electronics

MDPI

Article

Threat Analysis and Distributed Denial of Service (DDoS) Attack Recognition in the Internet of Things (IoT)

Mohammed Hasan Ali [1,2], Mustafa Musa Jaber [3,4], Sura Khalil Abd [3,5], Amjad Rehman [6], Mazhar Javed Awan [7,*], Robertas Damaševičius [8,*] and Saeed Ali Bahaj [9]

1 Computer Techniques Engineering Department, Faculty of Information Technology, Imam Ja'afar Al-sadiq University, Najaf 10023, Iraq; mh180250@gmail.com
2 Faculty of Computer Science and Mathematics, University of Kufa, Najaf 10023, Iraq
3 Department of Computer Science, Dijlah University College, Baghdad 10022, Iraq; mustafa.musa@duc.edu.iq (M.M.J.); sura.khalil@duc.edu.iq (S.K.A.)
4 Department of Medical Instruments Engineering Techniques, Al-Farahidi University, Baghdad 10022, Iraq
5 Department of Computer Science, Al-Turath University College, Baghdad 10022, Iraq
6 Artificial Intelligence and Data Analytics Lab, College of Computer and Information Sciences (CCIS), Prince Sultan University, Riyadh 11586, Saudi Arabia; rkamjad@gmail.com
7 Department of Software Engineering, University of Management and Technology, Lahore 54770, Pakistan
8 Faculty of Applied Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland
9 MIS Department College of Business Administration, Prince Sattam bin Abdulaziz University, Alkharj 11942, Saudi Arabia; s.bahaj@psau.edu.sa
* Correspondence: mazhar.awan@umt.edu.pk (M.J.A.); robertas.damasevicius@polsl.pl (R.D.)

Abstract: The Internet of Things (IoT) plays a crucial role in various sectors such as automobiles and the logistic tracking medical field because it consists of distributed nodes, servers, and software for effective communication. Although this IoT paradigm has suffered from intrusion threats and attacks that cause security and privacy issues, existing intrusion detection techniques fail to maintain reliability against the attacks. Therefore, the IoT intrusion threat has been analyzed using the sparse convolute network to contest the threats and attacks. The web is trained using sets of intrusion data, characteristics, and suspicious activities, which helps identify and track the attacks, mainly, Distributed Denial of Service (DDoS) attacks. Along with this, the network is optimized using evolutionary techniques that identify and detect the regular, error, and intrusion attempts under different conditions. The sparse network forms the complex hypotheses evaluated using neurons, and the obtained event stream outputs are propagated to further hidden layer processes. This process minimizes the intrusion involvement in IoT data transmission. Effective utilization of training patterns in the network successfully classifies the standard and threat patterns. Then, the effectiveness of the system is evaluated using experimental results and discussion. Network intrusion detection systems are superior to other types of traditional network defense in providing network security. The research applied an IGA-BP network to combat the growing challenge of Internet security in the big data era, using an autoencoder network model and an improved genetic algorithm to detect intrusions. MATLAB built it, which ensures a 98.98% detection rate and 99.29% accuracy with minimal processing complexity, and the performance ratio is 90.26%. A meta-heuristic optimizer was used in the future to increase the system's ability to forecast attacks.

Keywords: Internet of Things (IoT); cybersecurity; attack recognition; distributed denial of services; sparse convolute network; long-short term network; intrusion detection system

check for updates

Citation: Ali, M.H.; Jaber, M.M.; Abd, S.K.; Rehman, A.; Awan, M.J.; Damaševičius, R.; Bahaj, S.A. Threat Analysis and Distributed Denial of Service (DDoS) Attack Recognition in the Internet of Things (IoT). *Electronics* **2022**, *11*, 494. https://doi.org/10.3390/electronics11030494

Academic Editors: Il-Gu Lee, Kyungmin Go and Jung Hoon Lee

Received: 30 November 2021
Accepted: 7 February 2022
Published: 8 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Internet of Things (IoT) [1] has a collection of network devices that are interconnected via near-field communication (NFC), Bluetooth, and Wi-Fi connections [2]. The IoT devices are widely utilized in smart appliances (thermostats, refrigerators, etc.), security

systems, health care, computer peripherals, military, agriculture, etc. [3,4]. These IoT devices utilize the Internet Protocol (IP) to transmit the information from source to destination. This IP protocol identifies the computer to allow fast communication without requiring human intervention. However, IoT devices change human life in different applications, and the threats to IoT lead to a significant security risk. Intruder detection systems now in use may not provide adequate protection against today's sophisticated threats. Therefore, in this study, the threat of IoT intrusion has been assessed by employing a sparse convolute network to counter threats and attacks. Every IoT device has specific characteristics [5], such as large data gathering, physical and virtual environment connection, complex environment creation, centralized architecture. These characteristics enable the IoT to function efficiently, but it causes threat actors abuse in communication.

The SLR Blockchain paper suggested the world population utilizes around 10 billion IoT devices [6] to improve their lifestyle. The high utilization of IoT devices faces IoT security issues in the fast expansion of smart appliances because of connecting to the network. The device-linked IoT devices consist of home automation; thermostats, printers, refrigerators, etc., are operated with the help of artificial intelligence like Google Assistant and Amazon Alexa [7,8]. Hence, hijacking [9] these devices is easy by sending spam emails, conscripted into a botnet, and privacy leaks. By considering these, IoT devices are developed by considering the security-related features [10]. However, the IoT device utilization increases and data transmission is uncountable. Significant data transmission and billions of connections lead to difficulties while managing and tracking data security [11,12].

The IoT devices are susceptible to weaponization and hijacking for the Distributed Denial of Service (DDoS) attacks [13], man-in-the-middle attack [14], targeted code injection [15], and pose estimation [16]. In addition to this, the IoT devices are remotely controlled by the bad actors, creating a significant impact while transmitting data in the network. Therefore, managing and protecting the IoT security is more essential to reduce the intermediate attacks in a network environment. Then, the security threats are reduced by the complete network, segmentation of IoT devices, monitoring, inspection and policy enforcement, and taking immediate automatic actions if the network is influenced by attacks.

The IoT security has been achieved using the Intrusion Detection System (IDS) [17,18]. The IDS system uses various devices and software applications that help to monitor the network and predict malicious activities. Suppose the system or IoT devices face any security information and event management (SIEM) control with [19] their activities. SIEM integrates multiple source outputs and alarm filtering techniques to differentiate malicious activities [20]. The intrusions are prevented by four types: network-based, wireless intrusion, network behavior analysis, and host-based intrusion prevention system. These four types continuously monitor the entire network, wireless network, and software packages, and the suspicious traffic is predicted successfully. The prevention process uses Harris Hawks Sparse Auto-Encoder Networks for detecting the speeches [21].

As an effective extractor, the GA and 5-fold cross-validation (CV) methods used to establish the CNN model structure identify the bagging (BG) classifier. The deep feature subset of the selected CNN model is utilized to test the performance of the BG classifier with a 5-fold CV. An enhanced detection rate was achieved using a hybrid CNN/BG learning technique, including the GA, FCM, CNN extractor, and CNN extractor. The very reliable validation findings produced by the 5-fold CV technique for the proposed algorithm imply that NIDS can be used in a real-world computer network setting. These methods successfully predict the intrusion activities by examining network protocol with the predetermined and pre-configured attack patterns. Therefore, the intrusion detection system should be designed according to the threat patterns. The patterns are classified into misuse intrusion patterns [22] (it helps to predict the entire known threats by comparing the matching patterns) and anomaly intrusion [23] (this intrusion detects based on the network behavior). Sometimes, this system develops by combining the misuse and anomaly intrusion patterns to reduce the intermediate access. By considering these patterns, statistical

analysis, evolutionary algorithm, protocol verification, rule-based, and machine learning techniques [24] are introduced to detect and prevent intrusion activities. The general description of these methods is illustrated in Table 1. Intrusion detection systems (IDS) are cutting-edge security mechanisms that use advanced approaches to guard computer networks against intrusions in process or illicit accesses that have already occurred. In addition, the IDS must be designed to survive large-scale ethical hacking and real-time testing to be effective in cyber security activities.

Table 1. Intrusion detection techniques.

Technique	Description
Statistical Analysis [25]	This analysis compares the current behavior with the set of predetermined baselines.
Evolutionary algorithm [26]	It develops the application path used to predict the model average, error, and different behaviors according to the conditions.
Protocol Verification [27]	The suspicious activities are predicted by checking the protocol field. However, the false-positive rate is produced due to the unspecified protocols.
Rules-based [28]	This technique predicts the intrusions by comparing them with the signatures.
Machine learning technique [29]	Evaluating the hypothesis with a set of nodes and the feedback process predicts the intrusions.

As discussed in Table 1, different techniques are incorporated in the IoT network to predict intrusion activities. Machine learning techniques provide satisfactory results because the network is trained using sets of intrusion data, characteristics, and suspicious activities, which helps identify and track the attacks, mainly Distributed Denial of Service (DDoS) attacks. By considering the impact of machine learning techniques in the intrusion threat analysis process, these evolutionary techniques are incorporated to manage the reliability against the attacks.

The main contributions of the paper are:

1. Security and privacy concerns were a problem in this IoT paradigm because of threats and attacks.
2. This IoT paradigm was plagued by security and privacy concerns due to intrusion threats and attacks.
3. The use of training patterns in the network successfully classifies the standard and the threats.

Then, the rest of the paper is organized as follows; Section 2 discusses the various research opinion regarding intrusion and threat analysis in IoT networks. Section 3 explores the working process of machine learning with the evolutionary technique-based threat analysis process. Section 4 discusses the efficiency of the introduced system and concludes in Section 5.

2. Related Work

Particle swarm optimization with gradient descent algorithm (PSO-Light) was utilized in [30] to detect the intrusion activities in the IoT. This system resolves the poor scalability and low detection rate while recognizing intrusion activities. The PSO-Light algorithm derives the features from input data and feeds them into the one-class support vector machine to identify the malicious data. This process is applied to the UNSW-NB15 dataset, and the PSO-Light approach recognizes the shellcode, backdoor, and worm activities with a maximum detection rate. Improving classification methods or balancing classes in the training data (data preparation) before feeding the data into a machine learning algorithm are strategies for dealing with imbalanced datasets. As a result of its broader applicability, the latter technique is preferred. Problems arise because data collection and analysis can be time-consuming and expensive, and we often work with less relevant information

than desired. As a result, we may not collect enough representative instances from the minority population.

The Passban intelligent intrusion detection system was created in [31] to prevent IoT devices from intrusion activities. Passban helps identify malicious traffic such as SSH Brute force, Port scanning, and SYN flood attacks. This system resolves the existing accuracy and false positive rate challenges with a high detection rate.

Three-layer supervised intrusion detection was developed in [32] to detect the weakest IoT devices in smart home applications. First, the IoT device's normal and abnormal behaviors are classified, malicious packets are identified at the time of the attack, and attacks like denial of service (Dos), spoofing, man-in-the-middle, and replay attacks are detected successfully. This process detects multistage attacks with a minimum false positive rate. A genetic optimized deep belief network (GA-DBN) algorithm was introduced in [33] to create an effective intrusion detection model. This work predicts various types of attacks using different number genetic algorithm iterations and multiple hidden layers. The optimized classifiers classify the attacks with the maximum detection rate on the NSL-KDD dataset. In addition, this process minimizes the computation complexity. 'Distributed denial of service,' a general term for these kinds of attacks, refers to them all. Botnets are online devices used to flood a target website with fake traffic. Many internet businesses are vulnerable to DDoS attacks, and the consequences can be severe. A security that is predictable, reliable, effective, and trustworthy saves money. Operating and capital costs save money because it does not need to utilize a third-party scrubbing center, hire additional IT security staff, or purchase more bandwidth. As a result of restricted access, real users may not find information or carry out their desired actions. A blemish could be thrown on their record.

The two-tier classification model and dimension reduction algorithm is applied in the Internet of Things Backbone network [34] to predict the anomaly-related intrusion detection. This process is intended to detect the remote to local and user root attacks by utilizing the linear discrimination and component analysis approach. The extracted features proceed with the help of K-nearest neighbor and Naïve Bayes to predict the suspicious actives, introducing a two-stage artificial intelligence (AI) related intrusion detection process in [35] to detect the abnormal activities in software-defined IoT (SD-IoT). This system aims to detect the signature and unknown attacks in SD-IoT. The features are selected according to the bat algorithm with binary differential mutation and optimized random forest approach weights. This process detects abnormal activities with high accuracy and lower overhead.

Stochastic Petri Net (SPN) is used [36] for different attack strategies for developing the intrusion detection system. This process improves the network lifetime using a set of parameter values and reduces intruder involvement in the IoT. This system considers several failure conditions to detect malicious attacks using 128 mobile sensor nodes and analyzing and protecting network traffic [37] using ensemble intrusion detection techniques and statistical flow features. This paper focuses on the protocol-related malicious activities and the attacks detected using naïve Bayes, decision trees, neural networks. This system was developed using NIMS and UNSW-NB15 datasets, and different potential characteristics were extracted. Malicious activities are removed from the derived features based on the correlation coefficient and entropy features. Thus, the system ensures the minimum false positive and high detection rates.

Multi-agent and multilayered game processes in the IoT are formulated in [38] to detect intrusions. This system aims to prevent and avoid security-related vulnerabilities using multilayered game formulation. This process is incorporated with the trust model to make the trust communication process. The system ensures security with minimum delay and maximum accuracy and throughput.

Azeez et al. [39] used an upgraded hashing-based Apriori algorithm implemented on the Hadoop MapReduce framework, capable of discovering and detecting network intrusions using association rules in mining algorithms. The proposed method was evaluated on the KDD dataset.

Deep convolution neural networks are applied in [40] to identify the intrusions in the intelligent Internet of vehicles. The data-driven approach is linked with the rode side unite (RSU) load behavior to prevent attacks. These features are extracted according to the convolution neural network that avoids RSU attacks.

Machine learning techniques are utilized in [41] to detect malicious bots in the IoT. This system aims to reduce the misclassification of malicious activities using compelling network traffic features. The corrAUC approach is applied to select the parts that work according to the wrapper technique. Components are chosen based on Shannon entropy and TOPSIS, which helps to classify malicious nodes in Bot-IoT. A self-recurrent neural network based on wavelets with multidimensional radial wavelons is proposed for network intrusion detection in [42]. The results demonstrate that recurrent architectures based on wavelets outperform their counterparts not only in terms of attack detection and classification but also in terms of overall performance.

A Local–Global Best Bat Algorithm for Neural Networks (LGBA-NN) approach was proposed in [43] to select the best feature subsets and hyperparameter values for efficient detection of botnet attacks on the IoT. Enhanced BA was also used for neural network hyperparameter tuning and weight optimization to categorize ten separate botnet assaults and one benign target class. The proposed LGBA-NN method was evaluated on an N-BaIoT dataset that included comprehensive real-time traffic data from benign and malicious target classes.

A malware detection approach based on a stacked ensemble of dense (fully connected) CNNs in the first stage classification with a machine learning-based meta-learner in the final stage classification was proposed in [44]. The approach was evaluated on the Classification of Malware with PE headers (ClaMP) dataset. A method for detecting network intrusions based on multistage deep learning image recognition was introduced in [45]. The network flow features are converted into four-channel pictures (Red, Green, Blue, and Alpha). The images are then used to train and evaluate the pre-trained deep learning network ResNet50. The suggested method is tested against two publicly accessible benchmark datasets, UNSW-NB15 and BOUN Ddos. In Sodhro et al. [46], the ETPC algorithm is presented, implemented on hardware, and then compared to several traditional TPC approaches. In Muzammal et al. [47], in a fog computing environment, an ensemble technique with data fusion is presented to work with medical data acquired via BSNs. A group of sensors has been assembled to provide high-quality activity data, and the data has been combined. Table 2 describes existing methods for network intrusion detection with their advantages and disadvantages.

Table 2. Comparison of network intrusion detection methods.

Method	Advantages	Disadvantages
PSO-Light	The increased part of computational complexity is caused by building complex networks operation.	The disadvantages of the particle swarm optimization (PSO) algorithm are that it is easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process.
GA-DBN	Genetic Algorithms are faster and more efficient when compared to the traditional methods of brute-force search. Genetic Algorithms are proven to have many parallel capabilities.	GA requires less information about the problem, but designing an objective function and getting the representation and operators right can be difficult. GA is computationally expensive, i.e., time-consuming.
SD-IoT	It enables centralized management of networking devices and helps in the automation of networking devices. It provides improvements to end-users.	Every device used on a network occupies a space on it, making it almost impossible to manage the actual devices.
SPN	Petri nets can be used as a hierarchical model. This is because they can be used at all levels, including networks, register transfer functions, gates, etc.	The existing policies are that many control places and associated arcs are added to the initially constructed Petri net model, which significantly increases the complexity of the supervisor of the Petri net model.
LGBA-NN	A bat algorithm (BA) is a heuristic algorithm that operates by imitating the echolocation behavior of bats to perform global optimization.	Mesh networking is much harder to do work; the overall overhead of every node having a full copy of the AI program makes it very expensive.

In summary, according to various researcher opinions, intruders and threats are detected with the help of machine learning techniques. Taking advantage of the minimum false positive rate, maximum detection rate, and minimum complexity in this work, and the optimized machine learning technique is utilized to analyze threat activities in the IoT. The above literature methods are analyzed and found that most of the problems occurred in the network. It is to avoid and get threat detection solutions and solve by using evolutionary sparse convolution network (ESCNN) intrusion and threat activities in the IoT.

3. Materials and Methods

Intrusion Detection Using Optimized Sparse Convolution Neural Networks

This section discusses the optimized sparse CNN-based intrusion detection in the IoT. As discussed earlier, intrusions are prevented in any type such as host, network, wireless, etc. These kinds of data have been utilized to extract anomaly features using introduced approaches.

Figure 1 shows the database load balancer to act as a middleman between the database and the applications that use it. A single database endpoint, increased query throughput, reduced latency, and better usage of database server resources are all aims of database load balancing. Changing the training dataset's composition is the most common approach to an unbalanced classification problem. Because we are sampling an existing data sample, strategies to alter the class distribution in the training dataset are sometimes referred to as sampling methods or resampling methods. It collects and translates a dataset into relevant, useable information that constitutes data processing in research. A researcher, data engineer, or data scientist can manually or automatically transform raw data into a more understandable format, such as a graph, report, or chart. A dataset of this size is a training dataset to train a machine learning model. The second set of data, referred to as a validation or testing set, might supplement the first. A training set, training dataset, or learning set is another name for training data. There are several ways to evaluate the effectiveness of a learning model. Still, one of the most common is to use a train test dataset, which divides a dataset into training and testing datasets. The full dataset is utilized for training and testing a specific model in a more advanced method.

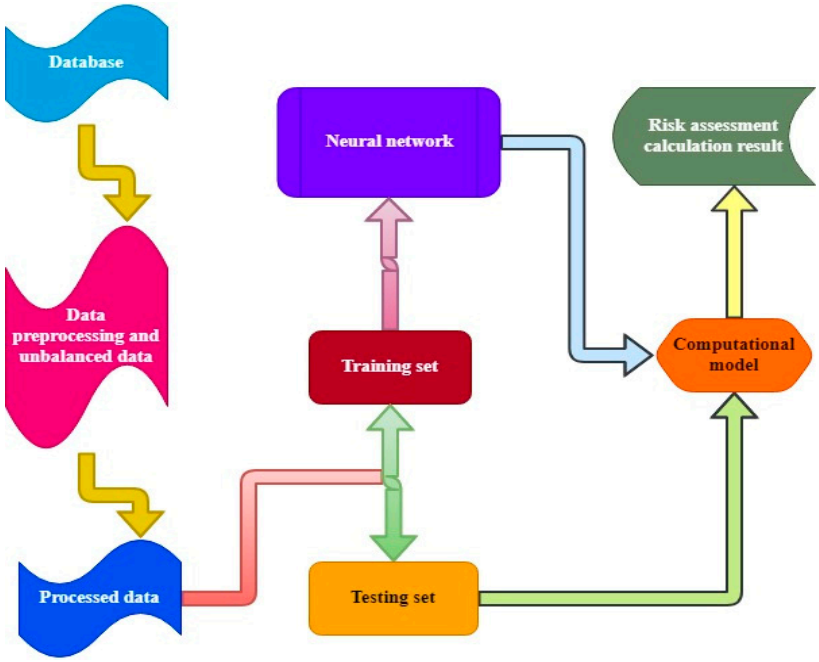

Figure 1. Data balancing model structure and its process.

Datasets in neural networks are essentially data sets that may be analyzed and predicted by computers as if they were a single entity. Quantitative risk analysis is used for

risks that need additional investigation. Modern biology relies heavily on computational models. To synthesize current knowledge, assess opposing hypotheses qualitatively and quantitatively, and assist the understanding of complex data, they give a framework within which to do so.

The extracted features are more helpful in predicting intrusions and threats with minimum complexity and maximum detection rate. The intrusion detection system is then illustrated in Figure 2.

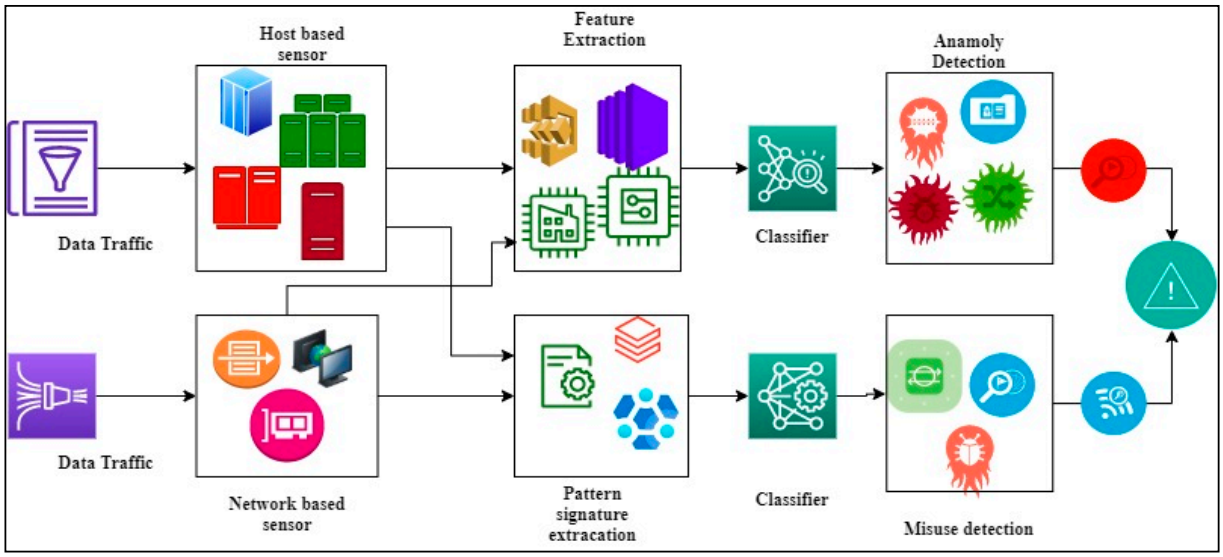

Figure 2. Intelligent Intrusion Detection System in the Internet of Things (IoT).

This study's main objective is to detect threat and intrusion activities from the data traffic presented in the network and host reliably. Intrusion detection systems (IDS) are primarily designed to protect an IoT network from external threats, rapid response, and high-volume data processing requirements for an IDS intended for IoT-based smart settings. Network security technology such as an Intrusion Detection System (IDS) was designed to discover vulnerabilities in a target application or computer. The IDS is a listening-only device, as mentioned. The goal is achieved according to Equation (1), that is, the output of convolution layer feature map $O \in \mathbb{C}^{(g-s+1)*(w-s+1)\times n} = \mathfrak{K} \times \Im$ defined by

$$O(y,x,j) = \sum_{i=1}^{m} \sum_{u,v=1}^{s} \mathfrak{K}(u,v,i,j)\Im(y+u-1, x+v-1, i) \quad (1)$$

The objective is obtained from feature map $\Im$ in $\mathbb{C}^{g*w*m}$; input feature map height and width are denoted as g and w, convolutions kernel is $\mathfrak{K}$, with size s, and n number of the output channel. The network uses zero paddings and one stride during the threat identification process. The threat should be detected in a reliable and fast manner according to the sparse matrices. For this, the feature tensor should change according to sparse multiplication matrix-like $\Im$ to $\mathfrak{J} \in \mathbb{C}^{g*w*m}$ and kernel $\mathfrak{K}$ to $\Re \in \mathbb{C}^{s*s*m*n}$ to $\mathfrak{P} \in \mathbb{C}^{m*m}$. The kernel operation is performed with kernel $\mathfrak{K}$ and input $\Im$, which O R*J replaces, defined as follows.

$$\mathfrak{K}(u,v,i,j) \approx \sum_{k=1}^{m} \Re(u,v,k,j)\, \mathfrak{P}(k,i) \quad (2)$$

$$\mathfrak{J}(y,x,i) = \sum_{k=1}^{m} \mathfrak{P}(i,k)\Im\,(y,x,k) \quad (3)$$

Then, for channel i, decompose the tensor ($\Re$ (.,.,i,.)$\in$ $\mathbb{C}^{s*s*n}$) into the product of matrix ($\mathfrak{S}_i \in \mathbb{C}^{q_i*n}$) and tensor ($\mathfrak{W}_i \in \mathbb{C}^{s*s*q_i}$) according to number base (q_i) that is defined in Equation (4).

$$\Re(u,v,i,j) \approx \sum_{k=1}^{q_i} \mathfrak{S}_i(k,j)\mathfrak{W}_i(u,v,k) \tag{4}$$

$$\mathfrak{V}_i(y,x,k) = \sum_{u,v=1}^{s} \mathfrak{W}_i(u,v,k)\mathfrak{J}(y+u-1,x+v-1,i) \tag{5}$$

From the denser decomposition process, the sparse convolution operation is performed using Equation (6).

$$O(y,x,j) \approx \sum_{i=1}^{m}\sum_{k=1}^{\mathfrak{W}_i} \mathfrak{S}_i(k,j)\mathfrak{V}_i(y,x,k) \tag{6}$$

Here, $O(y,x,j)$ is formulated according to the single matrix multiplication of $\mathfrak{S}_i(k,j)$ and $\mathfrak{V}_i(y,x,k)$. The first two dimensions, $\mathfrak{S}_i(k,j)$ from $\Re(u,v,i,j)$ and $\mathfrak{V}_i(y,x,k)$ from $\mathfrak{V}_i(y,x,k)$, are utilized from the sensor during this computation. This sparse convolution kernel value ensures the output of the threat's activities from a user action.

However, the computation complexity should be reduced during the threat and intrusion detection process. The system's complexity is measured by counting the number of multiplications. Generally, the convolution network requires $mns^2(g-s+1)(w-s+1)$ multiplications; but this work reduces the complexity using the sparse kernel process; therefore, complexity is computed from non-zero sparse matrix γ and decomposition of a matrix.

$$\left(\gamma mn + \textstyle\sum_{i=1}^{n} q_i\right)s^2(g-s+1)(w-s+1) + m^2gw \tag{7}$$

After reducing computation complexity, the matrix formulation problem is reduced by performing decomposition, defined in Equations (2) and (3). Then, the fine-tuning process is applied to the network to improve the threat detection accuracy and specificity. In the fine-tuning phase, the objective function Equation (8) is used to minimize the deviation while predicting threats in IoT.

$$minimize_{\mathfrak{P},\mathfrak{W}_i,\mathfrak{S}_i}\ \mathcal{L}_{net} + \lambda_1 \sum_{i=1}^{m} \parallel \mathfrak{S}_i \parallel_1 + \lambda_2 \sum_{i=1}^{m}\sum_{j=1}^{q_i} \parallel \mathfrak{S}_i(j,.) \parallel_2 \tag{8}$$

The deviation should minimize using the logistic loss function $\mathcal{L}_{net}$ in network output. Element wise matrix is in $\parallel . \parallel_1$ and $\parallel . \parallel_2$. Based on the above discussion, the objective of the work is achieved; that is, reliable and minimum computation complexity is achieved while detecting threats in IoT. Further, the system's effectiveness improved using an effective training process that uses long-short term memory neural networks (LSTM).

The training process aims to predict user behavior while attempting to perform IoT actions. The user behavior and features are used to detect the intruder and inside threat. Here, user behavior features are extracted according to the function of LSTM that helps predict anomalous activity.

Consider the IoT network with a set of users such as $\{u_1, u_2, \ldots . u_k\}$; each user has several actions (A) in a day $\mathcal{J}$. The user actions are represented as $A = \left[A_{u_{k,1}},\ A_{u_{k,2}}, \ldots, A_{u_{k,j}}\right]$. During the training process, u_k actions $A_{u_{k,1}}$ in j day is derived that was utilized for the network training process; according to the u_k and $A_{u_{k,1}}$ Neural network extract features. Then, the derived features are analyzed and the matrix (fixed-size) $\mathfrak{M}^{u_{k,j}}$ constructed, which contains user behavior-related temporal features. By utilizing these features, threat and regular activities are classified using the sparse convolution network in the testing phase. Then, the overall network training process is illustrated in Figure 3.

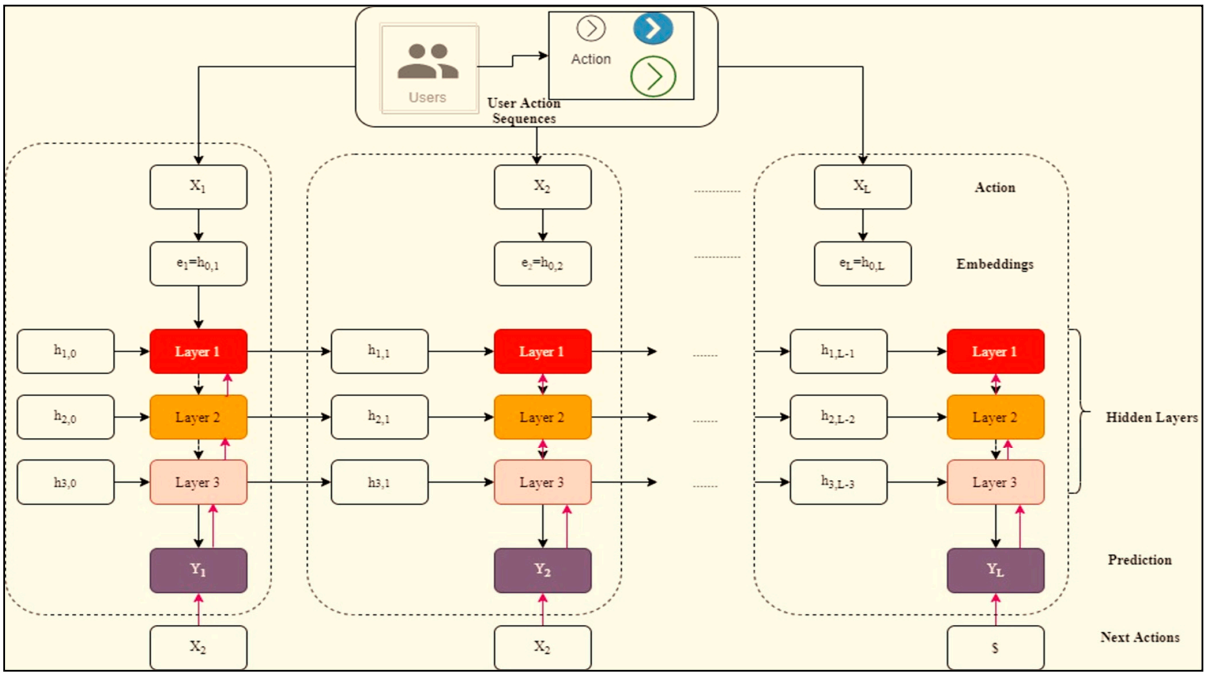

Figure 3. Intruder actions training using long-short term network.

Figure 3 illustrates that the LSTM based training process consists of input, embedding, hidden, and output layers. Each layer performs a specific function, and respective user behavior features are extracted. As discussed earlier, the output $\mathfrak{y}_t^{u_{k,j}}$ $(1 \leq t \leq T)$ at time instance t is obtained by processing user actions $A_{u_{k,j}} = \left\{X_1^{u_{k,j}}, X_2^{u_{k,j}}, \ldots, X_T^{u_{k,j}}\right\}.X_t^{u_{k,j}} (1 \leq t \leq T)$ in the LSTM network hidden layer $h_t^{u_{k,j}} (0 \leq l \leq 3, 1 \leq t < T)$. The dictionary should be created for IoT users and individual actions, which helps identify user behavior features. If the user logs into the IoT device after an hour, that is represented as one, and logged off the IoT device after an hour, defined as 2. These actions are converted to the one-vector format to get the exact user behavior in the hidden layer process. The network generally has the input, weight, and bias values used to predict the output. The three hidden layer process is defined as follows:

$$\mathcal{I}_{l,t}^{u_{k,j}} = \sigma\left(\mathfrak{Z}_l^{(i,x)}\, h_{l-1,t}^{u_{k,j}} + \mathfrak{Z}_l^{(i,h)} h_{l,t-1}^{u_{k,j}} + \mathfrak{b}_l^i\right) \tag{9}$$

$$\mathfrak{f}_{l,t}^{u_{k,j}} = \sigma\left(\mathfrak{Z}_l^{(\mathfrak{f},x)}\, h_{l-1,t}^{u_{k,j}} + \mathfrak{Z}_l^{(\mathfrak{f},h)} h_{l,t-1}^{u_{k,j}} + \mathfrak{b}_l^{\mathfrak{f}}\right) \tag{10}$$

$$\mathfrak{o}_{l,t}^{u_{k,j}} = \sigma\left(\mathfrak{Z}_l^{(\mathfrak{o},x)}\, h_{l-1,t}^{u_{k,j}} + \mathfrak{Z}_l^{(\mathfrak{o},h)} h_{l,t-1}^{u_{k,j}} + \mathfrak{b}_l^{\mathfrak{o}}\right) \tag{11}$$

$$\mathfrak{g}_{l,t}^{u_{k,j}} = tanh\left(\mathfrak{Z}_l^{(\mathfrak{g},x)}\, h_{l-1,t}^{u_{k,j}} + \mathfrak{Z}_l^{(\mathfrak{g},h)} h_{l,t-1}^{u_{k,j}} + \mathfrak{b}_l^{\mathfrak{g}}\right) \tag{12}$$

$$\mathfrak{c}_{i,t}^{u_{k,j}} = \mathfrak{f}_{l,t}^{u_{k,j}} \odot \mathfrak{c}_{i,t-1}^{u_{k,j}} + \mathcal{I}_{l,t}^{u_{k,j}} \odot \mathfrak{g}_{l,t}^{u_{k,j}} \tag{13}$$

$$h_{l,t}^{u_{k,j}} = \mathfrak{o}_{l,t}^{u_{k,j}} \odot \tanh\left(\mathfrak{c}_{i,t}^{u_{k,j}}\right) \tag{14}$$

The above computations are utilized for training the features derived from the user actions. Here, $\mathfrak{c}_{i,0}^{u_{k,j}}$ and $h_{l,0}^{u_{k,j}}$ values are zero for the entire three layers $one \leq l \leq 3$, $\odot$ represented as the element-wise multiplication and $\sigma(.)$. It is denoted as the sigmoid function. These functions are applied to the hidden representation $_{al,t}^{u_{k,j}}$ to identify the output in hidden units. Along with the value, $\mathcal{I}_{l,t}^{u_{k,j}}$ is updated, and $\mathfrak{f}_{l,t}^{u_{k,j}}$ values are forgotten for getting the $\mathfrak{o}_{l,t}^{u_{k,j}}$ output value. This process is repeated to investigate the user actions as $A = \left[A_{u_{k,1}}, A_{u_{k,2}}, \ldots A_{u_{k,j}}\right]$ for getting the exact output value $y_{l,t}^{u_{k,j}}$. At last, the cross-entropy

loss value is estimated by collating output $y_{l,t}^{u_{k,j}}$ with input $x_{t+1}^{u_{k,j}}$. Here, dropout process is applied to reduce the overfitting data that help to improve the overall recognition accuracy. The training process helps to derive the feature vectors $\mathfrak{H}^{u_{k,j}} = \left\{h_{3,1}^{u_{k,j}}, h_{3,2}^{u_{k,j}}, \ldots, h_{3,T}^{u_{k,j}}\right\}$. Then, the extracted features are transferred into the fixed-size illustration because it has to be given the input to the sparse CNNs.

The user u_k any sequence actions $A_{u_{k,j}}$ are defined in maximum (N^{u_k}) and minimum length (n^{u_k}) because the sequences are eliminated from this process with low length compared to n^{u_k}. This process helps minimize unwanted computation and to maximize threat detection time. Therefore, zeros are pad between n^{u_k} to (N^{u_k}) to reach the extract features to maximum length. This process is performed to convert the $\mathfrak{H}^{u_{k,j}} = \left\{h_{3,1}^{u_{k,j}}, h_{3,2}^{u_{k,j}}, \ldots, h_{3,T}^{u_{k,j}}\right\}$ feature to matrix $\mathfrak{M}^{u_{k,j}} - (N^{u_k} * V^{u_k})dimension$. Then, the formed $\mathfrak{M}^{u_{k,j}}$ is given as input to the sparse convolution matrix to analyzing user behavior to predict the threat and everyday activities.

Consider that the IoT network has a different number of nodes, in which one node is treated as a server node, and the remaining nodes are a client for data transmission and analytic processes. Here, traffic is continuously monitored to eliminate the modification on live traffic; every user action (data transmission) server responds to the client sender node by providing replies. The sensor node's behavior must be analyzed to eliminate the intermediate action during this process. Then, the IoT communication behavior and attacks are illustrated in Figure 4.

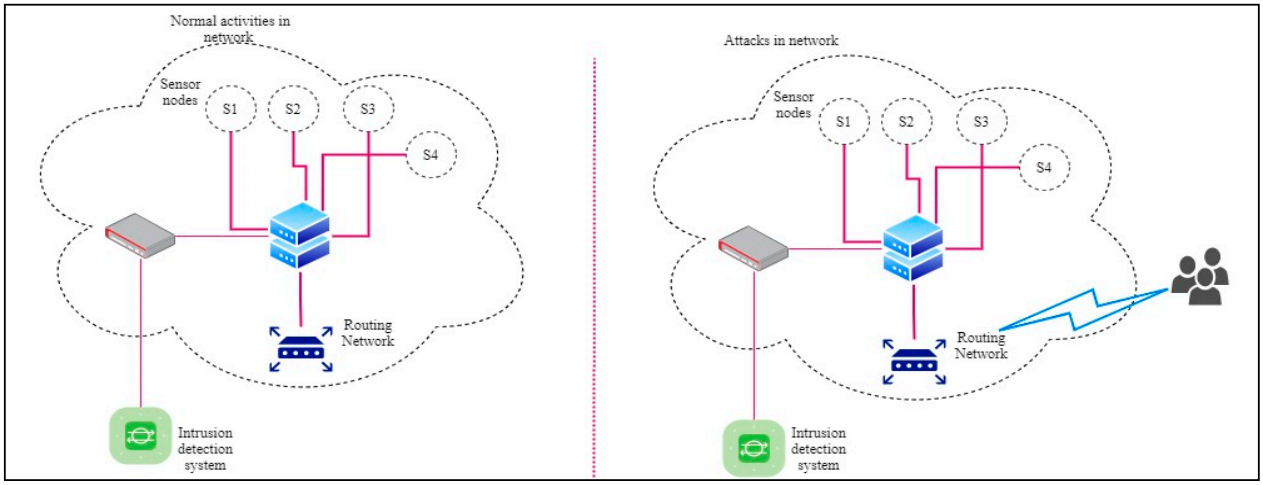

Figure 4. IoT Experimental Structure of Intrusion Detection System.

As shown in Figure 4, the attacker attacks the server node because it analyzes transmitted data, login, and other responding processes. The DDoS attacks happened using a single host among the 10 million packets transferred. Therefore, the attack must be detected according to user actions in a day. According to the above process, features like several nodes, neighbors, leaving, joining, etc., are extracted as features. Those derived features are more valuable to predict the intermediate access. In addition to this, the evolutionary computation algorithm is incorporated to predict the threat activities with minimum loss and high accuracy value. The evolutionary algorithm chooses the best solution for automatically created solutions using the fitness value. Here, the multi-objective evolutionary algorithm is used to find the optimal solution (Pareto set). The predicted solution x is greater than the other solution y, supposing the network does not have any more excellent value; at least one less than the values are presented as the optimal solution. Here, sensor node features are continuously examined; if the server node characteristics face any changes, the alarm should be ringed to treat as intruder and attack. Then, the efficiency of the system is evaluated using experiments.

This paper focuses on DDoS by applying Bayesian network models with incomplete data. The above algorithm 1 can be used to re-estimate a parameter when a new set of data $d = \{d_0, d_1, \ldots, d_n\}$ becomes available, some of which may be partially observed. Algorithm 1 depicts a procedure that is in use while DDoS is running. New security data samples are added to the parameter when they arrive with thresholds th. The current characteristics of DdoS can be reflected in an improved modeling tool, which is critical for enhancing performance based on moment factor mf.

Algorithm 1:

Start
Initialize network values n, $d = \{d0, d1, \ldots, dn\}$, momentum factor mf, threshold th;
Output parameter;
$n \leftarrow 0; \Delta th \leftarrow d = \{d0, d1, \ldots, dn\}$;
While $\Delta th \leftarrow d > n$ do
$n = n + 1$;
For each $d = \{d0, d1, \ldots, dn\}$;
Re-estimate variables mf, th;
End for;
$mf \leftarrow d$;
$n++$;
End;
Return;
Print parameter value;
Stop

The industrial benchmark for the system is presented first. This system is mapped in MATLAB using a Bayesian network value n for risk assessment. We compare our proposed method ESCNN to the reference work in terms of accuracy and dynamic range during our threat assessment experimentation.

4. Results and Discussion

This section evaluates the effectiveness of the evolutionary sparse convolution network (ESCNN) intrusion or threat detection system discussed in Section 3. This system uses the DDoS Evaluation Dataset [48] for evaluating the introduced system efficiency. The dataset aim to manage the network security on various attacks and traffic. The algorithm was developed to reduce the network overhead using various DDoS attack-related feature examinations. Here, 2313 samples are utilized as training, 490 samples for validation, and 502 samples for testing. Then, the data samples used in threat detection activities are illustrated in Table 3.

Table 3. Data sample description.

Type of Attacks	Data Samples	Percentage
Distributed Denial of services (DDoS)	2138	65%
Normal	1180	35%

This dataset handles various DDoS attacks such as NTP, LDAP, DNS, NetBIOS, MSSQL, TFTP, SYM, WebDDoS, etc. These attacks are executed at a specific time. The collected samples are trained using the LSTM network, and the obtained results are illustrated in the confusion matrix shown in Figure 5.

Figure 5 represents the confusion matrix value of training, testing, validation, and overall confusion matrix. The confusion matrix formed according to the *false positive rate (FP)* (indicates the correct classification of the regular events: yellow box) and *true positive rate (TP)* (measures the right category of attack events: green box). Then, the adequate training and learning process improves the overall classification rate up to 99.6%. It was able to detect the DDoS attacks in IoT network traffics. The practical computation of this process improves the general network security and alerts the data transmission team in the earlier stage by avoiding network disruptions. Further, the performance of the system is

evaluated using *accuracy (Acc)*, measuring the exact detection from entire data instances; *Detection Rate (DR)*, intrusion instances ratio; *False Alarm Rate (FAR)*, misclassification of normal instance; *Precision (Pre)*, how many attacks are classified correctly; and *Recall (Re)*, detecting how many attacks are done in the model return.

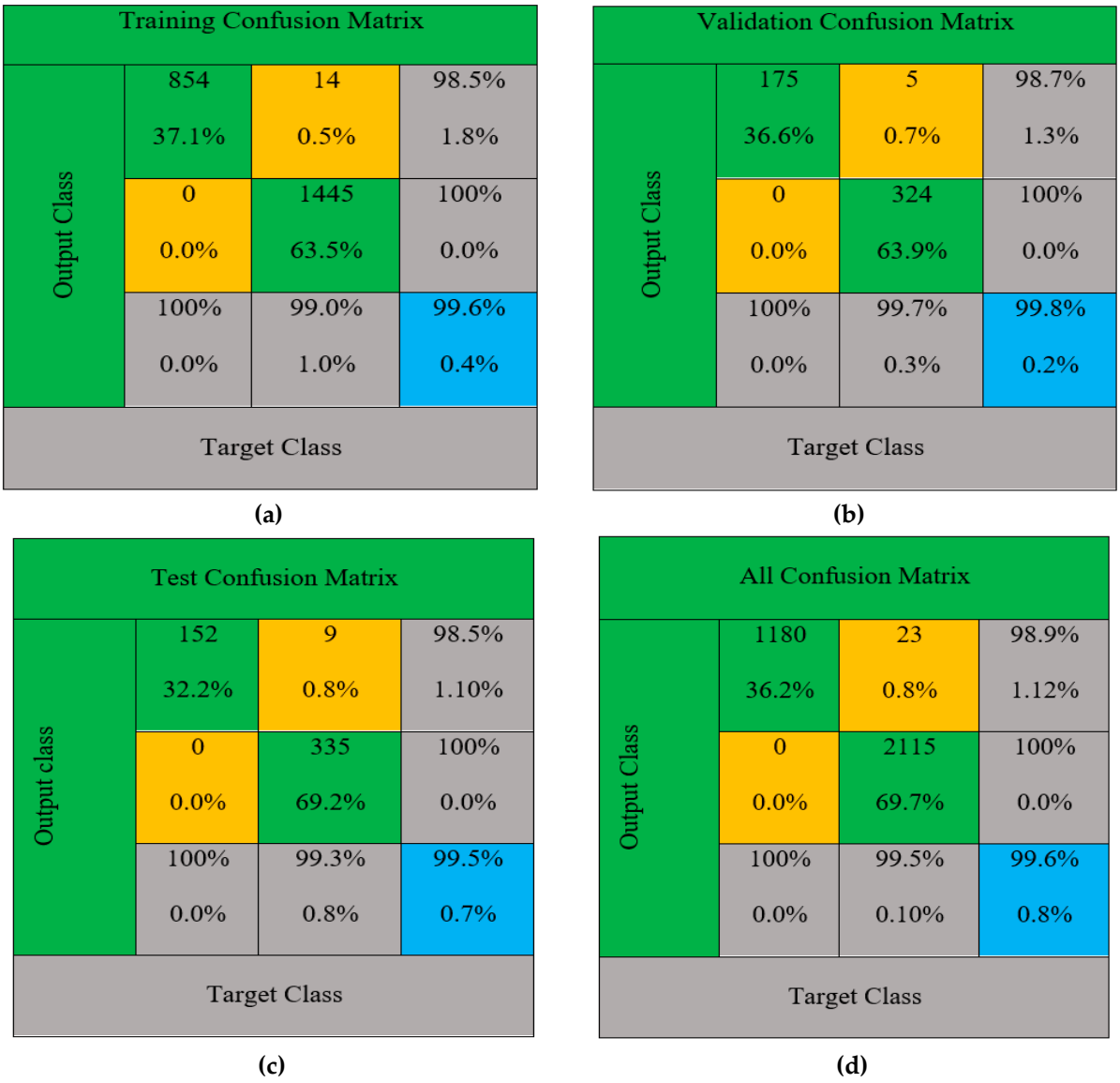

Figure 5. Confusion matrix of classification results: (**a**) training, (**b**) validation, (**c**) test, and (**d**) full.

$$Accuracy = \frac{True\ positive + True\ Negative}{True\ positive + True\ Negative + False\ Positive + False\ Negative} \quad (15)$$

$$Detection\ Rate\ (DR) = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (16)$$

$$False\ Alarm\ Rate\ (FAR) = \frac{False\ Positive}{True\ Negative + False\ Positive} \quad (17)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (18)$$

$$Recall = \frac{True\ Postive}{True\ Positive + False\ Negative} \quad (19)$$

The discussed evolutionary sparse convolution network (ESCNN) classifies the abnormal activities in a reliable and the fastest manner. The successful formulation of sparse matrix features from data traffic reduces the computation complexity with maximum accuracy. The obtained accuracy result is illustrated in Figure 6. The introduced ESCNN approach compared with existing research approaches such as Particle swarm optimization with gradient descent algorithm (PSO-Light) [30], genetic optimized deep belief network (GA-DBN) algorithm [33], Two-tier classification model, and dimension reduction algorithm (TT-DR) [34].

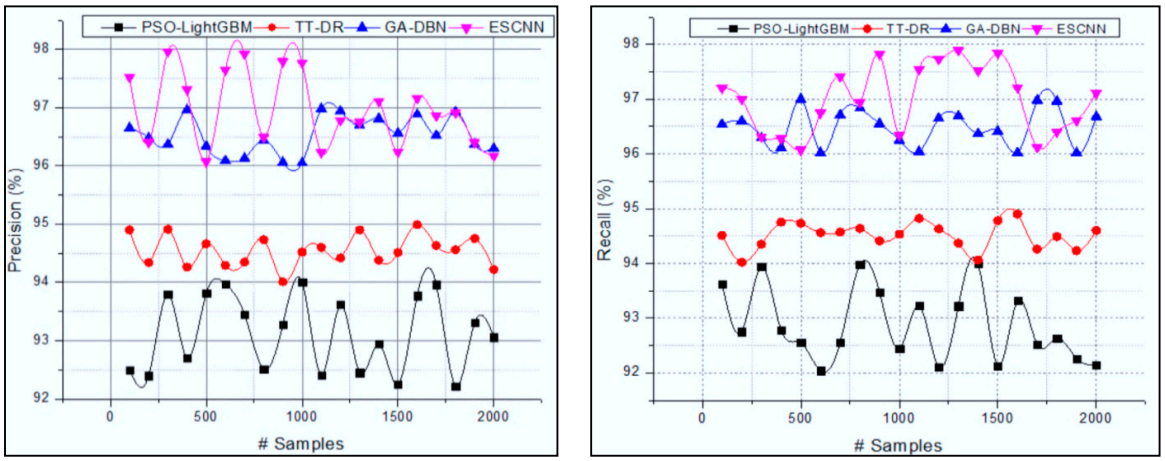

Figure 6. Classification results: (**a**) Precision and (**b**) Recall.

According to Figure 6a, the ESCNN approach predicts the abnormal activities, i.e., threat in IoT environment, by analyzing the user action sequences $A_{u_{k,1}}, A_{u_{k,2}}, \ldots, A_{u_{k,j}}$ Using the LSTM layers, the extraction of $\mathfrak{H}^{u_{k,j}} = \left\{h_{3,1}^{u_{k,j}}, h_{3,2}^{u_{k,j}}, \ldots, h_{3,T}^{u_{k,j}}\right\}$ helps to identify the normal and abnormal activities while the user tries to execute the IoT environment. A sparse matrix is generated $\mathfrak{from\ the\ features}\ \mathfrak{M}^{u_{k,j}} - (N^{u_k} * V^{u_k})$. That minimizes the computation complexity while extracting different activities in the IoT. The successful identification of user behavior improves overall precision. Specific abnormal events are predicted from the analyzed behavior using practical computation of $h_{l,t}^{u_{k,j}} = \mathfrak{o}_{l,t}^{u_{k,j}} \odot \tanh\left(\mathfrak{c}_{i,t}^{u_{k,j}}\right)$. The system minimizes the deviations in the fine-tuning phase $minimize\ _{\mathfrak{P},\mathfrak{W}_i,\mathfrak{S}_i}\ \mathcal{L}_{net} + \lambda_1 \sum_{i=1}^{m} \parallel \mathfrak{S}_i \parallel_1 + \lambda_2 \sum_{i=1}^{m} \sum_{j=1}^{q_i} \parallel \mathfrak{S}_i(j,.) \parallel_2$. The system improves the threat prediction rate and minimizes the false alarm rate (Figure 7).

The effective computation of $\mathfrak{K}(u,v,i,j)\mathfrak{F}(y+u-1, x+v-1, i)$ sparse multiplication and decomposition of denser and convolution operations help to identify the data traffic feature map. Moreover, the evolutionary algorithm minimizes the computation problem, and fine-tuning process helps to improve the overall attacks prediction rate.

The LSTM training process in different layers $\mathfrak{c}_{i,t}^{u_{k,j}} = \mathfrak{f}_{l,t}^{u_{k,j}} \odot \mathfrak{c}_{i,t-1}^{u_{k,j}} + \mathcal{I}_{l,t}^{u_{k,j}} \odot \mathfrak{g}_{l,t}^{u_{k,j}}$ helps to reduce the false attack prediction rate. The minimum false alarm rate directly indicates the ESCNN approach maximizes the overall attack detection accuracy and detection rate shown in Table 4. The proposed ESCNN approach recognizes the network attacks with a maximum detection rate (98.9%). In addition to this, the method classifies the normal and abnormal activities with high recognition accuracy (99.29%).

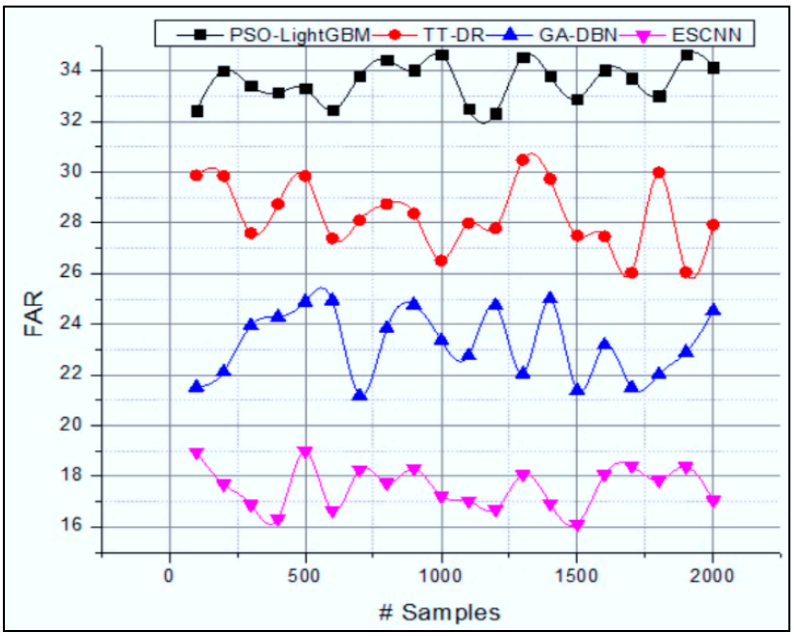

Figure 7. False Alarm Rate.

Table 4. Attack Detection Rate.

Methods	Accuracy (%)	Detection Rate (%)
PSO-Light [30]	93.56	94.29
GA-DBN [33]	94.18	95.92
TT-DR [34]	93.90	94.23
ESCNN	99.29	98.98

As seen in Figure 8, an application or system's data threat detection refers to the systems and procedures used to identify current or potential risks. "Intrusion Detection System" refers to these devices (IDS).

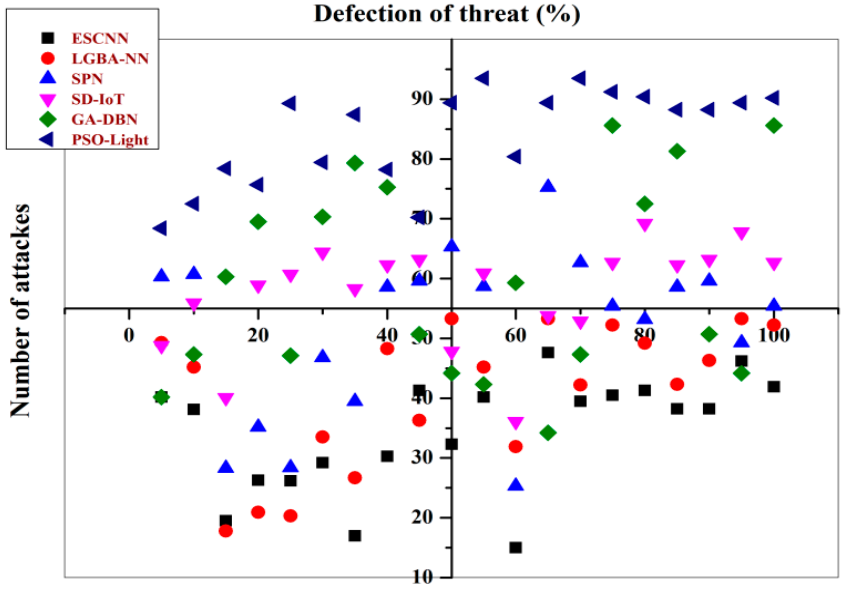

Figure 8. Threat defections on attackers.

There is always an unlawful system intrusion in any danger to data. These detective systems monitor a network system's actions, traffic, identification, and assaults when they

are used. Both software and hardware can be used in the creation of them. When data are "in use," this kind of data protection is used. Masking is the act of covering or concealing something entirely or in part. The process of obscuring or hiding real-time access to data collection via dynamic data masking does not modify the actual data. When accessing the data, the process is running. As a safeguard against unwanted access, it is used.

By storing all atomic operations for a given set of convolution kernel elements in an instruction book, sparse convolution is shown in Figure 9. It is possible to train sparse neural networks from scratch with a fixed number of parameters using ESCNN. When training an ESCNN network, the weight values and sparse topology are optimized and combined to suit the distribution of data better. The main principle behind this is to start with a sparse network. As stated above, it provides superior threat detection and the correcting ratio is 90.26% compared to other approaches in the literature.

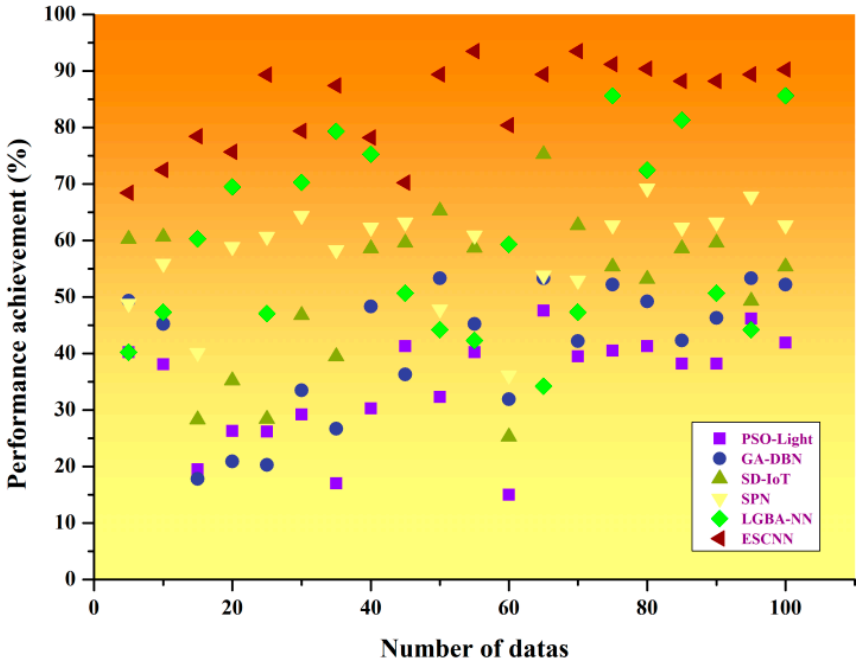

Figure 9. Overall performance analysis.

5. Conclusions

This paper analyzed the evolutionary sparse convolution network (ESCNN) intrusion and threat activities in the Internet of Things (IoT). Here, the DDoS Evaluation Dataset information is utilized to process the discussed intrusion detection system. The collected data are split into training, testing, and validation set. The data are trained according to the different layers of long–short term networks, improving attack detection accuracy. With the help of trained information, testing details are classified by extracting the feature and forming the sparse matrix construction. This process improves the overall attack detection accuracy with a minimum false alarm rate. The MATLAB tool implemented the system, ensuring 98.98% detection rate and 99.29% accuracy with minimum computation complexity, and the performance ratio is 90.26%. The limitation of the study is to ensure high reliability, fast computation, and reduced computation complexity. In the future, the system's effectiveness will be improved using a metaheuristic optimizer to estimate the global solution to attack prediction through recent work using big data approaches [49–52] and deep learning CNN architectures models [53–55].

Author Contributions: Conceptualization, A.R.; Data curation, M.H.A.; Formal analysis, M.M.J., S.K.A., M.J.A., R.D. and S.A.B.; Funding acquisition, R.D.; Investigation, M.H.A., M.M.J. and S.K.A.; Methodology, A.R.; Software, M.H.A.; Validation, S.K.A., A.R., M.J.A., R.D. and S.A.B.; Writing—original draft, M.H.A., M.M.J., S.K.A., A.R., M.J.A. and S.A.B.; Writing—review & editing, M.J.A. and R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset are available at https://www.unb.ca/cic/datasets/ddos-2019.html (accessed on 30 November 2021).

Conflicts of Interest: The authors declare no conflict of interest.

Table of Abbreviations

NTP	Network time protocol
LDAP	Lightweight directory access protocol
DNS	Domain name system
NetBIOS	Network Basic Input/Output System
MSSQL	Microsoft SQL Server
TFTP	Trivial File Transfer Protocol

References

1. Saba, T.; Haseeb, K.; Ahmed, I.; Rehman, A. Secure and energy-efficient framework using Internet of Medical Things for e-healthcare. *J. Infect. Public Health* **2020**, *13*, 1567–1575. [CrossRef]
2. Awan, M.J.; Masood, O.A.; Mohammed, M.A.; Yasin, A.; Zain, A.M.; Damaševičius, R.; Abdulkareem, K.H. Image-Based Malware Classification Using VGG19 Network and Spatial Convolutional Attention. *Electronics* **2021**, *10*, 2444. [CrossRef]
3. Noshad, Z.; Javaid, N.; Saba, T.; Wadud, Z.; Saleem, M.Q.; Alzahrani, M.E.; Sheta, O.E. Fault Detection in Wireless Sensor Networks through the Random Forest Classifier. *Sensors* **2019**, *19*, 1568. [CrossRef]
4. Ahmad, A.M.; Sulong, G.; Rehman, A.; Alkawaz, M.H.; Saba, T. Data Hiding Based on Improved Exploiting Modification Direction Method and Huffman Coding. *J. Intell. Syst.* **2014**, *23*, 451–459. [CrossRef]
5. Javaid, S.; Javaid, N.; Saba, T.; Wadud, Z.; Rehman, A.; Haseeb, A. Intelligent Resource Allocation in Residential Buildings Using Consumer to Fog to Cloud Based Framework. *Energies* **2019**, *12*, 815. [CrossRef]
6. Hussain, M.; Javed, W.; Hakeem, O.; Yousafzai, A.; Younas, A.; Awan, M.J.; Nobanee, H.; Zain, A.M. Blockchain-Based IoT Devices in Supply Chain Management: A Systematic Literature Review. *Sustainability* **2021**, *13*, 13646. [CrossRef]
7. Rehman, A.; Haseeb, K.; Saba, T.; Lloret, J.; Ahmed, Z. Mobility Support 5G Architecture with Real-Time Routing for Sustainable Smart Cities. *Sustainability* **2021**, *13*, 9092. [CrossRef]
8. Saba, T.; Rehman, A.; Latif, R.; Fati, S.M.; Raza, M.; Sharif, M. Suspicious Activity Recognition Using Proposed Deep L4-Branched-Actionnet With Entropy Coded Ant Colony System Optimization. *IEEE Access* **2021**, *9*, 89181–89197. [CrossRef]
9. Haseeb, K.; Almustafa, K.M.; Jan, Z.; Saba, T.; Tariq, U. Secure and Energy-Aware Heuristic Routing Protocol for Wireless Sensor Network. *IEEE Access* **2020**, *8*, 163962–163974. [CrossRef]
10. Awan, M.J.; Farooq, U.; Babar, H.M.A.; Yasin, A.; Nobanee, H.; Hussain, M.; Hakeem, O.; Zain, A.M. Real-Time DDoS Attack Detection System Using Big Data Approach. *Sustainability* **2021**, *13*, 10743. [CrossRef]
11. Saba, T.; Haseeb, K.; Din, I.U.; Almogren, A.; Altameem, A.; Fati, S.M. EGCIR: Energy-Aware Graph Clustering and Intelligent Routing Using Supervised System in Wireless Sensor Networks. *Energies* **2020**, *13*, 4072. [CrossRef]
12. Rashid, M.; Khan, M.A.; Alhaisoni, M.; Wang, S.-H.; Naqvi, S.R.; Rehman, A.; Saba, T. A Sustainable Deep Learning Framework for Object Recognition Using Multi-Layers Deep Features Fusion and Selection. *Sustainability* **2020**, *12*, 5037. [CrossRef]
13. Ferooz, F.; Hassan, M.T.; Awan, M.J.; Nobanee, H.; Kamal, M.; Yasin, A.; Zain, A.M. Suicide Bomb Attack Identification and Analytics through Data Mining Techniques. *Electronics* **2021**, *10*, 2398. [CrossRef]
14. Khan, A.Y.; Latif, R.; Latif, S.; Tahir, S.; Batool, G.; Saba, T. Malicious Insider Attack Detection in IoTs Using Data Analytics. *IEEE Access* **2019**, *8*, 11743–11753. [CrossRef]
15. Saba, T.; Haseeb, K.; Shah, A.A.; Rehman, A.; Tariq, U.; Mehmood, Z. A Machine-Learning-Based Approach for Autonomous IoT Security. *IT Prof.* **2021**, *23*, 69–75. [CrossRef]
16. Ali, S.F.; Aslam, A.S.; Awan, M.J.; Yasin, A.; Damaševičius, R. Pose Estimation of Driver's Head Panning Based on Interpolation and Motion Vectors under a Boosting Framework. *Appl. Sci.* **2021**, *11*, 11600. [CrossRef]
17. Saba, T.; Sadad, T.; Rehman, A.; Mehmood, Z.; Javaid, Q. Intrusion Detection System Through Advance Machine Learning for the Internet of Things Networks. *IT Prof.* **2021**, *23*, 58–64. [CrossRef]

18. Odusami, M.; Misra, S.; Adetiba, E.; Abayomi-Alli, O.; Damasevicius, R.; Ahuja, R. An Improved Model for Alleviating Layer Seven Distributed Denial of Service Intrusion on Webserver. *J. Physics Conf. Ser.* **2019**, *1235*. [CrossRef]
19. Saba, T. Intrusion Detection in Smart City Hospitals using Ensemble Classifiers. In Proceedings of the 2020 13th International Conference on Developments in eSystems Engineering (DeSE), Liverpool, UK, 14–17 December 2020; pp. 418–422.
20. Mujahid, A.; Awan, M.; Yasin, A.; Mohammed, M.; Damaševičius, R.; Maskeliūnas, R.; Abdulkareem, K. Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. *Appl. Sci.* **2021**, *11*, 4164. [CrossRef]
21. Ali, M.H.; Jaber, M.M.; Abd, S.K.; Rehman, A.; Awan, M.J.; Vitkutė-Adžgauskienė, D.; Damaševičius, R.; Bahaj, S.A. Harris Hawks Sparse Auto-Encoder Networks for Automatic Speech Recognition System. *Appl. Sci.* **2022**, *12*, 1091. [CrossRef]
22. Haafza, L.A.; Awan, M.J.; Abid, A.; Yasin, A.; Nobanee, H.; Farooq, M.S. Big Data COVID-19 Systematic Literature Review: Pandemic Crisis. *Electronics* **2021**, *10*, 3125. [CrossRef]
23. Awan, M.J.; Yasin, A.; Nobanee, H.; Ali, A.A.; Shahzad, Z.; Nabeel, M.; Zain, A.M.; Shahzad, H.M.F. Fake News Data Exploration and Analytics. *Electronics* **2021**, *10*, 2326. [CrossRef]
24. Rehman, A.; Haseeb, K.; Saba, T.; Lloret, J.; Sendra, S. An Optimization Model with Network Edges for Multimedia Sensors Using Artificial Intelligence of Things. *Sensors* **2021**, *21*, 7103. [CrossRef]
25. Rehman, A.; Haseeb, K.; Saba, T.; Lloret, J.; Tariq, U. Secured Big Data Analytics for Decision-Oriented Medical System Using Internet of Things. *Electronics* **2021**, *10*, 1273. [CrossRef]
26. Khan, M.A.; Abuhasel, K.A. An evolutionary multi-hidden Markov model for intelligent threat sensing in industrial internet of things. *J. Supercomput.* **2021**, *77*, 6236–6250. [CrossRef]
27. Gerard, A.; Latif, R.; Latif, S.; Iqbal, W.; Saba, T.; Gerard, N. MAD-Malicious Activity Detection Framework in Federated Cloud Computing. In Proceedings of the 2020 13th International Conference on Developments in eSystems Engineering (DeSE), Liverpool, UK, 14–17 December 2020; pp. 273–278.
28. Dange, S.; Chatterjee, M. IoT Botnet: The Largest Threat to the IoT Network. In *Data Communication and Networks*; Springer: Singapore, 2020; pp. 137–157.
29. Waheed, N.; He, X.; Ikram, M.; Usman, M.; Hashmi, S.; Usman, M. Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–37. [CrossRef]
30. Liu, J.; Yang, D.; Lian, M.; Li, M. Research on Intrusion Detection Based on Particle Swarm Optimization in IoT. *IEEE Access* **2021**, *9*, 38254–38268. [CrossRef]
31. Eskandari, M.; Janjua, Z.H.; Vecchio, M.; Antonelli, F. Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices. *IEEE Internet Things J.* **2020**, *7*, 6882–6897. [CrossRef]
32. Anthi, E.; Williams, L.; Slowinska, M.; Theodorakopoulos, G.; Burnap, P. A Supervised Intrusion Detection System for Smart Home IoT Devices. *IEEE Internet Things J.* **2019**, *6*, 9042–9053. [CrossRef]
33. Zhang, Y.; Li, P.; Wang, X. Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network. *IEEE Access* **2019**, *7*, 31711–31722. [CrossRef]
34. Pajouh, H.H.; Javidan, R.; Khayami, R.; Dehghantanha, A.; Choo, K.-K.R. A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks. *IEEE Trans. Emerg. Top. Comput.* **2016**, *7*, 314–323. [CrossRef]
35. Li, J.; Zhao, Z.; Li, R.; Zhang, H. AI-Based Two-Stage Intrusion Detection for Software Defined IoT Networks. *IEEE Internet Things J.* **2019**, *6*, 2093–2102. [CrossRef]
36. Al-Hamadi, H.; Chen, I.-R.; Wang, D.-C.; Almashan, M. Attack and Defense Strategies for Intrusion Detection in Autonomous Distributed IoT Systems. *IEEE Access* **2020**, *8*, 168994–169009. [CrossRef]
37. Moustafa, N.; Turnbull, B.; Choo, K.-K.R. An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 4815–4830. [CrossRef]
38. Khan, B.U.I.; Anwar, F.; Olanrewaju, R.F.; Pampori, B.R.; Mir, R.N. A Novel Multi-Agent and Multilayered Game Formulation for Intrusion Detection in Internet of Things (IoT). *IEEE Access* **2020**, *8*, 98481–98490. [CrossRef]
39. Azeez, N.A.; Ayemobola, T.J.; Misra, S.; Maskeliūnas, R.; Damaševičius, R. Network Intrusion Detection with a Hashing Based Apriori Algorithm Using Hadoop MapReduce. *Computers* **2019**, *8*, 86. [CrossRef]
40. Nie, L.; Ning, Z.; Wang, X.; Hu, X.; Cheng, J.; Li, Y. Data-Driven Intrusion Detection for Intelligent Internet of Vehicles: A Deep Convolutional Neural Network-Based Method. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 2219–2230. [CrossRef]
41. Shafiq, M.; Tian, Z.; Bashir, A.K.; Du, X.; Guizani, M. CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques. *IEEE Internet Things J.* **2021**, *8*, 3242–3254. [CrossRef]
42. Alarcon-Aquino, V.; Ramirez-Cortes, J.M.; Gomez-Gil, P.; Starostenko, O.; Garcia-Gonzalez, Y. Network Intrusion Detection Using Self-Recurrent Wavelet Neural Network with Multidimensional Radial Wavelons. *Inf. Technol. Control* **2014**, *43*, 347–358. [CrossRef]
43. Alharbi, A.; Alosaimi, W.; Alyami, H.; Rauf, H.; Damaševičius, R. Botnet Attack Detection Using Local Global Best Bat Algorithm for Industrial Internet of Things. *Electronics* **2021**, *10*, 1341. [CrossRef]
44. Damaševičius, R.; Venčkauskas, A.; Toldinas, J.; Grigaliūnas, Š. Ensemble-Based Classification Using Neural Networks and Machine Learning Models for Windows PE Malware Detection. *Electronics* **2021**, *10*, 485. [CrossRef]
45. Toldinas, J.; Venčkauskas, A.; Damaševičius, R.; Grigaliūnas, Š.; Morkevičius, N.; Baranauskas, E. A Novel Approach for Network Intrusion Detection Using Multistage Deep Learning Image Recognition. *Electronics* **2021**, *10*, 1854. [CrossRef]

46. Sodhro, A.H.; Sangaiah, A.K.; Sodhro, G.H.; Lohano, S.; Pirbhulal, S. An Energy-Efficient Algorithm for Wearable Electrocardiogram Signal Processing in Ubiquitous Healthcare Applications. *Sensors* **2018**, *18*, 923. [CrossRef]
47. Muzammal, M.; Talat, R.; Sodhro, A.H.; Pirbhulal, S. A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks. *Inf. Fusion* **2020**, *53*, 155–164. [CrossRef]
48. Canadian Institute for Cybersecurity. DDoS Evaluation Dataset (CIC-DDoS2019). 2019. Available online: https://www.unb.ca/cic/datasets/ddos-2019.html (accessed on 29 November 2021).
49. Javed Awan, M.; Shafry Mohd Rahim, M.; Nobanee, H.; Munawar, A.; Yasin, A.; Mohd Zain Azlanmz, A. Social Media and Stock Market Prediction: A Big Data Approach. *Comput. Mater. Contin.* **2021**, *67*, 2569–2583. [CrossRef]
50. Awan, M.J.; Gilani, S.A.H.; Ramzan, H.; Nobanee, H.; Yasin, A.; Zain, A.M.; Javed, R. Cricket Match Analytics Using the Big Data Approach. *Electronics* **2021**, *10*, 2350. [CrossRef]
51. Javed Awan, M.; Shafry Mohd Rahim, M.; Nobanee, H.; Yasin, A.; Ibrahim Khalaf, O.; Ishfaq, U. A Big Data Approach to Black Friday Sales. *Intell. Autom. Soft Comput.* **2021**, *27*, 785–797. [CrossRef]
52. Awan, M.; Khan, R.; Nobanee, H.; Yasin, A.; Anwar, S.; Naseem, U.; Singh, V. A Recommendation Engine for Predicting Movie Ratings Using a Big Data Approach. *Electronics* **2021**, *10*, 1215. [CrossRef]
53. Awan, M.J.; Rahim, M.S.M.; Salim, N.; Mohammed, M.A.; Garcia-Zapirain, B.; Abdulkareem, K.H. Efficient Detection of Knee Anterior Cruciate Ligament from Magnetic Resonance Imaging Using Deep Learning Approach. *Diagnostics* **2021**, *11*, 105. [CrossRef]
54. Awan, M.J.; Bilal, M.H.; Yasin, A.; Nobanee, H.; Khan, N.S.; Zain, A.M. Detection of COVID-19 in Chest X-ray Images: A Big Data Enabled Deep Learning Approach. *Int. J. Environ. Res. Public Health* **2021**, *18*, 10147. [CrossRef]
55. Awan, M.J.; Rahim, M.S.M.; Salim, N.; Rehman, A.; Nobanee, H.; Shabir, H. Improved Deep Convolutional Neural Network to Classify Osteoarthritis from Anterior Cruciate Ligament Tear Using Magnetic Resonance Imaging. *J. Pers. Med.* **2021**, *11*, 1163. [CrossRef] [PubMed]