Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

A Swarm- Optimized Hybrid Approach to Feature Selection in IDS

Maan Y Anad Alsaleem*

Directorate of Education in Nineveh , Al-Shurta , Mosul, Iraq .Email: maanyounis1983@gmail.com

ARTICLE INFO

Article history:
Received: 21/10/2025
Rrevised form: 25 /11/2025
Accepted : 01 /12/2025
Available online: 30 /12/2025

Keywords:

Artificial Bee Colony
Genetic Algorithm
Ensemble Learning
Anomaly Detection

ABSTRACT

The network intrusion detection (IDS) strategies are important to securing our systems and networks from unauthorized behaviors. An IDS routinely deals with large amounts of data transmission that contain non-informative and duplicate features, which implication the performance of the machine learnind model negatively. In this paper, we proposed on hybrid two optimization techniques Artificial Bee Colony (ABC) and Genetic Algorithm (GA) to feature selected data. we used Random Forest (RF) and XGBoost classifiers to evaluate how well they perform with the reduced features. Experiments are conducted on three datasets: NSL-KDD, UNSW-NB15, and CIC-IDS2017. Experimental results show that the ABC-GA algorithm reduces the number of features (up to 70-88%) while maintaining detection accuracy, with accuracy reaching 97% in NSL-KDD, 92% in UNSW-NB15, and 94% in CIC-IDS2017.

https://doi.org/ 10.29304/jqcsm.2025.17.42565

1. Introduction

Nowadays, computers and networks have become ubiquitous in people's lives [1, 2]. However, with the increasing number of cyber-attacks launched by hackers, intruders, and cyber organizations, network security has become increasingly critical [3, 4]. To ensure the safe operation of networks, network intrusion detection has become a major research focus in recent years [5, 6]. Due to the large number of features in network intrusion detection datasets, the time efficiency and accuracy of intrusion detection are reduced [7]. Therefore, feature selection has become an important means of improving the performance of network intrusion detection [8, 9]. Researchers have conducted extensive research on this topic, such as traditional feature selection methods (associative feature selection, information acquisition, etc.) and swarm intelligence optimization algorithms such as swarm algorithms, ant colonies, and gray wolf algorithms, to solve the feature selection problem [10–12]. [7] proposed.

The Artificial Bee Colony (ABC) algorithm is an intelligent optimization algorithm proposed by Karaboga et al. [13] in 2005. In recent years, the ABC algorithm has been widely used in various fields [14, 15]. Because the ABC algorithm has good optimization performance, it is more suitable for feature selection problems [16]. However, when dealing with problems with a large number of data dimensions, the ABC algorithm is prone to local optimization, and the searched solution is often not optimal [17]. The genetic algorithm (GA) is a random optimization algorithm that starts with a random solution and then evolves it according to basic processes derived from nature, such as mating and mutation, until it reaches a solution close to the optimal one [18, 19]. To improve the feature selection algorithm, the artificial bee colony (ABC) algorithm was combined with the genetic algorithm.

Maan Y Anad Alsaleem*

Email: maanyounis1983@gmail.com

Communicated by ‘sub etitor’

Both algorithms have their own method for searching for solutions. ABC is better at exploring, while the genetic algorithm is stronger at exploring, as it uses crossing and mutation. Therefore, in this paper, ABC was combined with GA to create the ABC-GA algorithm for feature selection

2. Related Works

Many papers have addressed the problem of attack classification in intrusion detection systems. Manual feature selection methods have been used in some studies, which involves selecting relevant features using the researcher's experience and domain knowledge. This allows researchers to focus on the features that are most important for intrusion detection [20, 21]. Other researchers have used hybrid feature selection methods such as the information measure of features (IMF) and the uncertainty measure of features (UMF), which help improve performance and provide a broader view of the importance of features [22]. Another method is feature selection based on the correlation coefficient, which measures the linear correlation between features to determine their dependence on each other, allowing the algorithm to focus on the features most closely related to the target variable (intrusion or normal behavior) [23,24]. In [25] reported using a combination of a support vector machine and a genetic algorithm to reduce the number of features in the KDD CUP-99 dataset from 41 to 10. Experiments showed that this method improved true detection rates while significantly reducing false alarm ratesIn another study, [27] proposed using the Pigeon Inspired Optimizer (PIO) algorithm to reduce the number of features on the KDD CUP-99, NSL-KDD, and UNSW-NB15 datasets. Reducing the number of features from 41 to 7 reduced training time, maintained accuracy, and reduced model building time. In a study on the CUP-99 and UNSW-NB15 datasets, logistic regression was used for feature selection. The study found that 18 influential features for the CUP-99 dataset and 20 features for the UNSW-NB15 dataset were effective in model training [28]. While these methods reduce dimensionality and improve or maintain performance, their effectiveness depends on the selected features and they do not always guarantee optimal results, especially in the case of imbalanced data. Many traditional strategies rely on heuristic rules that limit their ability to represent complex relationships between features. And a variety of metaheuristic algorithms are now being utilized to select features in IDS systems, most of the research has focused on a single (e.g. ABC, GA, PSO, ACO, GWO) or hybrid algorithms whose functionalities in the search space are somewhat similar to each other. To date, however, there has been little into the use of hybrid models that take advantage of a between exploration and exploitation.

1. Artificial Bee Algorithm Enhanced by Genetic Factors (ABC-GA)

3.1 Artificial Bee Algorithm (ABC)

It is an optimal algorithm based on a model of the intelligence of a swarm of bees' foraging behavior. When a bee finds food during its search, it returns to the hive with a sample to inform the rest of the worker bees of the food's location and direction. The bee performs a waggle dance in a specific direction and a specific number of times to indicate the food's location. It consists of three types of bees: worker bees, forager bees, and scout bees. Worker bees explore solutions surrounding their current food location, while forager bees choose the best solution, taking into account food quality. Finally, scout bees search randomly when a solution fails to improve. The number of worker bees equals the number of solutions. Each solution is encoded as a binary feature vector:

$$X = \{x1, x2, \ldots, xn\}, xi \in \{0,1\}$$

Where $x_i = 1$ means the feature is selected, $x_i = 0$ means it is not selected n is food sources and d is the total number of original features.

During the employed-bee phase, each bee i generates a neighboring solution v_{ij} for feature j using:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

Bee i updates its solution using a random bee k ($k \neq i$) and a random factor $\phi_{\text{ij}} \in [-1,1]$; if the new solution is fitter, it replaces the old one.

3.2 Genetic Algorithm (GA)

To avoid local optima, the GA applies crossover and mutation. The Crossover operation combines two solutions X^a and X^b to generate a new one:

$$X_i^{new} = \begin{cases} X_i^a & if\ \ rand\ < p_c \\ X_i^b & otherwise \end{cases}$$

Where p_c is the crossover probability.

To create diversity, mutation is used to flip feature values by small percentages:

$$X_i^{mut} = \begin{cases} 1 - X_i & if\ \ rand\ < p_m \\ X_i & otherwise \end{cases}$$

Where p_m is mutation probability

3.3 The Proposed ABC-GA Algorithm

The proposed algorithm, ABC-GA, takes steps from the genetic algorithm and integrates them into an artificial bee colony to optimize the selection of the required feature vector. Initially, randomly generated a set of binary solutions, for each solution a subset of features vector is representing. Figure 1 illustrates proposed method (ABC-GA).

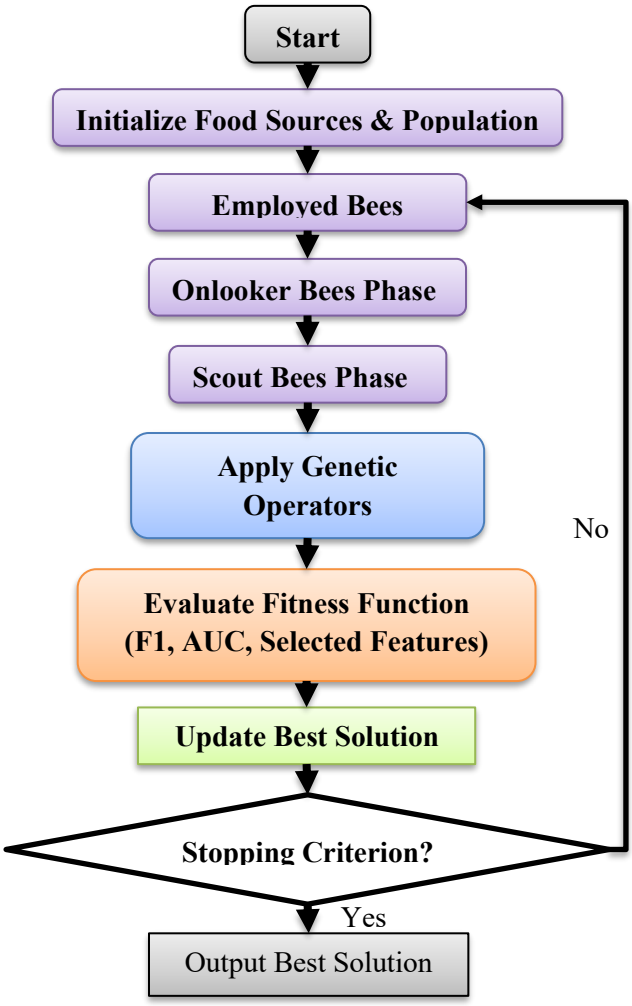

Figure 1- Proposed ABC-GA Based Feature Selection

In the used bee stage, each bee generates a neighbor solution using the neighborhood equation; if it is better fit, it is replaced with the old solution. In the observation bee stage, solutions are selected probability of validity, as defined as follows:

$$pi = \frac{fitness_i}{\sum_{i=1}^{N} fitness_{\ j}}$$

High fitness solutions are likely to be improved. In the scouting phase, stagnant solutions are replaced with random solutions to maintain diversity. Hence, crossover and mutation improve exploration and prevent premature convergence. The fitness function balances accuracy with feature reduction, which is defined as follows:

$$\text{Fitness} = \alpha(1 - \text{F1}) + \beta \frac{\text{SelectedFeatures}}{\text{TotalFeatures}} + \gamma(1 - \text{AUC})$$

where:

- $F1$ represents the F1-score,
- AUC denotes the Area Under the ROC Curve,
- $SelectedFeatures$ is the number of chosen features, and $TotalFeatures$ is the total number of available features,

- $\alpha > \beta > \gamma$ are weighting coefficients that prioritize detection performance over feature minimization.

The weights were adjusted through preliminary experiments, and the values that gave the best performance were chosen.

The iterative process continues until a termination condition is satisfied, either reaching the maximum number of iterations or achieving convergence stability. Finally, the algorithm outputs the optimal feature subset, which ensures the highest detection accuracy with the least number of features.

4 Experiments and Results

4.1 Dataset

Three publicly available datasets were used: UNSW-NB15, CIC-IDS2017, and NSL-KDD [28][29][30].

The UNSW-NB15 dataset was constructed with the IXIA PerfectStorm platform, simulating a combination of legitimate and attack traffic in 2 days, resulting in approximately 100 GB of recorded network traffic. There were nine categories of attack included in the dataset. Argus and Bro-IDS generated features for each traffic flow, categorizing as temporal, content and statistical features from the network traffic, yielding a total of 47 features.

In contrast, the CIC-IDS2017 dataset was developed to create a more realistic simulation of user behavior. The CICFlowMeter was employed to capture various features and metadata, including but not limited to: IP addresses, ports, timestamps, and application protocols (HTTP, HTTPS, FTP, and SSH). Network traffic was generated from about 25 virtual users through the B-Profile framework and attempted to mimic natural human activity.

The dataset NSL-KDD, a cleaned and balanced dataset based on the original KDD Cup '99 dataset, removing duplicate records. It consists of four subsets (KDDTrain+, KDDTest+, KDDTest-21, KDDTrain+_20%) that focus on the main types of penetration based on the original dataset: DoS, Probe, R2L, and U2R.

Table 1- Summary of IDS Datasets Used

Dataset	Number of Samples	Number of Features	Attack Categories
NSL-KDD	125,973	41 (122 after encoding)	DoS, Probe, R2L, U2R
UNSW-NB15	2,540,044	47 (196 after encoding)	attack types 9
CIC-IDS2017	2,830,743	80	attack types 14

4.2 Parameter Settings

The artificial bee colony (ABC) and genetic algorithm (GA) were implemented using commonly used settings for population size, number of iterations, and search operators, as summarized in Table 2. For the evaluation stage, the default parameter settings implemented in the scikit-learn and XGBoost libraries were used. This allows the evaluation to focus on the effectiveness of the proposed feature-selection mechanism rather than the classifier's specific settings.

Table 2- Parameter Settings for ABC and GA

ABC		GA	
Parameter	**Value**	**Parameter**	**Value**
Number of Food Sources	20	Population Size	30
Employed Bees	20	Crossover Probability	0.9
Onlooker Bees	20	Mutation Probability	0.02
Limit	15	Generations	25
Maximum Iterations	40		

4.3 Results

The evaluation relied on three datasets, divided into 70% training and 30% testing. Performance was examined in three progressive setups. baseline using all available features, optimized input through the ABC–GA feature selection scheme, and the same selection integrated with ensemble learning for comparative analysis.

Table 3 - Results without feature selection (All Features)

Dataset	Classifier	Accuracy	Precision	Recall	F1	AUC	MCC
NSL-KDD	RF	0.93	0.92	0.87	0.89	0.95	0.82
	XGBoost	0.94	0.93	0.89	0.91	0.96	0.85
UNSW-NB15	RF	0.86	0.84	0.80	0.82	0.90	0.71
	XGBoost	0.88	0.86	0.82	0.84	0.92	0.78
CIC-IDS2017	RF	0.87	0.85	0.79	0.82	0.91	0.74
	XGBoost	0.89	0.87	0.82	0.84	0.93	0.77

Table 3 shows that the models achieved good performance when using all features, but some metrics remained substandard. For example, on the NSL-KDD set, the XGBoost model achieved an accuracy of 94% with F1 = 0.91, while the Random Forest model achieved an accuracy of only 93%. In contrast, the UNSW-NB15 set showed a relative decline, with accuracy not exceeding 88% with XGBoost. On CIC-IDS2017, performance ranged between 87–89%.

Figure 2 shows the effect of the ABC-GA algorithm in reducing the number of features for all datasets. In the NSL-KDD set, the features were reduced from 122 to only 30, i.e., approximately 75% of the features were deleted. In the UNSW-NB15 set, the reduction was greater, as the number decreased from 196 to only 23 features, a reduction of more than 88%. In CIC-IDS2017, the number of features decreased from 80 to 38, i.e., approximately half.

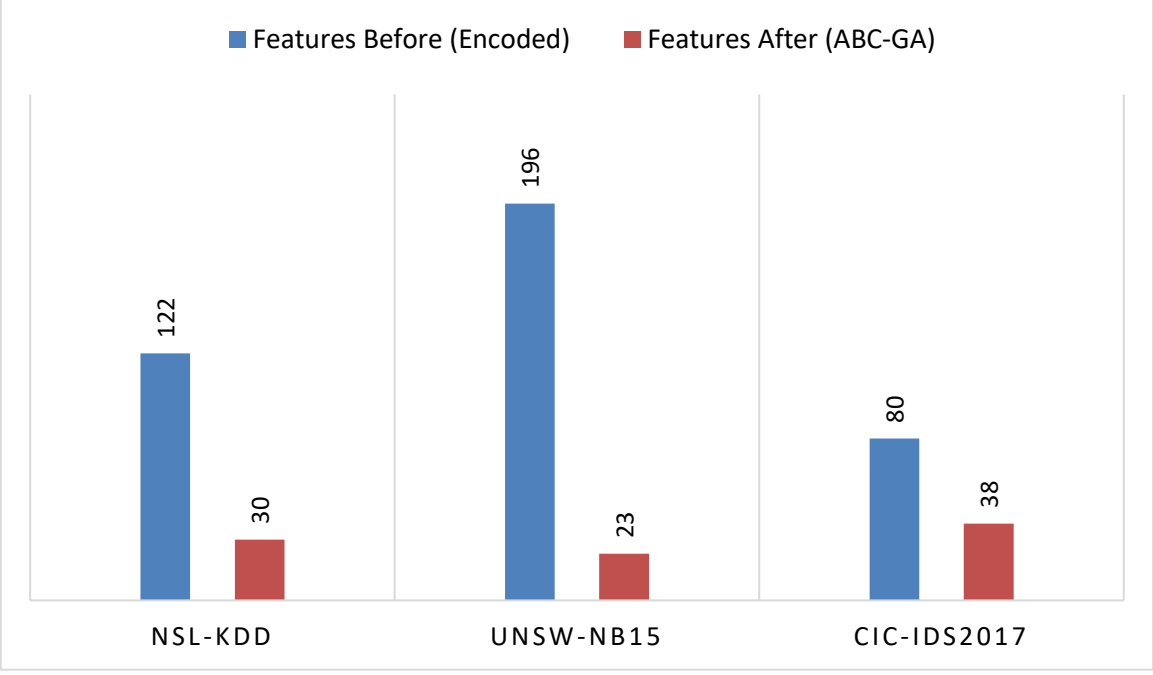

Figure 2- feature Reduction by ABC-GA

After applying the ABC-GA algorithm for feature selection, the performance improved significantly, as shown in Table 4. The accuracy of XGBoost on the NSL-KDD set rose to 96% with F1 = 0.93, while the accuracy on the CIC-IDS2017 set reached approximately 92% with AUC = 0.96.

Table 4 - Results after feature selection using ABC-GA

Dataset	Classifier	Accuracy	Precision	Recall	F1	AUC	MCC
NSL-KDD	RF	0.95	0.94	0.90	0.92	0.97	0.87
	XGBoost	0.96	0.95	0.92	0.93	0.98	0.89
UNSW-NB15	RF	0.89	0.87	0.84	0.85	0.93	0.77
	XGBoost	0.91	0.89	0.86	0.87	0.94	0.82
CIC-IDS2017	RF	0.91	0.89	0.85	0.87	0.95	0.81
	XGBoost	0.92	0.90	0.87	0.88	0.96	0.83

When feature selection was combined with ensemble learning, the model achieved the best performance, as shown in Table 5. The classification accuracy for NSL-KDD reached 97% with F1 = 0.95, while the performance for UNSW-NB15 rose to 92% with MCC = 0.85. The CIC-IDS2017 set achieved an accuracy of 94% and an AUC = 0.96.

Table 5 - Results with hybrid approach (ABC-GA + Ensemble Learning)

Dataset	Accuracy	Precision	Recall	F1	AUC	MCC
NSL-KDD	0.97	0.96	0.94	0.95	0.98	0.93
UNSW-NB15	0.92	0.90	0.88	0.89	0.95	0.85
CIC-IDS2017	0.94	0.92	0.89	0.90	0.96	0.87

5 Discussions

Three experimental setups were compared: models trained using all available features, models using features selected by the ABC–GA algorithm, and a hybrid approach combining ABC–GA selection with ensemble learning.

When all features were used, the models achieved reasonable accuracy but still showed signs of overfitting and noise. For instance, on NSL-KDD, XGBoost reached 94% accuracy (F1 = 0.91), while Random Forest scored 93%. In UNSW-NB15, accuracy peaked at 88%, and CIC-IDS2017 achieved between 87–89%.

When ABC-GA was used, the number of features was reduced severely: NSL-KDD: 122 features to 30 (~75% reduction). UNSW-NB15: 196 features to 23 (~88% reduction). CIC-IDS2017: 80 features to 38 (~52% reduction).

In representing the performance, the model's accuracy improved after feature reduction for all datasets. The performance of the xgboost model on NSL-KDD is 96% accuracy (F1 = 0.93). CIC-IDS2017 outcome is improved to 92% accuracy and 0.96 AUC.

After processing the features again through ensemble learning, the hybrid combination provided the highest accuracy overall compared to using classifiers individually:

For NSL-KDD 97% accuracy improvement, F1 = 0.95. UNSW-NB15: 92% accuracy, MCC = 0.85. CIC-IDS2017: 94% accuracy, AUC = 0.96.

The improvement was not limited to accuracy; reducing features also decreased overall data redundancy, decreased computing time, and increased detection of rare attacks (R2L, U2R). In several scenarios, we were able to reduce the features by 70% without a notable loss in recall/precision.

6 Conclusions

This paper proposes a new Intrusion Detection System based on a hybrid feature selection. The method is structured in two stages in the first, the Artificial Bee Colony (ABC) algorithm is applied, and in the second stage, a Genetic Algorithm (GA) is integrated as a wrapper. Three common Intrusion Detection System datasets (NSL-KDD, UNSW-NB15, and CIC-IDS2017) are used with the Random Forest (RF) and XGBoost (XGB) classifiers to evaluate the proposed hybrid feature selection method in terms of accuracy, F1-score, AUC, MCC, and the number of selected features. From the results, it is noted that the ABC-GA has achieved superior performance in terms of accuracy, AUC, and F1-score metrics for all three datasets compared with several recent state-of-the-art methods. In addition, the method significantly reduces the number of selected features in all datasets. The ABC-GA hybrid outperforms the other compared methods. It reduces the number of features to 30 out of 122, 23 out of 196, and 38 out of 80 in NSL-KDD, UNSW-NB15, and CIC-IDS2017 datasets, respectively, while maintaining high detection accuracy reaching 97%, 92%, and 94%.

References

[1] Z. Halim and S. Shafique, “An effective genetic algorithm-based feature selection method for intrusion detection systems,” Computers & Security, 2021.
[2] P. Nimbalkar and S. Nimbalkar, “Feature selection for intrusion detection system in Internet-of-Things,” ICT Express (Elsevier), 2021. Available: ScienceDirect
[3] O. Almomani, “A Feature Selection Model for Network Intrusion Detection System,” Symmetry, 2020.
[4] M.-T. Nguyen, H. Fang, et al., “Genetic convolutional neural network for intrusion detection,” Future Generation Computer Systems (Elsevier), 2020.
[5] Y. Yin, J. Jang-Jaccard, et al., “IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15,” Journal of Big Data, 2023 (open PDF).
[6] A. H. Farooqi, S. A. Chaudhry, et al., “Enhancing Network Intrusion Detection Using an Ensemble Voting Classifier (DRX),” Sensors, 2023.
[7] Y. Alotaibi and M. Mehmood, “Ensemble-Learning Framework for Intrusion Detection to Improve Efficiency,” Sensors, 2023.
[8] H. Yu, F. Wang, et al., “A feature selection algorithm for intrusion detection system based on enhanced heuristic optimization,” Expert Systems with Applications, 2024.
[9] S. A. Ajagbe, A. Adeniyi, et al., “A Comparison Study of ML Models Using Feature Selection on UNSW-NB15,” SN Computer Science, 2024.
[10] G. Balhareth, A. Alshahrani, et al., “Optimized Intrusion Detection for IoMT Networks with Tree-based Classifiers and Filter FS,” Sensors, 2024.
[11] J. Bo, Y. Tong, et al., “Boosting Few-Shot Network Intrusion Detection with Adaptive Feature Fusion,” Electronics, 2024.
[12] E. M. Maseno, Y. He, et al., “Hybrid wrapper feature selection method based on genetic algorithm for IDS,” Journal of Big Data, 2024 (open PDF).
[13] M. Wang, X. Jiang, et al., “Learn-IDS: Bridging Gaps between Datasets and Models for Network IDS,” Electronics, 2024.
[14] G. Nassreddine and A. Baz, “Ensemble Learning for Network Intrusion Detection Based on Correlation and XGBoost-Embedded FS,” Computers, 2025.
[15] E. Emirmahmutoğlu and A. Genç, “A feature selection-driven ML framework for anomaly-based attack detection,” Peer-to-Peer Networking and Applications, 2025.
[16] X.-Y. Gong, Z.-H. Zhou, et al., “Feature selection method for network intrusion based on hybrid metaheuristic (HMDOA),” Computers & Security, 2025.
[17] Y. Gao, J. Li, et al., “IR-IDS: A network intrusion detection method based on causal feature selection and robust anomaly classification,” Computers & Security, 2025.
[18] J. Maldonado, R. Sosa, et al., “An evolutionary wrapper to support intrusion detection with adaptive model selection,” Computers & Security, 2025.
[19] M. Alharthi, F. Medjek, and D. Djenouri, “Ensemble Learning Approaches for Multi-Class IDS for the IoV: A Comprehensive Survey,” Future Internet, 2025.
[20] N. Islam, F. Farhin, I. Sultana, M. S. Kaiser, M. S. Rahman, M. Mahmud, and G. H. Cho, “Towards machine learning based intrusion detection in IoT networks,” Computers, Materials & Continua, vol. 69, no. 2, pp. 1983–1997, 2021.
[21] A. M. Banaamah and I. Ahmad, “Intrusion detection in IoT using deep learning,” Sensors, vol. 22, no. 21, p. 8417, 2022, doi: 10.3390/s22218417.
[22] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, “Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions,” Electronics, vol. 11, no. 11, p. 3330, 2022, doi: 10.3390/electronics11203330.
[23] S. A. Arhore, Intrusion Detection in IoT Systems Using Machine Learning, Doctoral dissertation, National College of Ireland, Dublin, 2022.
[24] M. Baich, T. Hamim, N. Sael, and Y. Chemlal, “Machine learning for IoT based networks intrusion detection: A comparative study,” Procedia Computer Science, vol. 215, pp. 742–751, 2022.
[25] B. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. J. Golkar, and A. Ebrahimi, “A hybrid method consisting of GA and SVM for intrusion detection system,” Neural Computing and Applications, vol. 27, no. 6, pp. 1669–1676, 2016.
[26] H. Alazzam, A. Sharieh, and K. E. Sabri, “A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer,” Expert Systems with Applications, vol. 148, Art. no. 113249, 2020.
[27] C. Khammassi and S. Krichen, “A GA-LR wrapper approach for feature selection in network intrusion detection,” Computers & Security, vol. 70, pp. 255–277, 2017.
[28] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, ON, Canada, 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.
[29] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in Proc. 2015 Military Communications and Information Systems Conf. (MilCIS), Canberra, Australia, 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.
[30] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), Funchal, Portugal, 2018, pp. 108–116. doi: 10.5220/0006639801080116.