D-ACO/GA - A bio-inspired strategy for feature selection in anomaly traffic detection in smart Internet of Things environments.

Ant. Helder G. L. Júnior, Dr. Joaquim Celestino Jr, Dr. Gustavo Campos
State University of Ceara (UECE), Fortaleza, CE, Brazil
helder.leoncio@aluno.uece.br
joaquim.celestino, gustavo.campos @uece.br

***Abstract*—With the growth of the Internet of Things (IoT) and its adoption, organizations are confronted with security challenges. This highlighted the need for controls to reinforce information security. Such growth comes with a significant increase in cyber threats accompanied by delays in detection. Consequently, the use of artificial intelligence techniques, especially machine learning and bio-inspired algorithms, like the ant colony algorithm (ACO) and the Genetic algorithm (GA), have become essential for enhancing IoT security by identifying and countering anomalies. This paper proposes an intrusion detection strategy, D-ACO/GA, for IoT environments. The approach combines both ACO and GA with supervised machine learning classification techniques. Like this, ACO and GA are applied to data subset selection, targeting the identification of crucial characteristics for anomaly detection. Through this comprehensive strategy, we achieve a sensible accuracy of 0.9937% with the 14 features selected by ACO, and 0.9886% accuracy with the 12 features selected by GA.**

***Index Terms*—IoT, Artificial Intelligence, IDS, Ant Colony Optimization, Genetic algorithm.**

I. INTRODUCTION

In the world of information and communication technologies, there have been some disruptive events, such as the invention of the Internet of Things (IoT). The IoT, with its ubiquitous proposal to connect objects to the global computer network, appears to be an ongoing development process rather than a single event. It represents a rapidly evolving paradigm within modern telecommunications networks, with a particular focus on wireless networks. At its core, this concept revolves around the idea of the ubiquitous presence of smart objects in our surroundings. Examples of these devices include wireless identification tags (RFID), sensors, actuators, and smartphones [1].

IoT devices are becoming increasingly popular. They are firmly establishing themselves in the technological environment and permeating everyday life. They address various aspects, including social, environmental, health, industrial, privacy, and information security issues. The increasing prevalence of IoT in our environment brings major concerns and, therefore, challenges to the security of Internet of Things applications at various levels [2].

There are many threats to digital architectures, whether in a traditional computer system or in an IoT network. The problem is particularly pronounced in these devices due to their open and heterogeneous architecture. This complexity makes protecting these types of devices much more challenging, consequently creating numerous security challenges [3].

The use of mechanisms to enhance security in IoT networks is a topic discussed in detail in the literature by 4, 2, 5. These authors address the topic from various perspectives, including architectural improvements, solutions for intrusion detection systems with machine learning capabilities, blockchain-based applications, and even mechanisms inspired by nature.

To ensure IoT security, it's essential to reprocess and scrutinize data during transmission to remove non-conforming or redundant information that could compromise security. Feature Selection (FS) techniques are used to process the vast amount of data in the network, enhancing the efficiency and accuracy of the communication system and preventing errors and system downtime. [4]

Among these AI methods, bioinspired algorithms and evolutionary computation are techniques inspired by natural and biological processes and phenomena, such as the Ant Colony Algorithm (ACO) and Genetic Algorithm (GA). These algorithms mimic the behavior of natural systems to solve complex problems. [6]. They can be used to select the most relevant features from a large set of attributes for efficient information analysis. These techniques can also be employed to reduce the data's dimensionality, facilitating its interpretation and visualization. [7].

Ant colony optimization (ACO) is a metaheuristic that can be applied to a variety of problems, including feature selection [7]. It works by mimicking the behavior of foraging ants and applying the same principles of cooperation and collaboration between individual ants to explore the search space. ACO uses pheromone trails to guide the search process, indicating the probability of reaching a solution [8].

Genetic algorithms are another metaheuristic inspired by the natural evolution of species and were devised by [9]. Using biological terms such as genes and phenotypes, these computational algorithms simulate evolutionary processes. The essence is that the "best" individuals pass their hereditary characteristics to new generations, emulating natural selection.

This study contributes to information security field by elaborating on a strategy for feature selection, aimed at developing a model for intrusion detection in the Internet of Things, named

D-ACO/GA. The approach combines bioinspired mechanisms with supervised machine learning techniques to propose a new intelligent mechanism for intrusion detection in IoT environments. The approach introduces a robust and scalable optimization technique that can efficiently search through potential feature sets to address the intrusion detection problem.

The structure of this paper is laid out as follows. Section II delves into prior research on feature selection techniques in the realm of intrusion detection. Section III introduces the Ant Colony Algorithm and Genetic algorithms. Section IV unveils our proposed D-ACO/GA approach for feature selection in intrusion detection, while Section V showcases computational experiments and offers a succinct discussion on the outcomes. Concluding remarks and directions for future research are presented in the final section.

II. LITERATURE REVIEW

Work by Aghdam et al. [7] proposed a research on the problem of intrusion detection in computer networks. The goal presented in this work is to identify important resources for building an efficient and effective Intrusion Detection System (IDS). The work proposes an IDS model based on Ant Colony Optimization (ACO). In this proposal, ACO is presented as a feature selection strategy (FS) to reduce the dimensionality of the dataset and identify relevant features without compromising prediction accuracy. For this modeling, a graph representation is used, where the search for a minimum-cost path is postponed, with the features as the vertices while the edges represent the algorithm's selection of the next feature.

Nur [10], present a new hybrid IDS for IoT using the Binary Grey Wolf Optimizer (BGWO) for feature selection and Naive Bayes (NB) as the classification method. BGWO is a bio-inspired method that examines the hunting behavior and social leadership of grey wolves in nature. According to the GWO author, it is divided into four groups: alpha (α), beta (β), delta (δ), and omega (ω). The fittest wolves are alpha, beta, and delta, leading other wolves into the search space. The hunting process is carried out by the swarm using information from α, β, and δ to determine the prey's position, calculate the best solutions within the population, and perform an optimal search.

In a study by Alweshah [4], a new model for feature selection called Feature Selection (FS) is introduced. This model uses the Emperor Penguin Colony method (EPC) and a K-nearest neighbors (KNN) classifier to address feature selection in IoT intrusion detection systems. The algorithm is inspired by penguin behaviors, specifically their location and temperature-related behaviors. It was tested on nine IoT datasets like Baby Monitor, Danmini Doorbell, and others. The evaluation focused on six metrics: classification accuracy, fitness value, selection size, recall, precision, F-measure, and convergence speed.

In the study by [11], a wrapper-based ACO methodology is used for feature selection and simultaneous classification. ACO integrates correlation information, Gini classification, and pheromone learning to refine attribute selection over iterations. The ACO is further improved using a combination of three heuristics addressing redundancy, relevance, and feature correlation. Pearson correlation and Gini importance measure also enhance the feature selection process. The effectiveness of the algorithm is evaluated using a Random Forest classifier and eight reference datasets, including Biodegradation, Cardiotocography, and others.

In [12], the authors detail the exploration and application of Genetic Algorithms (GA) for feature selection. Particularly, a binary GA was used for dimensional reduction to enhance the performance of the concerned classifiers. They proposed that one hundred features were extracted from a set of images found in the Flavia dataset. The extracted features include Zernike Moments (ZM), Fourier Descriptors (FD), Legendre Moments (LM), Hu's 7 Moments (Hu7M), Texture Properties (TP), and Geometrical Properties (GP). The main contribution of the authors is the detailed documentation of the GA Toolbox in MATLAB, and the development of a GA-based feature selector using a novel fitness function, kNN-based classification error, which enabled the GA to obtain a combinatorial set of features.

III. THE ANT COLONY ALGORITHM

This algorithm is proposed by Dorico [8]. ACO is motivated by the collective behavior of real ants as they search for food, using the same principles of cooperation and collaboration to explore the search space. These agents employ a bioinspired strategy based on the principle that ants operate individually with simple rules. During their journeys, they deposit pheromones, and collectively, they have the ability to detect variations in the concentration of these pheromones in the vicinity. As a result, they tend to move in the direction where the concentration is higher. The functioning of ACO can be comprehended through three expressions as depicted in formulas (1), (2), and (3).

A. Heuristic Information

$$P_{xy}^{k} = \frac{\left[\Gamma_{xy}\left(t\right)\right]^{\alpha} . \left[\eta_{xy}\left(t\right)\right]^{\beta}}{\sum_{|\epsilon N_{x}^{y}} \left[\Gamma_{xy}\left(t\right)\right]^{\alpha} . \left[\eta_{xy}\left(t\right)\right]^{\beta}} \tag{1}$$

The Heuristic information is the expectation of transitioning to the next edge, where P_{xy}^{k} represents the probability of a specific ant k leaving point x and choosing to reach point y, taking into account the influence of pheromone in α, the distance in β, the pheromone value present along the path between x and y in $\Gamma_{xy}\left(t\right)$, the inverse of the distance between x and y represented by $\eta_{xy}\left(t\right)$, and the summation of all products between pheromone and the paths taken by ant k departing from edge x and ending at each of the points y.

B. Pheromone Update

$$\Delta\Gamma_{xy}^{k} = \frac{Q}{d_{k}} \tag{2}$$

The pheromones present along a specific trail indicate the quality of the path traversed by the ant and serve as a memory for other ants within the colony. When other ants cross the same edge, they are more likely to follow the same path if there is a higher concentration. The calculation of the pheromone deposited on each route is presented in equation (2), where $\Delta\Gamma_{xy}^{k}$ characterizes the pheromone deposited by ant k after passing through the route between points x and y, the constant Q, and d_k representing the total distance traveled by ant k.

$$\Gamma_{k}^{xy} = (1-\sigma)\,.\Gamma_{k}^{xy}\,(t-1) + \sum_{k=1}^{m} \Delta\Gamma_{k}^{xy}(t) \tag{3}$$

Finally, as each ant moves through the graph, it leaves a trail of pheromones on the edges it traverses. At each interaction, it is updated by equation (3), where Γ_{k}^{xy} corresponds to the pheromone updated in the path between x and y, σ characterizes the pheromone evaporation rate, $\Gamma_{k}^{xy}\,(t-1)$ reflects the total pheromone from the previous iteration between edges x and y, and $\sum_{k=1}^{m} \Delta\Gamma_{k}^{xy}(t)$ represents the summation of all ants k that have passed through edges x and y. This process is repeated until the algorithm converges to an optimal solution.

IV. The Genetic Algorithm

Genetic algorithms, developed by Holland, [9] are inspired by processes observed in natural species evolution. Fundamentally, much like in the biological theory of natural systems, the "fittest" individuals survive and produce offspring that inherit their characteristics. These descendants typically resemble, or have similar phenotypes to, their predecessors. Drawing a parallel, a genetic algorithm starts with a population of randomly generated individuals, assesses each one, selects the "best" ones, and then performs genetic operations like crossover and mutation to form a new population. [13] This adaptive process is versatile and can address various optimization challenges, such as feature selection, for example.

In [13], the author provides a summary of the following steps to construct a Genetic Algorithm (GA):

1) **Chromosome Representation:** Decide how chromosomes will be represented, commonly using strings or vectors.
2) **Initial Population Creation:** Randomly generate an initial set of solutions, called the population, consisting of a number "p" of configurations.
3) **Solution Evaluation:** Assess each solution (individual) in the population using an objective function to determine its quality or fitness.
4) **Genetic Operations:** Apply genetic operators to produce a new generation of solutions:
    a) **Selection:** Choose the fittest individuals for reproduction based on their evaluation.
    b) **Crossover:** Combine traits from two selected parents to produce new individuals.
    c) **Mutation:** Introduce minor random changes in some individuals to ensure diversity and explore new solutions.
5) **Iteration:** Repeat the evaluation and genetic operations steps for multiple generations or until a stopping criterion is met.
6) **Conclusion:** At the end of the iterations, the best individual from the last population is considered as the solution to the problem.

V. Proposed D-ACO/GA Method

This section presents the algorithm D-ACO/GA for IoT model. The Figure 1 depict the proposed approach. It's divided into the following parts including, data preparation, feature selection, classification and detection, and performance evaluation.

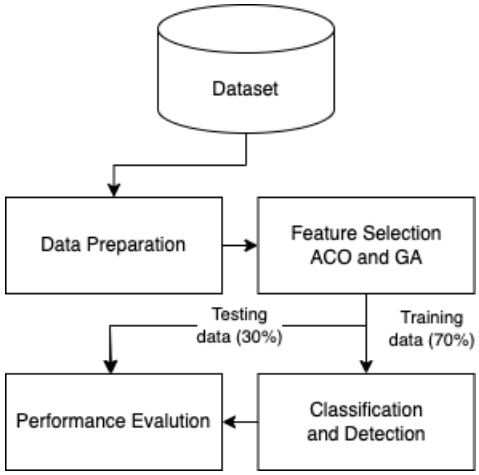

Fig. 1. Proposed D-ACO/GA system.

A. Data preparation

The effectiveness of the learning algorithm relies heavily on the nature of the input data it receives. To enhance the quality of the data, several data preprocessing techniques are employed, including transformation, normalization, and sampling. From IoT devices, a diverse array of data is gathered, containing numerous features. However, not all of these features contribute meaningfully to the classification task; some are considered redundant and irrelevant. Consequently, these superfluous and non-informative features are removed. Additionally, numerical feature values within the dataset may exhibit fluctuations. To mitigate this, a normalization process is applied to confine these values to a range between 0 and 1. In this research, the MinMax scaling function is utilized for this purpose. The dataset may exhibit an imbalance, with instances of intrusive traffic significantly outnumbering instances of normal traffic. This imbalance adversely impacts the classifier's performance. To address this issue, dataset must be rebalanced through random sampling.

B. Feature selection using ACO

In this section, we introduce a semi-supervised feature selection algorithm using ACO. Within this framework, graph nodes symbolize features, while edges represent the relationships

between them. Our proposed algorithm combines a learning algorithm with a filter-based technique. Subsequent sections will further explore the characteristics of the proposed D-ACO/GA, which leverages a heuristic linear approach for learning. The overarching structure of this algorithm is depicted in Algorithm 1.

Algorithm 1 ACO-based Feature selection for IDS

```
input: matrix of distances between the features
output: the best subset of features
procedure ACO
    while not termination condition is met do
        for each interaction in the application do
            init of the parameters: α, β, Q, p, and τ0 > m=n
            – number of ants is equal to the number of towns
            for generated ant population do
                calculate partial fitness for each ant
            end for
            bestsolutions ← partial fitness
        end for
        update pheromone trails
    end while
end procedure
```

The model feature selection process begins with the calculation of the Pearson correlation between the dataset attributes, aiming to create a distance matrix used in the ACO algorithm. The algorithm runs until a stopping condition is met. The initial parameters include alpha, beta, rho, Q, and the number of ants.

Once the parameters are properly configured, a population of ants is generated and distributed among the dataset attributes. After this distribution, a set of features is selected based on Equation 1. Subsequently, the fitness function is calculated, which is based on the relationship between the classification accuracies obtained by each ant and the number of selected attributes.

The result of the fitness function is compared to the previous iteration to determine if there has been an improvement. If so, the best selection is stored, and the pheromones deposited by the ants are updated according to Equations 2 and 3. When the stopping condition is met, the algorithm returns an optimized dataset, ready to be used in model classification and detection.

C. Feature selection using GA

The Genetic Algorithm method is well-suited for feature selection. According to the algorithm's structure s depicted in Algorithm 1, each feature corresponds to a gene, and solutions are represented as binary arrays indicating the status of each feature. Thus, solutions equate to the population. The algorithm begins with random solutions. The initial population size is set, and the maximum number of generations is set to ensure the algorithm's convergence. Initially, the fitness function is applied to the current population with a vector of accuracies, $accGa$, representing accuracies for different trials or instances, the average accuracy (or mean accuracy) is computed as:

$$\text{avg\_acc} = \frac{1}{n}\sum_{i=1}^{n} accGa_i$$

where: avg_acc is the average accuracy. n is the number of elements in the $accGa$ vector. $accGa_i$ is the accuracy of the i^{th} element in the $accGa$ vector.

After calculating avg_acc, it's appended to a list named fitness_results. , and this is followed by parent selection using the roulette wheel selection method. Subsequently, parents are chosen with probability:

$$\text{probabilities}_i = \frac{\text{inverse\_fitness}_i}{\sum_j \text{inverse\_fitness}_j}$$

where j ranges over all individuals in the set, such that to calculate the inverse fitness value for each j:

$$\text{inverse\_fitness}_i = \frac{1}{1 + \text{fitness}_i}$$

where i is the index of a specific individual in the set.

Uniform crossover occurs between parents to generate a pool of children; each gene has a 50% chance to be flipped. After this, mutation takes place with a 5% probability for each feature to be mutated. Mutation introduces appropriate diversification into the search process. Subsequently, the fitness function is applied again. The algorithm continues looping until the maximum generation is reached. The final result is the best one among the top solutions across all generations.

Algorithm 2 Genetic Algorithm for Feature Selection

```
input: PopSize NumGen, Mutation and Crossover rate,
Selection method
output: the best subset of features
procedure GA
    initialize population
    for each generation in n_generations do
        compute current population fitness
        identify best individual of generation
        select parents with tournament
        perform crossover among parents
        apply mutation to offspring
        update population
        store generation information
    end for
end procedure
```

D. Classification and detection

Utilizing the feature subset derived from the feature selection phase, we conduct classification and detection tasks using a machine learning classifier. In this research, our primary focus is on binary classification, specifically centered on assessing the likelihood of an attack occurrence. The classifiers employed in this study encompass K-Nearest Neighbors

(KNN), Decision Trees, Random Forest, and Naive Bayes. This subsection provides a concise overview of these machine learning classifiers.

1) KNN: K-Nearest Neighbors (KNN) is one of the simplest supervised machine learning classifiers available. This classification method leverages statistical measures to assess the proximity of data instances. The 'K' in KNN represents the number of neighbors considered for classification. Instances with a high degree of similarity are assigned to the same class. KNN enables the labeling of new instances based on the patterns observed in previously labeled ones.

2) Random Forest: Random Forest is a powerful machine learning algorithm known for its ensemble learning approach, reducing overfitting, handling missing and categorical data, feature importance assessment, and strong performance in various data scenarios.

3) Decision Tree: The Decision Tree classifier is widely used for classification tasks, thanks to its interpretability, versatility (handling various data types), simplicity, and ability to capture non-linear patterns. It's also robust to outliers, scalable, and can be a foundation for ensemble methods, making it valuable in many domains.

4) Naive Bayes: Naive Bayes is a probabilistic machine learning algorithm used for classification. It relies on the assumption of feature independence, simplifying calculations. It's efficient, making it suitable for real-time applications. Naive Bayes is commonly used in spam email detection and sentiment analysis, among other tasks. Its ability to handle high-dimensional data and its speed make it a valuable tool in various machine learning scenarios.

VI. Experiments and Results

The proposed D-ACO/GA approach was applied one standard Iot dataset to evaluate their effectiveness in terms accuracy and number of guaranteed features selected. The dataset employed in this study is the BoTNeTIoT-L01 dataset Alhowaide [14], which comprises data collected for our research. This dataset was generated by monitoring network traffic from nine IoT devices within a local network infrastructure utilizing a central switch. It contains valuable insights into two significant Botnet attacks, specifically, the Mirai and Gafgyt attacks. The dataset comprises twenty-three attributes along with a class label and features shown in tables IV, set within an unbalanced dataset framework. In this context, the value 0 denotes instances of attacks, while the value 1 signifies normal samples.

All tests were run on a personal computer using Jupyter Lab with Python 3.9. All of the tests were run on a personal computer platform with 16 gigabytes of RAM and an eight-core Inter(R) i7(R) CPU running Ubuntu Linux 20.04 professional.

Based on the methodology described in section V and the parameters shown in tables I and II, the following results were obtained in the feature selection module using ACO and GA. Table III presents a comparative evaluation of the algorithms, highlighting the performance of ACO in most metrics, except for the number of features selected.

In contrast, GA shows a reduced feature selection but does not perform well in other metrics. The execution time for both is notably different. In figure 2, it can be observed that ACO, over the course of 50 iterations, selects 5 top accuracy values, while GA selects only 2 specific values. The features selected by the ACO and GA can be seen in IV

TABLE I
PARAMETERS SETTING FOR ACO

Parameter	***Value***
Number of Ants	24
Model Training Epoch	50
alpha	1
beta	0.2

TABLE II
PARAMETERS SETTING FOR GA

Parameter	***Value***
Population Size	50
Number of Generations	50
Selection Mechanis	Roulette
Mutation Rate	0.05
Crossover Rate	0.5

TABLE III
EVALUATE THE PERFORMANCE ACO & GA

Evaluate	**ACO**	**GA**
Features selected	14	12
Accuracy	0.9937	0.9886
Accuracy mean	0.9829	0.9786
Precision	0.9753	0.9007
Recall	0.9973	0.9743
F1score	0.9853	0.9191
Roc Curve	0.9944	0.9617
Total Time	31631.37s	111692.20s

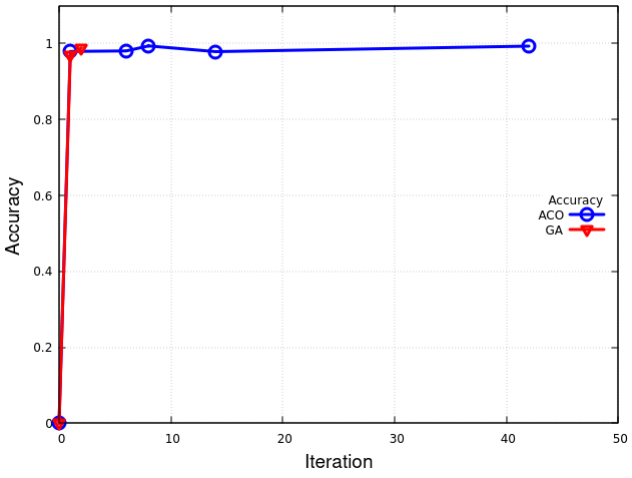

Fig. 2. INTERACTIONS x ACCURACY

In the task of network traffic classification for IoT devices, the ACO and GA algorithms demonstrated their effectiveness

TABLE IV
COMPARISON OF FEATURE SELECTION USING ACO AND GA

Original Features	Selected by ACO	Selected by GA
MI_dir_L0.1_weight	-	X
MI_dir_L0.1_mean	-	X
MI_dir_L0.1_variance	-	-
H_L0.1_weight	X	-
H_L0.1_mean	X	X
H_L0.1_variance	-	X
HH_L0.1_weight	-	-
HH_L0.1_mean	-	X
HH_L0.1_std	X	-
HH_L0.1_magnitude	X	-
HH_L0.1_radius	X	-
HH_L0.1_covariance	X	X
HH_L0.1_pcc	X	-
HH_jit_L0.1_weight	-	X
HH_jit_L0.1_mean	X	X
HH_jit_L0.1_variance	X	-
HjHp_L0.1_weight	-	-
HjHp_L0.1_mean	-	X
HjHp_L0.1_std	X	X
HjHp_L0.1_magnitude	X	X
HjHp_L0.1_radius	X	-
HjHp_L0.1_covariance	X	X
HjHp_L0.1_pcc	X	-

based on the selected features in Table IV. By utilizing cross-validation techniques, they identified the most data segments for the process, as shown in Figure 3. According to the results, the classifiers achieved an accuracy rate exceeding 0.99, except for Naive Bayes.

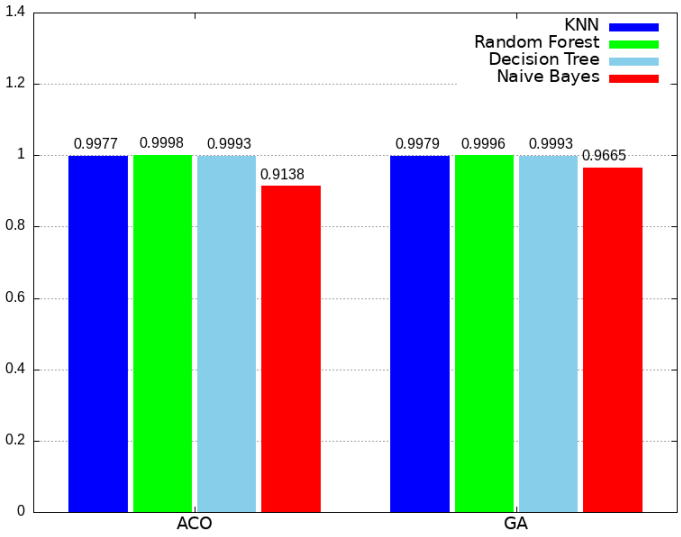

Fig. 3. FINAL ACCURACY OF CLASSIFICATION MODELS

VII. CONCLUSIONS AND FUTURE WORKS

In this work, an IDS model for IoT environments based on an ant colony optimization algorithm for attribute selection was presented. The development of this approach involved a meta-heuristic that imitates the ants' cooperative behavior to find a subset of characteristics capable of producing a satisfactory attack detection model, combined with supervised machine learning algorithms for classifying this type of anomaly. Thus, the model described with D-ACO/GA can be considered a promising and effective method for intrusion detection in Internet of Things device networks. As for future directions, an approach is proposed to improve the effectiveness and scalability of the IDS model: the adoption of the Apache Spark distributed processing environment, enabling simultaneous execution of the model on multiple processing nodes, optimizing detection speed, and efficiently handling large volumes of data.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[3] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the internet of things (iot): A security taxonomy for iot," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 163–168.

[4] M. Alweshah, A. Hammouri, S. Alkhalaileh, and O. Alzubi, "Intrusion detection for the internet of things (iot) based on the emperor penguin colony optimization algorithm," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 09 2022.

[5] G. Raja, A. Ganapathisubramaniyan, G. Anand, and Gowshika, "Intrusion detector for blockchain based iot networks," in *2018 Tenth International Conference on Advanced Computing (ICoAC)*, 2018, pp. 328–332.

[6] A. Darwish, "Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 231–246, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2314728818300631

[7] M. Hosseinzadeh Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *International Journal of Network Security*, vol. 18, pp. 420–432, 05 2016.

[8] M. Dorigo and T. Stützle, *Ant Colony Optimization*. The MIT Press, 06 2004. [Online]. Available: https://doi.org/10.7551/mitpress/1290.001.0001

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. A Bradford Book, Apr. 1975. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20path=ASIN/0262581116

[10] I. M. Nur and E. Ülker, "A novel hybrid iot based ids using binary grey wolf optimizer (bgwo) and naive bayes (nb)," 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:232119562

[11] T. Joshi, A. Lahorkar, G. Tikhe, H. Bhosale, A. Sane, and J. K. Valadi, "An improved ant colony optimization with correlation and gini importance for feature selection," in *Communication and Intelligent Systems*, H. Sharma, M. K. Gupta, G. S. Tomar, and W. Lipo, Eds. Singapore: Springer Singapore, 2021, pp. 629–641.

[12] O. Babatunde, L. Armstrong, J. Leng, and D. Diepeveen, "A genetic algorithm-based feature selection," *International Journal of Electronics Communication and Computer Engineering*, vol. 5, pp. 889–905, 07 2014.

[13] R. Linden, *Algoritmos genéticos*. Rio de Janeiro: Ciência Moderna Ltda., 2012.

[14] A. Alhowaide, I. Alsmadi, and J. Tang, "Features quality impact on cyber physical security systems," 10 2019, pp. 0332–0339.