https://dx.doi.org/10.4314/dujopas.v10i2c.7
ISSN (Print): 2476-8316
ISSN (Online): 2635-3490
Dutse Journal of Pure and Applied Sciences (DUJOPAS), *Vol. 10 No. 2c June 2024*

Improving Intrusion Detection System Accuracy Using Deep Neural Network

[1]Abdullahi Ya'u Gambo, [2]Farouk Lawan Gambo, [2]Aminu Aliyu Abdullahi, [1]Nasima Ibrahim, [1]Yusuf Isyaku Maitama, [3]Zahrau Ahmad Zakari

[1]Department of Computer Science
Kano State Polytechnic
Kano,
Nigeria.

[2]Department of Computer Science
Federal University Dutse
Dutse,
Nigeria.

[3]Department of Computer Science
Sa'adatu Rimi University of Education
Kano,
Nigeria.

Email: aygambo@gmail.com

Abstract

Internet of Things (IoT) has emerged as an intelligent network that connects objects to the Internet, allowing them to interact with each other without human intervention. The accessibility of IoT devices through unprotected networks subjected them to security vulnerabilities and various malicious attacks. While traditional Intrusion Detection Systems were introduced to address IoT security issues, there is need for intelligent intrusion detection methods. This study attempts to address and mitigate these security challenges by enhancing the performance and efficiency of IDSs with proposed Deep Neural Network (DNN) model. The study use a Deep Neural Network (DNN) and processed IoTID20 datasets for the detection of intrusion. The performance of the system is evaluated using performance metrics; Accuracy, Precision, recall and F1-Score. The optimal result accuracy obtained is 99.04%. The proposed model has demonstrated a potential improvement of Intrusion Detection Systems.

Keywords-Deep Neural Network, Internet of Things, Intrusion Detection Systems, IoTID20 dataset, Machine Learning

INTRODUCTION

Internet-of-Things (IoT) has emerged as an intelligent network that connects objects to the Internet, allowing them to interact with each other without human intervention (Chen, Xu, Liu, Hu, & Wang, 2014). This network employs smart sensors that wirelessly connect to various objects, enabling the creation of IoT devices and applications in domains such as smart environments, smart homes, and smart cities (Li, Xu, & Zhao, 2015; Sherasiya & Upadhyay, 2016). However, as IoT applications grow, significant challenges have arisen, with security being a paramount concern. Due to the accessibility of IoT devices through unprotected networks like the Internet, they are vulnerable to various malicious attacks.

Author for Correspondence

In addressing the issue of IoT security, researchers have implemented Intrusion Detection Systems (IDSs) as an additional layer of defense (Hajar, Al-Kadri, & Kalutarage, 2021). IDSs monitor network traffic to identify malicious or attack-related activities, providing timely alerts (Otoum & Nayak, 2021). These systems are crucial in safeguarding against both internal and external threats. While traditional IDSs have been used, recent studies have explored the integration of machine learning (ML) techniques to enhance their performance. Conventional ML techniques, like the Classification and Regression Tree (CART), offer improved accuracy but have limitations when dealing with complex datasets. Shallow learning approaches, commonly used in traditional detection methods, require extensive feature engineering and selection and can struggle with intricate datasets (Thapa, Liu, Kc, Gokaraju, & Roy, 2020). Deep learning, a subset of ML, presents a promising solution by utilizing multi-layered neural networks to automatically learn hierarchical representations of features from extensive datasets. Compared to traditional machine learning classifiers, they exhibit superior performance, particularly in capturing abstract and high-dimensional features crucial for identifying potential threats.

Dhillon & Haque, (2020) propose a novel approach to address the limitations of conventional network intrusion detection systems (NIDSs) in handling the immense volume of network traffic and the unpredictability of real-world scenarios. Their method leverages deep transfer learning, amalgamating various deep neural network architectures like DNNs, convolutional neural networks (CNNs), and long short-term memory (LSTM) models. By harnessing prior knowledge from models trained on extensive datasets, their approach achieves remarkable accuracy and speed, even in resource-constrained environments, as demonstrated on the UNSW-15 dataset.

Moreover, deep learning techniques have shown promising results in bolstering IoT security. (Jose & Jose, 2021) explore the application of deep learning algorithms for intrusion detection in IoT devices, emphasizing the criticality of security measures in the data-driven IoT landscape. Through experimentation with different deep learning models such as feed-forward artificial neural networks (ANNs), autoencoders (AEs), deep belief networks (DBNs), and LSTMs, they highlight the potential of deep learning in predicting and thwarting attacks on IoT networks. Similarly, (Khan et al., 2021) present a deep learning-based IDS tailored for MQTT-enabled IoT networks, showcasing its superiority over classical machine learning models and state-of-the-art deep learning techniques in detecting intrusions within MQTT data packets.

Additionally, researchers like (Shareena, Ramdas, & AP, 2021) focus on combating specific threats, such as IoT distributed denial-of-service (DDoS) botnet attacks, through deep learning-based IDSs. By leveraging realistic datasets and sophisticated deep neural network architectures, they achieve remarkable accuracy and precision in detecting malicious activities within IoT environments. Furthermore, studies like that of (Alzughaibi & El Khediri, 2023; Jose & Jose, 2023) continue to explore the potential of deep learning techniques in enhancing intrusion detection accuracy and robustness, particularly in resource-constrained IoT settings and cloud environments. These endeavors underscore the growing significance of deep learning in fortifying cybersecurity measures against evolving cyber threats. To the best of our knowledge, our model achieved higher accuracy as compared to the recent studies (Alzughaibi & El Khediri, 2023; Jose & Jose, 2023).

METHODOLOGY

The stages that are involve include data acquisition, data processing, model development and evaluation.

A. Data Acquisition

The IoTID20 dataset's testbed is a composite setup comprising a variety of IoT devices and interconnected elements. To create the IoTID20 dataset, a typical smart home environment was emulated, featuring the inclusion of the SKT NGU smart home device and an EZVIZ Wi-Fi camera. These two IoT devices were linked to a smart home Wi-Fi router. Additionally, other devices such as laptops, tablets, and smartphones were connected to this same smart home router. Notably, the SKT NGU and EZVIZ Wi-Fi camera functioned as IoT victim devices, while all other devices within the testbed assumed the role of attacking devices. The IoTID20 dataset was developed by adapting Pcap files accessible on the website (Ullah & Mahmoud, 2020).

The process involved the utilization of the CICflowmeter application (Lashkari, Gil, Mamun, & Ghorbani, 2017) to extract features from Pcap files and convert them into CSV format, resulting in the creation of the IoTID20 dataset. Subsequently, the next phase encompassed assigning labels to each instance within the IoTID20 dataset. This dataset comprises a total of 80 network features, complemented by three distinct label features. These label features encompass binary, category, and sub-category designations, which provides an overview of the binary, category, and sub-category labels employed within the IoTID20 dataset.

B. Data Preprocessing

The first step in building the proposed model is by pre-processing the IoTID20 dataset. The dataset comprises of 625,783 instances with 86 distinct features. After dropping rows with missing values, the number of instances reduced to 625415. The number of features becomes 83 after creating feature Matrix X and to 73 after handling constant features. And finally to 39 after applying hierarchical clustering (Table 1).

TABLE 1. ***Dataframe shapes during pre-procsseing stages***

S/N	Pre-processing stage	Shape of the Data frame
1	Initial stage	(625,783, 86)
2	Dropna() method	(625,415, 86)
3	Creation of feature Matrix X	(625,415, 83)
4	Handling constant features	(625,415, 73)
5	Hierarchical clustering	(625,415, 39)

C. Model Development

A DNN is an advanced model of the classical Feed Forward Neural network, with multiple hidden layers using the non-linear activation function, ReLU (Gambo et al., 2021). It's utilized in (Vinayakumar et al., 2019) for detecting attacks due to its ability to handle the continuous change in network behaviour and rapid evolution of attacks.
The proposed DNN model is created using the Sequential class from tensorflow.keras.models. The model consists of a series of layers as follows:

Input Layer: The input layer is defined using the Flatten layer. This layer is used to flatten the input data, which has the shape (1, 38), to a one-dimensional array.

Hidden Layers: The DNN has multiple hidden layers, each defined using the dense layer with 64 units. These layers apply a linear transformation followed by an activation function to

produce the output. The activation function used is the rectified linear unit (ReLU) activation function.

Dropout: This layer applies dropout regularization with a rate of (ranges 0.2 to 0.4), which helps prevent overfitting. These layers (Dense and Dropout) are repeated twice to add more hidden layers to the model.

Output Layer: The output layer is defined using a dense layer with a single unit and a sigmoid activation function. This is suitable for binary classification tasks where the output represents the probability value between 0 and 1.

The number of epochs (iterations) to train the model: In the experiment, the model were trained for 100, 200, and 300 epochs. The number of samples per gradient update. It determines how many samples are processed before the model's weights are updated. In the experiment, the model updated its weights after processing every 5000 samples (Table 2).

D. Model Training and Evaluation

The training process for the DNN model is configured by compiling the model with the specified loss function, optimizer, and metrics. The optimizer adjust the model's weights based on the calculated loss, and monitored the accuracy during training to assess the model's performance.

TABLE 2 ***The Proposed DNN parameters***

S/N	Parameter	Value
1	Hidden Layers	3 Dense Layers with three Dropout layers inserted in between
2	No of Neurons	38 for input layer, 64 for each layer hidden layer and 1 for output layer
3	No of features	38
4	No of classes	1 neuron for Binary Classification
5	Loss function	BinaryCrossEntropy
6	Optimization Algorithm	Adam
7	Batch size	5000
8	Epochs	100 to 300
9	Training-testing ration	70%(samples)-30%(samples)

Loss Function: The loss function is set to 'binary cross entropy'. This is a commonly used loss function for binary classification problems (Gambo et al., 2021; Gambo, Wajiga, Garba, & Aliyu, 2021). It measures the difference between the predicted probabilities and the true binary labels, and it is optimized to minimize this difference during training.

Optimizer: The optimizer is set to 'Adam'. Adam (Adaptive Moment Estimation) is an optimization algorithm that is widely used for training neural networks. It adapts the learning rate based on the gradients of the model parameters, which helps in faster convergence and better performance. It combines the advantages of two other optimization methods, AdaGrad and RMSProp, to provide efficient and adaptive gradient-based optimization

Metrics: The metrics parameter is set to 'accuracy'. This specifies that during training and evaluation, the model's performance will be evaluated based on the accuracy metric. Accuracy measures the proportion of correctly classified samples out of the total number of samples.

RESULTS AND DISCUSSION

The result presented here is based on the optimal performance achieved by the DNN model. The optimal result achieved through processing IoTID20 data with DNN algorithm. The DNN model layers used is summary presented in table 3.

TABLE 3 *Summary of the proosed model*

Layer (type)	Output shape	Parameter No
Flatten_1 (Flatten)	(None, 38)	0
Dense_1 (Dense)	(None, 64)	2496
Dropout_1 (Dropout)	(None, 64)	0
Dense_2 (Dense)	(None, 64)	4160
Dropout_2 (Dropout)	(None, 64)	0
Dense_3 (Dense)	(None, 64)	4160
Dropout_3 (Dropout)	(None, 64)	0
Dense_3 (Dense)	(None, 1)	65
Total parameters: 10,881 Trainable parameters: 10,881 Non-trainable parameters: 0		

Using dropout of 0.4 the accuracy were around 98.88 to 98.92. It started to converge from 98.89 to 98.92 when the number of epoch dropped from 300 to 100. And using dropout of 0.3 the accuracy were around 98.85 to 99.2. It started to converge when the epoch was increased from 100 to 300. The highest accuracy recorded was at dropout of 0.2 and 300 epoch.

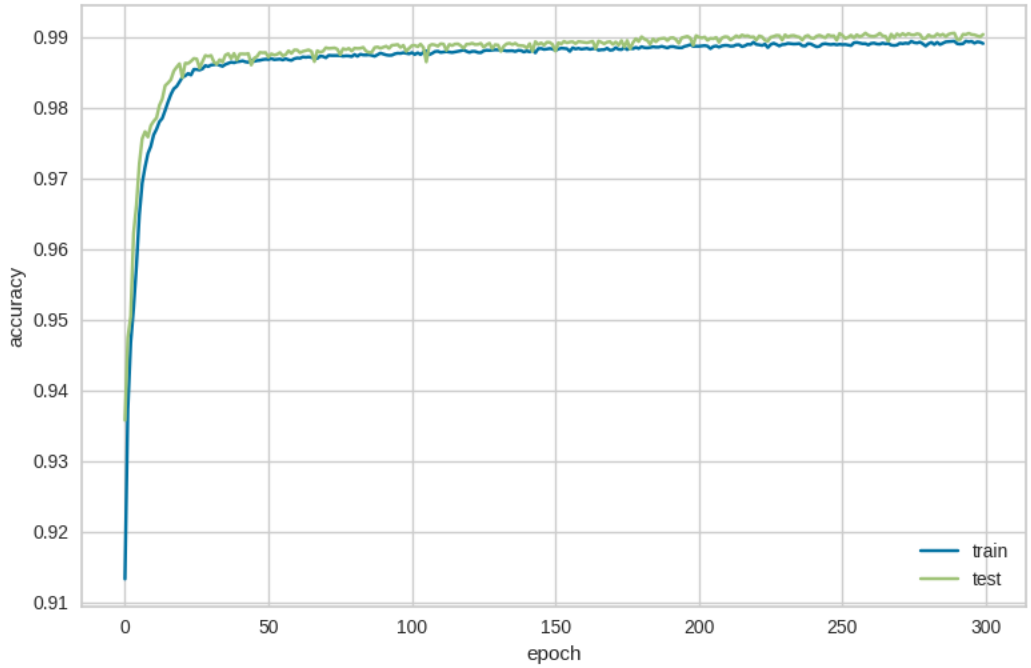

Figure 1. Accuracy of 99.04 at 0.2 dropout and 300 epoch

The proposed DNN Model achieved high accuracy and other evaluation metrics, surpassing the results reported in the literature. For example, the proposed model achieved an accuracy of 98.85% to 99.06% under different dropout and epoch values, as shown in Table 4. This is higher than the accuracy reported in other studies, such as (Shareena et al., 2021) which is (94.00%), (97.13%) in (Khan et al., 2021), (94.61%) in (Jose & Jose, 2023) and (98.97%) in (Alzughaibi & El Khediri, 2023). The accuracy 99.04 is recorded at dropout of 0.2 and 300

epoch, the precision 99.1%, recall of 99.9% more than 98.8% and F1 score of 99.5% more than 99.38% as recoded in (Alzughaibi & El Khediri, 2023). See figure 1 for the result of accuracy 300 epoch and 0.2 dropout

TABLE 4 ***DNN algorithm in the literature and the proposed model***

S/N	Dataset used	Accuracy	References
1	UNSW-NB	88.00	Dhillon & Haque, (2020)
2	KDD Cup 99, CIC IDS-2017	92.50	Vinayakumar et al., (2019)
3	BoT-IoT	94.00	Shareena et al.,(2021)
4	CIC IDS-2017	94.61	Jose & Jose, (2023)
5	MQTT-IoT-IDS2020	97.13	Khan et al., (2021)
6	CSE-CIC-IDS2018	98.97	Alzughaibi & El Khediri, (2023)
7	**IoTID20**	**98.85-99.06 (Under different dropout and epoch values)**	**Proposed Model**

CONCLUSIONS

The study underscores the dual nature of IoT networks, offering both convenience and posing significant security challenges. It emphasizes the imperative of employing intelligent intrusion detection techniques to safeguard these networks effectively. Notably, the research highlights the pivotal role of selecting appropriate datasets in intrusion detection studies. Evaluation reveals that the proposed DNN algorithm outperforms existing literature, exhibiting higher accuracy rates. The applicability of this algorithm extends beyond the current study's dataset, provided that the new dataset shares similar features and characteristics. Nonetheless, potential limitations arise, including dependency on dataset size and quality, class distribution, and attack complexity. Additionally, real-time intrusion detection systems, necessitating low latency and high throughput, may pose challenges for the proposed model. Thus, further exploration is warranted to assess its adaptability and scalability across diverse intrusion detection scenarios.

As a recommendation, it is essential to continue refining intelligent intrusion detection methods to address evolving IoT security concerns effectively. Researchers should prioritize dataset curation, ensuring datasets accurately reflect real-world scenarios for robust model training and evaluation. Moreover, efforts should concentrate on mitigating the identified limitations, including dataset dependencies and real-time applicability challenges. Future research endeavors should focus on enhancing the proposed DNN algorithm's generalizability and scalability across various intrusion detection contexts. Collaborative initiatives between academia, industry, and policymakers are vital to foster innovation and develop practical solutions that bolster IoT network security.

REFERENCES

Alzughaibi, S., & El Khediri, S. (2023). A Cloud Intrusion Detection Systems Based on DNN Using Backpropagation and PSO on the CSE-CIC-IDS2018 Dataset. *applied sciences, 13*(4), 2276. DOI:10.3390/app13042276

Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things journal, 1*(4), 349-359. DOI:10.1109/JIOT.2014.2337336

Dhillon, H., & Haque, A. (2020). *Towards network traffic monitoring using deep transfer learning.* Paper presented at the 2020 IEEE 19th International Conference on Trust, Security and

Privacy in Computing and Communications (TrustCom), 1089-1096. DOI: 10.1109/TrustCom50675.2020.00144

Gambo, F., Wajiga, G. M., Shuib, L., Garba, E. J., Abdullahi, A. A., & Bisandu, D. B. (2021). Performance Comparison of Convolutional and Multiclass Neural Network for Learning Style Detection from Facial Images. *EAI Endorsed Transactions on Scalable Information Systems, 9*(35). DOI:10.4108/eai.20-10-2021.171549

Gambo, F. L., Wajiga, G. M., Garba, E. J., & Aliyu, A. (2021). A deep learning solution for learning style detection using cognitive-affective features. *Computer Science, 3*(1), 32-43.

Hajar, M. S., Al-Kadri, M. O., & Kalutarage, H. K. (2021). A survey on wireless body area networks: Architecture, security challenges and research opportunities. *Computers & Security, 104*, 102211. DOI:10.1016/j.cose.2021.102211

Jose, J., & Jose, D. V. (2021). *Performance analysis of deep learning algorithms for intrusion detection in IoT.* Paper presented at the 2021 International Conference on Communication, Control and Information Sciences (ICCISc). DOI:10.1109/ICCISc52257.2021.9484979

Jose, J., & Jose, D. V. (2023). Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset. *International Journal of Electrical and Computer Engineering, 13*(1), 1134. DOI: 10.11591/ijece.v13i1.pp1134-1141

Khan, M. A., Khan, M. A., Jan, S. U., Ahmad, J., Jamal, S. S., Shah, A. A., . . . Buchanan, W. J. (2021). A deep learning-based intrusion detection system for mqtt enabled iot. *Sensors, 21*(21), 7016. DOI:10.3390/s21217016

Lashkari, A. H., Gil, G. D., Mamun, M. S. I., & Ghorbani, A. A. (2017). *Characterization of tor traffic using time based features.* Paper presented at the International Conference on Information Systems Security and Privacy, 2017. SciTePress, 253-262. DOI:10.5220/0006105602530262

Li, S., Xu, L. D., & Zhao, S. (2015). The internet of things: a survey. *Information systems frontiers, 17*, 243-259. DOI:10.1007/s10796-014-9492-7

Otoum, Y., & Nayak, A. (2021). As-ids: Anomaly and signature based ids for the internet of things. *Journal of Network and Systems Management, 29*, 1-26. DOI:10.1007/s10922-021-09589-6

Shareena, J., Ramdas, A., & AP, H. (2021). Intrusion detection system for iot botnet attacks using deep learning. *SN Computer Science, 2*(3), 205. DOI:10.1007/s42979-021-00516-9

Sherasiya, T., & Upadhyay, H. (2016). Intrusion detection system for internet of things. *Int. J. Adv. Res. Innov. Ideas Educ.(IJARIIE), 2*(3).

Thapa, N., Liu, Z., Kc, D. B., Gokaraju, B., & Roy, K. (2020). Comparison of machine learning and deep learning models for network intrusion detection systems. *Future Internet, 12*(10), 167. DOI:10.3390/fi12100167

Ullah, I., & Mahmoud, Q. H. (2020). *A scheme for generating a dataset for anomalous activity detection in iot networks.* Paper presented at the Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33. DOI:10.1007/978-3-030-47358-7_52

Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7*, 41525-41550. DOI:10.1109/ACCESS.2019.2895334