Journal of Innovations in Computer Science and Engineering
Shahid Beheshti Unversity

January 2025, Special Issue on AI 4 All- 2

A Hybrid Approach for Intrusion Detection in Computer Systems Using Optimized Deep Neural Networks

Yusuf Nahi Salman, Code ORCID: 0000-0002-8439-8591
dept. of Computer Engineering, Tehran South Branch, Islamic Azad University, Tehran, Iran
Maral Kolahkaj✉, Code ORCID: 0000-0002-5235-0227
dept. of Computer Engineering, Sousangerd Branch, Islamic Azad University, Sousangerd, Iran, Maral.kolahkaj@iau.ac.ir

Abstract— The issue of intrusion in security presents a fundamental challenge that can lead to serious damage in IT systems. Intrusion Detection Systems (IDS) serve as effective tools for identifying intrusion activities and generating alerts. However, traditional IDS methods often face issues such as low accuracy and long training times. Therefore, enhancing the performance and efficiency of these systems is crucial. The proposed approach in this study leverages evolutionary optimization algorithms combined with machine learning approaches to improve accuracy and training speed in IDS and better manage large volumes of data. This combination leads to the development of an Evolutionary Neural Network (ENN) that enhances and optimizes IDS performance. In this approach, BUZOA and Ant Colony Optimization (ACO) algorithms are used for feature selection, and decision tree, k-nearest neighbor, support vector machine, and deep neural network algorithms are used for classification and intrusion detection. The dataset used in this research is from the CICDDOS2019 database, containing 54,000 samples and 22 initial features. The experimental results indicate that among the metaheuristic algorithms BUZOA and ACO, and their combinations with decision tree, k-nearest neighbor, and support vector machine, the BUZOA-CNN hybrid algorithm with an average RMSE of 0.0117 and an accuracy of 96.32% performs better than other algorithms.

Keywords— Intrusion Detection, Cyber Attacks, BUZOA Optimization Algorithm, Ant Colony Optimization, Deep Learning.

I. Introduction

Intrusion poses a critical challenge in cybersecurity, threatening the integrity of IT systems [1-5]. An intrusion incident can steal or destroy data in seconds, damage hardware, cause significant financial losses, compromise critical IT infrastructure, and potentially lead to failure in cyber warfare [6-9]. To combat these threats, Intrusion Detection Systems (IDS) are essential tools for monitoring and identifying abnormal and malicious activities in network traffic, generating alerts, and reporting suspicious behavior [10-14].

Traditional IDS methods often face problems such as low accuracy and long training times [13, 14], necessitating the development of more robust and intelligent systems [15-18]. Machine learning techniques, especially Artificial Neural Networks (ANNs), are promising in addressing these challenges as they can learn from data to understand system behaviors and make general decisions based on new information [21-32]. However, ANNs encounter significant optimization issues, such as falling into local minima and slow convergence rates during training, which require more powerful optimization techniques [34-39].

Therefore, the main objective of this research is to achieve more accurate results in the process of intrusion detection and evaluation by achieving the following goals:

- Gathering various intrusion detection and evaluation indicators using supervised learning methods.
- Increasing the speed of attack detection by utilizing feature reduction techniques.
- Reducing false negative and false positive rates in attack detection.
- Enhancing the ability to detect new and unknown intrusions using an improved neural network combined with feature selection methods.

In this study, an evolutionary algorithm called BUZOA [15], combined with a deep neural network called BUZOA-CNN, is proposed to develop advanced intrusion detection methods for IDS. This hybrid system, known as the Evolutionary Neural Network (ENN), aims to solve the optimization problems of ANNs, such as local minima and slow convergence rates [40-44]. The BUZOA-CNN approach is designed to efficiently identify new and emerging attack patterns, making it a crucial tool for organizations from small offices to multinational corporations. The proposed system offers the following advantages in terms of application:

Submit Date: 2024-12-01
Accept Date: 2025-05-05
✉ Corresponding author

- Improved efficiency in intrusion detection: Significantly enhances detection efficiency compared to manual systems.
- Ability to manage large volumes of data: Effectively processes extensive data.
- Near real-time alerts: Generates timely alerts to minimize damage.
- Automatic responses: Enables automated responses such as disconnecting users or deactivating accounts.
- Increased deterrence: Enhances deterrence against potential intrusions.
- Comprehensive reporting: Provides detailed reports of detected intrusions.

The main contributions of this research include:

- Proposing a new solution to improve the efficiency of intrusion detection systems: Using an evolutionary optimization algorithm called BUZOA to enhance the training process of learning algorithms and proposing a new Evolutionary Neural Network (ENN) by combining BUZOA and CNN to improve the detection capability of new and unknown attacks.
- Improving the accuracy and speed of intrusion detection: Using feature selection techniques to reduce data dimensions and eliminate ineffective features, leading to reduced computational load and increased classification accuracy.
- Reducing false negative and false positive rates in attack detection by utilizing optimization techniques.
- Evaluating and comparing the performance of the proposed system: Conducting comprehensive experiments to evaluate the performance of the BUZOA-CNN system compared to traditional methods and other optimization algorithms, and analyzing the results obtained from the experiments to examine the advantages and disadvantages of the proposed system.

The structure of the paper is organized as follows: In Section II, we delve into the background of the study. Section III outlines the proposed method. Section IV presents the implementation and evaluation of the proposed approach, and finally, Section V concludes the study.

II. Related work

In [15] the Buzzard Optimization Algorithm, inspired by the natural behavior of buzzards was presented. The algorithm effectively balances global and local search techniques, offering high accuracy and suitable computational time in finding optimal solutions. The algorithm requires precise parameter tuning, and improper settings can reduce its performance. Additionally, its efficiency in complex, high-dimensional problems need further investigation.

[19] offers an efficient mechanism for DDoS detection in large-scale IoT networks using ensemble learning. By combining multiple machine learning models, it enhances detection accuracy and reliability. The approach necessitates processing and analyzing large data volumes, which may require significant computational resources. Additionally, coordinating and integrating various machine learning models can be complex and time-consuming.

[20] enhanced DDoS attack detection in Software-Defined Networks (SDN) using a combination of K-means clustering and hybrid ensemble techniques. This approach provides high accuracy and fast detection of DDoS attacks. The method has high computational complexity, requiring significant processing resources. Moreover, the effectiveness heavily depends on the precise tuning of K-means and ensemble algorithm parameters.

[23] discussed adaptive DDoS attack detection using hybrid machine learning methods in software-defined IoT networks. The approach offers high detection accuracy by automatically adapting to network traffic changes. Challenges include the complexity of implementation and the need for precise parameter tuning of hybrid models. Continuous model updates to maintain detection accuracy may also consume additional resources.

[33] presented a hybrid deep learning method for detecting DDoS attacks in IoT environments. Utilizing deep neural networks, it achieves high detection accuracy and can learn from large datasets. The method requires substantial computational resources for training deep models, leading to long training times. The complexity of deep learning architectures can also make implementation and maintenance challenging.

[43] focused on real-time DDoS attack detection using hybrid machine learning approaches in IoT environments. The method allows for quick and efficient real-time analysis of network data, enhancing detection speed. Real-time data processing requirements may necessitate advanced and expensive hardware. Coordinating different machine learning models in real-time can also pose significant challenges.

[45] explored optimized deep learning techniques for detecting DDoS attacks in cloud-based IoT systems. By optimizing deep learning architectures, the approach achieves high detection accuracy and efficiency. The method demands high computational and storage resources, leading to potentially significant operational costs. Furthermore, precise parameter tuning of deep learning models requires specialized knowledge and expertise.

III. proposed approach

DDoS attacks exhibit complex and dynamic patterns, making their detection challenging. The primary objective of this study is to detect DDoS attacks in IoT networks. The selection of methodologies is based on the inherent characteristics of DDoS attacks, the challenges associated with their detection, and the need for high accuracy and efficiency in network environments.

This study designs and implements an CNN-based IDS trained by BUZOA. ANNs can solve multiple issues faced by existing intrusion detection approaches. They identify typical user behavior and detect statistically significant deviations from it.

Moreover, ANNs have open, flexible, and extensible structures, enabling them to create a general knowledge model of the environment's behavior. The proposed approach primarily involves estimating neuron parameters to enable the ANN to define the relationship between input patterns and the target output through a training process. Training a neural network is a complex task and requires ANN training, which can be defined as an optimization problem. Solving the training procedure requires obtaining responses from a linear constraint with a nonlinear optimization problem. Therefore, various evolutionary algorithms (EAs) are employed to solve this problem.

Essentially, EAs are stochastic population-based search algorithms aimed at finding an acceptable or nearly optimal solution in multimodal search spaces. EAs exhibit global convergence and strong robustness. They can overcome the local minima problem in multimodal search spaces before reaching the global optimum with fast convergence. The proposed IDS can be divided into three main modules: data input, CNN network, and BUZOA optimization algorithm, which can be described as follows:

1. **Data Input Module**: In the initial stage of the proposed framework, the dataset includes predefined training and testing sets, which are used as inputs for the next CNN model. Before feeding the data into the CNN module, it must be mapped into the range between 0 and 1 (0, 1) to make it usable for the subsequent module.
2. **CNN Module**: In the second stage, the CNN module receives the training features from the data input module. The CNN module is designed as an MLP, which is a feedforward neural network comprising an input layer, hidden layers, and an output layer. The inputs from the data input module (training dataset) are provided as input training patterns to the CNN model for training. This training process is conducted by sending the weights to the BUZOA module. The BUZOA module acts as an independent system to update synaptic weights after each iteration. In each iteration of the training process, the BUZOA module sends its individuals as a set of weights to an ANN module. Subsequently, it evaluates the individuals based on a training dataset and then returns their fitness values. The training process stops upon reaching the maximum number of iterations. After that, the knowledge base (weights and biases) is updated.
3. **Testing Phase**: In the third stage, when the CNN is trained with the training dataset, the test inputs from the test dataset are fed into the trained ANN to predict the output. The CNN testing process can be considered as checking the predicted output against the closest match to each of the target classes.

A. Data Preprocessing

One of the most critical stages of a data mining method is data preprocessing. Preprocessing determines the results of the proposed approach and is crucial enough to lead to either the best or the weakest results. Therefore, the first stage after data collection in this study will be data preprocessing.

B. Feature Selection

The large volume of data and numerous features in DDoS detection may lead to reduced model efficiency and increased computational costs. Feature selection can improve model efficiency by eliminating irrelevant or redundant features.

This study compares two optimization techniques for feature selection, namely BUZOA-based feature selection and Ant Colony Optimization (ACO)-based feature selection, and examines their impact on model performance. This process is especially important when designing ML models for large-scale systems that produce high-dimensional data.

- **BUZOA-based Feature Selection**: BUZOA is an evolutionary algorithm known for its ability to identify the most significant features with high accuracy and minimal feature redundancy. Due to its nature of optimization search, this algorithm can be effectively used to select important features from the feature set. In this approach, each scavenger represents a subset of features, and the optimization search process seeks to find the best combination of features to improve model performance.
- **ACO-based Feature Selection**: ACO is a nature-inspired optimization method that effectively selects the most relevant features for classification. In this method, ants explore various paths to select features and choose the best paths to improve accuracy.

Combining these optimization algorithms ensures that only the most relevant features are retained, improving both accuracy and computational efficiency.

C. Deep Neural Network Module

The CNN consists of parallel processing elements that are fully interconnected and map a set of inputs to a set of desired outputs. In a CNN, neurons in layers are arranged through unidirectional branches in the feedforward direction. The input layer has several neurons corresponding to the number of input features, but the number of hidden layers is limited to one hidden layer with M biases. Hidden neurons are initially connected to input neurons with initial weights. During the training process, these weights are adjusted, and the neural network model is built concurrently. The third layer is the output layer, which includes only one neuron for each class. The output neurons can be determined according to the expected classification result values: 1 for normal connection (correct class) and 0 for abnormal connection (incorrect class) with a single bias. Typically, each input to the network is multiplied by its corresponding weight, and the weighted sum function produces a weighted sum, which is then passed through a transfer function called the activation function. The sum function is calculated by adding the products of subsequent inputs, initial weights, and the bias weight.

IV. Experiments and Implementation

A. Dataset

To achieve the detection and classification of attacks in a large-scale IoT network, it is essential first to identify and collect datasets related to DDoS attacks, each with specific characteristics. The CICDDoS2019 dataset is one of the most comprehensive

and up-to-date datasets for DDoS attacks, ensuring that the model is trained on realistic attack scenarios. In this study, the collected dataset comprises raw data with 494,022 rows based on 22 different features considered for middleware, categorized by specific attack types. Given the vast amount of data and the potential issues in analysis using the proposed system, 54,000 different cases were selected based on the same 22 features, divided into three basics, content, and traffic categories, to determine the type of attack. It should be noted that eight different attack types are considered, with their names and assigned numbers presented in Table 1. The inclusion of this dataset ensures that the proposed model is applicable to real-world IoT network environments.

Table 1: Attacks considered in the dataset

Number	Attack Name
1	normal
2	back
3	Ipsweep
4	ftp_write
5	nmap
6	smurf
7	neptune
8	multihup

B. Data Preprocessing

In this study, preprocessing is conducted according to established methods, which include:

- **Noise and Outlier Removal**: During data collection, some columns might contain unreasonable data, which need to be identified and corrected at this stage.
- **Data Sorting**: Data must be sorted in a way that MATLAB can understand and read. The number of rows indicates different network states, while the number of columns indicates the features related to the network that led to the occurrence or non-occurrence of an attack.
- **Data Labeling**: Since MATLAB processes data numerically, all variables must be labeled numerically. Some variables are qualitative and need to be evaluated alongside numerical variables, so all data are labeled numerically.
- **Data Normalization**: After converting all data to numerical variables, they can be easily loaded into MATLAB. Given that data scales are different, they must be converted to a standard form using the following formula:

$$\bar{x} = \frac{(x-xmin)(d2-d1)}{xmax-xmin} + d1 \qquad (1)$$

For our data, d_1=0 and d_2=+1. Thus, all data are normalized within this range.

C. Data Partitioning

In this research, 70% of the data is used for training the network, and the remaining 30% is used to test the model. This partitioning is completely random to ensure all data are used for both tasks. The randperm function in MATLAB can automatically generate random indices, and the corresponding data are placed in their respective matrices. With 54,000 final data rows, 70% (37,800 data points) are used to create the classifier model, and 30% (16,200 data points) are used to test the model.

D. Classification

As previously mentioned, this research uses several classifiers such as decision trees, KNN, SVM, and CNN to classify the selected features. It is worth noting that suitable feature selection and the elimination of unnecessary features are performed using BUZOA and Ant Colony Optimization algorithms.

E. Evaluation Metrics

The evaluation metrics in this research include accuracy, sensitivity, precision, F-score, and error rate. Positive samples are denoted by PPP and negative samples by NNN. The number of positive samples correctly identified by the classifier is indicated by true positives (TP). True negatives (TN) refer to the number of negative samples correctly labeled by the classifier. False positives (FP) represent the number of negative samples incorrectly labeled as positive, and false negatives (FN) denote the number of positive samples incorrectly labeled as negative.

- **Accuracy**: The accuracy of the classifier on the dataset is the percentage of samples correctly labeled by the classifier:

$$\text{accuracy} = \frac{TP+TN}{P+N} \qquad (2)$$

- **Sensitivity**: Known as the true positive rate, sensitivity is defined as:

$$\text{sensitivity} = \frac{TP}{TP+FN} \qquad (3)$$

- **Precision**: Widely used in classification, precision is considered a measure of correctness (the percentage of tuples labeled as positive that are actually positive):

$$\text{precision} = \frac{TP}{TP+FP} \qquad (4)$$

- **F-score**: Combines precision and sensitivity and is defined as:

$$\text{F-score}=\frac{2\times\text{precision}\times\text{sensitivity}}{\text{precision}+\text{sensitivity}}) \qquad (5)$$

F. Results Presentation

This sub-section initially displays the results of combining the DT-BUZOA, BUZOA-KNN, and BUZOA-SVM approaches in Table 2.

Table 2: Comparison of Approaches

Appr	TP	TN	FP	FN	Acc	Pre	Rec	F1	ErrRate
BUZOA – DT	279	110	29	31	0.87	0.90	0.90	0.9	0.13
BUZOA – KNN	259	119	42	36	0.83	0.86	0.88	0.87	0.17
BUZOA – SVM	296	129	21	9	0.93	0.93	0.97	0.95	0.06

As shown in the table above, the combined BUZOA-Decision Tree (BUZOA-DT) approach achieved a precision of 0.906 and an error rate of 0.134, successfully predicting DDoS attacks by reducing the number of features in the database. The KNN algorithm applied to the BUZOA algorithm resulted in a precision of 0.861 with an error rate of 0.171. The BUZOA-SVM combined approach reached a precision of 0.934 and an error rate of 0.066, outperforming the BUZOA-Decision Tree and KNN approaches. Thus, the table indicates that the BUZOA-SVM approach provides the best performance among the three approaches, with the highest accuracy, precision, recall, and the lowest error rate.

Next, to evaluate the performance of the proposed approaches, the ACO algorithm is used in combination with the Decision Tree, KNN, and SVM algorithms instead of the BUZOA algorithm. The comparison of these approaches in terms of the number of features and detection precision is illustrated in Figure 1.

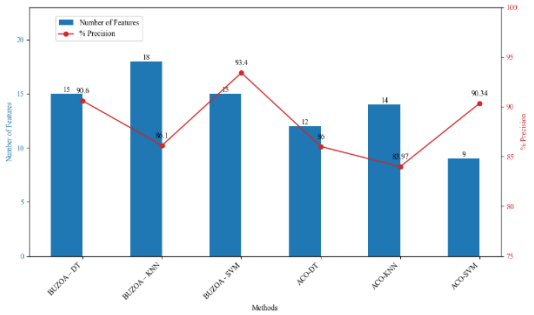

Fig. 1: Comparison of Approaches Using ACO and BUZOA

As shown in Figure 1, the combined BUZOA-DT approach with a precision of 0.906 outperforms the ACO-DT approach, which has a precision of 86.0027%. Furthermore, the BUZOA-DT approach selected only 15 features from the 22 effective features for DDoS attack detection in the network, while the ACO-DT approach selected 12 features. The results presented in Figure 1 indicate that using the KNN algorithm in combination with BUZOA and ACO ultimately resulted in higher precision for the BUZOA algorithm, selecting 18 features out of the 22.

Additionally, comparing the results of applying the SVM algorithm to the heuristic algorithms BUZOA and ACO shows that using the SVM algorithm in combination with BUZOA and ACO ultimately resulted in higher precision for the BUZOA algorithm, selecting 15 features.

Finally, the CNN algorithm is used for feature classification. Since the BUZOA-SVM combined approach has the best performance, selecting 15 features, these 15 features are also used in the CNN neural network method. Before presenting the results of applying the CNN algorithm to the selected features, the necessary settings for this algorithm must be made. In the fully connected layer, we will have the feature vector (using the activation function in MATLAB) and use it as a deep feature. In this work, the Stochastic Gradient Descent (SGD) algorithm is used to train the CNN. The parameters used for the SGD algorithm are provided in Table 3. The number of complete epochs is set to 30 for network training.

Table 3: Initial Settings for the CNN Model

Parameter	Value
Initial learning rate	0.0582
Learning rate decay	0.1
Number of iterations	100
Optimizer	Adam
Network architecture	ResNet50

Additionally, various training functions were tested to determine the training function, and the results are presented in Table 4.

Table 4: Comparison of Correlation of Different Deep Neural Network Training Functions

No.	Training Function	Definition	Correlation Coefficient
1	Trainoss	One-step secant	0.883
2	traincgb	Conjugate gradient backpropagation with Powell-Beale restarts	0.632
3	Traincgp	Conjugate gradient backpropagation with Polak-Ribiére updates	0.715
4	Trainscg	Scaled conjugate gradient	0.806
5	Trainbr	Bayesian regularization	0.936
6	Trainbfg	BFGS quasi-Newton	0.859
7	Trainlm	Levenberg-Marquardt	0.913
8	Traingda	Gradient descent with adaptive learning rate	0.678
9	Traingdm	Gradient descent with momentum	0.547
10	Traingdx	Gradient descent with momentum and adaptive learning rate	0.819

As seen in Table 4, the trainbr training function is used to train the proposed deep network approach.

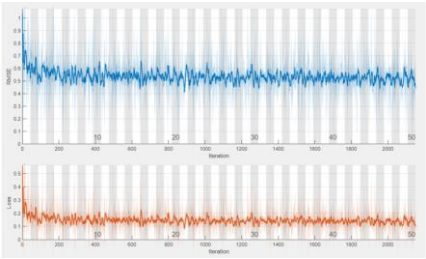

Fig. 2: Convergence Chart of the Proposed Approach

In Figure 2, it can be observed that the CNN model, after 2200 iterations, reached an average RMSE of 0.0117 and an accuracy of 96.32% for intrusion detection in computer systems.

Figure 3 shows the correlation chart of the proposed CNN network based on the distribution of training and testing data. The correlation coefficient is R = 0.91934.

For a better understanding of the results presented in the table above, the bar chart in Figure 4 is used to compare the precision of the approaches in intrusion detection.

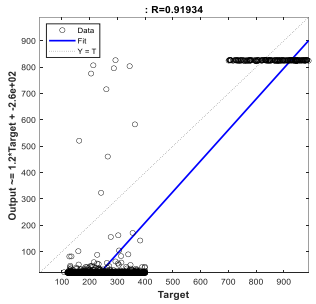

Fig. 3: Correlation Chart of the Proposed CNN Network

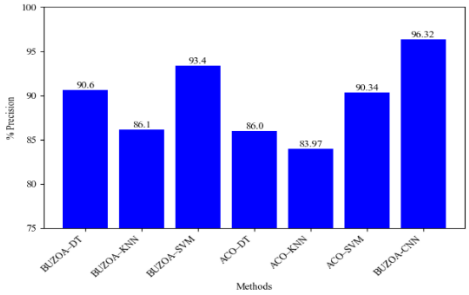

Fig. 4: Comparison of Detection Precision of Approaches

V. Conclusion

With the rapid advancement of information technology and internet networks, the issues of security and intrusion are expanding day by day. Intrusion detection systems are tasked with identifying and detecting any unauthorized use, misuse, or damage by both internal and external users. The goal of these systems is to discover and identify attacks, detect security flaws in the system, and alert the system administrator. In this research, we aimed to improve the performance of classification algorithms in detecting DDoS attacks in IoT networks by using ensemble learning techniques and combining classification algorithms in data mining. The collected data was sourced from the CICDDOS2019 database, which contains 54,000 samples and 22 initial features. For feature evaluation, two optimization algorithms, BUZOA and ACO, were used for feature selection, while classification algorithms such as Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and deep neural network CNN were employed for behavior classification and detection.

The results demonstrated that among the heuristic algorithms BUZOA and ACO applied to Decision Tree, K-Nearest Neighbors, and Support Vector Machine, the BUZOA-SVM combined approach exhibited the highest performance with an accuracy of 0.934. Furthermore, it was observed that the BUZOA-SVM approach selected only 15 features from the 22 significant features for intrusion detection. Hence, it can be asserted that the BUZOA-SVM combined approach achieved the best performance by ultimately selecting 15 features. In the next step, with these 15 features and the initial parameter settings of the CNN deep learning algorithm, classification and estimation of attacks and intrusions were carried out. The results indicated that this approach achieved an average RMSE error of 0.0117 and an accuracy of 96.32% in intrusion detection. Consequently, it was determined that the CNN algorithm outperformed the other algorithms used.
For a conclusion, why these methods suit the research objective?

- Scalability: The selected approaches are suitable for handling large datasets and real-world IoT environments.
- High Accuracy: The integration of feature selection and learning techniques enhanced detection accuracy while reducing unnecessary features.
- Generalizability: The proposed method is adaptable to real-world applications and diverse network environments.
- Computational Efficiency: By selecting the most relevant features, computational complexity was reduced while maintaining high detection performance.

These choices ensure that the proposed method effectively identifies DDoS attacks with high accuracy while optimizing feature selection and computational efficiency, making it a highly practical solution for IoT network security.

References

[1] J. Chen, H. Hu, M. Li, and Y. Li, "A review of machine learning for the Internet of Things," *Neural Computing and Applications*, vol. 33, pp. 7441-7458, 2021.
[2] J. Huang, and M. Xu, “A review of DDoS detection methods in IoT using machine learning,” *Journal of Cloud Computing*, vol. 10, no. 1, p. 33, 2022.
[3] J. Kaur, and R. Singh, “A comprehensive review on DDoS attack detection using deep learning in IoT environments,” *Wireless Personal Communications*, vol. 130, pp. 2301-2317, 2024.
[4] B. Smith, and L. Wang, “Deep learning approaches for detecting DDoS attacks in IoT systems,” *Future Generation Computer Systems*, vol. 144, pp. 184-196, 2023.
[5] T. Liu, and X. Wu, “A comparative study of machine learning algorithms for DDoS detection in IoT,” *Expert Systems with Applications*, vol. 234, p. 119027, 2023.
[6] T. X. Pham, Q. V. L. Nguyen, and V. Q. Nguyen, "A survey on IoT attack detection using machine learning," 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), pp. 289-294, 2019.
[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.
[8] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993-1001, 1990.
[9] McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification," *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41-48, 1998.
[10] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18-28, 2009.
[11] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303-336, 2014.
[12] Lazarevic, V. Kumar, and J. Srivastava, "Intrusion detection: A survey," *Managing Cyber Threats*, pp. 19-78, 2005.
[13] L. Zhang, W. Chen, and H. Yang, “Enhanced DDoS detection in IoT environments using hybrid deep learning techniques,” *Computer Networks*, vol. 225, p. 109212, 2024.
[14] J. Wang, and Q. Zhang, “A comprehensive DDoS detection model in IoT systems using multi-objective optimization and deep learning,” *Journal of Parallel and Distributed Computing*, vol. 186, pp. 68-79, 2024.
[15] A. Arshaghi, M. Ashourian, L, Ghabeli, " Buzzard Optimization Algorithm: A Nature-Inspired Metaheuristic Algorithm," *Majlesi Journal of Electrical Engineering*, Vol. 13, No. 3, pp. 83-98, 2019.
[16] M. A. Mohammad and M. Kolahkaj, "Detecting Network Anomalies Using the Rain Optimization Algorithm and Hoeffding Tree-based Autoencoder," *2024 10th International Conference on Web Research (ICWR)*, Tehran, Iran, Islamic Republic of, pp. 137-141, 2024.

[17] W. Dong, and F. Gao, “Evaluating the performance of deep learning models for DDoS detection in IoT environments,” *Computers & Security*, vol. 120, p. 102738, 2022.
[18] S. K. Sharma and X. Wang, "Live forensics of distributed IoT systems: A proposed approach for DDoS attack," 2019 IEEE International Conference on Big Data (Big Data), pp. 5773-5778, 2019.
[19] J. Zhao, and L. Feng, “Efficient DDoS detection mechanism in large-scale IoT networks using ensemble learning,” *Expert Systems with Applications*, vol. 240, p. 119804, 2024.
[20] Bhardwaj, A. K. Bindal, and M. K. Soni, "Enhanced DDoS attack detection using K-means clustering and hybrid ensemble technique in SDN," *IEEE Access*, vol. 9, pp. 92703-92713, 2021.
[21] J. Ni, X. Lin, and X. S. Shen, "Efficient and secure service-oriented authentication supporting multiple services in vehicular social networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4224-4235, 2019.
[22] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Applied Soft Computing*, vol. 10, no. 1, pp. 1-35, 2010.
[23] M. Feng, and J. Li, “Adaptive DDoS attack detection using hybrid machine learning methods in software-defined IoT networks,” *Information Sciences*, vol. 662, pp. 584-595, 2024.
[24] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6225-6232, 2010.
[25] Z. Xiao, P. Yang, and X. Zhang, "A new ensemble anomaly detection method based on a Gaussian mixture model," *IEEE Access*, vol. 8, pp. 154523-154535, 2020.
[26] W. Zhang, L. Cui, Z. He, Y. Wen, and Y. Zhang, "Optimized K-means clustering algorithm for outlier detection," 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 353-356, 2017.
[27] Mahmud, and M. Rahman, “Evaluating machine learning techniques for DDoS detection in large-scale IoT environments,” *Journal of Cloud Computing*, vol. 13, no. 1, p. 49, 2024.
[28] N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems," 2015 Military Communications and Information Systems Conference (MilCIS), IEEE, pp. 1-6, 2015.
[29] R. M. Noor and R. Hassan, "Machine learning model for DDoS attack detection," *Procedia Computer Science*, vol. 179, pp. 436-442, 2021.
[30] M. Othman, M. M. Kamaludin, M. F. Yusof, and S. H. Ariffin, "Machine learning approach for DDoS attack classification," *Journal of King Saud University-Computer and Information Sciences*, 2020.
[31] Z. Xu, C. Wang, and M. Song, "A DDoS attack detection and mitigation with software-defined Internet of Things framework," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 1, pp. 2-14, 2019.
[32] L. Liu, H. Zhu, and Y. Ma, "Intrusion detection based on hybrid model of rough set and neural network," 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 2596-2600, 2018.
[33] H. Wang, and J. Chen, “A hybrid deep learning approach for detecting DDoS attacks in IoT environments,” *Applied Soft Computing*, vol. 148, p. 110057, 2024.
[34] R. Singh, and P. Sharma, “A novel deep learning-based DDoS detection framework for IoT networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, pp. 502-513, 2024.
[35] F. Iqbal, A. Matrawy, and A. G. Banihani, "Detection of DDoS attacks using artificial neural networks," 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-6, 2017.
[36] J. Zhang and Z. Zhou, "A deep learning-based approach for DDoS attack detection in IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5663-5671, 2020.
[37] Roy, S. K. Das, and P. Deb, "A deep learning approach to mitigate DDoS attacks in fog computing environment," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 234-249, 2021.
[38] N. Patel, and K. Shah, “An effective DDoS detection mechanism using convolutional neural networks in IoT systems,” *Journal of Big Data*, vol. 11, no. 1, p. 42, 2024.
[39] M. Joshi, and K. Sharma, “Detecting DDoS attacks in IoT networks using hybrid deep learning techniques,” *Journal of Network and Systems Management*, vol. 32, pp. 152-168, 2024.
[40] Roy, and S. Bandyopadhyay, “A hybrid machine learning approach for DDoS detection in IoT networks,” *Journal of Information Security and Applications*, vol. 67, p. 102905, 2023.
[41] Y. Zhang, and J. He, “Anomaly-based DDoS detection using hybrid machine learning techniques in IoT networks,” *Journal of Network and Computer Applications*, vol. 189, p. 103234, 2022.
[42] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," 2017 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1-8, 2017.
[43] T. Roy, and S. Bhattacharyya, “Real-time DDoS attack detection using hybrid machine learning approaches in IoT,” *Journal of Supercomputing*, vol. 80, pp. 1249-1265, 2024.
[44] H. Gao, and Y. Sun, “Enhancing DDoS detection in IoT with optimized deep learning models,” *Journal of Information Security and Applications*, vol. 65, p. 102892, 2022.
[45] P. Sharma, and A. Jain, “Optimized deep learning techniques for detecting DDoS attacks in cloud-based IoT systems,” *International Journal of Information Technology*, vol. 16, pp. 905-920, 2024.

Dr. Maral Kolahkaj is an Assistant Professor in the Department of Computer Engineering at Islamic Azad University. She is an active researcher in the fields of data mining, recommender systems, artificial intelligence, and etc. She serves as a reviewer for ISI-indexed journals and has published numerous research articles in high-impact international journals and conferences. Her expertise includes machine learning, deep learning, and their applications in various domains.
https://orcid.org/0000-0002-5235-0227