International Journal of Safety and Security Engineering
Vol. 14, No. 6, December, 2024, pp. 1907-1912
Journal homepage: http://iieta.org/journals/ijsse

Optimizing Feature Selection for Intrusion Detection: A Hybrid Approach Using Cuckoo Search and Particle Swarm Optimization

Hadeel Qasem Gheni[1*], Wed Kadhim Oleiwi[1], Zahraa Al-Barmani[1], Mohammed A. M. Alabdali[2]

[1] Department of Computer Science, Science College for Women, University of Babylon, Hillah 51002, Iraq
[2] Department of Security, Information Technology College, University of Babylon, Hillah 51002, Iraq

Corresponding Author Email: wsci.hadeel.qasem@uobabylon.edu.iq

https://doi.org/10.18280/ijsse.140624

Received: 25 September 2024
Revised: 10 December 2024
Accepted: 20 December 2024
Available online: 31 December 2024

Keywords:
cuckoo search, particle swarm optimization, intrusion detection, IoT security, feature selection

ABSTRACT

Network security is crucial for preserving privacy and safeguarding private information. Recent laws relevant to current web services have increased the need for intrusion detection systems, which protect data and mitigate the impact of attacks. Effective feature selection is crucial for lowering dimensionality and enhancing detection accuracy to enhance the execution of IDS. The current study's objective is to offer a novel approach to feature selection by integrating Particle Swarm Optimization (PSO) and Cuckoo Search (CS) algorithms. This study evaluates the efficiency of various optimization strategies for feature selection in the CICIoT2023 dataset, which contains a wide variety of IoT attack scenarios, aimed at enhancing intrusion detection systems. To identify and prioritize the most significant features, the current methodology uses a hybrid feature selection framework that combines the advantages of both local search efficiencies from PSO and global search capabilities from CS. Following that, the chosen features are then fed into three classifiers: Multi-Layer Perceptron (MLP), Random Forest (RF), and AdaBoost. The experimental outcomes demonstrate that the CS-PSO hybrid model significantly improves attack detection accuracy, outperforming previously reported methods on the same dataset. This research contributes to advancing network security in IoT environments by addressing the growing demand for adaptive and effective IDS solutions, instilling greater confidence in the resilience of IoT networks.

1. INTRODUCTION

The number of sensitive and secret data has increased as a result of increased internet usage, which presents security risks to hackers attempting to break into networks [1]. Intrusion detection systems' significance, which is crucial for protecting data and reducing the harm caused by attacks and network system penetrations, has increased with the implementation of the latest advanced web services laws, including government, banking, email, and marketing services [2]. An intrusion occurs when someone gains access to a network or a specific computer with the intent to steal data, alter the system, sabotage, or destroy it through security weaknesses in the operating system [3]. Very effective methods for detecting intrusions are machine learning and deep learning which are accustomed to categorize attacks and identify abnormalities [4].

Optimization is one of the most important topics in many scientific fields, where, optimization algorithms refer to methods for determining the best course of action in a given situation, frequently in the context of engineering, machine learning, artificial intelligence, and other domains where optimization is essential [5]. The optimization algorithm is a technique that iteratively evaluates many solutions to produce the optimal result under particular circumstances [6].

The study's objective is to take the most key components from the used intrusion detection dataset to build a reliable and efficient intrusion detection system. These contributions are:

• The study suggests a hybrid feature selection framework that integrates the worldwide search capabilities of Cuckoo Search with the regional search efficiencies of Particle Swarm Optimization. The intention behind this combination is to efficiently determine and rank the key characteristics for intrusion detection.

• Three different classifiers get the chosen features: AdaBoost, Random Forest (RF), and Multi-Layer Perceptron (MLP). It is shown that the CS-PSO hybrid model significantly enhances these classifiers' performance by decreasing computational complexity and increasing classification accuracy.

• The CICIoT2023 dataset, which covers an extensive range of IoT attack scenarios, is used to assess the method. This dataset provides a thorough foundation for evaluating the suggested feature selection method's effectiveness in actual settings in the Internet of Things.

The work's remaining portion is described as like: Section 2 investigates the previous studies. Section 3 is methods and materials related. Section 4 explores the performance assessments and the suggested method. Section 5 addresses the conclusion and upcoming projects "form".

2. PREVIOUS STUDIES

Researchers from all throughout the world have long been interested in network security, particularly intrusion detection systems. This study makes use of the CICIoT2023 dataset, which has drawn extensive research attention.

Phan et al. [7] used a variety of methods as machine learning to evaluate the functionality of different feature selection techniques, encompassing, Information Gain (IG), Recursive Feature Elimination (RFE), Random Forest (RF), Logistic Regression (LR), and XGBoost. The analysis indicates that RFE is the most accurate feature selection technique, with an average accuracy of 95.55% across different models and configurations. RFE reaches its highest accuracy of 99.57% when paired with RF using 30 selected features. For scenarios with resource constraints, RFE remains the optimal choice, achieving 99.45% accuracy with only 5 selected features out of 46. RF is noted for its stability, providing consistent results across various models with accuracy ranging from 83% to 99.56%.

The difficulty of identifying attacks in expansive IoT networks is discussed in previous study [8]. It suggests a brand-new attack detection method that makes use of an enhanced Salp Swarm Algorithm (SSA) for feature selection and the Light Gradient Boosting Machine for classification. To choose the most relevant features and reduce dimensionality, the study improves the classic SSA by adding methods to better handle the high-dimensional feature space of IoT networks, meanwhile, IoT network data is classified using the LightGBM. The paper achieved 99.65% accuracy in binary classification and 0.9938% in multi-classification.

Mahdi et al. [9] proposed a methodology consisting of a two-stage feature selection model and an enhanced classification algorithm, resulting in exceptionally high accuracy rates. The study introduces a contrast threshold method to filter and select relevant features from the dataset. It integrates the “Select K Best” method with the Chi-Square (Chi2) test to further refine feature selection. This combination aims to enhance the relevance and quality of the features used for classification. The study uses stacking, an ensemble learning technique that combines predictions from multiple models, specifically logistic regression and the Stochastic Gradient Descent (SGD) classifier, to boost classification performance. The paper achieved 99.965% accuracy in binary classification.

The previous work [10] focused on enhancing, identifying, and reducing security risks, particularly Distributed Denial-of-Service (DDoS) attacks, within Internet of Things (IoT) networks by Choosing representative subsets of the data for training through random subset selection, removing irrelevant or redundant features to streamline the model through feature elimination, eliminating duplicate entries to reduce bias and noise through duplication removal, then, a two-level IDS architecture is developed, incorporating both binary and multiclass classifiers to effectively identify and classify DDoS attacks and their sub-classes within IoT networks by using CNN, DNN, LSTM, and RNN. The paper achieved 91.27% accuracy.

3. METHODOLOGY

3.1 The CICIoT2023 dataset

In 2023, the Canadian Institute for Cybersecurity (CIC) introduced the CICIoT2023 dataset, an innovative and compilation collection of IoT attack data [11]. This dataset captures attacks executed by compromised IoT devices targeting other IoT devices [12]. The CICIoT2023 dataset contains 232,885 connections across forty-seven different characteristics and encompasses thirty-three types of sub-attacks [13]. These attacks are categorized into 7 distinct classifications: DoS, DDoS, Web-based, Recon, Spoofing, Mirai, and Brute Force [11]. Dos has 4 sub-attacks, DDoS has 12 sub-attacks, Web-based has 6 sub-attacks, Recon has 5 sub-attacks, Spoofing has 2 sub-attacks, Mirai has 3 sub-attacks, while, Brute Force has no subtypes. The most dominant attack type in the dataset is DDoS with 169,276 instances and the least dominant is Brute Force with only 55 instances.

3.2 Cuckoo search optimization algorithm

Cuckoo Search (CS) is a developed meta-heuristic optimization algorithm that solves optimization issues by utilizing Levy flights' random walks and the brood parasitism of cuckoo species, drawing inspiration from nature [14]. Cuckoo bird species are brood parasites. They appear to lay eggs in the other host birds’ nests based on this. They procreate in this way to increase the likelihood that their eggs will survive. When host birds discover these kinds of eggs, they might relocate and build a new nest, or they'll throw out the alien eggs [15]. Certain species have unique egg-laying schedules; parasitic cuckoos frequently select nest sites where their host bird has recently produced eggs. After hatching, the chick increases its portion of nourishment by pushing out the host eggs. Cuckoo chicks can also imitate calls to increase their feeding opportunities [16]. Stochastic optimization is a technique used in CS; it simulates the compulsory brood parasitism behavior of cuckoo birds [17]. The CS uses the Lévy flight approach which is a strategy used in optimization and search algorithms inspired by the movement patterns of certain animals and insects [18]. It involves making random steps that follow a probability distribution with large tails is called a Lévy distribution. This means that the steps can vary widely in length, allowing the algorithm to investigate a wide region of the search space more effectively.

Based on the previous research [19], the core principles of CS are three highly idealized guidelines:

- Each cuckoo deposits one egg in an arbitrarily chosen nest.
- Retaining the nests with the best eggs for the following generation.
- There is a set quantity of provided nests, and each cuckoo's egg has a probability (between 0 and 1) of being detected by the host bird.

The algorithm of CS is as follows:

Initialization:

Set the number of nests to N and initialize them with random solutions.

Set the number of cuckoos M.

Define the probability Pa of egg discovery.

Egg Laying:

For each cuckoo i in M:

Randomly select a nest j from the N nests.

Place the cuckoo’s egg (new solution) in the selected nest j.

Egg Discovery:

For each nest j in N:

Determine whether the host bird finds the cuckoo’s egg based on Pa.

In case the bird discovers the egg (with probability Pa), replace the cuckoo’s egg with the host’s egg (current solution

in the nest).

Selection of Best Nests:

Evaluate the quality of the eggs in all nests.

To preserve the best nests for the upcoming generation, pick the ones that produce the best eggs.

Repeat:

Repeat steps 2-4 until the convergence requirements are met or for a predetermined number of iterations.

3.3 Particle swarm optimization algorithm

Particle Swarm Optimization (PSO) is a swarm-based stochastic algorithm that exploits concepts related to animal social behavior, such as fish schools and bird flocks [20]. In PSO, potential solutions are modeled as particles moving through the problem space like a flock of birds, every single particle updates its place based on the best locations of other particles as well as its own best prior position in the swarm, with some random variations, after all particles update their positions, the process repeats, over time, the swarm collectively converges towards the optimal solution [21]. Based on previous research [22], the particles adjust their state based on three principles:

- Maintaining their current momentum;
- Adapting according to their own most well-known position;
- Adjusting based on the most well-known site in the whole swarm.

Each particle in the swarm has an impact on its position depending on the role and the specific experience of the most optimist particle in its immediate surroundings [23]. The Particle Swarm Optimization (PSO) algorithm is outlined as follows:

Initialization

- Set initial velocity (Vi) and position (Xi) for each particle i.
- Define the inertia weight (w), cognitive coefficient (c1), and social coefficient (c2).

For Each Particle i in the Swarm

Keep Inertia

- Update the velocity by retaining a fraction of the current velocity:

$$Vi(t+1) = w * Vi(t)$$

Update Based on Personal Best

Adjust the velocity to move towards the particle's personal best position (pbest_i):

$$Vi(t+1) = Vi(t+1) + c1 * r1 * (pbest\_i - Xi(t))$$

where r1 is a random value between 0 and 1.

Update Based on Global Best

Further adjust the velocity to move towards the swarm's best position (gbest):

$$Vi(t+1) = Vi(t+1) + c2 * r2 * (gbest - Xi(t))$$

where r2 is another random value between 0 and 1.

Update Position

Based on the updated velocity, update the particle's location:

$$Xi(t+1) = Xi(t) + Vi(t+1)$$

Repeat

Repeat the previous steps for each particle, continuing until the necessary number of iterations is achieved or the stopping condition is met.

4. THE PROPOSED MODEL

As illustrated in Figure 1, the suggested intrusion detection model consists of four separate phases.

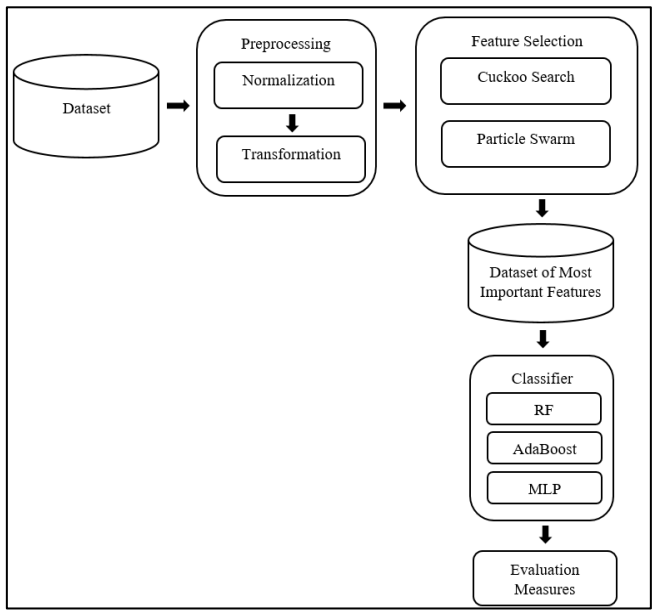

Figure 1. The proposed intrusion detection model

Preprocessing is the first step, which comprises normalizing the data within a given range by using the Min-Max Normalization Equation below and transforming category information into a numerical representation.

$$X_{new} = X_{old} - Min / Max - Min \quad (1)$$

where, X_{new} is the new value after normalization, X_{old} is the current value, Min is the minimum value in the feature, and Max is the maximum value in the feature.

In the second stage, which is dedicated to feature selection, the most crucial features from the CICIoT2023 dataset are determined by applying the PSO and CS optimization algorithms.

The third step involves using the chosen features to train and evaluate the preferred classifier. The review of the classification results from the previous stage is the final step.

When Cuckoo Search (CS) and Particle Swarm Optimization (PSO) algorithms work together for feature selection, the combination often leverages the strengths of both techniques to achieve an optimal subset of features. Each individual represents a potential solution, which is a binary vector where 1 indicates a selected feature and 0 indicates a rejected feature. A hybrid population is created. Some individuals are initialized randomly (as in CS), while others may use PSO's velocity and position-based approach for initialization. CS uses Lévy flights to explore the feature space. A cuckoo's current position is perturbed based on a Lévy distribution to generate a new solution. Each solution's quality is evaluated using a fitness function, and a poor-

performing solution in the population is replaced by a new solution if it is better. After that PSO updates the position (feature subset) of each particle and calculates the direction and size of the particle's movement depending on the best solution the particle has discovered thus far and the best solution the entire swarm has discovered. The particle's position (binary feature subset) is updated based on the new velocity. A sigmoid or thresholding function is often used to convert continuous positions to binary. CS and PSO periodically exchange solutions. For instance, the best solutions from PSO can be used to initialize or update nests in CS, and vice versa.

Employing both the CS and the PSO algorithm for feature selection may provide several benefits:

A feature selection procedure that leverages the advantages of both CS and PSO algorithms together, boosts model performance and strengthens the feature selection process overall.

The methods used by CS and PSO to explore the solution space differ. While PSO employs social behavior and swarm intelligence to converge toward optimal solutions, CS is well-known for its global search capabilities and ability to avoid local minima using a process inspired by brood parasitism.

Combining CS and PSO algorithms can enhance exploration and exploitation capabilities, leading to superior feature subsets. Through the identification of pertinent features and an improvement in the robustness of feature selection, which can improve machine-learning model performance. When two algorithms are combined, their shortcomings can be made up for, producing a more balanced result.

Once the most important features have been chosen, these features are trained and tested with different classifiers. In particular, Random Forest (RF), AdaBoost, and Multilayer Perceptron (MLP) are the classifiers used. After that, the model's accuracy in binary and multi-classification tasks is measured.

5. RESULT AND DISCUSSION

This study's primary goal was to assess the efficacy of the recommended intrusion detection methodology across several classifiers. The results are presented according to the research questions addressed.

The parameter settings in the hybrid implementation of the CSA and PSO involve several key variables that influence the behavior of both the CSA and PSO, as detailed in Table 1.

Table 1. Parameters setting

Parameter	Value
Number of nests	50
Number of particles	30
Discovery probability	0.25
Alpha (Step size scaling factor)	0.01
Beta (Lévy flight scaling factor)	1.5
Cognitive coefficient	1.496
Social coefficient	1.496
Inertia weight	0.5
Maximum iterations	100

Table 2 illustrates the features selected through the CS and PSO algorithms. The top features identified significantly contribute to model performance. The number of features selected by CS and PSO is 32, while the number of ignored features is 14.

Table 2. The selected features

No.	Significant Features	No.	Significant Features
1	flow_duration	17	psh_flag_number
2	Header_Length	18	ack_flag_number
3	Duration	19	ack_count
4	Rate	20	syn_count
5	Srate	21	fin_count
6	Drate	22	urg_count
7	fin_flag_number	23	rst_count
8	syn_flag_number	24	HTTP
9	DNS	25	SSH
10	UDP	26	IPv
11	DHCP	27	LLC
12	Tot-sum	28	Max
13	Min	29	AVG
14	Tot-size	30	Number
15	IAT	31	Radius
16	Variance	32	Weight

The most important features resulting from CS and PSO are trained and tested with RF, AdaBoost, and MLP. The training and testing split ratio are 70:30, where 30% goes toward testing and 70% goes toward training. Table 3 and Table 4 illustrate the performance obtained from training and testing the classifiers.

Table 3. Evaluation measures in binary classification

Classifier	Acc. (%)	Prec. (%)	Rec. (%)	F1-Score (%)
RF	99.66	99.43	99.77	99.61
AdaBoost	99.59	99.53	99.54	99.01
MLP	99.23	98.11	99.23	98.21

Table 4. Evaluation measures in multi-classification

Classifier	Acc. (%)	Prec. (%)	Rec. (%)	F1-Score (%)
RF	99.45	99.40	99.45	99.13
AdaBoost	98.18	97.10	98.20	97.84
MLP	98.73	97.32	98.73	97.52

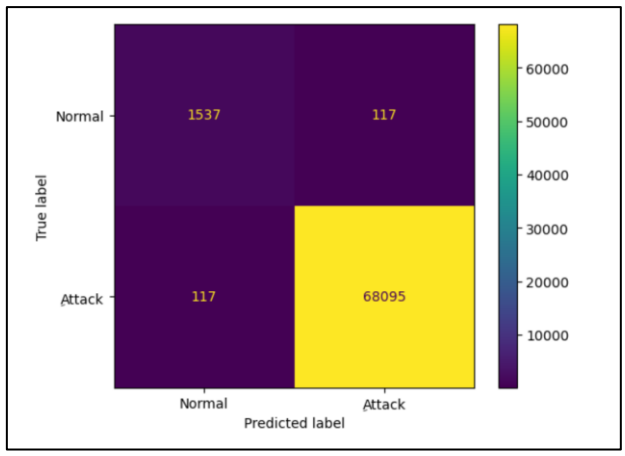

Figure 2. Confusion matrix of binary classification for RF classifier

It is clear from the above two tables that all the classifiers had superior results. Random Forest finds an equilibrium

between robustness, efficiency, and ease of use, making it a strong performer in many scenarios. If computational efficiency and handling of high-dimensional or noisy data are key considerations, Random Forest often emerges as the best choice, especially when compared to AdaBoost and MLP, which may require more tuning and preprocessing to achieve comparable results. Figure 2 and Figure 3 show the confusion matrix for the best classifier in terms of results, which is Random Forest.

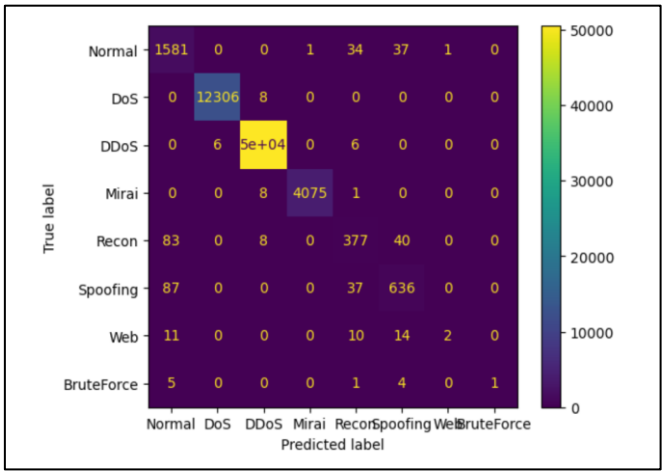

Figure 3. Confusion matrix of multi-classification for RF classifier

Table 5. Binary classification

Classifier	FPR	FNR
Random Forest	0.001	0.070
AdaBoost	0.002	0.085
MLP	0.003	0.158

Table 6. Multi classification

Classifier	FPR	FNR
Random Forest	0.001	0.130
AdaBoost	0.008	0.277
MLP	0.003	0.277

The False Positive Rate (FPR) and False Negative Rate (FNR) are crucial metrics for evaluating a classifier's performance, especially in applications where the cost of errors varies significantly. The FNR is the percentage of positive cases that are wrongly classified as negative, whereas the FPR is the percentage of many negative cases that are mistakenly labeled as positive. Table 5 and Table 6 show the proportions of FPR and FNR for the RF classifier.

6. CONCLUSION

The effectiveness of mixing Particle Swarm Optimization (PSO) and Cuckoo Search (CS) for feature selection in IDS applied to IoT networks has been effectively established in this study. We have created a strong hybrid feature selection framework that improves the performance of IDS models by utilizing the advantages of both optimization methods. The CICIoT2023 dataset showed significant improvements in Random Forest, AdaBoost, and Multi-Layer Perceptron classifier performance, resulting in better classification accuracy and lower computational complexity. Computational complexity, real-time processing capabilities, and hardware requirements are indeed crucial aspects for practical implementation. Future research will aim to explore the scalability of the method in real-time environments and evaluate its efficiency across different hardware setups. Additionally, optimizing the method for lower computational overhead while maintaining high detection accuracy will be a priority. Future research should explore the scalability of this method to diverse attack scenarios and datasets, focusing on its adaptability and robustness across various environments. Additionally, integrating this approach with real-time intrusion detection systems could enhance its practical applicability in dynamic, real-world settings. To further solidify its impact, actionable steps such as optimizing computational efficiency to handle large-scale data and implementing mechanisms for real-time threat analysis and decision-making should be prioritized. These improvements would not only boost efficiency but also increase the method's viability for use in contemporary cybersecurity frameworks.

REFERENCES

[1] Isife, O.F., Okokpujie, K., Okokpujie, I.P., Subair, R.E., Vincent, A.A., Awomoyi, M.E. (2023). Development of a malicious network traffic intrusion detection system using deep learning. International Journal of Safety & Security Engineering, 13(4): 587-595. https://doi.org/10.18280/ijsse.130401

[2] Gheni, H.Q., Al-Yaseen, W.L. (2024). Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 9: 100673. https://doi.org/10.1016/j.prime.2024.100673

[3] Meftah, S., Rachidi, T., Assem, N. (2019). Network based intrusion detection using the UNSW-NB15 dataset. International Journal of Computing and Digital Systems, 8(5): 478-487. https://doi.org/10.12785/ijcds/080505

[4] Praneeth, V., Kumar, K.R., Karyemsetty, N. (2021). Security: intrusion prevention system using deep learning on the internet of vehicles. International Journal of Safety and Security Engineering, 11(3): 231-237. https://doi.org/10.18280/ijsse.110303

[5] Gheni, H.Q., Al-Yaseen, W.L. (2023). Enhanced gaining-sharing knowledge optimization algorithm for 3D compression of intrusion detection dataset. In International Conference on Intelligent Systems Design and Applications, pp. 213-228. https://doi.org/10.1007/978-3-031-64650-8_21

[6] Al-Janabi, S., Alkaim, A. (2022). A novel optimization algorithm (Lion-AYAD) to find optimal DNA protein synthesis. Egyptian Informatics Journal, 23(2): 271-290. https://doi.org/10.1016/j.eij.2022.01.004

[7] Phan, V.A., Jerabek, J., Malina, L. (2024). Comparison of multiple feature selection techniques for machine learning-based detection of IoT attacks. In Proceedings of the 19th International Conference on Availability, Reliability and Security, Vienna, Austria, pp. 157. https://doi.org/10.1145/3664476.3670440

[8] Chen, W., Yang, H., Yin, L., Luo, X. (2024). Large-scale IoT attack detection scheme based on LightGBM and feature selection using an improved salp swarm algorithm. Scientific Reports, 14(1): 19165.

https://doi.org/10.1038/s41598-024-69968-2
[9] Mahdi, Z., Abdalhussien, N., Mahmood, N., Zaki, R. (2024). Detection of real-time distributed denial-of-service (DDoS) attacks on internet of things (IoT) networks using machine learning algorithms. Computers, Materials & Continua, 80(2): 2139-2159. https://doi.org/10.32604/cmc.2024.053542
[10] Hizal, S., Cavusoglu, U., Akgun, D. (2024). A novel deep learning-based intrusion detection system for IoT DDoS security. Internet of Things, 28: 101336. https://doi.org/10.1016/j.iot.2024.101336
[11] Wang, Z., Chen, H., Yang, S., Luo, X., Li, D., Wang, J. (2023). A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization. PeerJ Computer Science, 9: e1569. https://doi.org/10.7717/peerj-cs.1569
[12] Pirtama, A., Prasetia, Y., Saputra, R.I., Winanto, E.A. (2024). Improvement attack detection on internet of thinks using principal component analysis and random forest. Media Journal of General Computer Science, 1(1): 14-19. https://doi.org/10.62205/mjgcs.v1i1.8
[13] Mousa'B, M.S., Hasan, M.K., Sulaiman, R., Islam, S., Khan, A.U.R. (2023). An explainable ensemble deep learning approach for intrusion detection in industrial Internet of Things. IEEE Access, 11: 115047-115061. https://doi.org/10.1109/ACCESS.2023.3323573
[14] Joshi, A.S., Kulkarni, O., Kakandikar, G.M., Nandedkar, V.M. (2017). Cuckoo search optimization-A review. Materials Today: Proceedings, 4(8): 7262-7269. https://doi.org/10.1016/j.matpr.2017.07.055
[15] Huang, J., Gao, L., Li, X. (2015). An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes. Applied Soft Computing, 36: 349-356. https://doi.org/10.1016/j.asoc.2015.07.031
[16] Gandomi, A.H., Yang, X.S., Alavi, A.H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. Engineering with Computers, 29: 17-35. https://doi.org/10.1007/s00366-011-0241-y
[17] Rakhshani, H., Rahati, A. (2017). Snap-drift cuckoo search: A novel cuckoo search optimization algorithm. Applied Soft Computing, 52: 771-794. https://doi.org/10.1016/j.asoc.2016.09.048
[18] Huang, L., Ding, S., Yu, S., Wang, J., Lu, K. (2016). Chaos-enhanced Cuckoo search optimization algorithms for global optimization. Applied Mathematical Modelling, 40(5-6): 3860-3875. https://doi.org/10.1016/j.apm.2015.10.052
[19] Mohamad, A.B., Zain, A.M., Nazira Bazin, N.E. (2014). Cuckoo search algorithm for optimization problems—A literature review and its applications. Applied Artificial Intelligence, 28(5): 419-448. https://doi.org/10.1080/08839514.2014.904599
[20] Wang, D., Tan, D., Liu, L. (2018). Particle swarm optimization algorithm: An overview. Soft Computing, 22(2): 387-408. https://doi.org/10.1007/s00500-016-2474-6
[21] Gad, A.G. (2022). Particle swarm optimization algorithm and its applications: A systematic review. Archives of Computational Methods in Engineering, 29(5): 2531-2561. https://doi.org/10.1007/s11831-021-09694-4
[22] Bai, Q. (2010). Analysis of particle swarm optimization algorithm. Computer and Information Science, 3(1): 180-184.
[23] Wang, F., Zhang, H., Zhou, A. (2021). A particle swarm optimization algorithm for mixed-variable optimization problems. Swarm and Evolutionary Computation, 60: 100808. https://doi.org/10.1016/j.swevo.2020.100808