Journal of Theoretical and Applied Information Technology
15th June 2023. Vol.101. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN EFFICIENT ATTACK DETECTION FOR INTRUSION DETECTION SYSTEM (IDS) IN INTERNET OF MEDICAL THINGS SMART ENVIRONMENT WITH DEEP LEARNING ALGORITHM

FATIMAH SALEEM ABDULKAREEM[1], NOR FAZLIDA MOHD SANI[2]

[1,2]Department of Computer Science, Faculty of Computer Science and Information Technology,

Universiti Putra Malaysia, Serdang 43400, Selangor Malaysia

E-mail: [1]fatimahsaleem94@gmail.com, [2]fazlida@upm.edu.my

ABSTRACT

Recently, the Internet of Things (IoT) has been an invention for the creation of intelligent worlds. IoT is considered a widely recognized implementation that includes intelligent health care, intelligent transport, and intelligent grids. In any technology depending on the IoT model, in which the Internet of Medical Things (IoMT) is an important technique, privacy and secrecy are considered the major problems driven by numerous attacks triggered by intruders. The detection of unknown attacks is one of the main challenges in intrusion detection system (IDS). Researchers have performed multiple typing and detected anomaly traffic methods in the past decades without earlier understanding the attack signature specifically to the IoT environment. Therefore, an intrusion detection method for attacking and detecting anomalies in an IoT system must be enhanced. To achieve this, we measured the performance of three deep learning algorithms for normal and abnormal detection of IDS, and a comparison was made to select the best performance of the deep learning algorithm for detection in IDS, such as RNN, DBN and CNN. The CICIDS2017 dataset was used to analyze the performance of the existing intrusion detection system model. Additionally, the results of the deep learning algorithms will be evaluated using five confusion matrices, namely, accuracy, precision, recall, F1Score, and false-positive rate). It should be noted that the results showed a good average because most of them exceeded 90% of the total confusion matrix for all three deep learning algorithms that have been evaluated.

Keywords: *IoMT, Intrusion Detection, Anomaly Detection, Deep Learning.*

1. INTRODUCTION

The Internet of Things (IoT) comprises with a physical object network which the existence of the Internet does not limit to the personal computer (Pc) network. Still, we can find that the Internet will enter house tools, smartphones, medical devices, people, animals, and almost all the daily life tools by the predicted 50 billion devices by the end of 2020. All these huge equipment and tools have a database that considers one of the most critical security vulnerabilities if any attack attempts to take this data. One of the most popular IoT fields is the Internet of Medical Things (IoMT). IoMT is a combination of medical equipment and software that can be linked to IT systems through networking technologies. It will minimize needless hospital stays and the pressure on healthcare services by linking patients to their doctors and allowing medical records to be shared across a secure network. [1]

According to Frost and Sullivan, the IoMT global market was predicted to be $22.5 billion in 2016, with a forecast annual growth rate of $72.02 billion in 2021. The Internet of Medical Things is made of intelligent instruments, such as wearables and diagnostic and vital tracking appliances, strictly for use in the body, home, or neighborhood healthcare, clinical or hospital settings, related real-time locations, and telehealth other facilities. With the increase in the spread of the Internet of Things, users, and providers of IoT networks have raised many concerns. They face many challenges in securing the IoT environment and prevent data breaches. [2] [3]

2. RESEARCH BACKGROUND

The numerous potential attacks on the part of intruders contribute to privacy and security problems. Thus, an intrusion detection system to attack and locate anomalies in the IoT system is necessary. During this work, they also suggested an

intrusion detection system deep-learning Deep Belief Network (DBN). The performance analysis of the new IDS model with regard to attacks and anomaly detection uses the CICIDS2017 dataset. In terms of accuracy, recall, precision, F1 score, and detection rate, this proposed technique has shown improved results for all metrics.[1]

From [4] [5] suggested implementing the Deep Belief Network intrusion detection system as cyber-attacks are more recurrent and sophisticated. In the complex cyber-threat environment, existing intrusion detection solutions may not be adequate. The IDS framework can boost attack identification and reduce false-positive alerts by using deep learning technologies.

Medical Image one of the very important thing that need to be secure and privacy of the patient data [6]. The datasets that used in this paper is Chest X-Ray. A deep learning-based encryption and decryption network (DLEDNet) is suggested to complete the encryption and decryption of a medical image. The Cycle-GAN network transfers the medical picture from its original domain to the goal domain as the principal instructional network. The algorithm can secure a powerful safety level of medical images and more reliably encrypt and decrypt the image than other cutting-edge medical picture encryption methods. The paper limitation is that the weakness of the GAN network is unstable when applied for computer vision duties.

[7] suggested a technique to ensure that dense random neural grids are thoroughly analyzed to assess the potential for an attack from the metrics extracted on a network to secure Smart cities. In the case of an attack and anomaly class detection, intrusion into DOS, data control, malicious control, scan, tracing, miss-installation, and regularity, the DNN monitoring model, makes the system more empowering.

Several attacker's targeted the IoT Network Infrastructure & Intelligence Protection. [8] ~~They~~ proposed a new TR-IDS intrusion detection frame and add it to a sophisticated random forest algorithm's final classification. The TR-IDS uses both manually designed and payload enhancement functions. It uses two new approaches to NLP, word insert and text CNN, to extract excellent functionality from payloads. The word "installation technology" retains the semantic association between the bytes and lowers its feature, and text-CNN is used for deriving features from any payload. This system detects almost all attacks by the intrusion, BFSSH, and H5-0DoS, although some DDoS attacks are confused with normal traffic.

IoT malware is one of the critical problems that face by the security in IoT environment. This work [9] proposed that the RNN used execution procedure codes (Opcodes) for ARM-based IoT applications. It introduced a method that uses LSTM to hunt IoT malware for its sequence of Opcodes. This paper achieved 98 percent accuracy against training IoT malware. The dataset used in training was limited relative to the actual cyber threats in the world.

[10] suggested that the deep-NNN network be used to identify attacks into the IoT network to detect attacks on the IoT transportation layer. On three recent benchmarking datasets in DNN wireless and wireless network settings, the proposed solution's performance is assessed to provide good performance, particularly the wireless attack detection performance. In addition, the experimental findings show that experiments on the identification of anomalies should involve multiple confirmation approaches and data sets.

Anomaly based IDS using an Artificial Neural Network (ANN) was proposed to enhance the IDS model [3]. The IDS model uses feedforward and backpropagation algorithms, together with several other optimization techniques, to minimize overhead calculation and to keep its high performance. ANN-based IDS model doing parallel and at times better than other IDS models is the ANN-based IDS model (exactness and detection rate). It uses only two parameters (DR and Accuracy).

Based on the review of the current literature, it can be concluded that none of the articles conducted an evaluation of deep-learning algorithms regarding their efficiency in detecting the seven types of attacks, namely, (Benign, DoS/DDoS, Botnet, Brute Force, Web Attack, Infiltration, and Port Scan). Some articles evaluated the accuracy of deep-learning algorithms in detecting one attack, such as DoS, while others evaluated the accuracy of deep-learning algorithms in the detection of two or three attacks at the most. Hence, there is a necessity to conduct more research to fill this current gap found in the literature. Also, it is important to mention that none of the related articles compared the performance of DBN as opposed to CNN. Furthermore, previous articles utilized four metrics

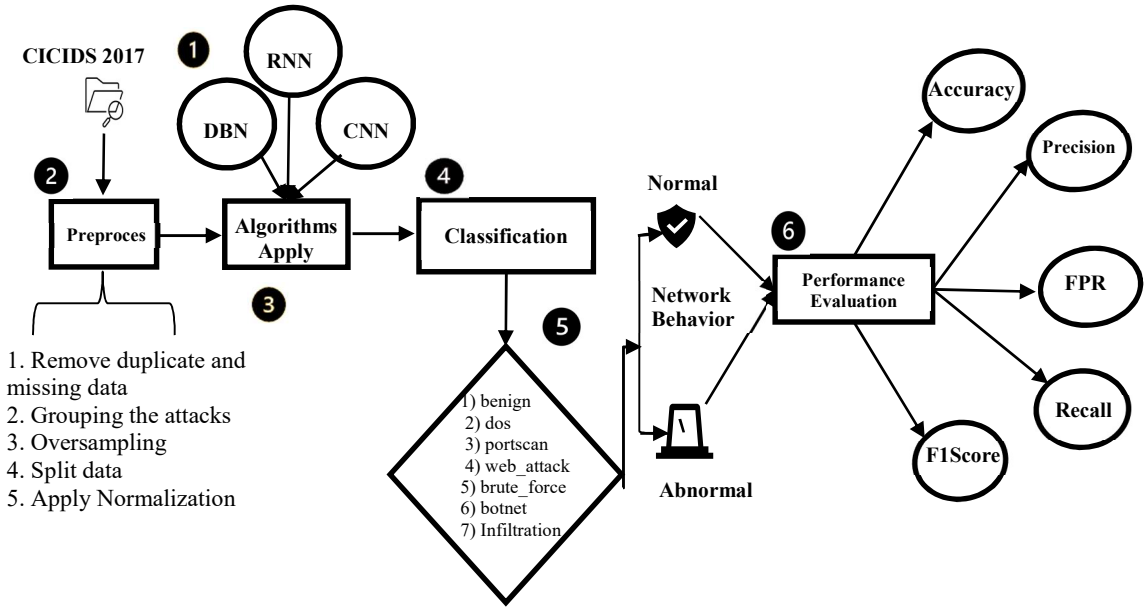

Figure 1: Proposed Methodology

in their evaluation at the most, despite the fact that there are five metrics that can be used as the basis for evaluating the performance of deep-learning algorithms.

3. METHODOLOGY

This work aims to discover many attacks on Internet networks of things. First step is converting the data into an appropriate format to be loaded to the deep learning program (TensorFlow Kit by Keras). After that, the five selected algorithms will be applied in this work (ANN, DNN, RNN, CNN, and DBN), and these algorithms and applying these algorithms in IDS will alert when abnormal data is detected. Later, a comparison will be made between the results of the five algorithms using the five-performance evaluation (accuracy, precision, recall, detection rate, and F1Score) to acknowledge the best available algorithm. Figure 1 show the methodology for this project's work. The method was divided into six steps which are (clarify the dataset, pre-processing, algorithms apply, classification, seven output groups, and performance evaluation) each step had been explained in detail in the sections below.

3.1 CICIDS 2017 Dataset Pre-Processing

This work used the CICIDS2017 (Canadian Institute for Cybersecurity 2017) dataset, a Benchmarking dataset. The dataset type is Multiclass, released in 2017, 2830540 is the total number of the distinct instances the number of the features 79, and the distinct classes number is 15. The presented of the data to understand the content of this dataset. The data took five continuous days (Monday - Friday) of attack traffic data. Table 1 will show all details of the dataset content. As shown in the Table 1 below the dataset has split into eight CSV files, so the first step that done here is to combine these files together. [11]

After reading the dataset, found that it contains various up-to-date multi-stages attacks such as Heartbleed and different types of DoS and DDoS attacks; also, they use Benign: Normal traffic behavior. In this way, can know the effectiveness of the system to detect normal and abnormal behavior. The file uses CSV format, making importing it into machine learning software easy. Table 2 show the instant appearance of the dataset. Moreover, consists of 80.3% normal traffic, with the remaining 19.7% being the fourteen types of attacks.

Table 1: Dataset Description

Files	Day Activity	Attacks
Monday-WorkingHours.pcap_ISCX	Monday	Benign (Normal human activities)
Tuesday-WorkingHours.pcap_ISCX	Tuesday	Benign , FTP-Pastor, SSH-Pastor
Wednesday-workingHours.pcap_ISCX	Wednesday	Benign, DoS Hulk, DoS GoldenEye, DoS slowloris, DoS Slowhttptest, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX	Thursday	Benign, WebAttacks
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX	Thursday	Benign, Infilteration
Friday-WorkingHours-Morning.pcap_ISCX	Friday	Benign
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX	Friday	Benign, PortScan
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX	Friday	Benign, DDos

Table 2: Dataset Content

DATA LABEL	NUMBER OF ENTRIES	DATA PERCENTAGE	DATA TYPE
BENIGN	2271320	**80.3%**	**NORMAL TRAFFIC**
DoS Hulk	230124	**19.7%**	**FOURTEEN TYPES OF ATTACKS**
PortScan	158804		
DDoS	128025		
DoS GoldenEye	10293		
FTP-Pastor	7935		
SSH-Pastor	5897		
DoS slowloris	5796		
DoS Slowhttptest	5499		
Bot	1956		
Brute Force	1,507		
XSS	652		
Infiltration	36		
SQL Injection	21		
Heartbleed	11		

Heartbleed: The attackers utilize the OpenSSL protocol, allowing the unauthorized attacker to access critical information and input malicious material into the OpenSSL memory.

SQL Injection: A SQL injection is a code injection approach for attacking applications that depend on data, including obnoxious SQL proclamations placed in the implementation section.

Infiltration: The assailants use infiltration techniques and software to penetrate the networked system information and acquire total unauthorized access.

XSS: The assailants frequently inject trustworthy websites and innocuous online apps to transmit malicious information.

Brute Force: The assailants try to gain privileged data, for example, by using trial and error, PINs and passwords.
Bot: An assailants employ Trojans to breach the safety of numerous victim computers, take charge of those devices and arrange each system in their Bot network to be utilized and managed remotely by the attackers.
SSH-Patator: The attackers use SSH-Patator to attempt brute force assaults to obtain credentials for the SSH login.
DoS Slowhttptest: An assailants utilize the HTTP Get request to bypass the number of HTTP connections allowed on the server, preventing different users from accessing and allowing the attackers to activate multiple HTTP connections a comparable server.
DoS slowloris: The attackers use Slow Loris tools to perform a DoS attack.
FTP-Patator: The assailants use FTP-Patator to attempt to perform brute force assaults to discover the FTP credentials for login.
DoS GoldenEye: The attackers use the GoldenEye tool to perform a DoS attack.
PortScan: The assailants try to gather data recognized by the target computer, such as operating system and running services through forwarding packs with various destinations.
DDoS: The attackers use several computers that cooperate to assault a victim's system.
DoS Hulk: The attackers use the HULK tool to execute DoS assaults on web servers that generate various traffic levels. Furthermore, the generated traffic may circumvent cache engines and target the immediate resource pool of the server.[11]

3.2 Pre-Processing

Pre-Processing steps:

1. **Remove duplicate and missing data:** remove all occurrences of null value (NaN) and duplicate since the dataset is big enough; this virtually does not affect the findings. The total number of observations before removing duplicates and Null values (2,830,743). A total number of observations after removing duplicate and Null values (2,425,727). [12]
2. **Grouping the attacks:** Generated Label_Category to combine the minority attack classes as having comparative behavior and characteristics into seven groups which are ('benign', 'botnet', 'brute_force', 'dos', 'portscan', 'web_attack', 'infiltration').
3. **Oversampling:** Due to the discrepancy in class variables, an algorithm will prefer to classify the class with the most occurrences of the majority class while simultaneously creating the illusion of high accuracy. As a pre-handling of imbalance dataset issue, we choose to use Synthetic Minority Oversampling Technique (SMOTE). [13] [14]
4. **Split data:** At this step decided to divide the dataset 80:20 ratio training: testing. Additionally, for the labels, we conduct a stratified split due to the imbalanced nature of the dataset. It also guarantees that the generated datasets have the same proportions of classes as the original, in contrast to random sampling, which arbitrarily divides the data. [15] [16]
5. **Apply Normalization:** The final stage of pre-processing. Normalization is important since feature values vary in size; some are [0;∞] while others are [0;1]. So, we make sure that they contribute an equal amount to the categorization by bringing all the characteristics into the same range. We did min-max normalization using the library of scikit-learn. Min-max normalization rescales all features to [0,1], using the formula below where x is the value for the original. [12]

$$x'=(x- \min(x))/(\max(x)- \min(x)) \quad (1)$$

3.3 Application Of Algorithms

Deep learning algorithms that used to do this work are (RNN, DBN, and CNN), we evaluate the work by training and testing deep learning models.

RNN it is a discriminatory DL algorithm, ideally suited for sequential processing in environments with results. Contrary to other neural networks, its performance depends more than on backpropagation instead of forward-propagation. A time layer is developed into an RNN to sequentially evaluate the data accompanied by learning about multi-dimensional variations in unrevealed variable units. Modifications to these unrevealed units refer to the data of the neural network, which results in constant changes and the present state of the neural network the figure 2 below show the simple RNN architecture. [9]

Journal of Theoretical and Applied Information Technology
15th June 2023. Vol.101. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

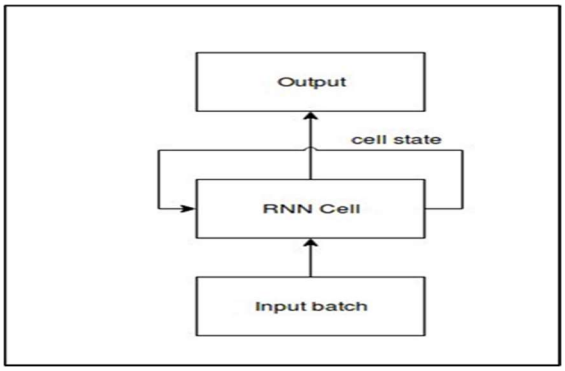

Figure 2: Simple RNN Architecture.

The CNN is a discrimination DL-based algorithm designed to reduce by using equal distribution and sparse interaction and sharing of parameters to decrease the data inputs needed by the traditional artificial nervous network (ANN), this makes CNN more scalable with less preparation time. In the CNN, three types of layers: convolutional layer, pooling layer, and activation unit. [17]

DBN it is a probabilistic generative model made up of stacked Boltzmann system modules (RBMs) as shown in figure 3 below. A DBN consists of stacked RBMs, which conduct greedy layer wise training for reliable performance in an unsupervised field. Training in a DBN is carried out layer by layer and each is done as an RBM trained over the previously trained layer. DBNs are a collection of RBM layers used for pre-workout phases and have also become a feed-forward network to finetune weight using another method. [1]

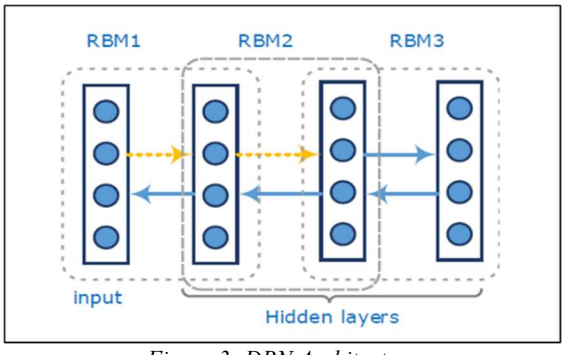

Figure 3: DBN Architecture

3.4 Classification (Seven Output Groups)

Classify the testing sets using the trained deep learning algorithms models, then compare the classification results to the testing sets' classification data to verify classification accuracy. The deep learning algorithms used the multi-class classification to recognize different types of attacks classes and benign classes.

We trained and evaluated each of the deep learning algorithms described above using the various label classes (original labels, first grouped (Label Category), and second, which is binary grouped (Label Category 2)) as shown in figure 4 and compare their performance using the performance evaluation given in Section 4. The Figure below shows the label classification in our work.

Label	Label_Category	Label_Category_2
BENIGN	benign	0
BENIGN	benign	0
BENIGN	benign	0
BENIGN	benign	0
BENIGN	benign	0
...	...	...

Figure 4: Labeling the Groups.

3.5 Network Behavior

The key parameter on which the anomaly detection systems depend is network behavior. Suppose the behavior of the network is in the predefined behavior. In that case, the network transaction is acknowledged, or the alarm in the anomaly detecting system is activated if any abnormal behavior will happen. Appropriate network output may be either pre-determined or learned from the network administrator requirements or conditions.

3.6 Performance Evaluation

To assess the algorithms, the results are compared using the performance metrics mentioned below. be evaluated against each algorithm.

A confusion matrix is created as a consequence of the categorization, and it is divided into four sections: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Accuracy, Precision, Recall, and F1Score performance metrics may be calculated using these numbers. The algorithms are compared using the performance metrics listed below.

Accuracy: The model's accuracy was measured based on the model's efficiency subset. Accuracy was one indicator of classification models assessment. The precision approximation is in equation 2.

$$Ac=(TP+TN)/(TP+TN+FP+FN) \quad (2)$$

Precision: means a positive degree of estimation. The balance of the true-positive that the model states the associated with demanded are a proportion of the total true positive variables. Equation 3 indicates the precision rate:

$$Pr=TP/(TP+FP) \quad (3)$$

Recall: the RE is called the TP value, which refers to the total positive values of the system states compared with the exact total positive values of the information. The reminder rate in equation 4 is presented:

$$Pr=TP/(TP+FN) \quad (4)$$

F1-score: can also be used for the efficiency prediction of the model. This is the weighted average of the recall and model precision. Equation 5 indicates the value of the F1-S:

$$F1\text{-}S=(2*TP)/(2*TP+FP+FN) \quad (5)$$

False positive rate: it is considered a positive example, but it is the real negative percentage of all negative samples Equation 6 indicates the value of the (FPR). The lower result it is the better.

$$FPR=FP/(FP+TN) \quad (6)$$

3. RESULTS AND DISCUSSION

Table 3 provide the results of the entire classification for the seven types of attacks that detected by using the three deep learning algorithms namely, CNN, RNN, and DBN of the CICIDS2017 dataset. The contribution of this work is CNN algorithm which is best suited for highly efficient and fast feature extraction from big data. Since CNN can automatically learn behavior from data enables intrusion detection system to identify, analyze, and classify data as normal or hostile.

The results of our final models are compared between the three algorithms according to the five-confusion matrix which are (Accuracy, Precision, Recall, FPR, and F1 Score). The Benign label refer to the normal behavior of the network traffic. In terms of detection the benign class, the CNN algorithm records the highest performance of accuracy (99.1%), precision (99.3%), recall (99.5%), and F1 score (99.4%) as compared to the other two algorithms as shown in figure 5 below. Additionally, DBN showed better performance in false positive rate of (1.9%). However, RNN recorded the lowest percentage performance.

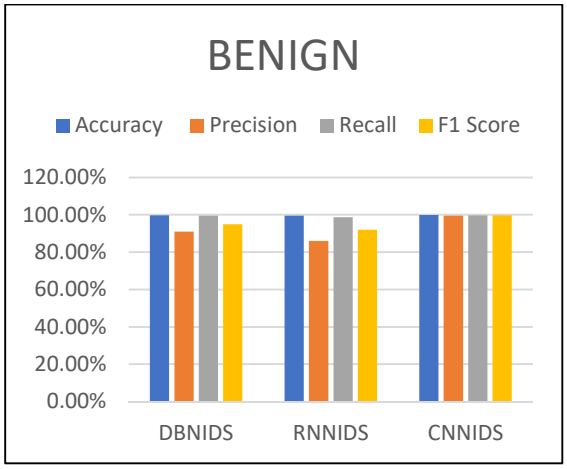

Figure 5: Benign Performance Analysis.

In terms of DOS attack, CNN and DBN both showed the same results of accuracy of (99.8%). Also, CNN recorded the highest percentage performance of precision (99.2%), recall (99.8%), false positive rate (0.1%), and F1 score (99.5%). In the other hand, RNN performed the lowest as compared with the other two algorithms. Figure 6 shows the results of DOS attack classification.

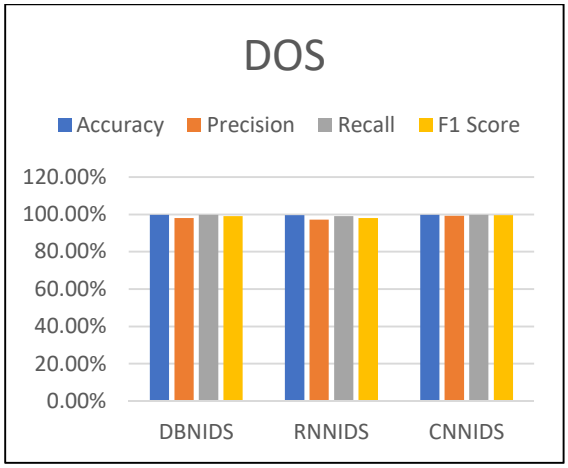

Figure 6: DoS Attack Performance Analysis.

In terms of PortScan detection attack CNN algorithm recorded the highest percentage performance as compared to the other two algorithms as shown in figure 7 below. The accuracy of CNN (99.4%), precision (93.3%), and F1 score (86.6%). However, in terms of false positive rate and recall, DBN performed better than CNN and RNN, (0.2%) and (85%) respectively.

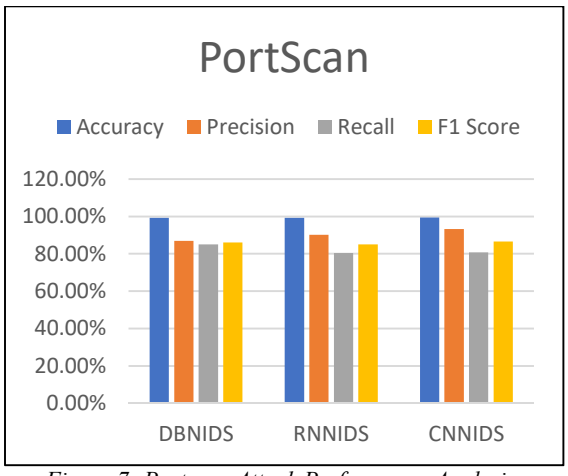

Figure 7: Portscan Attack Performance Analysis.

When it comes to Brute_force attack, in terms of accuracy, the three algorithms recorded the same result of (99.9%). CNN recorded the highest precision of (99.8%). In addition, CNN recorded the highest recall of (99.7%), F1 score (99.8%), and false positive rate (0.0035%). However, RNN had the lowest result in recall (98.7%), and false positive rate (0.03%). Among the three algorithms DBN recorded the lowest percentage in F1 score of (99%). Figure 8 shows the performance of the algorithms in terms of Brute_force attack detection.

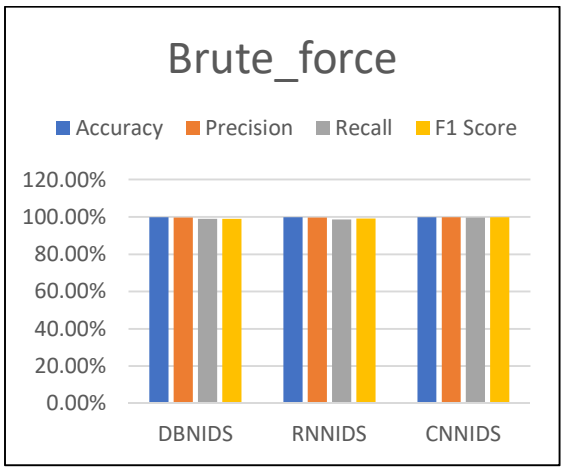

Figure 8: Brute_Force Attack Performance Analysis.

For the Web_attack detection performance evaluation CNN recorded the highest accuracy of (99.9%), precision (99.6%), recall (99.8%), false positive rate (0.006%), and F1 score (99.7%) as shown in figure 9 below. However, RNN showed the lowest performance among the three algorithms of accuracy of (97.9%), precision (99.1%), recall (98.2%), false positive rate (2.9%), and F1 score (98.96%).

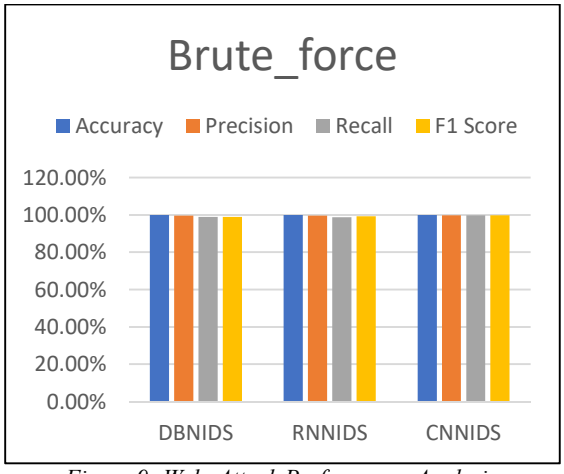

Figure 9: Web_Attack Performance Analysis.

In terms of Botnet attack detection, CNN algorithm performed the highest as compared to the other two algorithms as can be seen from figure 10 below of accuracy (99.8%), precision (99.1%), false positive rate (0.08%), and F1 score (96.5%). For DBN algorithm it had the highest recall of (97%). Between the three algorithms RNN recorded the lowest percentage performance of accuracy (99.3%), precision (80.01%), recall (94.09%), false positive rate (0.52%), and F1 score (86.4%).

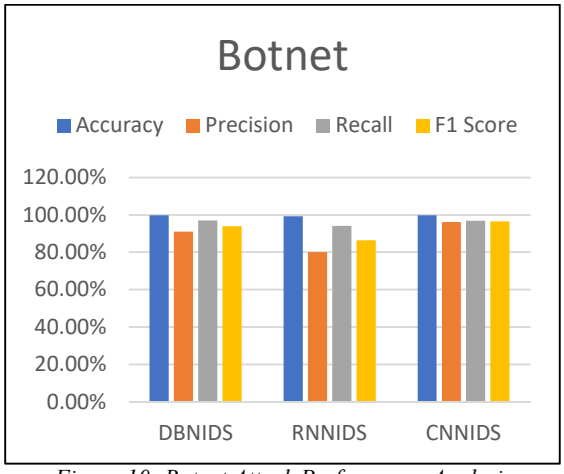

Figure 10: Botnet Attack Performance Analysis

Finally, for the Infiltration attack, all CNN, DBN, and RNN had the same accuracy of (99.9%). In addition, CNN recorded the highest precision

(99.6%), recall (99.5%), false positive rate (0.0079%), and F1 score (99.6%). In the contrary, RNN performed the lowest precision (99%), recall (99.3%), false positive rate (0.02%), and F1 score (99%). Figure 11 presents the performance of Infiltration attack classification.

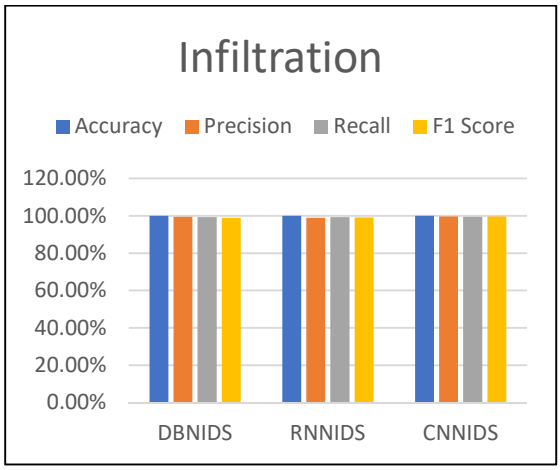

Figure 11: Infiltration Performance Analysis

Table 3: Performance Evaluation of Three Algorithms

Classifier	Accuracy	Precision	Recall	FPR	F1Score
BENIGN					
DBNIDS	98.6%	99%	98.7%	1.9%	99%
RNNIDS	97.9%	99.1%	98.2%	2.9%	98.96%
CNNIDS	99.1%	99.3%	99.5%	2.2%	99.4%
DoS					
DBNIDS	99.7%	98%	99.8%	0.2%	99%
RNNIDS	99.5%	97.1%	99.1%	0.4%	98.1%
CNNIDS	99.8%	99.2%	99.8%	0.1%	99.5%
PortScan					
DBNIDS	99.3%	87%	85%	0.2%	86%
RNNIDS	99.3%	90.2%	80.5%	0.19%	85%
CNNIDS	99.4%	93.3	80.8%	0.12%	86.6%
Brute_force					
DBNIDS	99.9%	99.6%	99%	0.01%	99%
RNNIDS	99.9%	99.6%	98.7%	0.03%	99.2%
CNNIDS	99.9%	99.8%	99.7%	0.0035%	99.8%
Web_attack					
DBNIDS	99.7%	91%	99.5%	0.2%	95%
RNNIDS	99.6%	86%	98.7%	0.35%	92%
CNNIDS	99.9%	99.6%	99.8%	0.006%	99.7%
Botnet					
DBNIDS	99.7%	91%	97%	0.21%	94%
RNNIDS	99.3%	80.01%	94.09%	0.52%	86.4%
CNNIDS	99.8%	96.1%	96.9%	0.08%	96.5%
Infiltration					
DBNIDS	99.9%	99.5%	99.3%	0.01%	99%
RNNIDS	99.9%	99%	99.3%	0.02%	99.1%
CNNIDS	99.9%	99.6%	99.5%	0.0079%	99.6%

4. CONCLUSION

In order to solve the data privacy and security in the IoMT environment, this work focused on improving an intrusion detection tool that can detect the attack and identify anomalies in the IoT framework. By test three deep learning algorithms based on IDS to detect normal and abnormal behavior . The algorithms are (RNN, DBN, and CNN). The algorithms are based on five confusion matrixes namely, (accuracy, precision, recall, F1Score, and detection rate). In order to evaluate the

efficiency of the current intruding device model, we used the CICIDS2017 dataset. The result showed that CNN had an overall better result among RNN and DBN algorithms, where most CNN algorithm results were more than 95%. For future researchers, can use the proposed algorithms to detect other types of attacks facing the IoT's systems.

REFERENCES:

[1] S. Manimurugan, S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, and R. Patan, "Effective attack detection in internet of medical things smart environment using a deep belief neural network," IEEE Access, vol. 8, pp. 77396–77404, 2020.

[2] M. Erza and K. Kim, "Deep Learning in Intrusion Detection System : An Overview," pp. 1–12.

[3] B. Subba, S. Biswas, and S. Karmakar, "A Neural Network based system for Intrusion Detection and attack classification," 2016 22nd Natl. Conf. Commun. NCC 2016, 2016,

[4] N. Gao, L. Gao, Q. Gao, and H. Wang, "An Intrusion Detection Model Based on Deep Belief Networks," Proc. - 2014 2nd Int. Conf. Adv. Cloud Big Data, CBD 2014, pp. 247–252, 2015,

[5] Q. Tian, D. Han, K. C. Li, X. Liu, L. Duan, and A. Castiglione, "An intrusion detection approach based on improved deep belief network," Appl. Intell., vol. 50, no. 10, pp. 3162–3178, 2020.

[6] Y. Ding et al., "DLEDNet: A deep learning-based image encryption and decryption network for internet of medical things," arXiv, pp. 1–13, 2020.

[7] D. K. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik, and P. K. Singh, "Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities," Trans. Emerg. Telecommun. Technol., no. June, pp. 1–26, 2020.

[8] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "TR-IDS: Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest," Secur. Commun. Networks, vol. 2018, 2018.

[9] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K. K. R. Choo, "A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting," Futur. Gener. Comput. Syst., vol. 85, pp. 88–96, 2018.

[10] B. A. Tama and K.-H. Rhee, "Attack classification analysis of IoT network via deep learning approach," Res. Briefs Inf. Commun. Technol. Evol., vol. 3, no. 15, pp. 1–9, 2017.

[11] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," no. Cic, pp. 108–116, 2018.

[12] K. Farhana, M. Rahman, and M. Tofael Ahmed, "An intrusion detection system for packet and flow based networks using deep neural network approach," Int. J. Electr. Comput. Eng., vol. 10, no. 5, pp. 5514–5525, 2020.

[13] C. Series, "Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset," 2019.

[14] A. A. Abdulrahman and M. K. Ibrahem, "Toward Constructing a Balanced Intrusion Detection Dataset Based on CICIDS2017," vol. 2, no. 3, pp. 132–142, 2020.

[15] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE - CIC - IDS2018 Big Data," J. Big Data, 2020.

[16] G. Kaur, A. Habibi Lashkari, and A. Rahali, "Intrusion Traffic Detection and Characterization using Deep Image Learning," Proc. - IEEE 18th Int. Conf. Dependable, Auton. Secur. Comput. IEEE 18th Int. Conf. Pervasive Intell. Comput. IEEE 6th Int. Conf. Cloud Big Data Comput. IEEE 5th Cybe, pp. 55–62, 2020.

[17] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," Electron., vol. 9, no. 7, 2020.