THE IDS SECURITY CHALLENGE SOLUTIONS OFFERED BY METAHEURISTIC OPTIMIZATION

Dušan Cvetković *,
[0009-0006-2436-4740]
Miodrag Živković,
[0000-0002-4351-068X]
Nebojša Bačanin Džakula
[0000-0002-2062-924X]

Singidunum University,
Belgrade, Serbia

Abstract:

The concern of this paper is to analyze the potential solutions offered by metaheuristic optimization for intruder detection systems, which have become standard due to their use throughout industries. Most recent trends have caused a large influx of potentially vulnerable devices, leading to the increasing challenge of properly detecting what constitutes a false positive or true positive detection. The aim of the research is twofold:

a) to identify aspects of the intrusion detection system that can be improved

b) to identify methods via which this improvement can be achieved.

The methodology of meta-research includes a comparative analysis of the systems based on secondary sources (papers published in prestigious journals) and accompanying references to the theoretical and industrial aspects.

The first step is to analyze the optimization techniques, chosen as the case studies, such as the genetic algorithm, firefly algorithm, chimp optimization algorithm, etc. In the next step, the paper diagnoses the security challenges faced by modern IDS solutions and discusses the proposed improvement (and optimizations) offered by the previously mentioned metaheuristic optimizations.

Keywords:

Intrusion Detection System, Classification Method, Metaheuristic Optimization, Meta-analysis.

INTRODUCTION

IDS is one of the key constituents of a comprehensive cybersecurity infrastructure. It is used for monitoring various communications, both within the confines of an organization's network and outside it, as outbound traffic. One of the most vital roles it performs is the detection of anomalous activity, regardless of whether it is a new unknown address attempting to access a server secured only for privileged users or it is a sudden large transfer of files to an unknown destination (device, server, or IP address). With a steep increase in the number of devices, the average user employs due to the development of IoT, systems like IDS, when it comes to parsing which alerts and detections are genuine and which are not, are faced with a great challenge.

Correspondence:
Dušan Cvetković

e-mail:
dusan.cvetkovic.24@singimail.rs

DOI: 10.15308/Sinteza-2025-29-34

Many attacks remain invisible to an automated system because they inhabit the border between the rational and emotional. In these cases, the strategy of AI deployment – regarding which algorithm would be best suited to uncover such attacks – must be considered in advance. The inherent cost-benefit analysis reveals that running an AI model is still expensive, especially if it is to be dedicated to a singular purpose. The cost can be both financial and technical, in terms of limited processing power available to a machine. When considering the different kinds of machine learning models, the classification models stand out as the most optimal solutions. Classification machine learning models focus on correctly labeling and identifying sets of data (a good example would be the separation of spam emails from legitimate correspondence).

Primarily, the model must be able to separate the intentional and unintentional insider malicious actors, [1] – the latter being users who become malicious actors by accident. When security policies and rules are not followed properly, they create an opening for malicious actors to exploit. Early detection of such occurrences is a way of improving the onboarding and security education of users [2]. On the opposite side, there are intentional malicious actors, the masquerader and the traitor. Masqueraders acquire access to the system through existing compromised credentials. Through impersonation, they seek to gain access to sensitive materials and vulnerable systems before abandoning their identity, and it is not uncommon for a user's unintentional action to create a potential opening for an attack by a masquerader. The most dangerous threat is the traitor – a legitimate user who leverages his access, knowledge and trust to further malicious activity within an organization [3]. A machine learning model can be based on different learning styles, specifically bagging and boosting. In this paper, the focus is on boosting learning models characterized by the sequential processing of data.

2. CHALLENGES FACED BY IDS

The majority of current-day intrusion detection systems use static rule definitions. Drawing data from either publicly available or paid databases, they rely on humans to define the criteria that will trigger an alert. A unique challenge posed by IDS systems is the sheer volume and diversity of data that must be processed and analyzed in a way so as not to disrupt communication and data traffic. In most cases, an IDS uses mirrored traffic instead of the actual real-time flow, although this leads to a slower response time.

The solution suggested in the paper is to use models similar to the CatBoost classification model that was additionally improved by an optimization algorithm. With this method, the machine learning model is able to learn what the typical activities of users and the network are, while any deviation from those standards would produce an alarm and force an investigation by an analyst. The procedure is highly effective in detecting the traitor intruder type. Since traitors are able to mask their malicious activity within their regular actions naturally, a holistic approach is necessary to face the challenges of spotting an obvious pattern in the activity log. Accordingly, if the traitor has sufficient technical expertise or privilege, it becomes trivial for him to bypass publicly available static rules.

LightGBM (light gradient boosting machine) can be used to deal with the large volume of data generated by the IDS. A greater data volume necessitates greater sampling, which in turn demands more resources. LightGBM communicates and compares the local best split points to find the single best one. Further, this reduces the sample sizes and gives better performance compared to similar algorithms. This comes with the compromise that configuring for such results is more challenging [4].

Extreme Gradient Boosting (XGBoost) is another potential technique that is applicable to the field of cybersecurity, specifically in the context of real-time systems, under which most security and industrial systems are classified. The key advantage is the ability to handle varied and complex datasets, which have to be contrasted and compared to come to a proper conclusion regarding the validity of a threat or alert [5].

3. METAHEURISTIC OPTIMIZATION SOLUTIONS

Metaheuristic optimization is a form of optimization that does not guarantee a globally optimal solution, although, in the field of cybersecurity, it could be argued that there is no such thing. As the field of optimization is still rapidly developing with many schools of thought based upon different approaches, this paper looks only into the ones who sought inspiration in animals and natural processes – algorithms like particle swarm optimization (PSO), artificial bee colony (ABC), firefly algorithm (FA), bat algorithm (BA), chimp optimization algorithm (ChOA), etc. There are other solutions that are inspired by mathematical approaches, such as the sine cosine algorithm, where the mathematical functions are used to bound values through each iteration and, at

the same time, to ensure smaller steps [6]. Swarm-based algorithms, on average, place a higher emphasis on exploration and, for that reason, will be considered above other solutions.

For example, ChOA was developed with the intent to emulate the hunting techniques and tactics employed by actual chimpanzees. This draws natural parallels with the red and blue team dynamics seen in cybersecurity when the blue team is hunting down the red team and employing various agents/tools in order to corner and identify the opponents. Combating threat actors requires multi-dimensional searches and trans-media analysis of log sources, which is also the strength of this algorithm. ChOA can be further enhanced with the incorporation of quasi-adaptive learning (QRL). In this way, the solution is split into two halves, where only one is created by applying the conventional unmodified ChOA, while the other uses QRL. Additionally, there is a rollback functionality added, meaning that in case of stagnation during an iteration, the altered ChOA can go back to a previous one to attempt a new solution. This novel algorithm was therefore named iteration stagnation aware ChOA (ISA-ChOA) [4]. An example of a pseudocode for ISA-ChOA is given in Listing 1.

Another potential swarm behavior algorithm that can be used is the firefly algorithm (FA). Like ChOA it requires a form of modification to produce a performance superior to the original one. In its unmodified form, FA simulates the behavior of a swarm of fireflies that are mutually attracted based on their luminosity. This luminosity is a fitness function of their position. The fireflies with lower brightness intensity will be attracted to the ones of higher intensity; if there are no such, they will move around randomly. This cycle of attraction is repeated until achieving a satisfactory convergence or until enough time has passed (enough iterations have been reached).

One of the drawbacks of FA is a lack of exploration power. The initial distribution and positioning of the fireflies are randomly generated, and this kind of randomness often leads to immediate convergence towards sub-optimal solutions. To add more diversity and enhance the ability of the algorithm to seek optimal solutions, the following modifications were added:

- Genetic operators – uniform crossover and Gaussian mutation
- Quasi-reflection – based learning mechanism
- Dynamically adjusting the step size parameter.

Genetic operators allow for diversification, both solutions and fireflies. By combining existing and random solutions, they make an improvement in exploration that leads to new solutions, among which a more optimal one is to be found [7].

Quasi-reflection works oppositely and is comparable to the form of proof by contradiction. It seeks out the opposite solution, using the inverse values of the ones in the fitness function, which have been proven to provide the current most optimal solution.

```
Set maximum number of agents N
Set stagnation criteria st
Produce N/2  of the population P
Produce remaining agents by applying QRL
Separate agents in to simulated chimp colonies
while t<T do
  Evaluate agent fitness
  Use colony appropriate strategy to update c, f and m
  for Each search agent a in P do
    for each search agent s do
      Determine appropriate search strategy
      Update agent position
    end for
  end for
  Check for stagnation
  if Stagnation confirmed then
    Apply soft rollback
  else
    Store solutions for soft rollback
  end if
end while
```

Listing 1. ISA-ChOA optimizer pseudo code [4]

Dynamically adjusting the step size parameter in this context means reducing it, i.e., the algorithm converges on a solution, thereby altering the scope from a global to a local search. This value is bounded by a minimum that the parameter is not allowed to exceed.

These improvements create a modified version of the algorithm that is referred to as genetic operators quasi-reflected FA (GOQRFA Listing 2). It allows for the exploration of previously unknown or under-promising solutions in the hope of finding an actual promising and optimal one. The replacement for the worst-performing solution is created by mutating it or merging it with another random solution. Every subsequent iteration continuously elevates this lower boundary of the worst solution and exploits the narrowing spectrum of the potentially optimal one [8].

The FA is still not ideal even when modified since it has a potential worst-case complexity of O(N) + O(N2·T), where in the equation N is the number of fireflies and T is the number of iterations. To avoid this complexity, the initial data must be considered and analyzed in terms of distribution. For this to be effective, the sorting algorithm must have a baseline complexity that is lower than that of the worst-case scenario for FA [9].

While MFA/GOQRFA solves the issue of over-localized searches and under-exploration, the red fox algorithm (RFO) prioritizes a global search. Unlike FA, where the individual fireflies will group up together, foxes will seek out new "food sources". In this context, the food source is a potential solution, the foxes will follow in a certain direction and evaluate it with a fitness function [10]. If the newly discovered value is better than

```
Metaheuristics parameter's values initial setting
 Population P production
 P evaluation with regard to the fitness function
 for i = 1 to max iteration count do
   for every individual do
     for every better individual do
       if individual is better then
         Obtain attraction in terms of distance
         Adjust position toward the better individual
       end if
     end for
   Evaluate and update individuals in population P
   Produce novel solution by applying genetic crossover mechanism
   Subject novel solution to mutation
   Replace the worst-performing individual with a novel generated solution
 end for
 Return top-performing solution
end for
```

Listing 2. Modified firefly algorithm pseudocode [10]

```
Generate 1/2 of the optimization group P
Apply QRL to initialize the later portion of solutions.
Define θ=0.8
while T>t do
  Assess group fitness
  Select a random value for ψ
  if ψ>θ then
    Apply FA search
  else
    Apply RFO search
  end if
  θ=θ-0.4
end while
```

Listing 3. HARFO pseudocode [3]

the previous one, the fox will stay in place; if not, it will return to their prior position. This is comparable to the behavior of the chimps in ISA-ChOA.

On the other hand, RFOs can struggle to find a balance between exploitation and exploration. Too much exploitation makes it run into similar problems as FA, while too much exploration leads to diminishing returns, where each iteration leads to no significant improvements or new solutions. These issues can be addressed in the same way as in FA, i.e., with quasi-reflexive learning. One-half of the initial population is initialized in the same way as in unmodified RFO, while the other half is initialized through quasi-reflexive learning, through a random function that operates within the upper and lower bounds of the region being searched. This version of RFO is known as the hybrid adaptive RFO (HARFO), which is shown in the form of a pseudocode in Listing 3 [3].

This is a non-exhaustive list of swarm-based algorithms. An analysis of other optimization solutions like bat algorithm, artificial bee colony, lizard algorithm, etc., is left for subsequent papers. Just like algorithms, the search for a solution must be limited to a certain scope, and first, it is necessary to find a locally optimal solution before expanding and evaluating other possibilities.

4. ANALYSIS

In the analysis, the paper considers three points deriving from the previously listed algorithms and their implementations: the challenges of training models with adequate realistic data; the proven improvement of outcomes when employing the optimization algorithms; and the cost and requirements that would precede any implementation.

The first issue created using these optimization models is the acquisition of and training on real-world data [3]. Much of the cited analysis was done on simulated data and based on publicly available datasets. However, as the security environment is different and uniquely designed for a certain system or corporation, that means that the variety, quality, and details within logs and datasets will greatly differ on a case-by-case basis. An additional concern is whether the system handles information from third parties or from users who have not consented to have their data used to train a machine learning model. Furthermore, as blind application of a model will not yield high-quality results and the precision required in cybersecurity, the full understanding of a model's decision-making must precede any evaluation of a model's performance. There is a clear separation between simpler models, which can be reduced to decision trees, and more complex models, which would require mathematical analysis and detailed interpretations [5].

Secondly, there is an empirically and experimentally verified improvement in performance when optimization algorithms are used on machine learning models. Specifically in the case of comparative analysis and multi-class classification of various types of attacks against IoT systems, "a custom-altered optimizer (..) resulting in the best performing models (..) attained a supreme accuracy level of 99.83%" [4]. These results show a large gap between non-optimized and optimized machine learning models. With the benefits of optimization clearly established, the cost and requirements have to be evaluated as well.

Thirdly, real-time mitigation as well as real-time attack detection is a key concern since the impact on customers, third parties, and alike must be minimal to nonexistent. This is a requirement for a modern and competitive security solution. The optimization algorithms would have to be evaluated in a real-world case, and that would mean analyzing real-world data, which is unpredictable and ever-changing based on trends and events outside of the programmer's or security analyst's control. Also, it would be in addition to the hardware requirements in terms of data throughput and processing requirements. During the learning stages of the algorithm, the data would have to be doubled to prevent any degradation or damage in case of false positives until the model establishes a baseline of activity.

The conclusion of the analysis emphasizes that the empirical data gained through simulation is twofold. First, it shows that the use of metaheuristic optimization algorithms guarantees better results, and second, that the practical and logistical concerns need to be resolved before any testing could be performed in a practical scenario or environment. Furthermore, either the legality of data harvesting would have to be legislated, or the data would need to be sufficiently anonymized. This paper does not consider anonymity to be absolute or even theoretically possible if there is enough data from a user available.

5. CONCLUSION

ISA-ChOA, GOQRFA, and HARFO are presented as potentially good solutions to the problems faced in the dynamic landscape of cybersecurity. By emulating a hunting scenario, solutions like ISA-ChOA and HARFO are appropriate for dealing with masqueraders since they lean more towards exploration and testing out new potential solutions. Meanwhile, GOQRFA leans more towards detecting both the accidental malicious actor and the traitor, by highlighting and focusing on the specific areas of interest. Thus, it can help identify exceptionally vulnerable areas or highly trafficked systems that can be leveraged during an exfiltration.

Since there is no single universal algorithm that can identify every single type of attacker efficiently, this paper proposes a hybrid solution defined as a parallel use of the three suggested metaheuristic optimization algorithms (FA, ChOA, RFO). The greatest risk to an IDS is a threat that it has no rules for – a zero-day exploit not yet in any database. By employing AI, a system is able to construct its database, which is both wholly unique to the company and has a use case relevant to the industry it operates. Clear optimization is a requirement for avoiding unnecessary waste of resources or inefficient searches that would interfere with regular operations. Although the investment in resources is non-trivial, the benefits and improved results are proven and verified. Therefore, algorithms like ISA-ChOA, GOQRFA, and HARFO present a potential solution for efficient intruder detection systems.

REFERENCES

[1] L. Liu, O. De Vel, Q.-L. Han, J. Zhang and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 2, pp. 1397-1417, 2018. doi: 10.1109/COMST.2018.2800740.

[2] F. L. Greitzer, J. Strozer, S. Cohen, J. Bergey, J. Cowley, A. Moore and D. Mundie, "Unintentional insider threat: contributing factors, observables, and mitigation strategies," in *2014 47th Hawaii International Conference on System Sciences*, Waikoloa, HI, USA, 2014. doi: 10.1109/HICSS.2014.256.

[3] D. Mladenovic, M. Antonijevic, L. Jovanovic, V. Simic, M. Zivkovic, N. Bacanin, T. Zivkovic and J. Perisic, "Sentiment classification for insider threat identification using metaheuristic optimized machine learning classifiers," *Scientific Reports,* vol. 14, no. 1, p. 25731, 2024. doi:10.1038/s41598-024-77240-w

[4] M. Antonijevic, M. Zivkovic, M. Djuric Jovicic, B. Nikolic, J. Perisic, M. Milovanovic, L. Jovanovic, M. Abdel-Salam and N. Bacanin, "Intrusion detection in metaverse environment internet of things systems by metaheuristics tuned two level framework," *Scientific Reports,* vol. 15, no. 1, p. 3555, 2025. doi:10.1038/s41598-025-88135-9

[5] P. Dakic, M. Zivkovic, L. Jovanovic, N. Bacanin, M. Antonijevic, J. Kaljevic and V. Simic, "Intrusion detection using metaheuristic optimization within IoT/IIoT systems and software of autonomous vehicles," *Scientific Reports*, vol. 14, no. 1, p. 22884, 2024. doi:10.1038/s41598-024-73932-5

[6] M. Dobrojevic, M. Zivkovic, A. Chhabra, N. S. Sani, N. Bacanin and M. M. Amin, "Addressing internet of things security by enhanced sine cosine metaheuristics tuned hybrid machine learning model and results interpretation based on shap approach," *PeerJ Computer Science,* vol. 9, p. e1405, 2023. doi: 10.7717/peerj-cs.1405

[7] S. Mirjalili and S. Mirjalili, "Genetic algorithm," *Evolutionary algorithms and neural networks: Theory and applications*, pp. 43-55, 2019. source: https://www.iieta.org/journals/jesa/paper/10.18280/jesa.560601

[8] N. Bacanin, M. Zivkovic, T. Bezdan, K. Venkatachalam and M. Abouhawwash, "Modified firefly algorithm for workflow scheduling in cloud-edge environment," *Neural computing and applications*, vol. 34, no. 11, pp. 9043-9068, 2022. DOI:10.1007/s00521-022-06925-y

[9] M. Zivkovic, M. Tair, N. Bacanin, S. Hubalovsky and P. Trojovsky, "Novel hybrid firefly algorithm: An application to enhance XGBoost tuning for intrusion detection classification," *PeerJ Computer Science,* vol. 8, p. e956, 2022. DOI:10.7717/peerj-cs.956

[10] N. Savanovic, A. Toskovic, A. Petrovic, M. Zivkovic, R. Damasevicius, L. Jovanovic, N. Bacanin and B. Nikolic, "Intrusion detection in healthcare 4.0 internet of things systems via metaheuristics optimized machine learning," *Sustainability*, vol. 15, no. 16, p. 12563, 2023. doi:10.1038/s41598-024-73932-5