Intrusion Detection Framework for the Internet of Things Using a Dense Random Neural Network

Shahid Latif, *Student Member, IEEE*, Zil e Huma, Sajjad Shaukat Jamal, Fawad Ahmed, Jawad Ahmad, *Senior Member, IEEE*, Adnan Zahid, Kia Dashtipour, Muhammad Umar Aftab, Muhammad Ahmad, and Qammer Hussain Abbasi, *Senior Member, IEEE*

***Abstract*—The Internet of Things (IoT) devices, networks, and applications have become an integral part of modern societies. Despite their social, economic, and industrial benefits, these devices and networks are frequently targeted by cybercriminals. Hence, IoT applications and networks demand lightweight, fast, and flexible security solutions to overcome these challenges. In this regard, artificial-intelligence-based solutions with Big Data analytics can produce promising results in the field of cybersecurity. This article proposes a lightweight dense random neural network (DnRaNN) for intrusion detection in the IoT. The proposed scheme is well suited for implementation in resource-constrained IoT networks due to its inherent improved generalization capabilities and distributed nature. The suggested model was evaluated by conducting extensive experiments on a new generation IoT security dataset ToN_IoT. All the experiments were conducted under different hyperparameters and the efficiency of the proposed DnRaNN was evaluated through multiple performance metrics. The findings of the proposed study provide recommendations and insights in binary class and multiclass scenarios. The proposed DnRaNN model attained attack detection accuracy of 99.14% and 99.05% for binary class and multiclass classifications, respectively.**

***Index Terms*—Cybersecurity, deep learning, dense random neural network (DnRaNN), Internet of Things (IoT), intrusion detection.**

Manuscript received September 23, 2021; revised November 1, 2021; accepted November 18, 2021. Date of publication November 24, 2021; date of current version June 13, 2022. This work was supported in part by EPSRC IAA under Award EP/R511705/1. The work of Sajjad Shaukat Jamal was supported by the Deanship of Scientific Research at King Khalid University through research groups program under Grant R.G.P. 2/48/42. Paper no. TII-21-4112. *(Corresponding author: Jawad Ahmad.)*

Shahid Latif, Jawad Ahmad, and Kia Dashtipour are with the School of Computing, Edinburgh Napier University, EH10 5DT Edinburgh, U.K. (e-mail: shahid.latif@ieee.org; j.ahmad@napier.ac.uk; K.Dashtipour@napier.ac.uk).

Zil e Huma is with the Department of Electrical Engineering, Institute of Space Technology, Islamabad 44000, Pakistan (e-mail: zilehuma@mail.ist.edu.pk).

Sajjad Shaukat Jamal is with the Department of Mathematics, College of Science, King Khalid University, Abha 61413, Saudi Arabia (e-mail: shussain@kku.edu.sa).

Fawad Ahmed is with the Department of Cyber Security, Pakistan Navy Engineering College, National University of Sciences and Technology, Karachi 75350, Pakistan (e-mail: fawad@pnec.nust.edu.pk).

Adnan Zahid is with the School of Engineering and Physical Sciences, Heriot-Watt University, EH14 4AS Edinburgh, U.K. (e-mail: a.zahid@hw.ac.uk).

Muhammad Umar Aftab and Muhammad Ahmad are with the Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan (e-mail: umar.aftab@nu.edu.pk; dr.ahmad@nu.edu.pk).

Qammer Hussain Abbasi is with the James Watt School of Engineering, University of Glasgow, G12 8QQ Glasgow, U.K. (e-mail: qammer.abbasi@glasgow.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3130248.

Digital Object Identifier 10.1109/TII.2021.3130248

I. Introduction

THE Internet of Things (IoT) can be generally described as an extensive network of interconnected smart devices that offer digital services to individuals and industries [1], [2]. The IoT plays a significant role in modern industries to acquire real-time information through multiple sensors and actuators. The worldwide acceptance of the IoT in modern industries is revolutionizing the current industry trends in terms of information collection, data analysis, and efficient monitoring of industrial operations. The IoT can enhance the productivity and efficiency of the smart industry through intelligent decision making and remote management. However, the rapid growth of IoT networks can increase security and privacy challenges. In 2016, a cyberattack was launched on power substations in Ukraine [3]. The supervisory control and data acquisition (SCADA) system of the smart grid was illegally accessed by the attackers through the information technology network. As a result, more than 225 000 general consumers faced a complete power blackout. Another attack reported in 2016 was a huge distributed denial-of-service (DDoS) attack that compromised accessibility of the Internet on the United States East Coast [4]. It was a malicious program that replicated itself by finding, attacking, and infecting vulnerable IoT devices.

The analysis of security and privacy risks in the IoT and the potential solutions to address these challenges became an emerging topic in the field of cybersecurity [5]. In existing networks, intrusion detection systems (IDSs) are the most commonly used security models. The IDS monitors real-time Internet traffic and identifies malicious activities within the traffic. The IDS can be classified as anomaly based or signature based. The anomaly based IDS analyzes the Internet traffic and compares it to the previously learned patterns to detect malicious activities [6]. On the other hand, the signature-based IDS identifies the intrusions by finding the relationship between the previously learned signatures of known attacks. The anomaly based IDS can identify recent and well-known attacks, but they often have a greater risk of false-positive rates. Signature-based methods are not generally effective IDS to detect new and unknown and threats [7].

In recent years, machine learning (ML) and deep learning (DL) techniques have been used for a variety of security solutions in IoT applications. ML/DL-based solutions with Big Data analytics can produce significant results in the field of cybersecurity. In several studies, DL-based algorithms have gained considerable attention over conventional learning methods because of their better pattern extraction abilities. Modern IoT networks necessitate lightweight security solutions due to the resource-constrained nature of IoT devices. Prior research findings that include evaluation of the suggested solutions using outdated cyber security datasets do not meet modern security criteria. To address these challenges, we propose a novel DL scheme for intrusion detection in an IoT environment. Following are the major contributions of this article.

1) It proposes a novel dense random neural network (DnRaNN) for intrusion detection and classification in IoT networks.
2) The proposed scheme is evaluated by conducting extensive experiments on the new generation IoT security dataset ToN_IoT.
3) The performance of the DnRaNN is evaluated in both binary class and multiclass scenarios through multiple performance evaluation parameters including accuracy, precision, recall, and F1 score.
4) Finally, it presents a detailed performance comparison of the proposed scheme with some well-known ML/DL-based intrusion detection.

The rest of this article is organized as follows. Section II presents a detailed overview of some recent intrusion detection schemes for IoT applications. Section III comprises research methodology that includes the mathematical modeling of a DnRaNN, and details of the proposed architecture. Section IV describes the implementation platform, dataset description, simulations, and discussion of results. Finally, Section V concludes this article.

II. Related Works

Cyberattack detection in IoT networks is generally a classification problem. ML/DL techniques can effectively solve this problem. This section presents an overview of some recent ML/DL-based studies for intrusion detection in the IoT.

An efficient attack detection framework significantly contributes to network security. False data injection (FDI) attack is one of the dangerous security problems in IoT networks. This attack aims to mislead IoT application platforms by falsifying the sensor's data from the perception layer. Aboelwafa *et al.* [8] presented a novel autoencoder for FDI attack detection in IoT networks. Researchers exploited the sensor data correlation in time and space, which can be helpful to identify false data. The results of the experiments demonstrate that the suggested scheme identified the FDI attacks with a higher accuracy. In IoT environments, ML approaches can play a significant role in ensuring security and authorization. In this context, Makkar *et al.* [9] developed an ML-based security framework for spam detection in IoT devices. In the proposed technique, researchers evaluated five ML models by using different parameters. The spam score is calculated by using the refined input features. The authors conducted extensive experiments on the REFIT smart home dataset to validate the efficiency of the proposed scheme. Jia *et al.* [10] presented a novel edge-centric ML-based IoT defense system for DDoS attacks on the IoT. The proposed model is used for the detection, identification, and classification of DDoS attacks in IoT environments. The authors generated a large dataset by using DDoS simulators; the SlowHTTPTest, BoNeSi along with the CICDDoS2019 dataset. They compared the proposed scheme with four well-known ML models. The suggested approach outperformed previous state-of-the-art DDoS attack detection systems, according to experimental data.

Several techniques for cyberattack detection and classification have been developed, but very few of them have focused on reducing the complexity of ML models. In the latest study, Huong *et al.* [11] proposed a novel edge cloud framework that performs attack detection on the edge layer. The proposed multi-attack detection mechanism named LocKedge reduces the complexity for deployment in resource-constrained devices while maintaining a higher accuracy. To analyze the effectiveness of the proposed model from multiple aspects, LocKedge was implemented in federated and centralized learning manners. Researchers analyzed the performance of the suggested framework using the BoT-IoT dataset. In experimental findings, the proposed approach demonstrated better performance than eight well-known ML/DL algorithms. In another new work, Bokka *et al.* [12] developed a deep neural network for cyberattack detection in the smart home. The authors utilized a DS2OS dataset in their experiments. The proposed scheme successfully detected denial-of-service, malicious control, and spying attacks in a smart home environment with higher accuracy, precision, and F1 scores. Smart industries are employing SCADA systems with their plants. However, the integration of SCADA with an industrial system can be vulnerable to multiple cyberattacks. Lu *et al.* [13] developed a deep belief network (DBN) with population extremal optimization (PEO) for anomaly detection in SCADA-based industrial control systems. In the proposed approach, PEO is utilized to determine the optimal parameters of the neural network. An ensemble learning approach is introduced for the accumulation of the DBN. The proposed model is evaluated on the water storage tank system dataset and the gas pipeline system dataset from the SCADA network traffic. DDoS attacks can destabilize the complete IoT system by sending malicious requests over the network. In other recent research, Rehman *et al.* [14] introduced a new attack detection approach called DIDDOS to detect real-time DDoS attacks. Researchers utilized recurrent neural networks (RNN), gated recurrent units, and sequential minimal optimization to detect and identify real-time DDoS attacks on IoT networks. The proposed framework is evaluated through several performance parameters including accuracy, precision, recall, and F1 scores.

The majority of the aforementioned studies discussed the validation of the proposed techniques utilizing datasets from previous generations. Most of these datasets are devoid of information concerning recent and specified ranges of threats. Another main challenge with the existing techniques is the compatibility of security solutions with resource-constrained IoT networks. To address these challenges, a lightweight dense random neural network (DnRaNN) is proposed. Extensive experimentation is conducted on the latest generation IoT security dataset ToN_IoT. Multiple performance parameters are defined to assess the efficiency and effectiveness of the proposed technique.

III. Research Methodology

This section discusses the detailed mathematical model of the proposed DnRaNN and highlights the performance evaluation metrics.

A. Random Neural Network (RaNN)

The RaNN was introduced by Gelenbe in 1989 [15]. An RaNN mimics the behavior of biological neurons with its "integrate and fire" system. The RaNN is used in several applications including image processing, optimization, communication systems, cybersecurity, classification, and pattern recognition [16]. The RaNN has several key advantages. First, it can efficiently represent signal transmission through the human brain. It provides easy-to-understand complex stochastic behaviors among neurons through simple mathematical modeling. Second, it has better generalization capabilities because of its probability constraints. Third, it has established mathematical properties that can simplify complex computations [16]. Fourth, it is an ideal algorithm for deployment in resource-constrained hardware and IoT devices because of its highly distributed nature [17], [18].

1) Mathematical Model of RaNN: Consider an RaNN model comprised of N neurons. Each neuron receives excitatory and inhibitory spikes from some external sources that can be sensors or cells. The arrival occurs to the cell $n \in \{1, \ldots, N\}$. The rate of occurrence for excitatory and inhibitory spikes can be denoted as rates $\psi_n^+$and ψ_n^-, respectively.

In the RaNN model, each neuron can be represented by its internal state $s_n(t)$ at time $t \geq 0$. If $s_n(t) > 0$. The arrival of an inhibitory spike to neuron n at time t can then cause the reduction of the internal state by one unit $s_n(t^+) = s_n(t) - 1$. If $s_n(t) = 0$, then the arrival of an inhibitory spike to a cell has no impact. On the other hand, the arrival of an excitatory spike will always increase the $s_n(t)$ by $+1$.

The neuron n is "excited" if its internal state $s_n(t) > 0$. It can "fire" a spike with probability $\mathcal{R}_n \Delta t$ in the interval $[t, t + \Delta t]$. Here, $\mathcal{R}_n > 0$ is its "firing rate," such that $\mathcal{R}_n^{-1}$ is the average firing delay of the excited nth neuron.

At time $t \geq 0$, neurons can interact in the following manners. If neuron α is excited, such that $s_\alpha(t) > 0$, then its internal state drops by 1, and we have $s_\alpha(t^+) = s_\alpha(t) - 1$. Neuron α can either send an excitatory spike to neuron β with probability $\varphi^+(\alpha, \beta)$, which results in $s_\alpha(t^+) = s_\alpha(t) - 1$ and $s_\beta(t^+) = s_\beta(t) + 1$, or it can send an inhibitory spike to neuron j with probability $\varphi^-(\alpha, \beta)$ so that $s_\beta(t^+) = s_\beta(t) + 1$ and $s_\beta(t^+) = s_\beta(t) - 1$ if $s_\beta(t) > 0$, else $s_\beta(t^+) = 0$ if $s_\beta(t) = 0$. In another case neuron α can "trigger" neuron β with probability $\varphi(\alpha, \beta)$ so that $s_\alpha(t^+) = s_\alpha(t) - 1$ and $s_\beta(t^+) = s_\beta(t) - 1$ if $s_\beta(t) > 0$.

When the neuron α triggers neuron β, both $s_\alpha(t^+) = s_\alpha(t) - 1$ and $s_\beta(t^+) = s_\beta(t) - 1$, and one of two things may happen. First, we have $s_n(t^+) = s_n(t) + 1$ with probability $\delta(\beta, n)$ so that neurons α and β together have incremented the state of neuron n. A trigger allows two neurons α and β to increase the excitation level of a third neuron n by $+1$, while neurons α and β are both depleted by -1. Second, the trigger moves on to the neuron n with probability $\sigma(\beta, n)$, and then, this sequence will be repeated.

Note that

$$\sum_{\beta=1}^{N} \left[\varphi(\alpha, \beta) + \varphi^-(\alpha, \beta) + \varphi^+(\alpha, \beta)\right] = 1 - \ell_\alpha. \quad (1)$$

When neuron α fires, the corresponding spike is lost, or it leaves the network with the probability ℓ_α.

Cells in the different layers of the human brain communicate with each other through simultaneous firing patterns of densely packed somas. An extended work of the RaNN algorithm was presented in [19] and [20], using a branch theory of stochastic networks. In the following, we will exploit this framework for DL.

2) Modeling of Soma-to-Soma Interactions: Let $\mu(n) = (\alpha_1, \ldots, \alpha_m)$ be any ordered sequence of distinct numbers where $\alpha_\beta \in \{1, \ldots, N\}$ and $\alpha_\beta \neq n$; obviously $2 \leq m \leq N - 1$

$$\gamma_n = \lim_{t \to \infty} \left[s_n(t) > 0\right]. \quad (2)$$

The probability of excitation for the neuron n can be described by the following expression:

$$\gamma_n = \frac{\varepsilon_n^+}{\mathcal{R}_n + \varepsilon_n^-} \quad (3)$$

where the variables in (3) are of the form

$$\varepsilon_n^+ = \psi_n^+ + \sum_{\beta=1, \beta \neq n}^{N} \mathcal{R}_\beta \gamma_\beta \varphi^+(\beta, n) + \sum_{\mu(n)} \mathcal{R}_{\alpha_1} \prod_{\beta=1,\ldots,m-1} \gamma_{\alpha_\beta} \varphi\left(\alpha_\beta, \alpha_{\beta+1}\right) \gamma_{\alpha_m} \delta\left(\alpha_m, n\right) \quad (4)$$

$$\varepsilon_n^- = \psi_n^- + \sum_{\beta=1, \beta \neq n}^{M} \mathcal{R}_\beta \gamma_\beta \varphi^-(\beta, n) + \sum_{\mu(n)} \mathcal{R}_{\alpha_1} \prod_{\beta=1,\ldots,m-1} \gamma_{\alpha_\beta} \varphi\left(\alpha_\beta, \alpha_{\beta+1}\right) \gamma_{\alpha_m} \varphi\left(\alpha_m, n\right). \quad (5)$$

Here, we set $\varphi(\alpha, \beta) = \sigma(\alpha, \beta)$ to simplify the expression. The simplified equation can be presented as follows:

$$\omega_{\beta\alpha}^+ = \mathcal{R}_\mathcal{R} \varphi^+(\beta, \alpha) \quad (6)$$

and

$$\omega_{\beta\alpha}^- = \mathcal{R}_\mathcal{R} \varphi^-(\beta, \alpha). \quad (7)$$

3) Clusters of Densely Connected Cells: Now, we consider the construction of clusters of densely interconnected cells. First, in a network $D(d)$ that contains d identically connected cells, the external excitatory and inhibitory arrival of spikes is represented by $\psi^+$and ψ^-. Each connected cell has a firing rate $\mathcal{R}$ and its state is denoted by γ. Each cell receives inhibitory input from the state of some cell $\hbar$, which does not belong to $D(d)$. Thus, for any cell $h \in D(d)$, we have an inhibitory weight $\omega_{\hbar}^- \equiv \omega_{\hbar,h}^- > 0$ from $\hbar$ to h.

For any $\alpha, \beta \in D(d)$, we have $\omega_{\alpha,\beta}^+ = \omega_{\alpha,\beta}^- = 0$, but whenever one of the cells fires, it triggers the firing of the other cells with $\varphi(\alpha,\beta) = \frac{\varphi}{d}$ and $\delta(\alpha,\beta) = \frac{1-\varphi}{d}$. As a result, we have

$$\gamma = \frac{\psi^+ + \mathcal{R}\gamma(d-1)\sum_{m=0}^{\infty}\left[\frac{\gamma\varphi(d-1)}{d}\right]^m \frac{1-\varphi}{d}}{r + \psi^- + \gamma_{\hbar}\omega_{\hbar}^- + \mathcal{R}\gamma(d-1)\sum_{m=0}^{\infty}\left[\frac{\gamma\varphi(d-1)}{d}\right]^m \frac{\varphi}{d}} \tag{8}$$

This can be reduced to

$$\gamma = \frac{\psi^+ + \frac{\mathcal{R}\gamma(d-1)(1-\varphi)}{d-\gamma\varphi(d-1)}}{\mathcal{R} + \psi^- + \gamma_{\hbar}\omega_{\hbar}^- + \frac{\mathcal{R}\gamma\varphi(d-1)}{d-\gamma\varphi(d-1)}} \tag{9}$$

where the aforementioned equation can be represented as a second-degree polynomial in terms of γ as follows:

$$\gamma^2\varphi(d-1)\left[\psi^- + \gamma_{\hbar}\omega_{\hbar}^-\right] + \gamma(d-1)\left[\mathcal{R}(1-\varphi) - \psi^+\varphi\right] - \gamma d\left(\mathcal{R} + \psi^- + \gamma_{\hbar}\omega_{\hbar}^-\right) + \psi^+ d = 0. \tag{10}$$

Under the conditions that $d \geq 2, \varphi \leq 1, \psi^+ > 0, \psi^- > 0$, and $\mathcal{R} > 0$ and $\psi^- \geq \psi^+$, this equation can easily be solved for its positive roots, where one of them is less than 1. This is the only one of interest because γ is a probability.

4) RaNN With Multiple Clusters: In this subsection, we describe the construction of a multiple cluster-based DL architecture. This DL framework contains K clusters $D(d)$ each with d hidden cells. For the kth cluster, $k = 1, \ldots, K$, the state of each of its identical cells is denoted by γ_k. In addition, U input cells do not belong to these K clusters and the state of the lth cell $l = 1, \ldots, L$ is denoted by $\bar{\gamma}_l$.

Each hidden cell in the clusters k, with $k \in \{1, \ldots, K\}$, receives inhibitory input from each of the L input cells. Thus, for each cell in the kth cluster, we have inhibitory weights $\omega_{l,k}^- > 0$ from the lth input cell to each cell in the kth cluster. The lth input cell will have a total inhibitory "exit" weight or total inhibitory firing rate $\mathcal{R}_l^-$to all of the clusters, which are of value

$$\mathcal{R}_l^- = d\sum_{k=1}^{K}\omega_{l,k}^-. \tag{11}$$

Then, from (9) and (10), we have

$$\gamma_k = \frac{\psi_k^+ + \frac{\mathcal{R}_k\gamma_k(d-1)(1-\varphi_k)}{d-\gamma_k\varphi_k(d-1)}}{\mathcal{R}_k + \psi_k^- + \sum_{l=1}^{L}\bar{\gamma}_l\omega_{l,k}^- + \frac{\mathcal{R}_k\gamma_k\varphi_k(d-1)}{d-\gamma_k\varphi_k(d-1)}} \tag{12}$$

yielding a second-degree polynomial for each of the γ_k as

$$w_k\gamma_k^2 + x_k\gamma_k + \gamma_k = 0 \tag{13}$$

where

$$w_k = \varphi_k(d-1)\left(\psi_k^- + \sum_{l=1}^{L}\bar{\gamma}_l\omega_{l,k}^-\right) \tag{14}$$

$$x_k = z_k - d\left(\psi_k^- + \sum_{l=1}^{L}\bar{\gamma}_l\omega_{l,k}^-\right) \tag{15}$$

$$y_k = d\psi_k^+ \tag{16}$$

$$z_k = \psi_k^+\varphi + \mathcal{R}_k\varphi - \psi_k^- d - \mathcal{R}_k - \psi_k^+\varphi_k d - d\varphi_k\mathcal{R}_k. \tag{17}$$

Its positive root can be described as

$$\xi_k = \frac{-(z_k - d\eta) - \sqrt{(e_k - d\eta)^2 - 4\varphi_k(d-1)(\psi^- + \eta)\,y_k}}{2\varphi_k(d-1)\left(\psi_k^- + \eta\right)} \tag{18}$$

where

$$\eta = \sum_{l=1}^{L}\bar{\gamma}_l\omega_{l,k}^-. \tag{19}$$

When all the parameters $d, \varphi_c = \varphi, \psi_c^+ = \psi^+, \psi_c^- = \psi^-, \mathcal{R}_k = \mathcal{R}, z_k = z$, and $y_k = y$, with $k = 1, \ldots, K$ are the same for all of the clusters, we will have

$$\xi(\eta) = \frac{-(z_k - d\eta) - \sqrt{(z - d\eta)^2 - 4\varphi(d-1)(\psi^- + \eta)\,y}}{2\varphi(d-1)(\psi^- + \eta)}. \tag{20}$$

The resulting activation function given by (20) is highly nonlinear and goes beyond the capabilities of conventional activation functions like the Sigmoid, Tansig, and Sine functions. The activation of the cluster (20) depends on parameters $d, \varphi, \psi^+, \psi^-$, and $\mathcal{R}$. The curves of $\xi(\eta)$ versus η under different parameters for a cluster with 20 cells satisfy the conditions derived for (10) ($d \geq 2, \varphi \leq 1, \psi^+ > 0, \psi^- > 0, \mathcal{R} > 0$, and $\psi^- \geq \psi^+$).

B. Proposed DnRaNN Architecture

The main idea of a DnRaNN is extracted from the structure of the human brain. Several important areas in the human brain contain dense clusters of cells. These clusters may contain similar cells or a variety of different cells. Because of the density of their arrangement, these clusters can allow for substantial communication through synapses and dendrites [21]. Therefore, a mathematical model of dense clusters is developed that can model both soma-to-soma interactions and synapses to create a DnRaNN whose proposed framework is presented in Fig. 1. This model contains one input layer, four cluster hidden layers, and one output layer. The framework of dense clusters allows the formulation of multilayer architecture, in which each layer contains a finite number of dense nuclei. The cells in each nucleus communicate with each other in a fully connected architecture by using synapse and soma-to-soma interactions. The communication framework among the hidden layers is based on a traditional multilayer feed-forward architecture. The nuclei of the first layer receive the excitation signal from an input layer. After that, every cell in each nucleus creates an

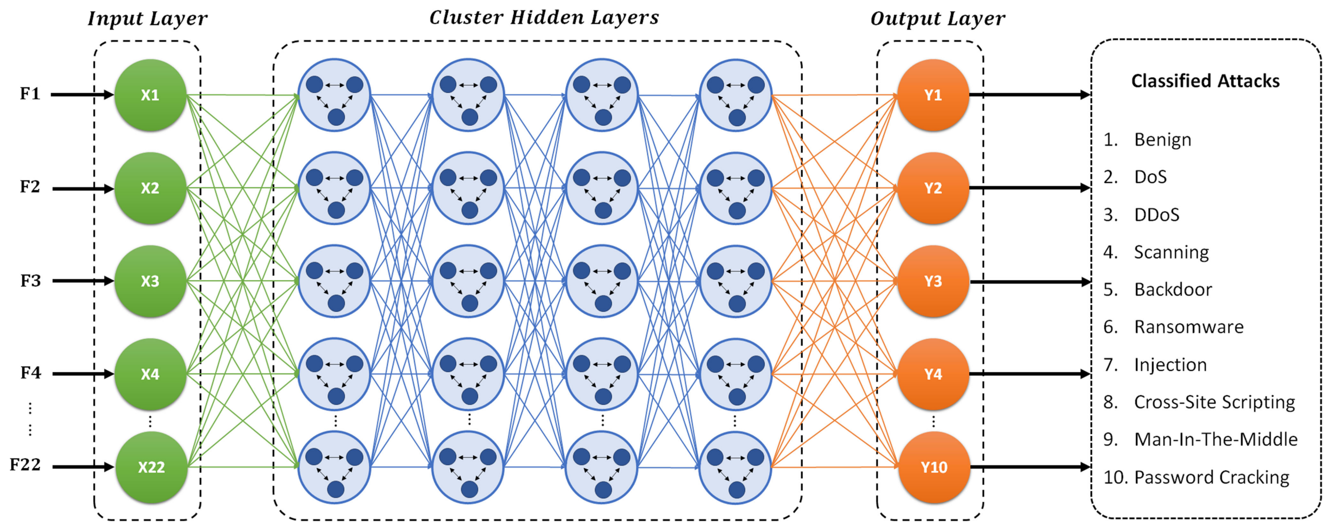

Fig. 1. Architecture of the proposed DnRaNN.

inhibitory projection to the layer above. Finally, the gradient descent algorithm efficiently trains the neural network.

C. Performance Evaluation Metrics

Several evaluation metrics such as accuracy, precision, recall, and F1 score are used to analyze the performance of the proposed DnRaNN.

1) Accuracy (AC): It determines the ratio of correctly predicted observations to the total number of observations by the IDS. It can be mathematically described as

$$\mathrm{AC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}. \tag{21}$$

2) Precision (PR): It determines the ratio of correctly predicted malicious observations to the total number of observations classified as an attack by IDS. It can be mathematically described as

$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{22}$$

3) Recall (RC): It determines the ratio of correctly predicted malicious observations to the correctly predicted malicious observations and incorrectly predicted normal observations. It can be mathematically described as

$$\mathrm{RC} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \tag{23}$$

4) F1 Score (F1): It determines the weighted average of precision and recall. It maintains the balance between PR and RC by considering positive and negative results. It can be mathematically described as

$$F1 = \frac{2\ \times(\mathrm{PR}\ \times \mathrm{RC})}{\mathrm{PR}+\mathrm{RC}}. \tag{24}$$

IV. Simulations and Results

This section presents the experimental methodology, simulations, and discussion of results.

A. Simulation Platform

The simulation and performance evaluation of the proposed DnRaNN are conducted on the Hewlett-Packard Pavilion Gaming Desktop TG01-2260xt workstation. This workstation contains an 11th generation Intel Core i7 11700 processor with 8-GB DDR4-2933 SDRAM. An Nvidia GeForce GTX 1660 Super (6-GB GDDR6) graphic card ensures the smooth execution of the simulations.

B. ToN_IoT Dataset

The ToN_IoT is a new generation open-source dataset. It was generated at the Cyber Range and IoT Labs of the University of New South Wales Australia [22]. It can be accessed from the ToN_IoT repository [23]. This dataset contains a total of 1 379 274 samples, of which 270 279 are normal and 1 108 995 are attack samples. The ten classes are benign, DoS, DDoS, scanning, backdoor, ransomware, cross-site scripting (XSS), data injection, man-in-the-middle, and password cracking attacks.

C. Dataset Preparation

Data preparation is an essential stage to prepare and clean the data before feeding it into any ML/DL algorithm, to accelerate the learning process and achieve a good accuracy. At this stage, several operations can be performed such as elimination of unnecessary features, conversion of nonnumerical features, and replacement of missing values with suitable data. In this article, we applied a two-step process for data preparation, which includes both data preprocessing and data normalization.

1) Data Preprocessing: In this stage, the categorical features with nominal values are converted into numerical values to ensure data compatibility with the input of the neural network. In this work, the categorical features are converted into numerical values by applying label encoding. Because data, time, and time stamp features have no impact and contribution on the output prediction, these columns are eliminated in preprocessing.

2) Data Normalization: Because model is biased toward large values, some features in the dataset contain larger values as compared to others, thereby it may reduce the accuracy of results. In data normalization, the data are mapped in the range between 0.0 and 1.0 without disturbing the normality of data behavior. In this work, we have used min–max scaling for data normalization [24].

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}}.$$

Here, x and y represent the original and normalized values, respectively. The minimum and maximum values of the features are represented by $x_{\max}$ and $x_{\min}$, respectively.

D. Hyperparameter Selection

Hyperparameters are the variables that determine the network structure and control the learning process. In our experiments, the main framework of the DnRaNN is fixed [25]. To ensure the best performance of the proposed DnRaNN, we determine the optimal hyperparameters by conducting extensive experiments. These hyperparameters include learning rate, momentum, number of epochs, and batch size.

1) Learning Rate: This parameter controls the learning speed of ML/DL models. A lower learning rate can help the model to learn better but can increase the training time, and the process can get stuck. On the other hand, a high learning rate can allow fast training but can cause a large output error. Therefore, the selection of an appropriate learning rate helps to ensure the optimum performance of the ML/DL model. In this article, we define five values of learning rates to obtain the best performance.

2) Momentum: This parameter helps to determine the direction of the next stage based on the knowledge of the previous stage. It helps to prevent oscillations in the model. Multiple experiments found that a suitable choice of momentum lies between 0.5 and 0.9.

3) Number of Epochs: In terms of ML/DL, an epoch refers to one cycle through the full training dataset. Usually, training an ML/DL model takes more than a few epochs. In our experiments, we conducted all the simulations for 100 epochs.

4) Batch Size: This parameter is related to the gradient descent algorithm, which controls the number of training samples to be processed before the model's internal parameters are changed. The popular default values for batch size are 32, 64, and 128.

E. Simulations and Result Analysis

The ToN_IoT dataset is split into training and testing datasets with the ratio of 80% and 20%, respectively. The class distribution of the ToN_IoT dataset is presented in Table I. The 22 most prominent features are used as input of the neural network. A detailed analysis is performed for both binary class and multiclass scenarios.

1) Result Analysis of Binary Classification: The performance of the proposed DnRaNN was first evaluated for the binary class scenario. Initially, we set the batch size as 32, momentum as 0.55, and a range of learning rates: 0.001, 0.005, 0.01, 0.10, 1.00, and 2.00. We performed a simulation for 100 epochs by keeping the batch size and momentum at fixed values and recorded the results at the aforementioned learning rates. The performance of the proposed model for the defined hyperparameters is presented in Table II. The best train and test accuracies attained are 99.22% and 99.15%, respectively, for the learning rate of 0.10. The other performance scores at this learning rate also indicate the satisfactory performance of the DnRaNN. For the learning rate of 0.001, 0.005, and 0.01, the model achieved an accuracy higher than 98%. At a higher learning rate of 1.00 and 2.00, the accuracy and other performance scores indicate a lower performance of the proposed model. A detailed performance comparison using these parameters is presented using the confusion matrices shown in Fig. 2.

At the second stage, we set the batch size as 64, momentum as 0.70, and the same range of learning rates as we had used in our first experiment. The performance using the defined hyperparameters is presented in Table III. The best train and test

TABLE I
CLASS DISTRIBUTION IN THE TON_IOT DATASET

Class Name	Total Samples	Training Samples	Testing Samples
Benign	270279	216223	54056
DoS	17717	14174	3543
DDoS	326345	261076	65269
Scanning	21467	17174	4293
Backdoor	17247	13798	3449
Ransomware	142	114	28
Injection	468539	374831	93708
Cross-Site Scripting	99944	79955	19989
Man-In-The-Middle	1295	1036	259
Password Cracking	156299	125039	31260

TABLE II
PERFORMANCE COMPARISON OF THE DNRANN FOR THE BINARY CLASS SCENARIO AT BATCH SIZE: 32, MOMENTUM: 0.55, AND EPOCHS: 100

Performance Parameters		Learning Rate					
		0.001	**0.005**	**0.01**	**0.10**	**1.00**	**2.00**
Training	**AC**	0.9834	0.9811	0.9867	**0.9922**	0.9398	0.8928
	PR	0.9905	0.9891	0.9945	0.9962	0.9620	0.9374
	RC	0.9889	0.9873	0.9889	0.9941	0.9632	0.9287
	F1	0.9897	0.9882	0.9917	0.9951	0.9626	0.9330
Testing	**AC**	0.9823	0.9842	0.9851	**0.9915**	0.9329	0.8725
	PR	0.9890	0.9904	0.9904	0.9923	0.9609	0.9231
	RC	0.9890	0.9900	0.9911	0.9907	0.9555	0.9178
	F1	0.9890	0.9902	0.9907	0.9927	0.9582	0.9205

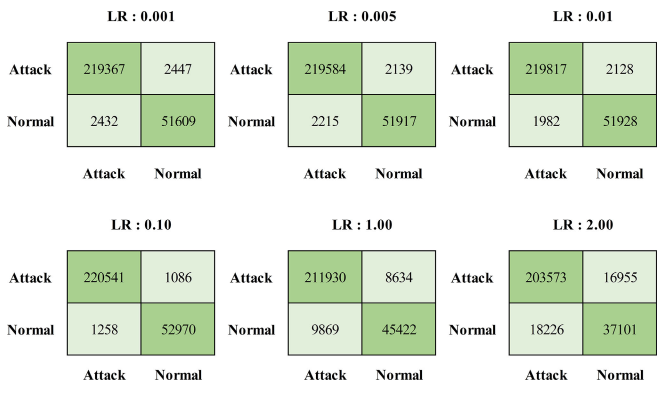

Fig. 2. Confusion matrices of the DnRaNN for binary class scenarios at batch size: 32, momentum: 0.55, and epochs: 100.

TABLE III
PERFORMANCE COMPARISON OF THE DNRANN FOR THE BINARY CLASS SCENARIO AT BATCH SIZE: 64, MOMENTUM: 0.70, AND EPOCHS: 100

Performance Parameters		Learning Rate					
		0.001	0.005	0.01	0.10	1.00	2.00
Training	AC	0.9723	0.9729	**0.9880**	0.9816	0.9325	0.9001
	PR	0.9828	0.9832	0.9935	0.9883	0.9550	0.9406
	RC	0.9827	0.9831	0.9916	0.9888	0.9614	0.9348
	F1	0.9828	0.9832	0.9925	0.9885	0.9582	0.9377
Testing	AC	0.9730	0.9783	**0.9920**	0.9803	0.9284	0.8791
	PR	0.9811	0.9863	0.9891	0.9876	0.9555	0.9232
	RC	0.9854	0.9867	0.9886	0.9880	0.9554	0.9267
	F1	0.9833	0.9865	0.9888	0.9878	0.9555	0.9250

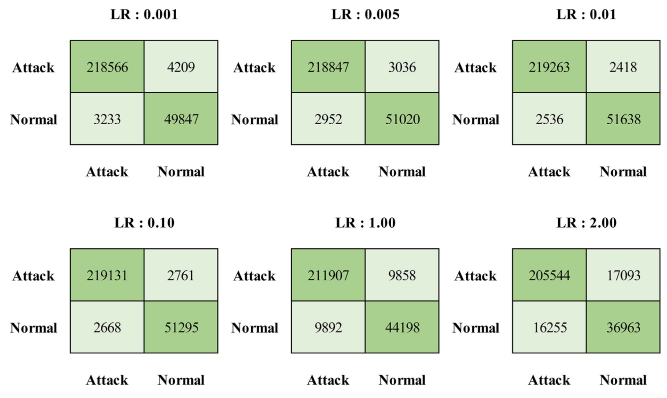

Fig. 3. Confusion matrices of the DnRaNN for binary class scenarios at batch size: 64, momentum: 0.70, and epochs: 100.

TABLE IV
PERFORMANCE COMPARISON OF THE DNRANN FOR THE BINARY CLASS SCENARIO AT BATCH SIZE: 32, MOMENTUM: 0.82, AND EPOCHS: 100

Performance Parameters		Learning Rates					
		0.001	0.005	0.01	0.10	1.00	2.00
Training	AC	0.9770	0.9725	0.9781	**0.9866**	0.9110	0.8699
	PR	0.9854	0.9841	0.9867	0.9917	0.9456	0.9206
	RC	0.9860	0.9816	0.9861	0.9916	0.9435	0.9174
	F1	0.9857	0.9829	0.9864	0.9917	0.9446	0.9190
Testing	AC	0.9721	0.9701	0.9753	**0.9817**	0.9011	0.8705
	PR	0.9825	0.9809	0.9864	0.9889	0.9434	0.9230
	RC	0.9829	0.9819	0.9829	0.9883	0.9329	0.9153
	F1	0.9827	0.9814	0.9846	0.9886	0.9381	0.9191

accuracies attained are 98.80% and 98.20%, respectively, for the learning rate of 0.01. The other performance scores at this learning rate indicate a superior performance of the DnRaNN. For the learning rates of 0.001, 0.005, and 0.10, the model achieved train and test accuracy higher than 97%. Again, for higher learning rates of 1.00 and 2.00, the accuracy and other performance scores indicate an unsatisfactory performance of the proposed model. A detailed performance comparison of these parameters is presented using the confusion matrices shown in Fig. 3.

At the third stage, we set the batch size as 32, momentum as 0.82 and the same range of learning rates as in our previous two experiments. The performance at the given hyperparameters is presented in Table IV. The best train and test accuracies are attained as 98.66% and 98.17%, respectively, for the learning rate of 0.10. The other performance scores at this learning rate indicate a good performance of the DnRaNN. For the learning rates of 0.001, 0.005, and 0.01, the model achieved train and test accuracies between 97% and 98%. Again, for the higher learning rates of 1.00 and 2.00, the accuracy and other performances were decreased. A detailed performance comparison for these

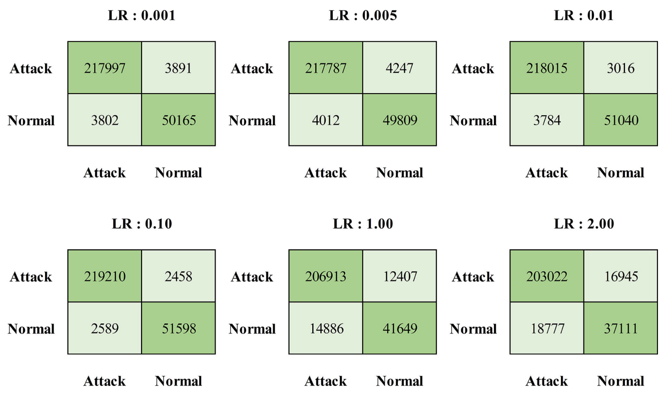

Fig. 4. Confusion matrices of the DnRaNN for binary class scenarios at batch size: 32, momentum: 0.82, and epochs: 100.

TABLE V
PERFORMANCE COMPARISON OF THE DNRANN FOR THE MULTICLASS SCENARIO AT BATCH SIZE: 32, MOMENTUM: 0.55, AND EPOCHS: 100

Performance Parameters		Learning Rate					
		0.001	0.005	0.01	0.10	1.00	2.00
Training	AC	0.9798	0.9740	**0.9911**	0.9808	0.9254	0.8677
	PR	0.9865	0.9838	0.9948	0.9879	0.9541	0.9198
	RC	0.9883	0.9839	0.9941	0.9883	0.9530	0.9152
	F1	0.9874	0.9838	0.9944	0.9881	0.9536	0.9175
Testing	AC	0.9720	0.9707	**0.9905**	0.9806	0.9145	0.8810
	PR	0.9843	0.9822	0.9913	0.9884	0.9495	0.9344
	RC	0.9808	0.9814	0.9908	0.9874	0.9438	0.9163
	F1	0.9825	0.9818	0.9917	0.9879	0.9467	0.9252

parameters is presented using the confusion matrices shown in Fig. 4.

The overall performance comparison for binary classification indicates that the proposed DnRaNN model achieved the optimum results at the learning rate of 0.10, batch size 32, and momentum 0.55.

2) Result Analysis of Multiclass Classification: As described earlier, the ToN_IoT dataset contains ten classes. Therefore, in the second phase of experimentation, we evaluated the efficiency of the proposed DnRaNN in a multiclass scenario. As in the binary class experiments, we initially set the batch size as 32, momentum as 0.55 and learning rates as 0.001, 0.005, 0.01, 0.10, 1.00, and 2.00. We executed the simulation for 100 epochs by keeping the batch size and momentum at fixed values and recorded the results at the aforementioned learning rates. The performance of the model at the predefined hyperparameters is presented in Table V. The results demonstrate that the proposed model attained a higher attack detection accuracy for the learning rate of 0.01. The best train and test accuracies are recorded as 99.11% and 99.05%, respectively. For the learning rates of 0.001, 0.005, and 0.10, the model achieved train and test accuracy between 97% and 98%. For higher learning rates of 1.0 and 2.0, the model attained low accuracy scores between 85% and 95%. As the model achieved higher performance scores at the learning rate of 0.01, we analyzed the attack detection performance for each class at this learning rate. The bar graph in Fig. 5 presents a comparative analysis of actual and predicted results of the proposed DnRaNN algorithm.

At the second stage, we set the batch size as 64, momentum as 0.70, and the same range of learning rates as in our

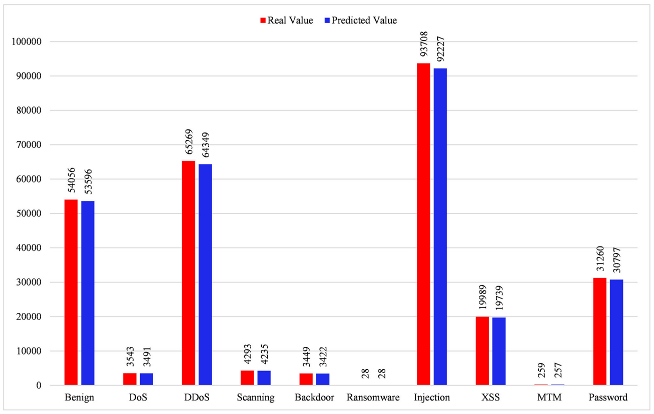

Fig. 5. Comparative analysis of multiclass classification at batch size: 32, momentum: 0.55, and epochs: 100.

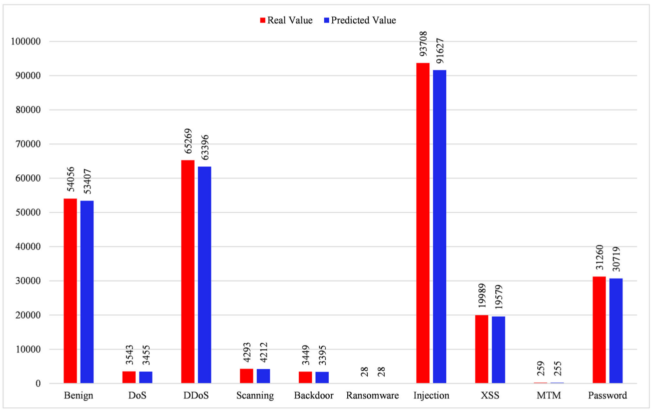

Fig. 6. Comparative analysis of multiclass classification at batch size: 64, momentum: 0.70, and epochs: 100.

TABLE VI
PERFORMANCE COMPARISON OF THE DNRANN FOR THE MULTICLASS SCENARIO AT BATCH SIZE: 64, MOMENTUM: 0.70, AND EPOCHS: 100

Performance Parameters		Learning Rate					
		0.001	**0.005**	**0.01**	**0.10**	**1.00**	**2.00**
Training	**AC**	0.9771	0.9752	**0.9907**	0.9791	0.9204	0.8793
	PR	0.9870	0.9846	0.9935	0.9856	0.9519	0.9237
	RC	0.9846	0.9845	0.9949	0.9885	0.9490	0.9264
	F1	0.9858	0.9845	0.9942	0.9870	0.9504	0.9250
Testing	**AC**	0.9723	0.9747	**0.9898**	0.9795	0.9017	0.8778
	PR	0.9846	0.9853	0.9932	0.9870	0.9340	0.9284
	RC	0.9809	0.9832	0.9942	0.9874	0.9444	0.9188
	F1	0.9827	0.9843	0.9937	0.9872	0.9392	0.9236

TABLE VII
PERFORMANCE COMPARISON OF THE DNRANN FOR THE MULTICLASS SCENARIO AT BATCH SIZE: 32, MOMENTUM: 0.82, AND EPOCHS: 100

Performance Parameters		Learning Rate					
		0.001	**0.005**	**0.01**	**0.10**	**1.00**	**2.00**
Training	**AC**	0.9735	0.9715	**0.9883**	0.9800	0.9052	0.8622
	PR	0.9851	0.9826	0.9929	0.9878	0.9445	0.9130
	RC	0.9820	0.9820	0.9926	0.9873	0.9372	0.9160
	F1	0.9835	0.9823	0.9928	0.9876	0.9408	0.9145
Testing	**AC**	0.9710	0.9741	**0.9877**	0.9795	0.8754	0.8504
	PR	0.9834	0.9824	0.9913	0.9890	0.9191	0.9103
	RC	0.9805	0.9855	0.9934	0.9855	0.9266	0.9030
	F1	0.9820	0.9839	0.9923	0.9872	0.9228	0.9066

first experiment. The performance of the proposed algorithm at predefined hyperparameters is presented in Table VI. The results demonstrate that the proposed model achieved higher attack detection accuracy for the learning rate of 0.01. The train and test accuracies are recorded as 99.07% and 98.98%, respectively. At the learning rates of 0.001, 0.005, and 0.10, the model attained train and test accuracies between 97.50% and 98%. As in the previous experiment, the model achieved lower performance scores at a higher learning rate. Again, the proposed model demonstrated a higher performance for the learning rate of 0.01, so we analyzed the attack detection performance for each class at this learning rate. The bar graph in Fig. 6 presents a comparative analysis of actual and predicted results of the proposed DnRaNN algorithm.

At the third stage, we set the batch size as 32, momentum as 0.82, and the same range of learning rates as in our previous two experiments. The performance of the proposed model at the predefined hyperparameters is presented in Table VII. The results demonstrate that the proposed model achieved a higher attack detection accuracy for the learning rate of 0.01. The best train and test accuracies are recorded as 98.83% and 98.77%, respectively. For other learning rates, 0.001, 0.005, and 0.10, the model attained train and test accuracies between 97.0% and 98%. As in the previous experiments, the model achieved lower performance scores at higher learning rates. Because our model demonstrated a higher performance at the learning rate of 0.01, we analyzed the attack detection performance for each class at this learning rate. The bar graph in Fig. 7 presents a comparative

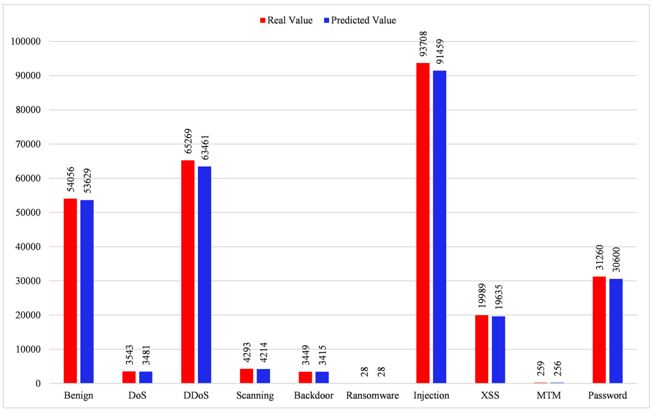

Fig. 7. Comparative analysis of multiclass classification at batch size: 32, momentum: 0.82, and epochs: 100.

analysis of actual and predicted results of the proposed DnRaNN model. In summary, the proposed DnRaNN gave a satisfactory performance with the new generation dataset ToN_IoT in both binary class and multiclass scenarios.

F. Performance Comparison With the Well-Known ML/DL-Based IDS

To analyze the effectiveness of the proposed DnRaNN further, we compared its performance with some well-known ML and DL-based algorithms for the ToN_IoT dataset. These algorithms

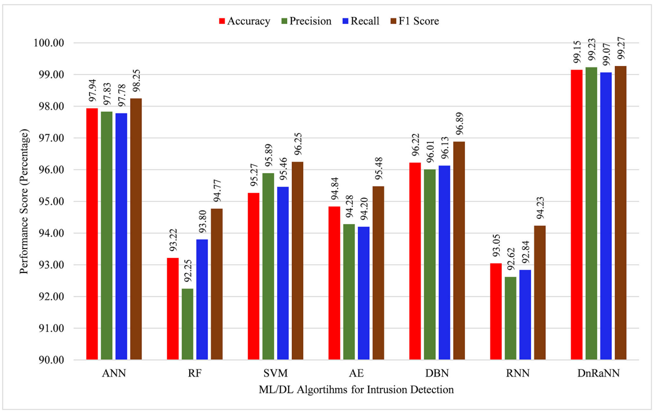

Fig. 8. Performance comparison of the state-of-the-art IDS for binary class scenario.

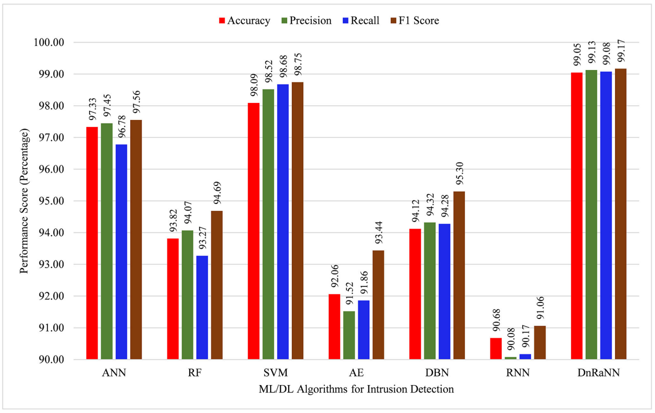

Fig. 9. Performance comparison of the state-of-the-art IDS for multiclass scenario.

include artificial neural network (ANN), support vector machine (SVM), random forest (RF), autoencoder (AE), deep belief network (DBN), and the RNN. For a better performance comparison, we maintained the training and testing dataset split ratio as 80% and 20%, respectively. All these algorithms were implemented in the Python simulation environment. We evaluated the performance of the proposed DnRaNN with other ML/DL algorithms in both binary class and multiclass scenarios. A detailed comparison for binary classification is shown in Fig. 8. According to this figure, the intrusion detection accuracy and other scores of ANN, SVM, and DBN for the ToN_IoT dataset are obtained between 95% and 97.50%. The intrusion detection performances of RF, AE, and RNN were less than 95%.

In the second phase, the performance of the ML/DL-based model is evaluated for the multiclass scenario. A brief comparison is presented in Fig. 9. The two classifiers ANN and SVM achieved a higher attack detection accuracy as greater than 96% and the performance scores for all the remaining classifiers were between 90% and 95%. In summary, the performance of the proposed DnRaNN is superior for both binary class and multiclass scenarios, as compared to the other ML/DL-based intrusion detection models.

V. Conclusion

In this article, a novel DnRaNN technique was proposed for cyberattack detection in an IoT environment. To evaluate the effectiveness of the proposed algorithm, extensive experiments were conducted on the new generation ToN_IoT dataset. We defined multiple parameters including accuracy, precision, recall, and F1 score for the performance evaluation. The efficiency of the proposed approach was evaluated in both binary and multiclass scenarios. In the binary class evaluation, the DnRaNN achieved a higher attack detection accuracy of 99.14%. For binary class scenarios, other performance scores are also higher than 99%. In multiclass evaluation, the DnRaNN obtained 99.05% attack detection accuracy. The proposed scheme successfully classified nine different attacks on the IoT with a high detection accuracy. To analyze the effectiveness of the proposed model further, its performance was compared with some well-known ML/DL-based intrusion detection schemes. The experimental findings of the proposed model proved its higher performance over other ML/DL-based schemes. To enhance the performance of cyberattack detection algorithm in hardware perspectives, a field programmable gate array (FPGA)-based accelerator can be integrated with DnRaNN.

References

[1] R. Kumar and R. Tripathi, "DBTP2SF: A deep blockchain-based trustworthy privacy-preserving secured framework in industrial Internet of Things systems," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 4, 2021, Art. no. e4222.

[2] S. Latif, Z. Idrees, J. Ahmad, L. Zheng, and Z. Zou, "A blockchain-based architecture for secure and trustworthy operations in the industrial Internet of Things," *J. Ind. Inf. Integr.*, vol. 21, 2021, Art. no. 100190.

[3] G. Falco, C. Caldera, and H. Shrobe, "IIoT cybersecurity risk modeling for SCADA systems," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4486–4495, Dec. 2018.

[4] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *Proc. 26th Secur. Symp.*, 2017, pp. 1093–1110.

[5] M. Abdel-Basset, V. Chang, H. Hawash, R. K. Chakrabortty, and M. Ryan, "Deep-IFS: Intrusion detection approach for industrial Internet of Things traffic in fog environment," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7704–7715, Nov. 2021.

[6] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K.-K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8852–8859, Sep. 2020.

[7] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4815–4830, Jun. 2019.

[8] M. M. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, and M. Gidlund, "A machine-learning-based technique for false data injection attacks detection in industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8462–8471, Sep. 2020.

[9] A. Makkar, S. Garg, N. Kumar, M. S. Hossain, A. Ghoneim, and M. Alrashoud, "An efficient spam detection technique for IoT devices using machine learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 903–912, Feb. 2021.

[10] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, "Flowguard: An intelligent edge defense mechanism against IoT DDoS attacks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9552–9562, Oct. 2020.

[11] T. T. Huong *et al.*, "Lockedge: Low-complexity cyberattack detection in IoT edge computing," *IEEE Access*, vol. 9, pp. 29696–29710, 2021.

[12] R. Bokka and T. Sadasivam, "Deep learning model for detection of attacks in the Internet of Things based smart home environment," in *Proc. Int. Conf. Recent Trends Mach. Learn., IoT, Smart Cities Appl.*, 2021, pp. 725–735.

[13] K.-D. Lu, G.-Q. Zeng, X. Luo, J. Weng, W. Luo, and Y. Wu, "Evolutionary deep belief network for cyber-attack detection in industrial automation and control system," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7618–7627, Nov. 2021.

[14] S. ur Rehman *et al.*, "DIDDOS: An approach for detection and identification of distributed denial of service (DDoS) cyberattacks using gated recurrent units (GRU)," *Future Gener. Comput. Syst.*, vol. 118, pp. 453–466, 2021.

[15] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural Comput.*, vol. 1, no. 4, pp. 502–510, 1989.

[16] Y. Yin, "Random neural network methods and deep learning," *Probability Eng. Informational Sci.*, vol. 35, no. 1, pp. 6–36, 2021.

[17] S. Latif, Z. Zou, Z. Idrees, and J. Ahmad, "A novel attack detection scheme for the industrial Internet of Things using a lightweight random neural network," *IEEE Access*, vol. 8, pp. 89 337–89 350, 2020.

[18] Z. E. Huma *et al.*, "A hybrid deep random neural network for cyberattack detection in the industrial Internet of Things," *IEEE Access*, vol. 9, pp. 55 595–55 605, 2021.

[19] K. Filus, J. Domańska, and E. Gelenbe, "Random neural network for lightweight attack detection in the IoT," in *Proc. Symp. Model., Analysis, Simul. Comput. Telecommun. Syst.*, 2020, pp. 79–91.

[20] E. Gelenbe, "Learning in the recurrent random neural network," *Neural Comput.*, vol. 5, no. 1, pp. 154–164, 1993.

[21] E. Gelenbe and Y. Yin, "Deep learning with dense random neural networks," in *Proc. Int. Conf. Man-Mach. Interact.*, 2017, pp. 3–18.

[22] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.

[23] N. Moustafa, "Ton_IoT datasets for cybersecurity applications based artificial intelligence," 2019. [Online]. Available: https://research.unsw.edu.au/projects/toniot-data-set

[24] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, 2020, Art. no. 105524.

[25] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," 2020, *arXiv:2003.05689*.