Fuzzy Systems and Soft Computing
ISSN : 1819-4362

AN EFFICIENT FRAMEWORK FOR INTRUSION DETECTION SYSTEM IN IOT USING DEEP LEARNING

Dr. K. Priya Assistant Professor, Department of Computer Applications and Technology, SRM Arts and Science College, Tamil Nadu, India

Abstract

Background: Security for these interconnected systems has become more critical as the number of devices connected to the Internet of Things (IoT) has increased. This research adds substantially to the current conversation about effective cybersecurity defences by addressing new risks in the ever-changing Internet of Things (IoT) environment. **Methods**: Deep Neural Networks (DNN) demonstrates a new approach to building an Intrusion Detection System (IDS) for the Internet of Things (IoT). The study highlights the effectiveness of DNN-IDS in detecting anomalies, especially when used with the NSL-KDD and TONNE-IoT datasets. **Findings**: The results of the studies were encouraging, DNN reached 0.8976. The Random Forest (RF) method for classification, efficient feature selection and extraction operations, is responsible for this remarkable accuracy. When looking at metrics like recall, accuracy, precision, and F1-score, it is clear that DNN-IDS perform exceptionally well in both the training and testing stages. **Novelty and applications**: The paper delves deep into the efficacy of DNN-IDS, highlighting their adaptability and resilience across different assessment criteria. This study demonstrates a new approach to building an Intrusion Detection System (IDS) for the Internet of Things (IoT).

Keywords:
Intrusion Detection System (IDS), Internet of Things (IoT), Deep Learning (DL), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN).

1. Introduction

Transportation, automated homes, medical care, production, warehouse management, safety, and Blockchain monitoring are just a few of the many industries that have been greatly affected by the new computer era brought about by the Internet of Things (IoT). The versatility of IoT devices is evident in their diverse applications, and it is projected that their number will reach 30.9 billion by 2025, indicating a remarkable growth rate exceeding 55% from the 13.8 billion devices in use in 2021, as forecasted [10]. This rapid expansion has attracted attention from researchers, inventors, entrepreneurs, and cybercriminals alike, all recognizing the immense commercial potential and reliance on IoT devices. The market demand for these devices is substantial due to their utility, prompting increased interest from prospective investors. Entrepreneurs and innovators are continually enhancing the appeal of the field by introducing novel applications that improve and simplify daily life [14].

However, the surge in IoT adoption has also created opportunities for cybercriminals to exploit security vulnerabilities in these devices. The exponential increase in cyber attacks on IoT devices is attributed to the sector's rapid growth and potential financial gains, as highlighted in research by various sources [15]. R. Williams et al.'s research reveals vulnerabilities in many fully functional IoT devices [16]. The broader threat landscape surrounding IoT devices is exacerbated by their common integration with other systems and appliances. This connectivity often serves as an entry point for malicious actors seeking access to associated resources. A notable example is the 2016 Dyn attack, where imaging IoT devices caused an internet outage affecting major platforms like Amazon, Twitter, and Netflix [11].

Most Internet of Things (IoT) gadgets and applications have inadequate security in their architecture. The creation of malware networks and the distribution of various forms of harmful software are both made possible by hacker groups taking advantage of this weakness. Due to their inherent complexity and restricted funds, typical techniques for identifying intrusions are not suitable for protecting IoT systems [12]. Within intelligent platforms, which predominantly utilise Wi-Fi networks to connect IoT devices, there is a critical requirement for heightened security in wireless transmission [1].

Certain devices lack crucial hardware security support, which results in their inability to detect cyberattacks. As a consequence, IoT devices become inefficient at defending against advanced threats [2]. The security issue in the IoT stems from the wide range of gadgets and detectors, wireless connections, insufficient device security design, limitations in resources, and the intrinsic intricacy of the IoT. The ability to communicate and share data inside systems powered by the IoT relies heavily on protocol routing. [13]. One well-known IPv6 networking method is the RPL protocol, which is ideal for low-power, low-loss networks.

However, this protocol encounters difficulties in dealing with advanced cyber threats [8]. The RPL system has several vulnerabilities that make it susceptible to attacks at the communication layer, including insufficient processing power, power reserve capacity, and targeted security frameworks [3]. The Sybil assault exemplifies how these networks are particularly prone to DoS attacks. The Sybil attack changes the way DODAG Information Objects (DIOs) are sent in RPL by breaking implicit assumptions on purpose, spreading fake identity information, and rendering core nodes useless while flooding the network with DIOs [4]. Aside from IDS, there are various countermeasures to address cyber-attacks on IoT devices [9].

The monitor-based technique entails the observation and recording of network traffic and communication activities for the purpose of detecting potential routing faults. In a given instance [5], individual nodes gauge the frequency of lost packets by monitoring the network traffic of adjacent nodes in both the downstream and upstream directions. The objective is to detect instances of selectively forwarding threats in wireless mesh networks (WMNs). These nodes across the forwarded path that act as intermediaries communicate with the source node in an acknowledgment-based mechanism to confirm packet receipt or indicate improper routing behaviour.

Many academics have utilised meta-heuristic techniques to tackle the problem of decreasing features in datasets with a high number of dimensions. A novel approach to anomaly detection in the IoT was presented in [6]. Their plan is to combine the random forest technique with two more optimisation algorithms, GWO and PSO. Although machine learning algorithms have achieved success in detecting anomalies, their effectiveness has been slightly hindered by the growing number of features and the volume of data. In contrast, multiple research suggest that deep learning approaches surpass machine learning methods, especially when applied to large datasets [7]. IDS for the IoT, also known as IoT, can greatly benefit from deep learning approaches because of the massive amounts of data generated by diverse devices with complicated properties.

Regarding anomaly-based IDS in network data, Deep Neural Networks (DNN) show a lot of promise for enhancing IoT security [17]. Considering how much data is produced by IoT devices, it is noteworthy that the data typically follows a consistent pattern. Any departure from this standard can be considered abnormal. One effective method for identifying and categorising such anomalous data is a well-trained DNN [18]. Building on this idea, our proposal focuses on developing an Internet of Things (IoT) intrusion detection system that uses deep neural networks to analyse anomalous data in real time. The primary goal of this study is to create a communication protocol-independent, instantaneous, DNN-based IDS for the IoT with the aim of promoting safer and more secure IoT settings.

Nevertheless, a great deal of work in the area of anomaly detection in the IoT using DML approaches remains unfinished. The issues include lowering the false negative rate, identifying crucial features for hostile attack detection, and extracting novel features from data. A DNN with mixed levels, termed DNN_IDS, is proposed as a solution to the aforementioned difficulties. The identification of anomalies in the IoT can be enhanced with the help of this network, which is designed to efficiently extract features. We have also created an ICM binaries variant of the MOICM to address the feature selection issue. A new method, IDS_DNNICM, has been developed by combining these two models; it successfully overcomes the problems and challenges mentioned.

In this study, creating a DNN that can more accurately detect anomalies in the IoT by obtaining characteristics from both minor data and higher-level data using a mixture of convolution layer techniques and Combining the method of RF with the DNN method to improve the validation material of IoT anomaly identification even further.

2. Methodology

2.1. Deep Learning in Intrusion Detection

The ANN's underlying principle is based on how the human brain functions biologically [19]. As nonlinear discriminating functions, neural networks (NN) can generate a variety of choices regarding boundaries in the space of features [20]. A lot of attention has been paid recently to using deep neural networks (DNN), which are an evolution of the shallow neural network (SNN), for intrusion detection. With its exceptional modelling and abstracting representation capabilities, DNN can mimic extremely complicated models. This opens up exciting possibilities for more accurate data representation and the creation of workable solutions. In our previous work [21], we compared six different machine learning algorithms, including supervised, semi-supervised, and unsupervised learning. We came to the conclusion that using DNN for the suggested approach is appropriate. A DNN is a type of NN that uses activation functions and weights applied to connections to produce its outputs [22]. We offer a method that improves classification accuracy by training the DNN with the NSLKDD dataset.

Figure 1 shows the system model, which describes the study's framework and, using a functional block diagram, shows how the whole system works. In this configuration, the Linux operating system is placed strategically between the gateway router and the organisation's network.

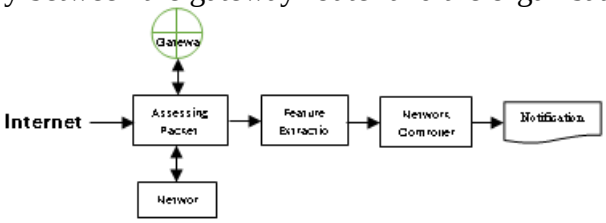

Figure 1. Architecture Diagram

Linux uses a packet sniffing technique to keep tabs on network traffic and analyse the results. After that, using a specialised technique, the data that has been sniffed is processed to extract pertinent features. After the features are extracted, the Linux operating environment's manager organises them into a list and sends it over the internet as a request via HTTP to the endpoint of the API backend. Data is retrieved from HTTP requests by the API backend controller and fed into the ML pipeline, which uses two main components: feature scaling to standardise the dataset and categorical data encoding to convert numerical values from categorical data. Following the completion of data pre-processing, a previously trained DNN architecture is fed the pre-processed data. After that, in response to the original HTTP request, the API background controller returns the prediction. As a last step, the Linux operating system controller will alert the network administrator if the HTTP response shows odd behaviour.

The table 1 is representing the dataset and it contains two classes which are train and test data. Totally 155,845 data taken as train data and 25,876 data taken for test purpose.

Table 1. Data set classification

	Non attack data	Denial-of-Service (DoS)	Remote to user (R2L)	User-to-Root (U2R)	Total
Train	85433	15644	23005	31763	155845
Test Data	9342	4535	6456	5543	25876

The Deep Neural Network (DNN) was constructed using the open-source software framework Keras, which is specifically built for developing ANNs. The built DNN has 16 layers (not including the output layer), and the hidden layers of each layer have different numbers of neurons. There is a plethora of essential hyper parameters that determine how well the final model works. Among these are the optimizer, learning rate, regularisation coefficient, activation function, number of neurons, value initializer, bias initializer, and quantity of hidden layers.

All hidden layers used the Rectified Linear Unit (ReLU) activating function, as described in Equation 1. When the input is positive, the activation function of ReLU directly produces the output; otherwise, it produces zero. This operation is similar to a piecewise linear function. This mechanism activates a node that is a corrected linear stimulation unit [23].

$$ReLU(s) = M(0,s) \quad (1)$$

The sigmoid function was used to turn on the return value layers, which transferred any real number to the (0,1) range. This particular function is used to transform the output of the DNN networks into a likelihood score. Equation 2 displays the sigmoid equation mathematically.

$$\Sigma(s) = \frac{1}{1+x^{-s}} \quad (2)$$

The set 0,1 is used for binary categorization jobs when the goal values fall underneath it. Using cross-validation, the deep neural network was trained through 100 epochs and then evaluated on untrained data. By comparing the initial training and a cross-valid accuracy curves, the 'earlier stopping' technique can identify underfitting and lessen the likelihood of overfitting. ML algorithms require hyperparameter modification for optimal performance. With the Keras Tuner module, values were fine-tuned for parameters including training rate, normalisation factor, the overall number of hidden layers, the number of neurons per concealed layer, and complete number of hidden layers. After training is complete, the DNN ML model is written to a JSON file for transport and storage purposes. On top of that, the model's weights are saved in an H5 file, which follows the HDF. Although the ML model is saved for future prediction purposes, there is a chance that data input dimension mismatches can halt the prediction process. As a result, using the same format for preprocessing data as training is essential.

2.3. Feature Extraction – Real Time

2.3.1 Configuration

The first step in executing instantaneous feature extraction is setting up the network. A Linux station with two interfaces for networks has been installed across the router that serves as the LAN to monitor packets of information as they go in and out of the network. This was accomplished by effectively bridging the two network ports on the Linux computer, which allowed it to intercept data packets going in and out of the workstation. Here, the C++ programming language serves as the foundation for the packet sniffing method.

2.3.2. Feature Extraction

The feature extraction process made use of the well-known and resource-efficient packet sniffing technology. In a network with two or more interfaces, this strategy can be applied to any machine. The LIBPCAP package and the C++ programming language were used to code the packet-capturing system. Key to this process is the LIBPCAP package, which offers an API for collecting data from networks at an advanced level. Using a text buffer, which is the programming language of C++ software, captures the packets. Not only that, but ICMP, TCP, and UDP packets were all taken care of by a newly-made packet analysis function.

2.4. Cost Analysis

As the amount of data increases, the effectiveness of deep learning (DL) algorithms also increases. Traditional machine learning (ML) approaches, on the other hand, often perform worse as data volumes increase. When faced with a large dataset with more than 100,000 instances, DL outperforms many ML algorithms, demonstrating its clear advantage in solving complicated issues. It should be noted, though, that DL implementation requires a lot of computing power in comparison to traditional ML methods, which means training times are much longer. The software and hardware used to train ML algorithms are listed in Table 6 below. The suggested deep neural network (DNN) took around 30 minutes to finish training according to the given criteria. The suggested DNN also has a lengthy hyper parameter tuning process because of all the linked hyper parameters.

Table 2. NSL-KDD data set classifications

Datasets	Traffics	Training	Testing
NSL-KDD	**Non attack Data**	75434	102322
	Attacked Data	64345	23564
TON-IoT	**Non attack Data**	500000	
	Attacked Data	534765	

Two datasets, NSL-KDD and TONNE-IoT, are used to exanimated the suggested technique. Below, we outline the characteristics of each dataset, which are organised into two types, and each dataset contains samples. There are two parts to the NSL-KDD dataset, KDD Train and KDD Test, which

together contain 139,779 samples. Table 2 details the two classes that make up the NSL-KDD dataset: normal samples and abnormal samples. The features that make up dataset number 41.
In contrast, the TONNE-IoT dataset includes operating systems and network traffic, and it is a next-generation collection of IoT/IIoT statistics. There are 1034765 records in this collection, including both normal and attack data. To be more precise, 461,043 records were culled from IoT traffic, with 534765 records representing attack traffic and 500,000 records representing regular traffic. Also, the TONNE-IoT dataset contains 88 attributes and includes 16 categories of attacks, including normal and aberrant samples.

2.5. Pre-processing

In this section, we will go over the main techniques used to prepare the NSL-KDD and TONNE-IoT datasets for analysis. Data cleaning is the first step in pre-processing; it entails removing any noisy data found while sampling the dataset. The second part of the pre-processing stage is filling in missing data with suitable alternatives. For numerical features, the average value is used for replenishment, whereas for string or batch data, the most frequent technique is used. The third stage is to use the label-encoder method to transform string or batch characteristics into their respective numerical kinds. In the last stage, data normalisation is addressed. All features are adjusted to fall within the range of [0, 1] using the Min-Max normalisation approach, as shown in Equation 3:

$$S_{nm} = \frac{S - S_{min}}{S_{max} - S_{min}} \qquad (3)$$

Each feature's lowest value, S_{min}, is represented by the variable S in Equation (12), while the highest value, S_{max}, is denoted by the variable S. Both datasets are prepared to be used with ML and DL techniques after the preparation procedures are finished.

3. Results and Discussion

The investigation of the proposed methods has proven that DNN-IDS can classify and extract features from anomaly detection datasets. As a reliable method for anomaly detection, DNN-IDS as proven its worth.

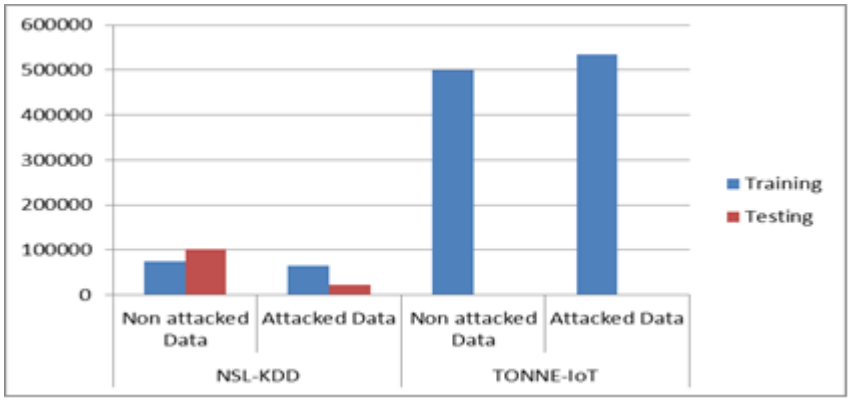

Figure 2. Comparison of dataset volumes

Figure 2 displays a summary of the results achieved from applying the DL method, DNN_IDS methodology, to the NSL-KDD and TONNE-IoT sample datasets.
As seen in the curves that follow, the following metrics were assessed as part of the DNN model's effectiveness evaluation: recall, precision, accuracy, the F1-s, and the confusion matrix (CM). Performance analysis in the presence of skewed class distributions is when these measures really shine. The formulas for recall, accuracy, precision, and F1-score are given in Equations 4 to 7 below.

$$\text{Accuracy} = \frac{TP+TN}{FP+TP+TN+FN} \qquad (4)$$

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (5)$$

$$\text{Recall} = \frac{TP}{FN+FP} \qquad (6)$$

$$\text{F1} = 2 * \frac{Precision*Recall}{Precision+Recall} \qquad (7)$$

Table 3. Performance comparison of proposed and existing methods.

Techniques	Accuracy	Precision	Recall	F1-score

XGBoost-SMOTE	0.9243	0.9800	0.8970	0.92370
SVM	0.8763	0.9324	0.7564	0.8376
K-Means	0.8543	0.9345	0.6453	0.7864
NB	0.7413	0.6098	0.7144	0.6577
OCSVM	0.7354	0.9543	0.6543	0.7864
DNN	0.8976	0.9863	0.8093	0.8923

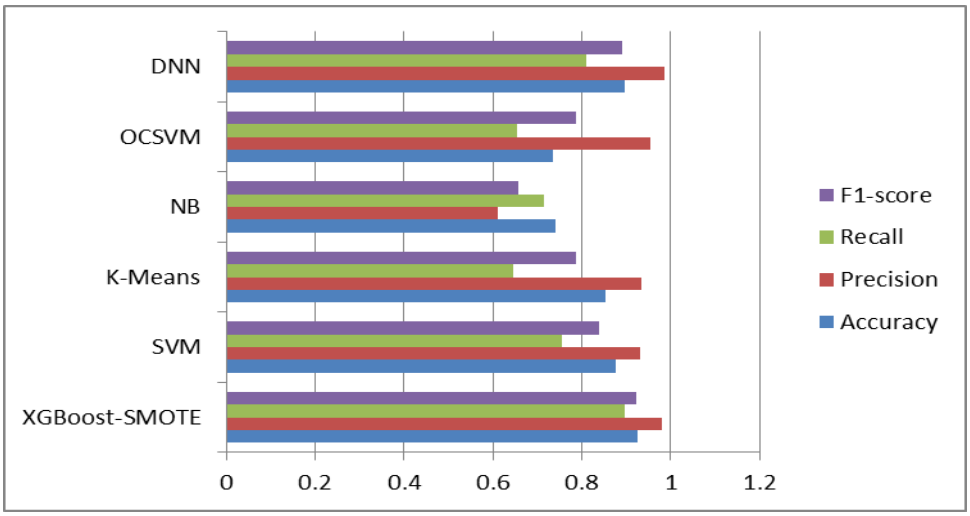

Figure 4. Performance comparison of proposed and existing methods

On the datasets NSL-KDD and TONNE-IoT the DNN techniques, according to their proposals, demonstrated rather strong performance with accuracies of 0.8976, respectively. Because of its efficient feature selection, feature extraction, and RF algorithm-based classification, DNN_IDS stood out with an impressive accuracy of 0.8976. When the F1-score was higher, techniques showed remarkable outcomes, but they performed well across the board. The accuracy, precision, recall, and F1-score for the training and testing processes with the NSL-KDD and TON-IoT datasets are shown in Table 3.

4. Conclusion

A lot of research has been done on the suggested methods, especially on DNN-IDS, and it is clear that they work for sorting and extracting features from datasets for anomaly detection. The study showcases DNN_IDS outstanding performance on the NSL-KDD and TONNE-IoT datasets, revealing it as a standout and trustworthy solution for anomaly detection. The experimental results showed that the DNN procedure performed well, with DNN reaching an accuracy of 0.8976, in line with their respective methodologies. The remarkable accuracy of 0.8976 achieved by DNN_IDS is largely attributable to its implementation of the RF (random forest) method for classification as well as its efficient feature selection and feature extraction processes. By looking at several measures like recall, accuracy, precision, and F1-score, we can see that DNN-IDS did a great job. An in-depth analysis of these methods' consistent and impressive successes is shown in Table 3. This analysis was carried out during the training and testing phases using the NSL-KDD and TONNE-IoT datasets. For the NSL-KDD and TONNE-IoT datasets in particular, the results show that DNN-IDS are effective in anomaly detection. Not only are these approaches quite accurate, but they also perform well across a variety of criteria, suggesting they could be useful tools for anomaly recognition and intrusion detection. The research adds significantly to the continuing discussion on efficient methods for strengthening cybersecurity defences in response to new and changing threats.

5. References

1. Apostolos Gerodimos, Leandros Maglaras, Mohamed Amine Ferrag, Nick Ayres, Ioanna Kantzavelou, IoT: Communication protocols and security threats, Internet of Things and Cyber-Physical Systems, Volume 3, 2023;1-13, ISSN 2667-3452. Available from:

https://doi.org/10.1016/j.iotcps.2022.12.003.
(https://www.sciencedirect.com/science/article/pii/S2667345222000293).
2. Mukhtar, B.I.; Elsayed, M.S.; Jurcut, A.D.; Azer, M.A. IoT Vulnerabilities and Attacks: SILEX Malware Case Study. Symmetry, 2023. Available from: https://doi.org/10.3390/sym15111978
3. Hussain, Muhammad Zunnurain, and Zurina Mohd Hanapi, "Efficient Secure Routing Mechanisms for the Low-Powered IoT Network: A Literature Review" Electronics 2023;12, 3: 482. Available from: https://doi.org/10.3390/electronics12030482
4. A. K. Mishra, D. Puthal and A. K. Tripathy, "A Secure RPL Rank Computation and Distribution Mechanism for Preventing Sinkhole Attack in IoT-based Systems," IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hoboken, NJ, USA, 2023;1-6, Available from: https://doi: 10.1109/INFOCOMWKSHPS57453.2023.10225895
5. Alazab, A.; Khraisat, A.; Singh, S.; Bevinakoppa, S.; Mahdi, O.A. Routing Attacks Detection in 6LoWPAN-Based Internet of Things. Electronics, 2023;12;1320. Available from: https://doi.org/10.3390/electronics12061320
6. Amiri, Z., Heidari, A., Navimipour, N.J. et al. Adventures in data analysis: a systematic review of Deep Learning techniques for pattern recognition in cyber-physical-social systems. Multimedia Tools Applications;22909-22973, 2023. Available from: https://doi.org/10.1007/s11042-023-16382-x
7. Bhavsar, M., Roy, K., Kelly, J. et al. Anomaly-based intrusion detection system for IoT application. Discov Internet Things, 2023;3;5. Available from: https://doi.org/10.1007/s43926-023-00034-5
8. Abed AK, Anupam A. Review of security issues in Internet of Things and artificial intelligence-driven solutions. Security and Privacy. 2023;6(3):e285. Available from: https://doi:10.1002/spy2.285.
9. Xiaoya Xu, Yunpeng Wang, Pengcheng Wang, "Comprehensive Review on Misbehavior Detection for Vehicular Ad Hoc Networks", Journal of Advanced Transportation, vol. 2022, Article ID 4725805, 27 pages, 2022. Available from: https://doi.org/10.1155/2022/4725805
10. O. Altay, Chaotic slime Mould optimization algorithm for global optimization, Artif. Intell. Rev. 55(5), 2022;3979:4040. Available from: https://doi.org/10.1007/s10462-021-10100-5
11. C. Bulla, M.N. Birje, Anomaly detection in industrial IoT applications using deep learning approach, in: Artificial Intelligence in Industrial Applications, Springer, 2022;127:147. Available from: http://dx.doi.org/10.1007/978-3-030-85383-9_9
12. T. Saba, A. Rehman, T. Sadad, H. Kolivand, S.A. Bahaj, Anomaly-based intru-sion detection system for IoT networks through deep learning model, Comput. Electr. Eng. 99 2022;107810. Available from: https://doi.org/10.1016/j.compeleceng.2022.107810
13. F. Hussain, R. Hussain, S. A. Hassan and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," in IEEE Communications Surveys & Tutorials, 2022;22,1686:1721. Available from: https://doi: 10.1109/COMST.2020.2986444
14. N. Balakrishnan, A. Rajendran, D. Pelusi, V. Ponnusamy, Deep Belief Network enhanced intrusion detection system to prevent security breach in the Internet of Things, Int. Things 14, 2021;100112. Available from: https://doi.org/10.1016/j.iot.2019.100112
15. M. Braik, A hybrid multi-gene genetic programming with capuchin search algorithm for modeling a nonlinear challenge problem: modeling industrial winding process, case study, Neural Process. Lett. 53(4), 2021;2873:2916. Available from: https://doi.org/10.1007/s11063-021-10530-w
16. L. Breiman, Random Forests, Machine Learning, 45(1) 2001;5-32. Available from: https://doi.org/10.1023/A:1010933404324
17. Elnakib, O., Shaaban, E., Mahmoud, M. et al. EIDM: deep learning model for IoT intrusion detection systems. J Supercomput 79, 2023;13241–13261. Available from: https://doi.org/10.1007/s11227-023-05197-0
18. P. Pathak, D. Singh, A. Saxena, K. Kumar, S. Singh Dari and D. Dhabliya, "Enhancing Cyber-Physical System Security with CGAN in Fog Environment," 2023 International Conference on Data Science and Network Security (ICDSNS), Tiptur, India, 2023;01-08, Available from: doi: 10.1109/ICDSNS58469.2023.10245323

19. Pikus, M.; Wąs, J. Using Deep Neural Network Methods for Forecasting Energy Productivity Based on Comparison of Simulation and DNN Results for Central Poland—Swietokrzyskie Voivodeship. Energies 2023;16,6632. Available from: https://doi.org/10.3390/en16186632
20. Mishra, V., Kane, L. A survey of designing convolutional neural network using evolutionary algorithms. Artif Intell Rev, 2023;56,5095:5132. Available from: https://doi.org/10.1007/s10462-022-10303-4
21. Tzu-Hao Harry Chao et al. ,Neuronal dynamics of the default mode network and anterior insular cortex: Intrinsic properties and modulation by salient stimuli.Sci. Adv.9,eade5732, 2023. Available from: https://doi:10.1126/sciadv.ade5732
22. Abed AK, Anupam A. Review of security issues in Internet of Things and artificial intelligence-driven solutions. Security and Privacy. 2023;6(3):e285. Available from: https://doi:10.1002/spy2.285
23. V. Thamilarasi, P. K. Naik, I. Sharma, V. Porkodi, M. Sivaram and M. Lawanyashri, "Quantum Computing - Navigating the Frontier with Shor's Algorithm and Quantum Cryptography," 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies, Pune, India, 2024, pp. 1-5, doi: 10.1109/TQCEBT59414.2024.10545283.
24. Xiaoya Xu, Yunpeng Wang, Pengcheng Wang, "Comprehensive Review on Misbehavior Detection for Vehicular Ad Hoc Networks", Journal of Advanced Transportation, vol. 2022, Article ID 4725805, 27 pages, 2022. Available from: https://doi.org/10.1155/2022/4725805