ISSN: 2633-4828 Vol. 5 No.4, December, 2023

International Journal of Applied Engineering & Technology

DEEP LEARNING OPERATIONS BY ELEPHANT HERD AND BAT ALGORITHM FOR IOT INTRUSION DETECTION

[1]Diwakar Kumar Chaudhary, [2]Dr. Pritaj Yadav and [3]Mrs. Kanchan Jha

[1]Research Scholar and [2]Associate professor, Computer Science and Engineering, Ravindranath Tagore University, Bhopal, India

[3]Lecture, J.S.C.S H/S Parsauni Murliyachak, Madhubani, India

[1]diwakarbright@gmail.com, [2]yadavpritaj@gmail.com and [3]kkanchan.jha@gmail.com

ABSTRACT

Internet of Things (IOT) brings flexibility and control in unfavorable conditions. IOT is adapted by various businesses to provide solutions for communication. But flexibility in computer networks increases the chance of attack. Hence security of data is highly desirable to build trust in the nodes, as many solutions collect data for research, monitoring purposes. This paper has proposed a model that improves the network security by optimizing the detecting session features. Deep learning operations were applied to identify the feature set for training and testing of machine. Proposed model uses the elephant Herd and BAT algorithm for feature detection. Selected features were further process for normalization of training values. Back propagation neural network was used in the proposed model for the training of IOT intrusion detection. Experiment was done on a real IOT dataset developed in 2020, and has five classes of attacks. Result shows that the proposed BAT based model has increased the precision, recall and detection accuracy parameters.

Keywords- Deep Learning, Intrusion Detection, Feature Optimization, Genetic Algorithm, Soft Computing.

I. INTRODUCTION

The Internet of Things (IoT) has become widely popular for its extensive potential in various sectors, such as healthcare, transportation, and smart cities. It involves connecting numerous physical devices, known as "things," within a network. These devices typically have limited computational power and storage capabilities. As the number of diverse devices increases in the IoT, a significant amount of data is generated, making IoT networks attractive targets for potential attackers.

The main challenges in IoT revolve around devices with constrained resources, low computational and storage capacities, and cybersecurity concerns [1]. Implementing essential security measures becomes challenging. In the current landscape, Machine Learning (ML) and Deep Learning (DL) techniques have emerged as suitable approaches for processing large amounts of data, leading to improved computational results. Various ML techniques are used to select optimal features from datasets, while DL methods automatically extract the best features during their application to a dataset [2].

Addressing cybersecurity in IoT networks has become a paramount concern, requiring the planning and implementation of effective Intrusion Detection Systems (IDS) at edge nodes. Over recent decades, both ML and DL-based IDSs have proven effective in detecting attacks within IoT networks. Thus, these approaches are gaining popularity in the cybersecurity realm, proving well-suited for identifying threats in IoT environments.

Nodes within an IoT network typically have limited capacity, constrained resources, and minimal manual control compared to traditional networks. These seemingly unassuming technological components often expose themselves to potential attacks, escalating concerns about their security due to the continuous emergence of new cyber threats. Various security mechanisms have been developed over the years, some effective against specific types of attacks [4]. Given the substantial data volume generated by the IoT, there is a need for efficient methodologies to detect attacks rapidly. Common forms of assaults on IoT communication channels include botnets, denial-of-service (DoS) attacks, man-in-the-middle attacks, infiltration, identity theft, data theft, ransomware, and more. Among these, botnet threats are particularly prevalent and challenging to completely thwart due to their evolving nature over time.

In [7], an enhancement in the Software-Defined Networking (SDN)-based intrusion detection and prevention system for IoT is presented. This mechanism utilizes SDN capabilities to create a proactive system designed for intrusion detection within IoT networks. The SDN-based system facilitates network programming by separating the control and data planes, providing a comprehensive global view of the network. With its programmability feature and holistic network perspective, SDN emerges as a superior alternative to address challenges encountered in ensuring the seamless operation of IoT [8,9].

II. RELATED WORK

In their study [8], Xiu Kan and team introduced a smart way to catch bad things happening in the Internet of Things (IoT). They used a mix of smart algorithms called Adaptive Particle Swarm Optimization Convolutional Neural Network (APSO-CNN). It's like having a bunch of tiny particles working together to adjust the settings of a special computer program that looks out for problems. They made sure this program learns well by checking how good it is using a set of data.

In another research [9], Fatani and team created a security system using a mix of deep learning (like how our brains learn) and optimization methods. They first built a way to understand important features using something called CNNs. Then, they used a special method called Growth Optimizer (GO) to pick the most useful features. To make the search for these features even better, they used something called Whale Optimization Algorithm (WOA). They tested this system on different datasets to see how well it could find and stop bad things.

Saied and others [10] did a big study to see which computer learning methods work best for catching problems in IoT networks. They tried out many different methods, like Adaptive Boosting, Gradient Descent Boosting, and others. They wanted to find the one that's most accurate and fast at stopping issues in IoT networks.

Zambare and team [11] developed a technique using a special computer program called Graph Neural Network (GNN) to spot problems in virtual networks. They tested it with data from car-hacking situations, like attacks that make a car's computer act weird. The GNN program looks at patterns and figures out what kind of attack is happening.

Kalyanam and others [12] came up with a smart way to make sure the security program on IoT devices works well without using too much computer power. They used a technique called threshold-based pruning, which is like trimming unnecessary things from a big tree. They tested this method on a common computer program called LeNet, using different datasets to check if it still works well after trimming.

In their research [13], T.-T.-H. Le and team introduced a new approach to explain why the security program thinks something is wrong. They used a mix of different methods to make sure the program not only catches problems but also tells us why it thinks there's a problem. They tested this approach using special datasets that simulate real-world situations with IoT devices.

K. A. Awan and team [14] tackled the challenge of finding and stopping bad things in IoT by using a special learning method called federated learning. This is like a group of friends working together to learn and share knowledge. They trained each IoT device to recognize and predict abnormal behavior using a trust dataset with information from knowledge, experience, and reputation. This method makes it easier for the devices to work together without using too much computer power.

III. PROPOSED METHODOLOGY

This section provides a summary of the proposed GINIDS (Genetic IoT Network Intrusion Detection System) methodology. Figure 1 illustrates the steps involved in training the model, with detailed explanations for each block. Additionally, Figure 2 presents a block diagram outlining the testing process for the trained model. For reference, Table 1 lists the different notations used in this work.

Table 1. BINIDS notation table.

Notations	Meaning
RID	Raw IOT Dataset
PID	Processed IOT Dataset
GP	Genetic Population
b	Number of chromosome in GP
m	Number of Feature in PID
GF	Chromosome Fitness
FF	Filter Features
Do	Desired Output
GIDMLM	Genetic Intrusion Detection Machine Learning Model

Pre-Processing of IOT Dataset

The dataset for the Internet of Things (IoT) network sessions includes both feature values and text collections. To enhance the efficiency of the algorithm, certain features that are repetitive or constant throughout the entire dataset are removed as part of the cleaning process. If RID represents the raw IoT dataset and PID denotes the processed dataset, the cleaning process can be expressed as:

PID←IOTDatasetCleaning(RID)------Eq. 1

Subsequently, the processed dataset undergoes normalization, as some feature values are in the range of 0 to 1000 while others are in the range of 0 to 1. To ensure consistency, all values are transformed to a 0 to 1 scale by taking the ratio with others:

PID←Normalization(PND)----------Eq. 2

Genetic Algorithm Optimization

In this proposed model two genetic algorithm was used for the feature seelction. First was BAT algorithm and other was elephant herd optimization. Both algorithm has few common steps population generation, fitness function and population updation. Modify population operations (Crossover and mutation) of the BAT and Elephant Herd algorithms are different.

Generate Population: In this step, a set of chromosomes is artificially generated using the Gaussian position function. Each chromosome is a set of feature positions with binary values, where 1 indicates inclusion in the training vector and 0 means the feature is not involved in training. Since genetic algorithms work dynamically, feature positions are generated using the Gaussian function. If the dataset has m features in the PID matrix and b chromosomes are generated, the genetic population (GP) is represented as a matrix of dimensions bxm.

GP←Generate_Population(b, m)----------Eq. 3

Fitness Function The fitness function assesses the fitness of each chromosome based on the accuracy of intrusion class detection. The evaluation is carried out by determining the detection accuracy of intrusion through the machine learning model. This accuracy value serves as the fitness value for the chromosome.

GF_f←Fitness(GP, PID) ----------Eq. 3

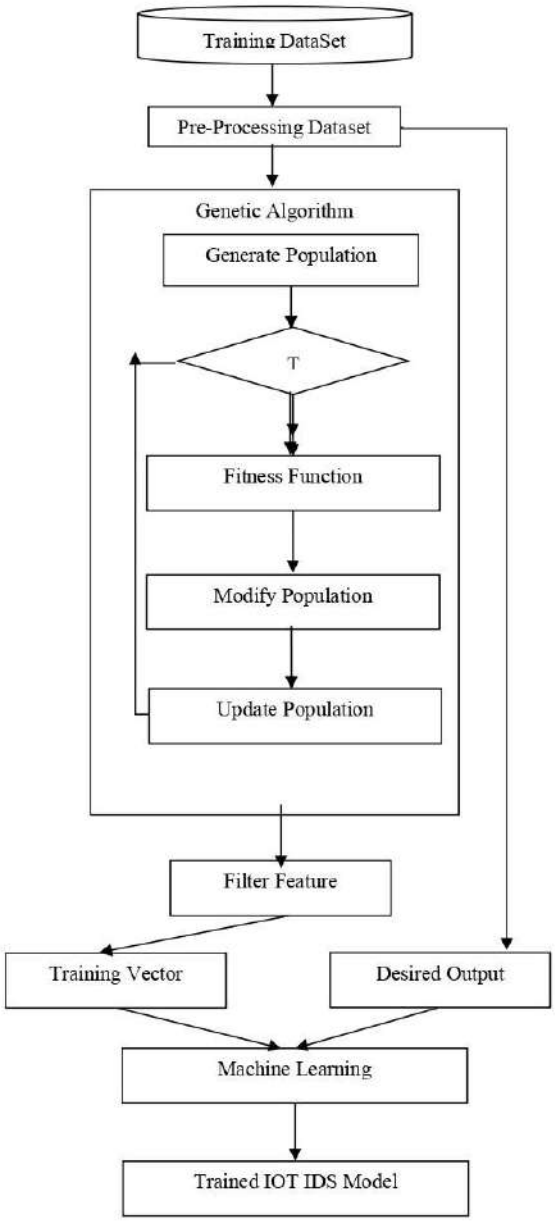

Fig. 1 Flow diagram of BINIDS proposed model.

BAT Algorithm Modification Process

Update BAT Location After obtaining the fitness value (F) through the fitness function, the values are sorted in descending order, and the best BAT (Chromosome) is identified from all the chromosomes in the population.

bfreq(i,1)=fmin+(fmax-bfreq(i,1))*beta;

bv(i,1)=bv(i,1)+(bpos(i,1)-agg_obj(i,1))*bfreq(i,1);

agg_obj(i,1) = w1*sum(Moth_pos(i,1))+w2*(100-fitness(i,1)*100)+w3*fitness(i,1)*100;

bpos(i,1)=bpos(i,1)+bv(i,1);

chg = floor(bfreq(i,1));

The success of the genetic algorithm relies on the alteration of chromosomes. Therefore, based on the parameter X, a specific number of random position values of BATs are modified. Notably, this operation excludes the best local BAT [19]. During this step, X positions of each BAT are randomly modified from zero to one or one to zero, guided by the best three local BAT feature sets.

Elephant Herd Optimization

Update Clan The best chromosome matriarch, M, is determined based on the fitness values of each elephant (chromosome) in the clan within the population [14]. Random changes are made to a number of statuses based on the feature set of the best matriarch, M. Cloning is performed by incorporating the best elephant set page into other elephants within the clan.

GP←Clan_update(GP, GP)

Separating Elephants with low fitness values are removed from the clan, taking the form of male elephants [15]. This elimination occurs after assessing the new clan fitness value.

Update Population These chromosomes are then tested for fitness values against parent chromosomes. If a child chromosome exhibits superior values, the parent is removed; otherwise, the parent remains. After this step, if the maximum iteration steps are reached, the process jumps to the filter feature block; otherwise, the fitness value of each chromosome is evaluated.

Filter Feature Upon completing the iterations, the best chromosome from the last updated population is identified. Features with a value of one in the chromosome are considered selected features for the training vector, while those with a value of zero are considered unselected.

FF←FilterFeatures(PID, GF, GP)

Training of Neural Network: The training of the neural network involves using the chosen features as input training vectors, along with the expected output during the training process. For every group of training vectors, the weights of the neurons are fine-tuned over a specific number of training cycles, which we call epochs. After this training, the GIDMLM Intrusion Detection trained neural network is employed to directly predict whether a session is classified as an attack or normal.

GIDMLM ← Train(FF, Do)

IV. Experiment and Results

Experimental setup: BINIDS and comparing model was developed on MATLAB software. Experimental machine having 4 GB ram, i3 6th generation processor. GINIDS has two models first is for BAT based approach BINIDS (BAT based IOT Network IDS) and second is was for Elephant DEIIDS (Dynamic Elephant based IOT IDS). Comparison of **GINIDS** was done with previous IOT malicious session detection model proposed in [19]. IOT dataset was taken from [20]. This dataset has 86 attributes, where three is the class of session and rest 83 is to training/testing features. Total number of sessions are 625784, with two and multiclass named sessions.

Table 2. Precision value based comparison of network intrusion detection models.

Dataset Size	Tsode	Ginids	
		DEIIDS	**BINIDS**
5000	0.923	0.9772	0.9858
10000	0.9984	0.976	0.9861
15000	1	0.9722	0.9865
20000	0.9236	0.9786	0.9894
25000	0.8365	0.9824	0.9846

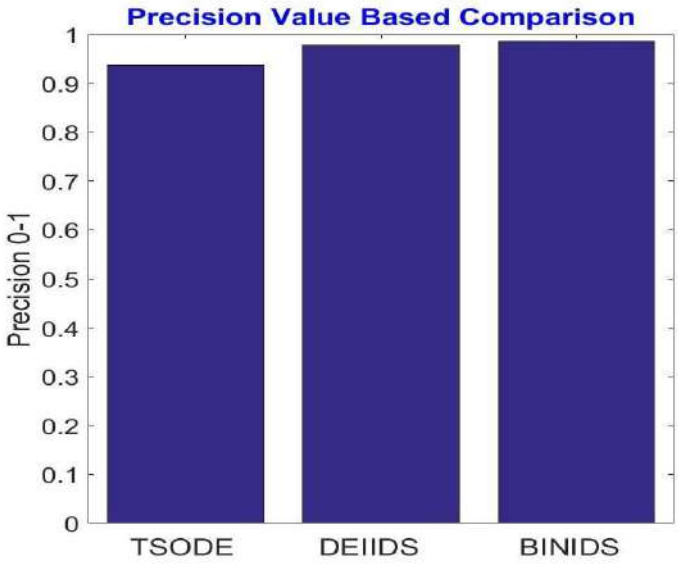

Fig. 2 Precision average values of IOT intrusion detection model.

Table 2 displays the precision values of IOT network intrusion detection models. It was discovered that the suggested GINIDS raises the detection's real alarm. Further it was found that use of elephant herd optimization for the training feature optimization has less learning accuracy in DEIIDS as compared to BAT algorithm BINIDS.

Table 3. Recall value based comparison of network intrusion detection models.

Dataset Size	**TSODE**	**GINIDS**	
		DEIIDS	**BINIDS**
5000	0.799	0.975	0.9761
10000	0.9346	0.962	0.9756
15000	0.942	0.985	0.9906
20000	0.7633	0.9908	0.9942
25000	0.8654	0.993	0.9718

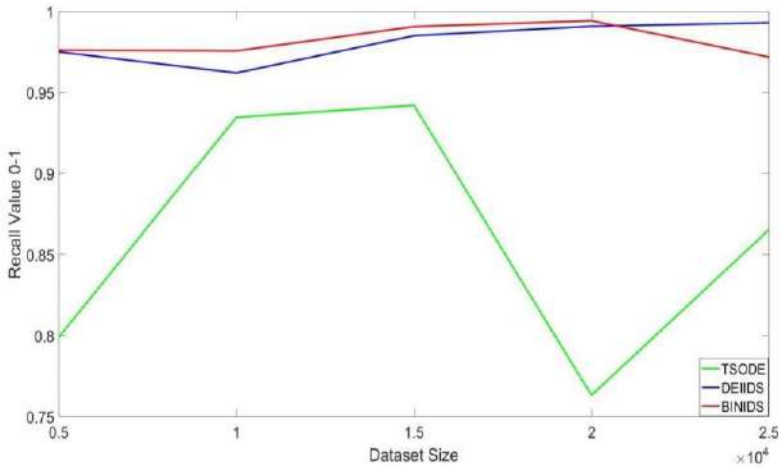

Fig. 3 Recall values of IOT intrusion detection model against different testing dataset.

In Table 3, the recall values for the proposed model are notably higher compared to previous existing work. Additionally, it was discovered that applying deep learning operations on the selected features transforms the values into more relevant and meaningful representations.

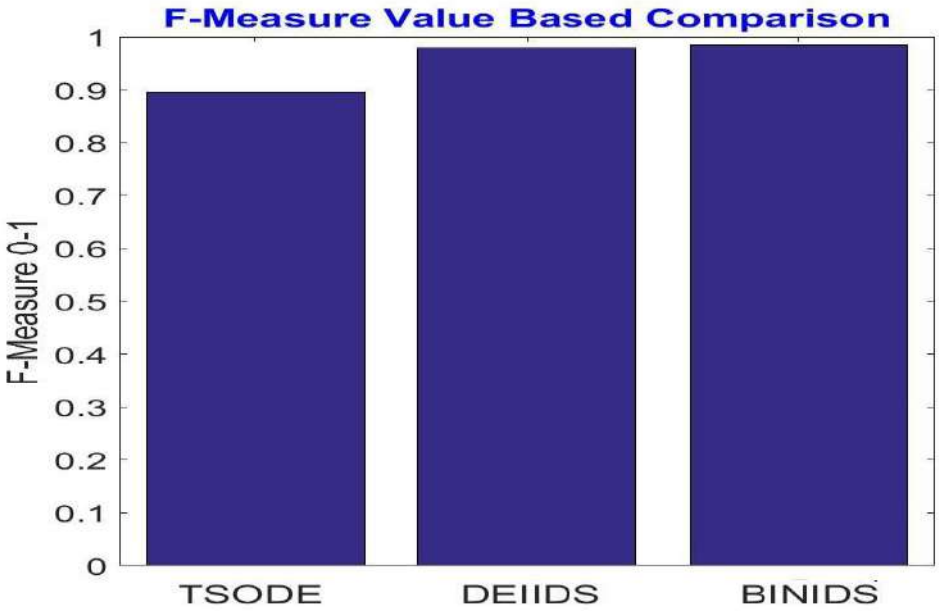

Fig. 4 F-measure average values of IOT intrusion detection model.

Table 4. F-Measure value based comparison of network intrusion detection models.

Dataset Size	**Tsode**	**Ginids**	
		DEIIDS	**BINIDS**
5000	0.8568	0.976	0.9809
10000	0.9655	0.969	0.9808
15000	0.97	0.9786	0.9885
20000	0.8359	0.9847	0.9918
25000	0.8507	0.9878	0.9781

The F-measure values in Table 4 indicate a high true alarm rate for intrusion class detection in the proposed model. The observation suggests that optimizing features and normalizing them contribute to the model's learning. Therefore, the combination of deep learning with feature optimization proves to be effective in detecting intrusion in IoT networks.

Table 5. Accuracy value based comparison of network intrusion detection models.

Dataset Size	**TSODE**	**GINIDS**	
		DEIIDS	**BINIDS**
5000	89.53	97.721	97.67
10000	95.38	98.07	98.8
15000	97.33	97.7	98.76
20000	89.66	98	98.92
25000	91.62	98.25	96.92

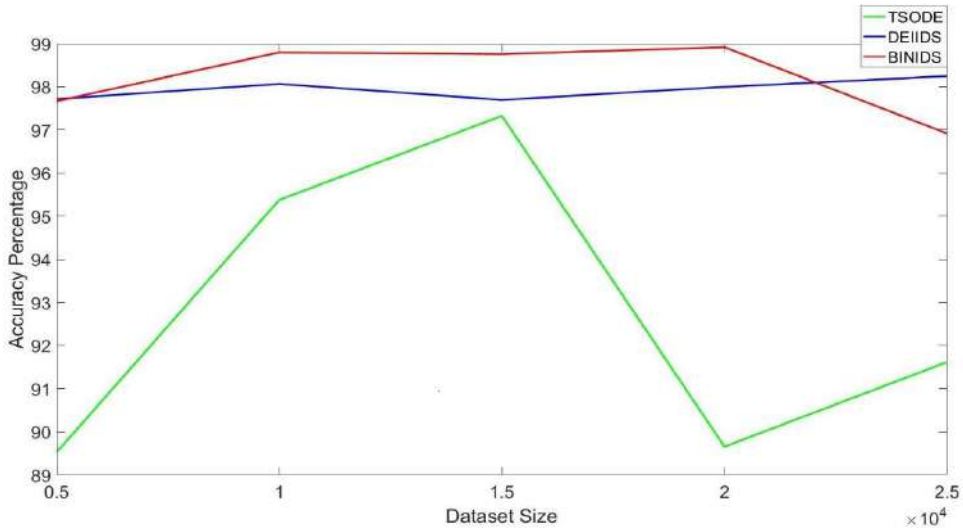

Fig. 5 Accuracy values of IOT intrusion detection model against different testing dataset.

Accuracy values of IOT network intrusion detection models shown in table 5. It was found that proposed model has increases the true alarm of the detection. Further it was found that use of elephant herd optimization for the training feature optimization has less learning accuracy in DEIIDS as compared to BAT algorithm BINIDS.

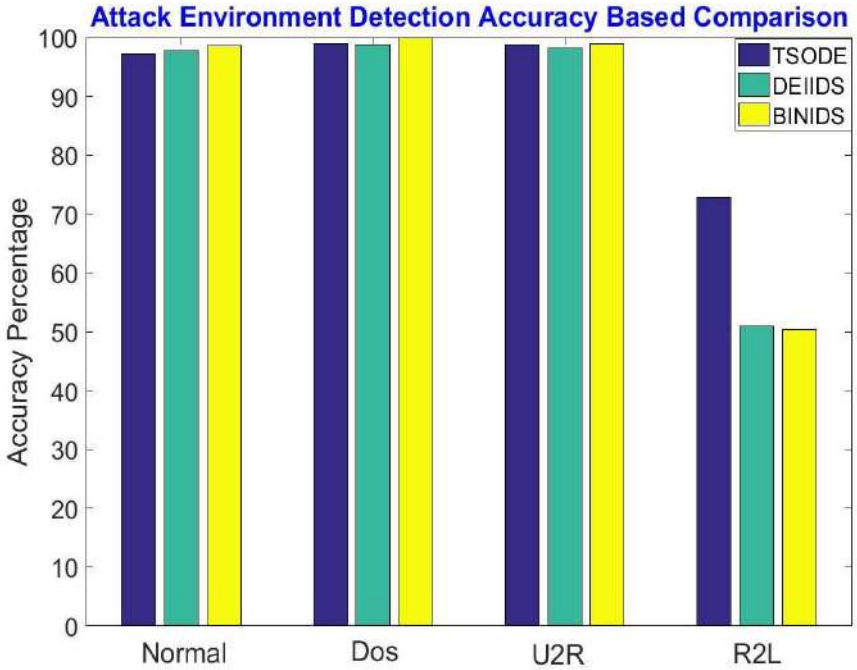

Fig. 6 Accuracy values of IOT intrusion detection model against different testing dataset.

Fig. 6 shows that accuracy values of the proposed model is high to detect all class of intrusion as well. Further it was found that use Deep learning operation on selected feature transform the values in more relevant values.

IV. CONCLUSION

This paper introduces a novel IoT network security model, aiming to extend the system's lifespan by promptly identifying and alerting against potential attack sessions. Additionally, the proposed model seeks to enhance the learning efficiency of the system by strategically reducing the feature set involved. An interesting observation from the study is the utilization of a genetic algorithm for optimization, which proved effective in reducing the learning epochs required for the system. Moreover, the research emphasizes the significance of normalization and the genetic algorithm operator, showcasing their effectiveness across a diverse range of intrusion class detection scenarios. Notably, the study reveals that the BAT algorithm surpasses the elephant herd optimization algorithm in terms of feature selection performance. The results of the experimentation highlight significant improvements in accuracy when detecting normal and various attack classes. Precision, recall, and F-measure values also exhibited enhancement, underscoring the overall effectiveness of the proposed work. Looking ahead, future scholars are encouraged to explore experimental setups within underwater IoT network environments.

REFERENCES

1. R. Vinayakumar, M. Alazab, S. Srinivasan, Q. V. Pham, S. K. Padannayil, and K. Simran, “A visualized botnet detection system based deep learning for the internet of things networks of smart cities,” IEEE Transactions on Industry Applications, vol. 56, pp. 4436–4456, 2020.
2. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep learning approach for intelligent intrusion detection system,” IEEE Access, vol. 7, pp. 41525–41550, 2019.
3. Smith S. IoT Connections to Reach 83 Billion by 2024, Driven by Maturing Industrial Use Cases. Accessed Apr. 2020;10:2021.
4. Awotunde J.B., Chakraborty C., Adeniyi A.E. Intrusion Detection in Industrial Internet of Things Network-Based on Deep Learning Model with Rule-Based Feature Selection. Wirel. Commun. Mob. Comput. 2021;2021:7154587. doi: 10.1155/2021/7154587.
5. Wazzan M., Algazzawi D., Albeshri A., Hasan S., Rabie O., Asghar M.Z. Cross Deep Learning Method for Effectively Detecting the Propagation of IoT Botnet. Sensors. 2022;22:3895. doi: 10.3390/s22103895.
6. Li Y., Liu Q. A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. Energy Rep. 2021;7:8176–8186. doi: 10.1016/j.egyr.2021.08.126.
7. Wani, A., Revathi, S.: Analyzing threats of IoT networks using SDNbased intrusion detection system (SDIoT-IDS). Commun. Comput. Inf.Sci. 828, 536–542 (2018).
8. Valdivieso Caraguay, ÁL., et al.: SDN: Evolution and Opportunities inthe Development IoT Applications. Int. J. Distrib. Sens. Netw. 10(5),735142(2014).
9. Kiani, F.: A survey on management frameworks and open challenges inIoT. Wirel. Commun. Mob. Comput. 1–33 (2018).
10. Xiu Kan, Yixuan Fan, Zhijun Fang, Le Cao, Neal N. Xiong, Dan Yang, Xuan Li. "A novel IoT network intrusion detection approach based on Adaptive Particle Swarm Optimization Convolutional Neural Network", Information Sciences, Volume 568, 2021, Pages 147-162.
11. Fatani, A.; Dahou, A.; Abd Elaziz, M.; Al-qaness, M.A.A.; Lu, S.; Alfadhli, S.A.; Alresheedi, S.S. Enhancing Intrusion Detection Systems for IoT and Cloud Environments Using a Growth Optimizer Algorithm and Conventional Neural Networks. Sensors 2023, 23, 4430.
12. Saied, M., Guirguis, S. & Madbouly, M. A Comparative Study of Using Boosting-Based Machine Learning Algorithms for IoT Network Intrusion Detection. Int J Comput Intell Syst 16, 177 (2023)

13. Zambare, P., Liu, Y. (2024). An Optimized Graph Neural Network-Based Approach for Intrusion Detection in Smart Vehicles. In: Puthal, D., Mohanty, S., Choi, BY. (eds) Internet of Things. Advances in Information and Communication Technology. IFIPIoT 2023. IFIP Advances in Information and Communication Technology, vol 683. Springer, Cham.
14. Kalyanam, L.K., Joshi, R., Katkoori, S. (2024). Layer-Wise Filter Thresholding Based CNN Pruning for Efficient IoT Edge Implementations. In: Puthal, D., Mohanty, S., Choi, BY. (eds) Internet of Things. Advances in Information and Communication Technology. IFIPIoT 2023. IFIP Advances in Information and Communication Technology, vol 683. Springer, Cham.
15. T. -T. -H. Le, R. W. Wardhani, D. S. C. Putranto, U. Jo and H. Kim, "Toward Enhanced Attack Detection and Explanation in Intrusion Detection System-Based IoT Environment Data," in IEEE Access, vol. 11, pp. 131661-131676, 2023, doi: 10.1109/ACCESS.2023.3336678.
16. K. A. Awan, I. Ud Din, M. Zareei, A. Almogren, B. Seo-Kim and J. A. Pérez-Díaz, "Securing IoT With Deep Federated Learning: A Trust-Based Malicious Node Identification Approach," in IEEE Access, vol. 11, pp. 58901-58914, 2023, doi: 10.1109/ACCESS.2023.3284677.
17. The rest of this article is organized as follows. Section II provides an overview of work done by other researchers in field of IOT intrusion detection. Section III describes our methodology for the proposed model, including the feature clustering model, and learning model. Section IV describes the evaluation results and analyzes the current state of the art. Finally, Section V concludes this article.
18. M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour and Y. Farhaoui, "An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security," in Big Data Mining and Analytics, vol. 6, no. 3, pp. 273-287, September 2023.
19. A. Fatani, M. Abd Elaziz, A. Dahou, M. A. A. Al-Qaness and S. Lu, "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," in IEEE Access, vol. 9, pp. 123448-123464, 2021, doi: 10.1109/ACCESS.2021.3109081.
20. Ullah I., Mahmoud Q.H. (2020) A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In: Goutte C., Zhu X. (eds) Advances in Artificial Intelligence. Canadian AI 2020.