IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DEEP LEARNING ALGORITHMS FOR INTERNET OF MEDICAL THINGS

Received March 10, 2020, accepted March 26, 2020, date of publication April 6, 2020, date of current version May 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986013

Effective Attack Detection in Internet of Medical Things Smart Environment Using a Deep Belief Neural Network

S. MANIMURUGAN[1], (Member, IEEE), SAAD AL-MUTAIRI[1], (Member, IEEE), MAJED MOHAMMED ABOROKBAH[1], NAVEEN CHILAMKURTI[2], (Senior Member, IEEE), SUBRAMANIAM GANESAN[3], (Senior Member, IEEE), AND RIZWAN PATAN[4]

[1]Faculty of Computers and Information Technology, University of Tabuk, Tabuk 47512, Saudi Arabia
[2]Department Computer Science and IT, La Trobe University, Melbourne, VIC 3086, Australia
[3]Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309, USA
[4]Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada 520007, India

Corresponding author: Rizwan Patan (prizwan5@gmail.com)

This work was supported by the University of Tabuk, Saudi Arabia

ABSTRACT The Internet of Things (IoT) has lately developed into an innovation for developing smart environments. Security and privacy are viewed as main problems in any technology's dependence on the IoT model. Privacy and security issues arise due to the different possible attacks caused by intruders. Thus, there is an essential need to develop an intrusion detection system for attack and anomaly identification in the IoT system. In this work, we have proposed a deep learning-based method Deep Belief Network (DBN) algorithm model for the intrusion detection system. Regarding the attacks and anomaly detection, the CICIDS 2017 dataset is utilized for the performance analysis of the present IDS model. The proposed method produced better results in all the parameters in relation to accuracy, recall, precision, F1-score, and detection rate. The proposed method has achieved 99.37% accuracy for normal class, 97.93% for Botnet class, 97.71% for Brute Force class, 96.67% for Dos/DDoS class, 96.37% for Infiltration class, 97.71% for Ports can class and 98.37% for Web attack, and these results were compared with various classifiers as shown in the results.

INDEX TERMS IoT, deep learning, anomaly detection, intrusion detection, DBN.

I. INTRODUCTION

The IoT is a sort of network which interfaces anything with the Internet dependent on a specified protocol over data sensing devices leading to data sharing and interchanges and allowing smart identification, tracing, positioning, administration, and monitoring. The IoT's regular definition is as a physical objects network. The internet is not just a PC network, however; it has advanced into a devices network of different sorts and sizes, home appliances, smart phones, vehicles, toys, cameras, medicinal tools, modern frameworks, people, animals, and structures, which are all associated, each sharing and communicating data dependent on specified protocols [1].

The IoT is an internet of three kinds of relations: (1). Human to human, (2) Human to machine/things, and (3) Things/machine to things/machine, all communicating over the internet [2]. The objective of the IoT is to allow things to be associated anytime, anywhere, with anything and anybody, desirably utilizing any paths/networks and any support [1]. The IoT has many applications. The commonly known applications comprise smart health services, smart transportation, and smart grids and structures [3]. The four-layer architecture of the IoT is shown in figure.1.

II. INTRUSION DETECTION

Intrusion detection is accepted to be an essential security system designed to manage attacks on networks and recognize malignant actions in computer network traffic. It assumes an imperative role in overall data security and supports in discovering, deciding, and detecting the unapproved use, duplication, modification, and demolition of data and data frameworks [4]. There are two regular security frameworks, the network security framework and the host

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

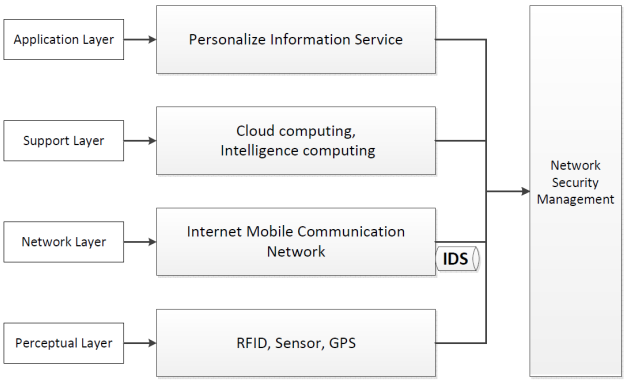

FIGURE 1. Architecture of IoT.

security frameworks which secure the fundamental network and systems from unapproved access, malfunction, destruction, and change. These two frameworks might comprise various coordinated security models; for example, firewalls, antiviruses, and Intrusion Detection Systems (IDS) which allow a network or system to be observed and raise an alert when malignant action happens [5].

Comprehensively, IDSs are classified into three techniques: misuse detection, anomaly detection, and hybrid. Misuse identification methods utilize predefined signatures of malignant actions to detect intrusion. Hence, they are utilized for identifying known attacks. Anomaly detection methods characterize typical patterns and detect malicious actions depending on their difference from ordinary patterns. In this way, anomaly-based identification techniques have the ability to identify zero-day attacks [6]. Hybrid methods exploit both anomaly and anomaly identification techniques. Through lessening the false positives of unknown attacks, hybrid methodologies target expanding identification rates of known intrusions [7].

Intruder detection was one significant advance in assuring the IoT networks security. Intrusion detection is therefore one of many systems for handling security interruptions that can be identified in any of the four architectural layers of IoT represented in Fig. 1. The Network Layer not only operates as the support for linking diverse IoT devices; it additionally facilitates network-based security defence systems like NIDS. There are numerous IDS techniques; for example, techniques dependent on statistical analytics, cluster analytics, ANN, or deep learning. Within these techniques, intrusion detection which is dependent on deep learning performs better than various other techniques, because deep learning has a high capacity for self-learning, self-adaption, generalization, and the identification of unknown attack activity [8].

III. ANOMALY DETECTION

The present world features a wide ranging IoT which is producing a vast measure of information, and anomalies are an essential part of each framework. These anomalies could be an indication of resources drain in an industrial framework,

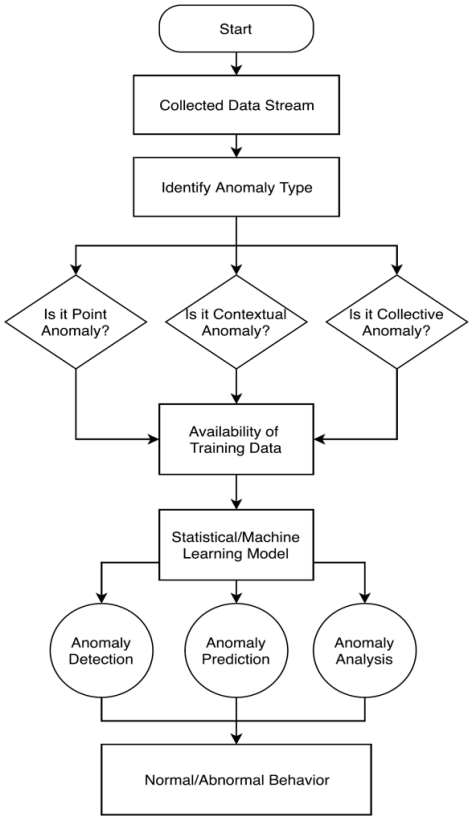

FIGURE 2. Anomaly detection flow chart.

an essential circumstance at an aeronautics platform to ignore unpredicted issues, or identifying unusual performance of medical instruments, and so on. Subsequently, having the option to identify the anomalies could enormously affect the total performance of any monitored model. The key difficulty in perceiving anomalies is describing the exact boundaries between abnormal/normal activities, as the accessibility of abnormal observations to train the models is usually insufficient. In practical situations, abnormal behaviour patterns have been minimally contrasted with normal behaviours [9]. Figure.2 shows a flow chart of anomaly detection.

In the anomaly identification framework (as shown in Fig. 2), the initial process is to understand the condition of the gathered data flow, which tends to be binary, discrete, or continuous, as well as the relationship framework. This relationship framework demonstrates whether it is time series information, spatial information, or graphical information. Identifying the kind of relationship supports the selection of the right method for detecting anomalies, examination, or expectations. The next step is to find the sort of anomaly from the predetermined set (for example: point anomaly, collective anomaly, or contextual anomaly).

The next process is to understand the presence of training information to design an anomaly identification framework. Based upon the presence of the information and its explanation, we might represent it as supervised, semi-supervised

or unsupervised. That data helps developers to select suitable anomaly identification strategies. In supervised training, the availability of the information with a class label and its basic type of learning is used to identify the abnormal conduct of the framework. In unsupervised learning, we have information but no solid output (for example, a class label). Also, in semi-supervised learning we have constrained models with a class label while the remaining information is unlabelled [9].

A. TYPES OF ANOMALIES

A significant part of an anomaly identification method is the concept of the required anomaly. Anomalies can be characterized into the following three classifications:

Point Anomalies: If an individual data model can be treated as abnormal regarding the remaining information, the model involves point anomalies. These are the most basic sort of anomalies and have been the target of most analysis on anomaly identification. In a practical example, credit card scam identification, let the dataset be compared to a person's credit exchanges. In order to simplify things, let us accept that the information is determined utilizing just one feature: *amount spent*. A cash transaction higher than the ordinary level which the individual would spend is a point anomaly.

Contextual Anomalies: If a data model is abnormal in a particular context (but not in another), it contains contextual anomalies (likewise stated as conditional anomalies). The concept of a context is caused by the structure in the dataset and must be determined as the segment of the issue definition. Every data model is characterized utilizing the accompanying two arrangements of features. Contextual attributesare utilized to decide the context (or neighbourhood) for that model; for example, in spatial datasets, the latitude and latitude of an area are the contextual attribute. In time-series information, time is the contextual attribute that decides the condition of a model on the total order.

Behavioural Attributes: These characterize the non-contextual attributes of a model; for example, in a spatial dataset defining the average rainfall of the whole world, the measure of rainfall in any area is a behavioural attribute. The anomaly conduct is resolved utilizing the qualities for the behavioural attribute inside the particular context. A data model may represent the contextual anomalies in the provided context. However, an equivalent data model (as far as behavioural attributes go) would be viewed as ordinary in a dissimilar context. This property is key in recognizing behavioural and contextual attribute for a contextual anomalies recognition method. The decision to implement a contextual anomalies identification method is made through recognizing the significance of the contextual anomaly in the objective application area.

Collective Anomalies: If an accumulation of pertained data model is anomalous regarding the whole dataset, it is known as the containing collective anomalies. The individual data model in this anomaly might not be anomalous by itself, but its event together as an accumulation is anomalous. Collective anomalies are investigated for arrangement, graph, and spatial information. It must be noted that while a point anomaly could happen in any dataset, a collective anomaly can only happen in a dataset in which data models are connected. The difference is that the event of a contextual anomaly relies upon the accessibility of context attributes in the information. A collective or a point anomaly could likewise be a contextual anomaly whenever considered in relation to the context. In this way, point or collective anomalies identification issues could become a contextual anomaly identification issue through consolidating the context information [10].

IV. PROPOSED ALGORITHM AND PROPOSED METHODOLOGY (DEEP BELIEF NETWORK

DBNs are generative techniques. A DBN comprises stacked RBMs which perform greedy layer-wise training to achieve solid execution in an unsupervised domain. In a DBN, training is achieved layer by layer, and each one is performed as an RBM trained over the past trained layer (DBNs are a group of RBM layers utilized for the pre-training stage and additionally turned into a feed-forward network for weight fine-tuning with a different approach.

The significant usage of RBMs is likely to be because there is a dearth of labeled data, and RBMs and auto-encoders can be pre-trained on unlabeled data and fine-tuned on a small amount of labeled data.

A greedy layer-wise training algorithm was used to train a DBN one layer at a time. The greedy layer-wise method was utilized because it optimizes each layer at a time greedily. After unsupervised training, there is usually a fine-tune stage, when a joint supervised training algorithm is applied to all the layers. It combines two ideas: 1) that the choice of initial parameters of a deep neural network can have a significant regularizing effect; 2) that learning about the input distribution can help with learning about the mapping from inputs to outputs. In the pre-training stage, the underlying features were trained by a greedy layer-wise unsupervised method, while a softmax layer was implemented in the fine-tuning stage to the top layer to enhance the features of the labelled samples [11]. Figure.3 represents the architecture of the DBN.

In order to visually represent the complexity, we standardized the SD as in equation.1:

$$\sigma^* = \frac{\sigma - \sigma_{min}}{\sigma_{max} - \sigma_{min}} \tag{1}$$

In RBM, v indicates every visible unit and h indicates every hidden unit. To decide the system, we sought to acquire the model's three parameters: $\theta = \{W, A, B\}$. These were the weight matrix W, hidden layer element bias B, and visible layer element bias A, individually.

Assume an RBM has m hidden cells and n visible cells, v_i indicates the i^{th} visible unit, h_i the j^{th} hidden unit, and the parameters structure is shown as in equation.2:

$$W = \left\{w_{i,j} \in R^{n \times m}\right\} \tag{2}$$

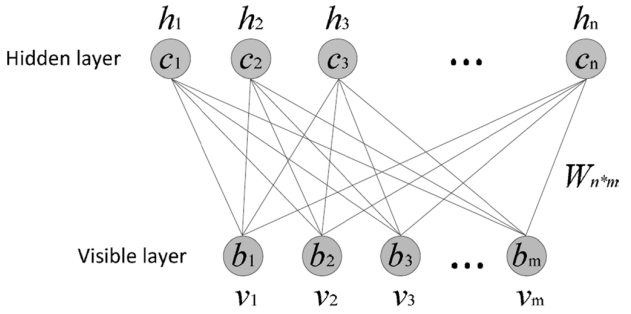

FIGURE 3. DBN architecture.

where $w_{i,j}$ indicates the weight among the i^{th} visible cell and j^{th} hidden cell from equation 3.

$$A = \{a_i \epsilon R^{\mathrm{m}}\} \tag{3}$$

where, a_i represents the bias threshold of the i^{th} visible cell from equation 4;

$$A = \{b_j \epsilon R^{\mathrm{n}}\} \tag{4}$$

where, b_j indicates the j^{th} visible cell bias threshold. For an order of (v, h) through a present condition, presuming that hidden and visible layer follow Bernoulli distribution, the energy equation of RBM is represented as in equation 5:

$$E(v, h \mid \theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i W_{ij} h_j \tag{5}$$

where, $\theta = \{W_{ij}, a_i, b_j\}$ were the RBM model's parameters, and the function of energy showed the value of energy among the estimation of every visible node and every hidden layer node. Due to the regularization and exponential of energy function, the joint likelihood distribution equation could be acquired in which the nodes set of visible layers and the nodes set of the hidden layers were in a specific condition separately (v, h) as in equation 6:

$$P(v, h \mid \theta) = \frac{e^{-E(v,h \mid \theta)}}{Z(\theta)} \tag{6}$$

$$Z(\theta) = \sum_{v,h} e^{-E(v,h \mid \theta)} \tag{7}$$

where, in equation 7, $Z(\theta)$ was the standardized factor or distribution function indicating the total energy exponents of every single available condition of the set of hidden nodes and visible layers [10].

The determination of the probability function is frequently utilized to obtain the parameters. Having presented the joint likelihood distributions $P(v,h|\theta)$, the marginal distributions $P(v|\theta)$ of the nodes set of the visible layers could be acquired through summations of the overall conditions of the hidden layer nodes set in equation 8:

$$P(v \mid \theta) = \frac{1}{Z(\theta)} \sum_{h} e^{-E(v,h \mid \theta)} \tag{8}$$

The marginal distributions indicate the likelihood with which the arrangement of nodes in the visible layers was in the specific level distribution. Because of the exceptional layer-layer connections and inter-layer connectionless form of RBM system, it has the accompanying significant conditions:

Having presented the condition of the visible cells, the enactment conditions of every hidden layers cell were restrictively autonomous. Here, the initiation likelihood of the j^{th} hidden element was as shown in equation 9:

$$P(h_j = 1 \mid v) = \sigma(b_j + \sum_i v_i W_{ij}) \tag{9}$$

Accordingly, once the condition of the hidden elements was specified, the initiation likelihood of the visible elements was additionally conditionally independent as represented in equation 10:

$$P(v_i = 1 \mid h) = \sigma(a_i + \sum_j W_{ij} h_j) \tag{10}$$

where, $\sigma(\mathrm{x})$ is the sigmoid function.

To decide the model of RBM, it was important to sort out the three parameters of the model: $\theta = \{W_{ij}, a_i, b_j\}$. The parameter arrangement utilized the logarithmic probability functions to take the subordinates of the parameters. From equation 8, $P(v \mid \theta) = \frac{1}{Z(\theta)} \sum_h e^{-E(v,h \mid \theta)}$, energy E is inversely proportional to probability P, and E was limited through expanding P.

The regular strategy for expanding the functional probability was the inclination raise technique that relates to the change of parameters as indicated by the accompanying equation 11:

$$\theta = \theta + \mu \frac{\partial \ln P(v)}{\partial \theta} \tag{11}$$

This iterative process expanded the probability P and reduced the energy E [11].

The flow of algorithm can be outlined as:

Step 1: Initiate the population and produce diverse number of hidden layers and the total neurons in every layer randomly;

Step 2: Compute the fitness rate as per Eq. 1, selected by the roulette technique, and keep the ideal individual in the present; interval crossover; variation;

Step 3: "Elite" holds, holding the individual with the best value of fitness in the process development;

Step 4: Find if the highest count of iterations has been achieved. Once achieved, the network structures generated are held, or repeat Step2-Step3 once more;

Step 5: Utilize the optimal networks structure for DBN and train the IDS system;

Step 6: Classify the testing sets through the trained DBN model, and lastly coordinate the results of classification with the classification data of the testing sets to validate the classification accuracy.

TABLE 1. Dataset description.

Files	Day Activity	Attacks
Monday- pcap_ISCX.csv	Monday	Benign (Normal individual activities)
Tuesday- pcap_ISCX.csv	Tuesday	Benign, SSH-Patator, FTP-Patator,
Wednesday-pcap_ISCX.csv	Wednesday	Benign, DoSGoldenEye, DoS Hulk, DoSSlowhttptest, DoSslowloris, Heartbleed
Thursday-Morning-WebAttack.pcap_ISCX.csv	Thursday	Benign, Web Attack – XSS, Web Attack – Sql Injection, Web Attack – Brute Force.
Thursday-Afternoon-Infilterations.pcap_ISCX.csv	Thursday	Benign, Infiltration
Friday-Morning.pcap_ISCX.csv	Friday	Benign, Bot
Friday-Afternoon- PortScan.pcap_ISCX.csv	Friday	Benign, PortScan
Friday-Afternoon- DDos.pcap_ISCX.csv	Friday	Benign, DDoS

TABLE 2. Dataset class labels with instances.

Class Labels	Number of instances
Heartbleed	11
Web Attack – Sql Injection	21
Infiltration	36
Web Attack – XSS	652
Web Attack – Brute Force	1507
Bot	1966
DoSSlowhttptest	5499
DoSslowloris	5796
SSH-Patator	5897
FTP-Patator	7938
DoSGoldenEye	10293
DDoS	41835
PortScan	158930
DoS Hulk	231072
BENIGN	2359087

V. IMPLEMENTATION AND RESULTS ANALYSIS

A. DATASET DESCRIPTION

The CICIDS2017 dataset was used to direct this work. Generally, many DDoS attack datasets have numerous impediments like non-pertinent information or redundancy that make them inconsistent. The CICIDS2017 dataset has recent network identical data. This dataset was gathered for five continuous days (Monday – Friday) with various attacks as well as normal information, as shown in Table.1, above. This dataset has the network information with and without attacks, which make it close to true network data. The dataset was uneven, so a duplicating technique was used as unevenness critically impacts the deep learning technique training so we had to ensure that the testing was balanced [12].

This work was implemented utilizing Keras on the Tensorflow package for deep learning on 64-bit Intel Core-i7 CPU with 16 GB RAM on the Windows 7 platform. The Machine learning algorithm was executed in MATLAB. Table.2 represents the instances with the class labels of the dataset.

Heartbleed Attack: The attackers use the OpenSSL protocol to embed malignant data within OpenSSL memory, giving the attacker unapproved permission to important information.

Web Attack–SQL Injection: An SQL injection is a code injection method, utilized to attack data-driven applications, including odious SQL proclamations embedded within a section area for implementation.

Infiltration: The attackers utilize infiltration strategies and software to infiltrate and obtain complete unapproved logins to the networked system information.

Web Attack – XSS: The attackers infuse generally trusted websites and benign web applications to forward malignant contents.

Web Attack – Brute Force: The attackers attempt to acquire privileged data; for example, PINs and passwords, utilizing trial-and-error.

Bot: The attackers utilize Trojans to break the protection of many victim machines, assuming responsibility for those machines and arranging each machine in the Bot network so it can be used and controlled by the attackers remotely.

DoSSlowhttptest: The attackers use the HTTP Get request to circumvent the count of HTTP connections permitted on the server, inhibiting various users from approaching and providing the attackers the chance to enable numerous HTTP connections with a similar server.

DoSslowloris: The attackers utilize Slow Loris tools to execute a DoS attack.

SSH-Patator: The attackers utilize SSH Patator to try to execute brute force attacks to find the SSH login passwords.

FTP-Patator: The attackers utilize FTP Patator to try to execute brute force attacks to find the FTP login passwords.

DoSGoldenEye: The attackers utilize the GoldenEye tool to execute a DoS attack.

DDoS: The attackers utilize numerous machines which work jointly to attack one victim machine.

PortScan: The attackers attempt to collect data identified with the victim machine like the type of OS and running services through forwarding packets with different destination points.

DoS Hulk: The attackers utilize the HULK tool to complete DoS attacks on web servers which create volumes of different and jumbled traffic. In addition, the produced traffic could bypass caching engines and attack the server's immediate resource pool.

Benign: Normal traffic behaviour [12].

VI. RESULTS AND DISCUSSION

The accuracy of the model was evaluated in terms of the subset of the performance of model. Accuracy was one of the measurements for assessing the classification models. Equation (12) represents the accuracy estimation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Precision implies the positive predicative rate. It is a proportion of the total true positives the model states correlated with the total positives it demands. The rate of precision is presented in equation 13:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

The recall is known as the TP value, which refers to the total positives in the system states contrasted with the exact total of positives in the information. The rate of recall is presented in equation 14:

$$Recall = \frac{TP}{(TP + FN)} \tag{14}$$

The F1 score could likewise be used to estimate model performance. It is the weighted average of the recall and precision of the model. The value of the F1 Score presented in Eq. (15) is:

$$F1Score = \frac{2 * TP}{2 * TP + FP + FN} \tag{15}$$

The detection rate (DR) represents the level of intrusion instances. The value of the detection rate is presented in equation 16:

$$DR = \frac{TP}{TP + FN} \tag{16}$$

TP: true positive, FP: false positive, FN: false negative, TN: true negative [13]–[16].

This paper combined the minority attack classes as having comparative behaviour and characteristics. Having combined comparable classes, the class of the predominant proportion of different attack labels seems to be enhanced. It can be seen from the table that the prevalence of the major class (Benign) was 83.34% where the minority class was 0.00039% (Heart bleed).

With such a major difference in prevalence values, the potential detectors might tend towards Benign. The Benign label was termed a normal label and the performance

TABLE 3. Normal Attack Detection (which includes the attack label "benign').

Sl. No.	Attack Detection Method	Accuracy	Precision	Recall	F1-Score	Detection Rate
1	DBNIDS[@]	99.37%	96.21%	98.34%	0.97	98.31%
2	SVMIDS[#]	98.45%	95.32%	97.15%	0.96	97.12%
3	RNNIDS[*]	97.00%	94.91%	95.59%	0.97	95.50%
4	SNNIDS[$]	92.00%	90.01%	90.25%	0.93	90.11%
5	FNNIDS[$$]	91.35%	89.08%	89.13%	0.90	89.01%

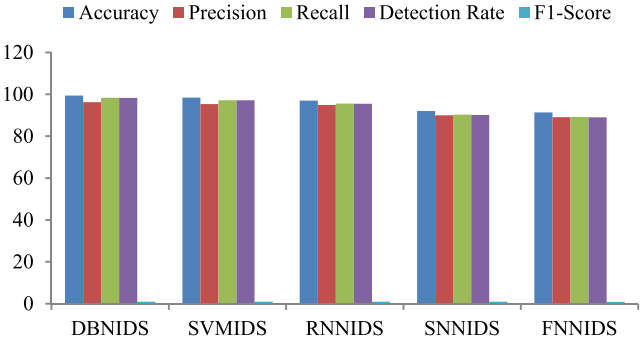

FIGURE 4. Performance analysis of Normal Attack Detection.

TABLE 4. Botnet Attack Detection (which includes the attack label "bot").

Sl. No.	Attack Detection Method	Accuracy	Precision	Recall	F1-Score	Detection Rate
1	DBNIDS[@]	97.93%	96.21%	98.54%	0.97	98.51%
2	SVMIDS[#]	96.84%	95.32%	97.55%	0.96	97.42%
3	RNNIDS[*]	95.01%	94.91%	95.69%	0.95	95.30%
4	SNNIDS[$]	90.00%	90.01%	90.75%	0.90	90.21%
5	FNNIDS[$$]	89.35%	89.08%	89.83%	0.89	89.01%

analysis of the presented DBN was evaluated and correlated with other detection techniques, as shown in fig.4 according to table. 3.

The Bot label was termed as Botnet ARES, a new label. This label contained 1966 instances, with a prevalence of 0.06%. Compared to the other conventional and existing techniques, the proposed method achieved better performance results in terms of all parameters, as shown in Table.4. An accuracy of 97.93% and a detection rate of 98.51% was achieved in relation to this Botnet ARES label. Fig.5. represents the Performance Analysis for Botnet ARES Attack Detection.

The FTP-Patator and SSH-Patator labels were combined as Brute Force labels because both the FTP-Patator and SSH-Patator labels have similar characteristics and behaviour. By combining these labels, we formed a new label with 13,835 instances and 0.48% prevalence.

DoS/DDoS was new label representing combinations of DDoS, DoSGoldenEye, DoSHulk, DoSSlowhttptest, DoSslowloris, and Heartbleed as represented in table.6. By combining all these labels, 294,506 instances with 10.4% prevalence were performed via the proposed method and better results were obtained in relation to all the proposed

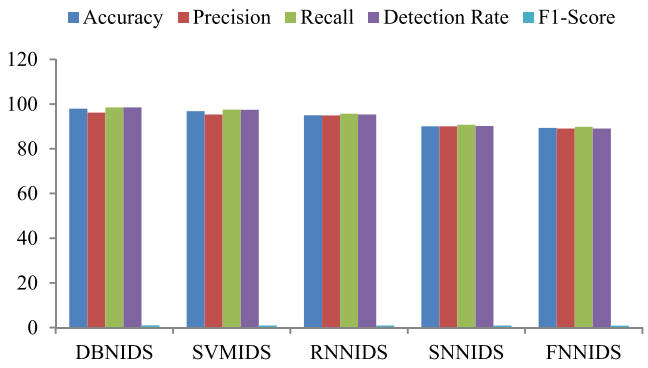

FIGURE 5. Performance analysis of Botnet ARES Attack Detection.

TABLE 5. Brute Force Attack Detection (which includes the attack labels "FTP-Patator & SSH-Patator").

Sl. No.	Attack Detection Method	Accuracy	Precision	Recall	F1-Score	Detection Rate
1	DBNIDS[@]	97.71%	96.21%	98.17%	0.97	98.01%
2	SVMIDS[#]	96.56%	95.32%	97.65%	0.96	97.23%
3	RNNIDS[*]	95.21%	94.91%	96.12%	0.95	96.11%
4	SNNIDS[$]	91.34%	90.01%	92.43%	0.91	92.32%
5	FNNIDS[$$]	90.56%	89.08%	91.65%	0.90	91.43%

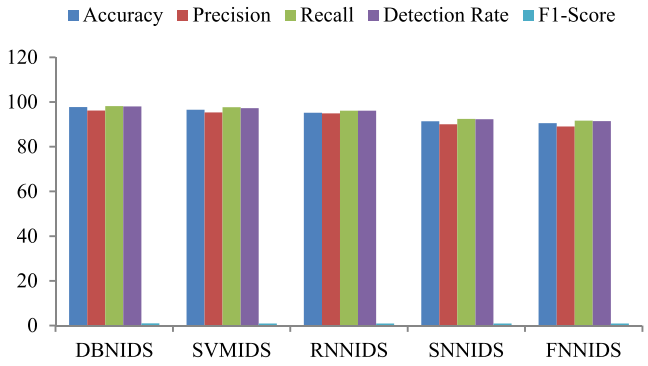

FIGURE 6. Performance analysis of Brute Force Attack Detection.

TABLE 6. Dos/DDos Attack Detection (which includes the attack labels "DDoS, DosGoldenEye, Dos Hulk, DoSSlowhttptest, DoSslowloris & Heartbleed").

Sl. No.	Attack Detection Method	Accuracy	Precision	Recall	F1-Score	Detection Rate
1	DBNIDS[@]	96.67%	95.21%	97.34%	0.97	97.31%
2	SVMIDS[#]	95.55%	94.32%	96.15%	0.95	96.12%
3	RNNIDS[*]	94.40%	93.91%	95.59%	0.94	95.50%
4	SNNIDS[$]	93.30%	92.01%	94.25%	0.93	94.11%
5	FNNIDS[$$]	92.25%	91.08%	91.13%	0.92	91.01%

parameters. Fig.7 represents the performance analysis for DoS/DDoS attack detection.

The performances in relation to the infiltration label and PortScan label were analyzed separately as shown in

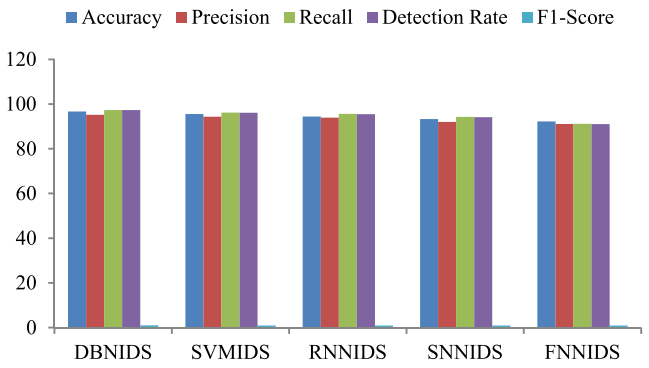

FIGURE 7. Performance analysis of Dos/DDos Attack Detection.

TABLE 7. Infiltration Attack Detection (which includes the attack label "Infiltration").

Sl. No.	Attack Detection Method	Accuracy	Precision	Recall	F1-Score	Detection Rate
1	DBNIDS[@]	96.37%	95.21%	96.74%	0.97	96.30%
2	SVMIDS[#]	95.45%	94.32%	95.65%	0.96	95.10%
3	RNNIDS[*]	94.00%	93.91%	94.59%	0.97	92.49%
4	SNNIDS[$]	93.00%	92.01%	89.45%	0.93	88.34%
5	FNNIDS[$$]	92.35%	91.08%	88.33%	0.90	87.31%

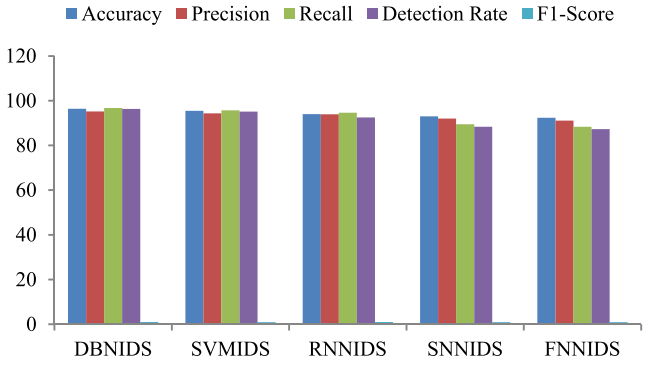

FIGURE 8. Performance analysis of Infiltration Attack Detection.

TABLE 8. PortScan Attack Detection (which includes the attack label "PortScan").

Sl. No.	Attack Detection Method	Accuracy	Precision	Recall	F1-Score	Detection Rate
1	DBNIDS[@]	97.71%	96.12%	96.24%	0.97	96.30%
2	SVMIDS[#]	96.65%	95.43%	95.05%	0.97	95.10%
3	RNNIDS[*]	94.43%	93.73%	94.39%	0.94	92.49%
4	SNNIDS[$]	91.01%	89.10%	89.05%	0.92	88.01%
5	FNNIDS[$$]	89.12%	88.33%	88.03%	0.90	87.31%

tables 7 and 8. These labels were not equivalent with the characteristics and conduct of the other labels. The infiltration attack had 36 instances with a 0.001% prevalence ratio which was the lowest prevalence of the total instances. The performance analyses of both the labels are represented in figures 8 and 9.

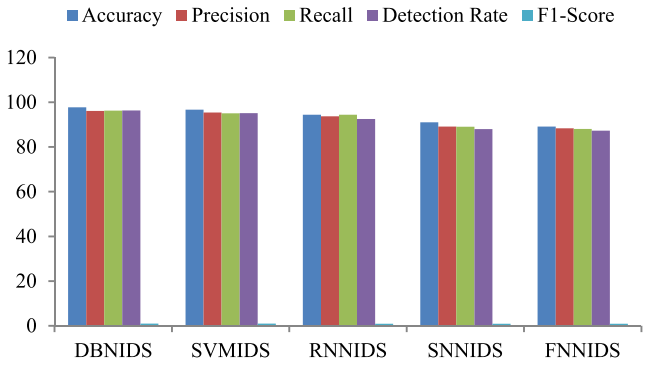

FIGURE 9. Performance analysis of PortScan Attack Detection.

TABLE 9. Web Attack Detection (which includes the attack label "Web Attack – Brute Force, Web Attack – SQL Injection & Web Attack – XSS").

Sl. No.	Attack Detection Method	Accuracy	Precision	Recall	F1-Score	Detection Rate
1	DBNIDS[@]	98.37%	97.21%	98.34%	0.97	98.31%
2	SVMIDS[#]	97.45%	96.32%	97.15%	0.96	97.12%
3	RNNIDS[*]	96.00%	94.93%	95.59%	0.97	95.50%
4	SNNIDS[$]	91.00%	90.05%	90.25%	0.93	90.11%
5	FNNIDS[$$]	90.35%	89.04%	89.13%	0.90	89.01%

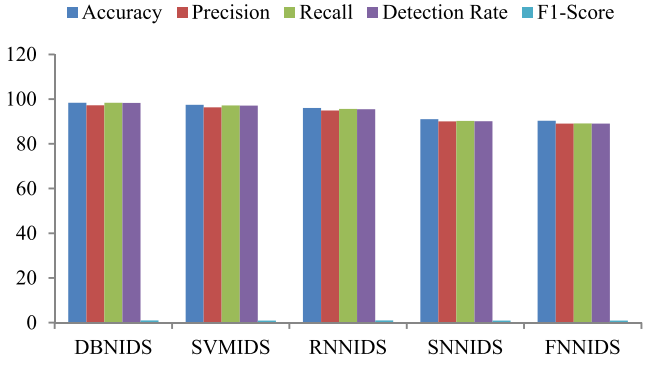

FIGURE 10. Performance analysis of Web Attack Detection.

The PortScan label had 158,930 instances with a 5.61%prevalence ratio with respect to the total instances. The proposed method on both the attacks performed better in all the parameters with an accuracy of 96.37 for infiltration attack and 97.71% for PortScan attack, and enhanced performance was evident for all the proposed parameters.

The Web Attack label included Web Attack-SQL Injection, Web Attack-Brute Force, and Web Attack-XSS, with 2,180 instances and a 0.07% prevalence ratio. Compared to the other attack labels, the proposed method achieved high performance results for all the proposed parameters. Figure.10 represents the performance analysis for Web attack detection. For the Normal attack labels the present technique accomplished 99.37% accuracy, and for the Web attack label the model accomplished 98.37% accuracy, as shown in table. 9, which are the highest performance levels obtained from this research.

VII. CONCLUSION

In this research, different types of attack and anomalies based on an intrusion detection system in the IoT were proposed and discussed. In evaluating the performance of the proposed deep learning model DBN-IDS system we used the CICIDS dataset for detection of attacks. Different attacks were presented in this dataset with many labels and numbers of attacks. In this paper we discussed the dataset in detail for the performance evaluation. DoS/DDoS, Botnet, Brute Force, Web Attack, Infiltration, and PortScan are types of attacks present in this dataset that could cause IoT system failure. The evaluation parameters utilized in the analysis were accuracy, recall, precision, detection rate, and F1-score. The proposed model obtained better results in terms of all parameters compared with the existing techniques. In future, the proposed IDS can be extended to detect other types of attacks against the IoT's systems, and various intrusion detection datasets. In addition, this proposed method can be used not only in intrusion detection, but also in classification and recognition.

REFERENCES

[1] K. P. Keyur and M. P. Sunil, "Internet of Things-IOT: Definition, characteristics, architecture, enabling technologies, application & future challenges," *Int. J. Eng. Sci. Comput.*, vol. 6, no. 5, pp. 6122–6131, 2016.

[2] 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," *Future Gener. Comput. Syst.*, vol. 85, pp. 88–96, Aug. 2018, doi: 10.1016/j.future.2018.03.007.

[3] M. Ali Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," 2018, *arXiv:1807.11023*. [Online]. Available: http://arxiv.org/abs/1807.11023

[4] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[5] M. Fahim and A. Sillitti, "Anomaly detection, analysis and prediction techniques in IoT environment: A systematic literature review," *IEEE Access*, vol. 7, pp. 81664–81681, 2019.

[6] A. Gurina and V. Eliseev, "Anomaly-based method for detecting multiple classes of network attacks," *Information*, vol. 10, no. 3, p. 84, Feb. 2019, doi: 10.3390/info10030084.

[7] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, Jun. 2019, doi: 10.1016/j.neucom.2019.02.056.

[8] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, pp. 31711–31722, 2019.

[9] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet Things*, vol. 7, Sep. 2019, Art. no. 100059.

[10] M. Roopak, G. Yun Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 0452–0457.

[11] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019.

[12] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.

[13] R. Vijayanand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Comput. Secur.*, vol. 77, pp. 304–314, Aug. 2018, doi: 10.1016/j.cose.2018.04.010.

[14] F. Hussain, R. Hussain, S. Ali Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," 2019, *arXiv:1904.05735*. [Online]. Available: http://arxiv.org/abs/1904.05735

[15] T. Mohamed, T. Otsuka, and T. Ito, "Towards machine learning based IoT intrusion detection service," in *Recent Trends and Future Technology in Applied Intelligence*, vol. 10868. Springer, 2018, pp. 580–585.
[16] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, Mar. 2019, doi: 10.3390/electronics8030322.

S. MANIMURUGAN (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees in computer science and engineering from Anna University, India.

He is currently working with the Computer Engineering Department, Faculty of Computers and Information Technology, University of Tabuk, Tabuk, Saudi Arabia. He has published nearly more than 70 research articles in several international and national forums, which include various ISI, Clarivate Analytics, Scopus, and IEEE indexed international conferences as well. His research areas are image processing, information security, visual cryptography, the IoT, and steganography.

Dr. Manimurugan is a Life Member of the Indian Society for Technical Education (MISTE). He also has been a celebrated an Editor and a Reviewer for many international journals, like Elsevier and Springer.

SAAD AL-MUTAIRI (Member, IEEE) received the B.Sc. degree from Al-Ahliyya Amman University, Jordon, and the M.Sc. and Ph.D. degrees from De Montfort University, U.K.

He is currently working as the Dean of the Deanship of the Information Technology, University of Tabuk, Saudi Arabia. His research interests are software engineering, context aware systems, cloud computing, cyber security, and steganography. He has published ample of articles in international refereed journals and conferences in his research areas.

MAJED MOHAMMED ABOROKBAH received the B.Sc. degree from Taif University, Saudi Arabia, the M.Sc. degree from Bradford University, U.K., and the Ph.D. degree from De Montfort University, U.K.

He has established the Robotics Center, University of Tabuk, Saudi Arabia, where he is currently the Dean of the Faculty of Computers and Information Technology. His research is in the areas of software engineering, context aware systems, cyber security, and steganography. He has published many articles in international journals and conferences, has organized various workshops and conferences in his research areas.

NAVEEN CHILAMKURTI (Senior Member, IEEE) received the Ph.D. degree from La Trobe University.

He is currently an Acting Head of Department, Computer Science and Computer Engineering, La Trobe University, Melbourne, VIC, Australia. He has published about 165 Journal articles and conference papers. His current research areas include intelligent transport systems (ITS), wireless multimedia, and wireless sensor networks.

He currently serves on the editorial boards of several international journals. He is an Associate Editor of the Wiley IJCS, SCN, Inderscience JETWI, and IJIPT. He is also the Inaugural Editor-in-Chief for the *International Journal of Wireless Networks and Broadband Technologies* launched, in July 2011.

SUBRAMANIAM GANESAN (Senior Member, IEEE) received the Ph.D. degree from the Indian Institute of Science, Bengaluru, India.

He is currently a Professor with the Department of Electrical and Computer Engineering, Oakland University, Rochester, MI, USA. He has over 30 years of teaching and research experience in digital computer systems. He was the Chair of the CSE Department, from 1991 to 1998. He worked with the National Aeronautical Laboratory, India, Ruhr University, Germany, Concordia University, Canada, and Western Michigan University, before joining at Oakland University. More than twenty students have obtained the Ph.D. degree under his guidance and eight students are currently doing the Ph.D. degree. He published a book on Java, in 2003. He developed a custom DSP board with software for his DSP book. He has published over 100 journal articles, more than 200 articles in conference proceedings, and three books.

Dr. Ganesan is a Senior Member of the IEEE Computer Society Distinguished Visiting Speaker, the IEEE Region four Technical Activities Member, and a fellow of ISPE. He received the Lifetime Achievement Award from ISAM, the Lloyd L. Withrow Distinguished Speaker Award from SAE, the Best Teacher Award from ASEE, the Best Teacher Award from Oakland University, the Best Service Award from ISPE, the Best Professor Award from ASDF, and the Albert Nelson Marquis Lifetime Achievement Award from Marquis Who's Who. He has been endorsed by Marquis Who's Who as a Leader in the fields of electrical and computer engineering.

RIZWAN PATAN received the B.Tech. and M.Tech. degrees from Jawaharlal Nehru Technological University Anantapur, India, in 2012 and 2014, respectively, and the Ph.D. degree in computer science and engineering from VIT University, Vellore, India, in 2017.

He was a Former Assistant Professor with the School of Computing Science and Engineering, Galgotias University, Delhi, India, from 2017 to 2019. He has been an Assistant Professor with the Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India, since 2019. He has published more than 20 SCI articles and 10 unpaid in the renowned Scopus indexed journals and presented 15 articles in national and international conferences, published book chapters in CRC Press, IGI global, Elsevier, and Edited as books. He has 20 Indian patents and one USA patent.

Dr. Rizwan apart from the academic stint, he has received many awards and accolades, which include the World Research Council and United Medical Council in the title of the Innovative Researcher on Big Data and IoT, in 2019. He is a Guest Editor of the *International Journal of Grid and Utility Computing* (Inder-science), Recent Patents on the *Computer Science*, the *Information Medical Unlock* (Elsevier), and the *Neural Computing and Applications* (Springer).