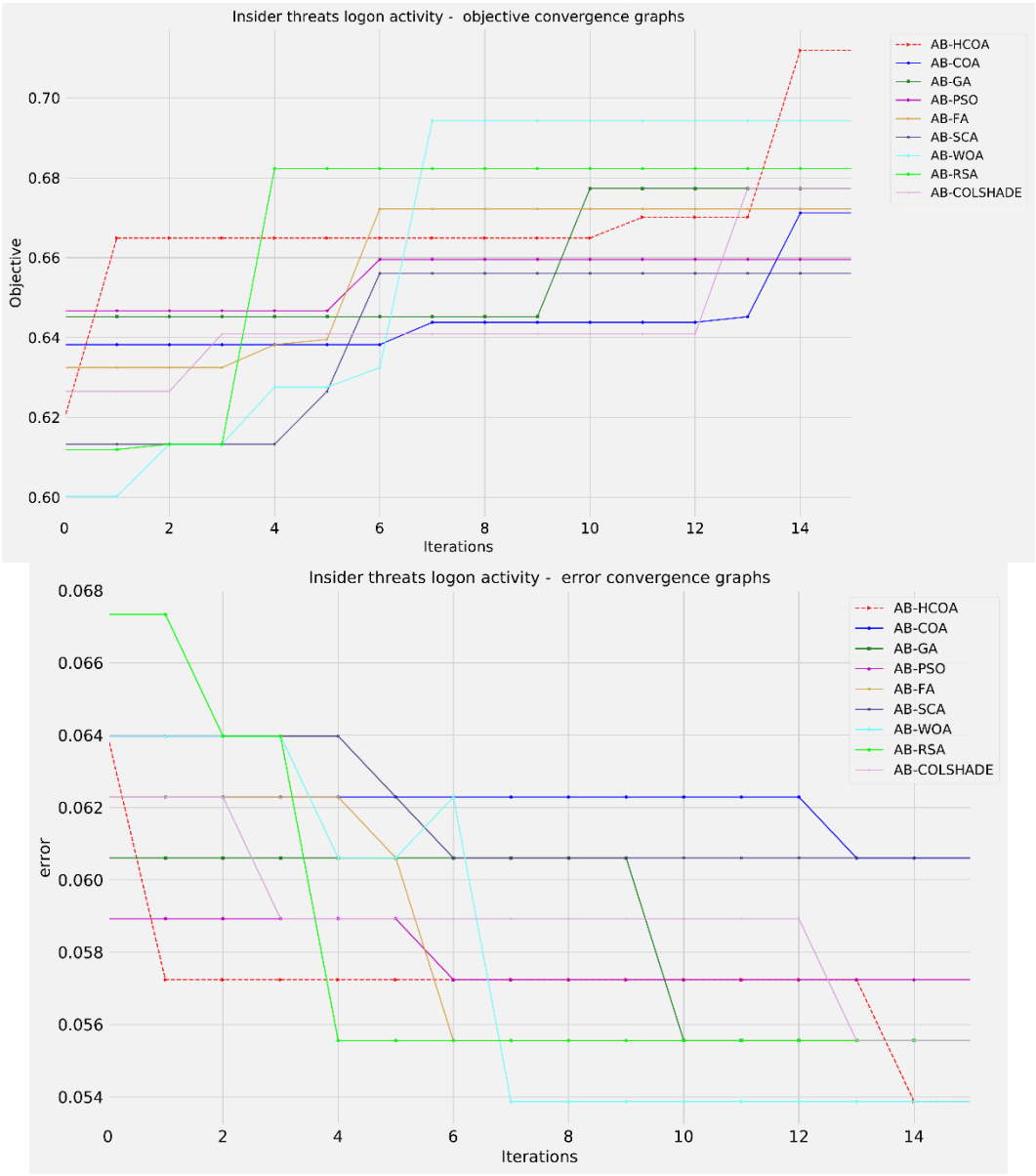


Figure 6. Objective and indicator function outcome convergence plots.

Detail metric comparisons between the best-constructed models are provided in Table 3. A clear dominance in terms of best outcomes is showcased by the introduced algorithm, with only the FA outperforming the algorithm in terms of precision for insider threat detection. However, this is to be somewhat expected as in accordance with the NFL [28] no single optimizer is equally suited to all challenges across all metrics.

Table 3. Metrics comparisons between the best performing models constructed by each optimizer.

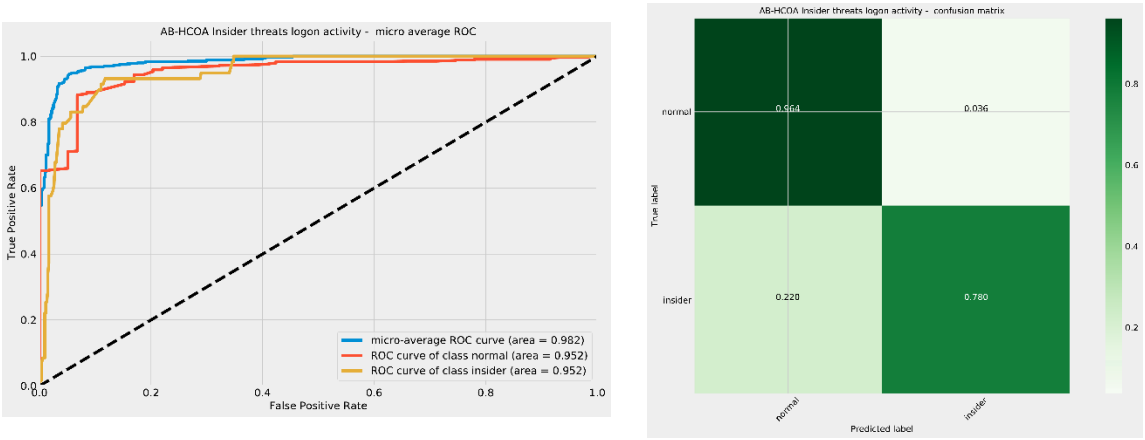
Method	metric	normal	insider	accuracy	macro avg	weighted avg
AB-HCOA	precision	**0.975425**	0.707692	0.946128	0.841559	**0.948832**
	recall	0.964486	**0.779661**	0.946128	**0.872073**	0.946128
	f1-score	**0.969925**	**0.741935**	0.946128	**0.855930**	**0.947279**
AB-COA	precision	0.969868	0.682540	0.939394	0.826204	0.9413297

 doi:10.20944/preprints202409.1500.v2

	recall	0.962617	0.728814	0.939394	0.845715	0.939394
	f1-score	0.966229	0.704918	0.939394	0.835573	0.940274
AB-GA	precision	0.964815	0.740741	0.944444	0.852778	0.942558
	recall	0.973832	0.677966	0.944444	0.825899	0.944444
	f1-score	0.969302	0.707965	0.944444	0.838633	0.943345
AB-PSO	precision	0.961326	0.745098	0.942761	0.853212	0.939849
	recall	0.975701	0.644068	0.942761	0.809884	0.942761
	f1-score	0.968460	0.690909	0.942761	0.829685	0.940892
AB-FA	precision	0.963100	**0.750000**	0.944444	**0.856550**	0.941933
	recall	0.975701	0.661017	0.944444	0.818359	0.944444
	f1-score	0.969359	0.702703	0.944444	0.836031	0.942873
AB-SCA	precision	0.964618	0.701754	0.939394	0.833186	0.938509
	recall	0.968224	0.677966	0.939394	0.823095	0.939394
	f1-score	0.966418	0.689655	0.939394	0.828037	0.938928
AB-WOA	precision	0.968343	0.736842	0.946128	0.852592	0.945348
	recall	0.971963	0.711864	0.946128	0.841914	0.946128
	f1-score	0.970149	0.724138	0.946128	0.847144	0.945714
AB-RSA	precision	0.966543	0.732143	0.944444	0.849343	0.943261
	recall	0.971963	0.694915	0.944444	0.833439	0.944444
	f1-score	0.969245	0.713043	0.944444	0.841144	0.943797
AB-COLSHADE	precision	0.964815	0.740741	0.944444	0.852778	0.942558
	recall	0.973832	0.677966	0.944444	0.825899	0.944444
	f1-score	0.969302	0.707965	0.944444	0.838633	0.943345
	support	535	59			

Graphical presentation of the outcomes for the best model in terms of confusion matrix and ROC are provided in Figure 7.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2024 doi:10.20944/preprints202409.1500.v2



(**a**) (**b**)

Figure 7. (**a**) Best performing model ROC; (**b**) Confusion matrix.

Additional sample additive explanations (SHAP) analysis [38] is conducted to determine the feature importance for the best models' decision-making process. The outcomes are presented in Figure 8. SHAP interpretation suggests that the time of the day, and the day of the week play important roles in user's activity being classified correctly. Additionally, the type of activity (logon or logoff) is considered. However, if an activity occurs on a weekend that is not considered important for the classification. It is likely due to this information being redundant with the day of the week being available.

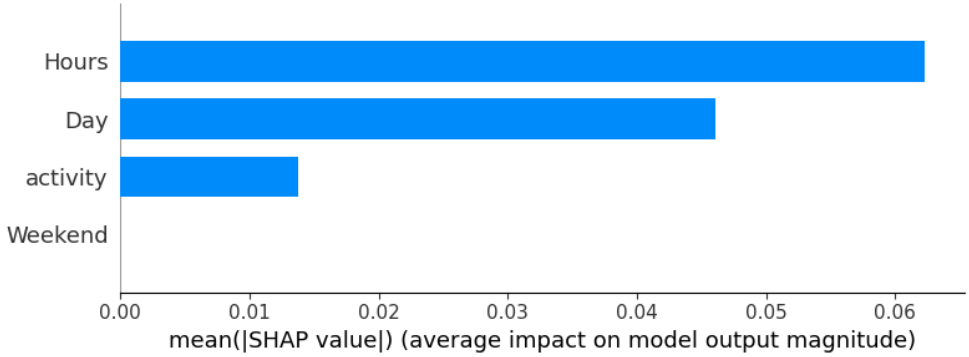


Figure 8. SHAP analysis outcomes for the best model.

Finally, the parameter selections made by each optimizer for the respective best-constructed model are provided in Table 4.

Table 4. Parameter selections for the respective best-performing model made by each optimizer.

Method	p1	p2	p3
AB-HCOA	34	4	1.941851
AB-COA	50	3	1.958571
AB-GA	42	2	1.557575
AB-PSO	30	2	1.522928
AB-FA	41	2	1.470723
AB-SCA	20	2	1.685248

 doi:10.20944/preprints202409.1500.v2

AB-WOA	50	2	1.815289
AB-RSA	50	2	1.839656
AB-COLSHADE	43	2	1.469586

6. Conclusions

This report presents our solutions for the problem of insider threat detection, based on login activity, within a greater context of cyber security, for the majority of normal users. Improved version of crayfish optimization algorithm is developed and applied to hyperparameter tunning of AdaBoost models to ensure better performances. Comparative analysis is conducted to compare efficiency of new design, with several contemporary optimizers applied on a simulated, publicly available dataset. Our models, optimized by the introduced modified algorithm, attained the best outcomes demonstrating accuracy of 94.6128% and an adaptive convergence rate that overcomes local minima and find the best solution. Additionally, the best-constructed model was subjected to SHAP analysis to determine the key contributing features.

There are few assumptions applied in our investigation. One of the key hypotheses was that the feature set, used for training, remains static and that the characteristics of insider threats do not change significantly over time. In real-world applications, insider behaviour could evolve, necessitating periodic updates to the feature set, and retraining of the model to maintain its effectiveness. Additionally, the study assumes that the labeled data used for training is accurate and true representative of actual insider threats. Mislabelling, or incomplete datasets, could negatively impact model accuracy, which is a common challenge in machine learning. Efforts were made to ensure data quality and coverage, but such challenges remain inherent to the approach. Future research will focus on further expanding the methodology, by incorporating other user actions in the classification. Computational constraint-associated limitations hope to be addressed as more powerful hardware becomes available on daily basis. Finally, other optimization tasks could be improved by the further introduction of novel, modified algorithms.

Author Contributions: Conceptualization, L.J. and N.B.; methodology, N.B.; software, N.B. and L.J.; validation, M.S. and F.M.; formal analysis, Ž.S.; investigation, Ž.S. and F.M.; resources, Ž.S.; data curation, M.S.; writing—original draft preparation, L.J.; writing—review and editing, M.S. and F.M.; visualization, L.J.; supervision, N.B.; project administration, M.S.; funding acquisition, Ž.S. All authors have read and agreed to the published version of the manuscript.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset was accessed on January 25, 2023. and is publicly available from https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/%2012841247.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. B. Gupta, B. Gupta, and S. Srinivasagopalan, *Handbook of research on intrusion detection systems* (Intrusion detection systems). Hershey, Pennsylvania: IGI Global, 2020.
2. "Insider threats and Insider Intrusion Detection," *International journal of recent technology and engineering*, vol. 8, no. 2S5, pp. 158-166, 2019, doi: 10.35940/ijrte.B1033.0782S519.
3. N. T N and D. Pramod, "Insider Intrusion Detection Techniques: A State-of-the-Art Review," *The Journal of computer information systems*, vol. ahead-of-print, no. ahead-of-print, pp. 1-18, 2024, doi: 10.1080/08874417.2023.2175337.
4. M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of information security and applications*, vol. 50, p. 102419, 2020, doi: 10.1016/j.jisa.2019.102419.

5. I. H. Hassan, A. Mohammed, and M. A. Masama, "Chapter 6 - Metaheuristic algorithms in network intrusion detection," in *Comprehensive Metaheuristics*, S. Mirjalili and A. H. Gandomi Eds.: Academic Press, 2023, pp. 95-129.
6. H. Jia, H. Rao, C. Wen, and S. Mirjalili, "Crayfish optimization algorithm," *The Artificial intelligence review*, vol. 56, no. Suppl 2, pp. 1919-1979, 2023, doi: 10.1007/s10462-023-10567-4.
7. J. Kosseff, "Defining Cybersecurity Law," *Iowa law review*, vol. 103, no. 3, pp. 985-1031, 2018.
8. A. Moallem, *HCI for Cybersecurity, Privacy and Trust : 6th International Conference, HCI-CPT 2024, Held as Part of the 26th HCI International Conference, HCII 2024, Washington, DC, USA, June 29 – July 4, 2024, Proceedings, Part I*, 1st 2024. ed. (Lecture Notes in Computer Science, 14728). Cham: Springer Nature Switzerland, 2024.
9. (2023). *2023-2030 Australian Cyber Security Strategy*. [Online] Available: https://www.homeaffairs.gov.au/about-us/our-portfolios/cyber-security/strategy/2023-2030-australian-cyber-security-strategy
10. D. J. B. Svantesson and D. Kloza, *Trans-Atlantic data privacy relations as a challenge for democracy* (European integration and democracy series ; volume 4). Cambridge [England] ;: Intersentia, 2017.
11. V. Papakonstantinou, "Cybersecurity as praxis and as a state: The EU law path towards acknowledgement of a new right to cybersecurity?," *The computer law and security report*, vol. 44, p. 105653, 2022, doi: 10.1016/j.clsr.2022.105653.
12. M. Lukings and A. Habibi Lashkari, *Understanding Cybersecurity Law and Digital Privacy : A Common Law Perspective*, 1st 2022. ed. (Future of Business and Finance). Cham: Springer International Publishing, 2022.
13. A. I. Savin and J. Trzaskowski, *Research handbook on EU internet law* (Research handbooks in European law series). Northampton: Edward Elgar Publishing, 2023.
14. J. H. Pohl, "International Data Transfers and Cybersecurity: Three Regulatory Approaches and Their Implications." United Kingdom: Cambridge University Press, 2023, pp. 134-160.
15. J. Ullrich, J. Cropper, P. Frühwirt, and E. Weippl, "The role and security of firewalls in cyber-physical cloud computing," *EURASIP Journal on Information Security*, vol. 2016, no. 1, p. 1, 2016, doi: 10.1186/s13635-016-0042-3.
16. S. Thapa and A. M. Dissanayaka, "The Role of Intrusion Detection/Prevention Systems in Modern Computer Networks: A Review," 2020.
17. L. Bilge and T. Dumitraş, "Before we knew it: an empirical study of zero-day attacks in the real world," presented at the Proceedings of the 2012 ACM conference on Computer and communications security, Raleigh, North Carolina, USA, 2012. [Online]. Available: https://doi.org/10.1145/2382196.2382284.
18. E. Džaferović, A. Sokol, A. A. Almisreb, and S. Mohd Norzeli, "DoS and DDoS vulnerability of IoT: A review," *Sustainable Engineering and Innovation*, vol. 1, no. 1, pp. 43-48, 2019, doi: 10.37868/sei.v1i1.36.
19. M. Kandias, A. Mylonas, N. Virvilis, M. Theoharidou, and D. Gritzalis, "An Insider Threat Prediction Model," Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 26-37, doi: 10.1007/978-3-642-15152-1_3.
20. T. J. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class AdaBoost ∗," *Statistics and Its Interface*, vol. 2, pp. 349-360, 2009.
21. J. H. Holland, "Genetic Algorithms," *Scientific American*, vol. 267, no. 1, pp. 66-73, 1992, doi: 10.1038/scientificamerican0792-66.
22. J. Kennedy and R. Eberhart, "Particle swarm optimization," 1995, vol. 4: IEEE, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.
23. Y. Xin-She and X. He, "Firefly Algorithm: Recent Advances and Applications," *arXiv.org*, 2013, doi: 10.48550/arxiv.1308.3898.
24. S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," *Knowledge-based systems*, vol. 96, pp. 120-133, 2016, doi: 10.1016/j.knosys.2015.12.022.
25. S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in engineering software (1992)*, vol. 95, pp. 51-67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.
26. L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Systems with Applications*, vol. 191, p. 116158, 2022/04/01/ 2022, doi: https://doi.org/10.1016/j.eswa.2021.116158.

27. J. Gurrola-Ramos, A. Hernàndez-Aguirre, and O. Dalmau-Cedeño, "COLSHADE for Real-World Single-Objective Constrained optimization Problems," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 19-24 July 2020 2020, pp. 1-8, doi: 10.1109/CEC48606.2020.9185583.
28. D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation,* vol. 1, no. 1, pp. 67-82, 1997, doi: 10.1109/4235.585893.
29. N. Bacanin, M. Zivkovic, L. Jovanovic, M. Ivanovic, and T. A. Rashid, "Training a Multilayer Perception for Modeling Stock Price Index Predictions Using Modified Whale Optimization Algorithm," in *Computational Vision and Bio-Inspired Computing*, Singapore, S. Smys, J. M. R. S. Tavares, and V. E. Balas, Eds., 2022// 2022: Springer Singapore, pp. 415-430.
30. A. Minic *et al.*, "Applying Recurrent Neural Networks for Anomaly Detection in Electrocardiogram Sensor Data," *Sensors,* vol. 23, no. 24, p. 9878, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/24/9878.
31. M. Zivkovic, L. Jovanovic, M. Ivanovic, A. Krdzic, N. Bacanin, and I. Strumberger, "Feature Selection Using Modified Sine Cosine Algorithm with COVID-19 Dataset," in *Evolutionary Computing and Mobile Sustainable Networks*, Singapore, V. Suma, X. Fernando, K.-L. Du, and H. Wang, Eds., 2022// 2022: Springer Singapore, pp. 15-31.
32. N. AlHosni *et al.*, "The XGBoost Model for Network Intrusion Detection Boosted by Enhanced Sine Cosine Algorithm," in *Third International Conference on Image Processing and Capsule Networks*, Cham, J. I.-Z. Chen, J. M. R. S. Tavares, and F. Shi, Eds., 2022// 2022: Springer International Publishing, pp. 213-228.
33. R. Damaševičius *et al.*, "Decomposition aided attention-based recurrent neural networks for multistep ahead time-series forecasting of renewable power generation," (in eng), *PeerJ Comput Sci,* vol. 10, p. e1795, 2024, doi: 10.7717/peerj-cs.1795.
34. S. A. Mirjalili and A. H. Gandomi, *Comprehensive metaheuristics : algorithms and applications*. London, England ;: Academic Press, 2023.
35. W. Luo and X. Yu, "Quasi-reflection based multi-strategy cuckoo search for parameter estimation of photovoltaic solar modules," *Solar Energy,* vol. 243, pp. 264-278, 2022/09/01/ 2022, doi: https://doi.org/10.1016/j.solener.2022.08.004.
36. J. Glasser and B. Lindauer, "Bridging the Gap: A Pragmatic Approach to Generating Insider Threat Data," in *2013 IEEE Security and Privacy Workshops*, 23-24 May 2013 2013, pp. 98-104, doi: 10.1109/SPW.2013.37.
37. B. Lindauer, "Insider Threat Test Dataset," ed: Carnegie Mellon University, 2020.
38. L. S. I. Lundberg S M, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems,* vol. 30, pp. 4765 - 4774, 2017, doi: https://doi.org/10.48550/arXiv.1705.07874.