IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 5, 2021, accepted April 12, 2021, date of publication April 21, 2021, date of current version April 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074664

Intrusion Detection System Based on Fast Hierarchical Deep Convolutional Neural Network

ROBSON V. MENDONÇA[1], ARTHUR A. M. TEODORO[1], RENATA L. ROSA[1], MUHAMMAD SAADI[2], DICK CARRILLO MELGAREJO[3], (Member, IEEE), PEDRO H. J. NARDELLI[3], (Senior Member, IEEE), AND DEMÓSTENES Z. RODRÍGUEZ[1], (Senior Member, IEEE)

[1]Departament of Computer Science, Federal University of Lavras, Lavras 37200-900, Brazil
[2]Department of Electrical Engineering, University of Central Punjab, Lahore 151001, Pakistan
[3]School of Energy Systems, Lappeenranta—Lahti University of Technology, 53850 Lappeenranta, Finland

Corresponding author: Pedro H. J. Nardelli (pedro.nardelli@lut.fi)

This work was supported in part by the Academy of Finland through ee-IoT under Grant 319009, through EnergyNet under Grant 321265 and Grant 328869, and through FIREMAN under Grant 326270/CHIST-ERA-17-BDSI-003, and in part by the Brazilian National Council for Scientific and Technological Development (CNPq).

ABSTRACT Currently, with the increasing number of devices connected to the Internet, search for network vulnerabilities to attackers has increased, and protection systems have become indispensable. There are prevalent security attacks, such as the Distributed Denial of Service (DDoS), which have been causing significant damage to companies. However, through security attacks, it is possible to extract characteristics that identify the type of attack. Thus, it is essential to have fast and effective security identification models. In this work, a novel Intrusion Detection System (IDS) based on the Tree-CNN hierarchical algorithm with the Soft-Root-Sign (SRS) activation function is proposed. The model reduces the training time of the generated model for detecting DDoS, Infiltration, Brute Force, and Web attacks. For performance assessment, the model is implemented in a medium-sized company, analyzing the level of complexity of the proposed solution. Experimental results demonstrate that the proposed hierarchical model achieves a significant reduction in execution time, around 36%, and an average detection accuracy of 0.98 considering all the analyzed attacks. Therefore, the results of performance evaluation show that the proposed classifier based on Tree-CNN is of low complexity and requires less processing time and computational resources, outperforming other current IDS based on machine learning algorithms.

INDEX TERMS Activation function, deep learning, intrusion detection system, Tree-CNN.

I. INTRODUCTION

The number of devices connected to the Internet has grown steadily [1]–[3]. Forecasts show that this number will reach 50 billion by 2022 [4], [5]. Accompanying this growth, vulnerabilities and virtual attacks are also expected to grow in the same proportion [6].

To prevent virtual attacks, an existing solution is the Intrusion Detection System (IDS). However, these systems generally work based on signatures [7]. The signatures could be a problem because if there is an unknown attack type, it is treated as legitimate access. To resolve this issue, several approaches [7]–[11] have been proposed for the intrusion detection methods used by IDS, based on traffic analysis and anomalies. Anomalous traffic is identified based on recognition patterns [12]. These patterns can be set manually or automatically by a Honeypot type system [13], [14] or by machine learning algorithms [15], [16]. Machine learning algorithms have been widely used for intrusion detection in computer networks, as in [17], [18], which compare the effectiveness of Support Vector Machine (SVM) [19], Naive Bayes (NB), Decision Tree (DT), and Multilayer algorithms Perceptron (MLP) in detecting distributed attacks. In [6], [20], the authors demonstrate the effectiveness of deep learning algorithms in detecting attacks of the Probe, Remote to Local, User to Root, and Denial of Service (DoS) types. However, these studies show inferior detection accuracy in actual environments.

Studies propose different methods for the detection of web attacks, obtaining an accuracy below 98%

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

[17], [21]–[23]. However, even when using techniques to increase the classification speed in these studies, such as the Principal Component Analysis (PCA), the time spent in generating the model is still considerable [24]. For security attacks in particular, it is essential to have fast and highly accurate models.

Manimurugan *et al.* [23] used Deep Learning to develop a network intrusion detection method. The method was based on Deep Belief Networks (DBN), which achieved an accuracy of 97.71 % for Brute Force attacks, 96.67 % for Dos/DDoS, 96.37 % for Infiltration, and 98.37 % for Web attacks. The intrusion detection method implemented by Manimurugan *et al.* [23] obtains high accuracy values for detecting network attacks. However, it does not reduce the training time to generate new detection models. As security attacks change over time, it is very useful to find a solution that takes as little training time as possible.

To improve the accuracy of a neural network model [25], an activation function is introduced, defining the output of a node. Thus, activation functions are a significant step in a deep neural network, providing a nonlinear property for the neural network. Therefore, the activation function is crucial for good behavior and learning performance [26]. In recent years, many activation functions have been proposed to replace those already known, such as Rectified Linear Units (ReLU) [27], including Randomized Leaky Rectified Linear Activation (RLReLU), Swish [28], Maxout [29], and others [30].

As an alternative to the previously mentioned activation functions, Soft-Root-Sign (SRS) [31] has shown promising results for obtaining faster training models. SRS can adaptively adjust the output through a pair of independent trainable parameters that present a better generalization performance and a faster learning speed. Thus, this activation function is chosen to be implemented in this work in order to increase the learning speed of the generated model.

In this work, a model for network intrusion detection was proposed, applying machine learning techniques with Tree-CNN that do not present the problem of data forgetting on the retraining phase of new models [32], in case a new model needs to be generated. It is important to note that there are few studies on the implementation and testing of SRS in the Tree-CNN algorithm with application in intrusion detection.

For the creation of the model, the accuracy results obtained from the machine learning algorithms NB, SVM, RF, MLP, Tree-CNN (ReLU), Tree-CNN (Softmax), and Tree-CNN (SRS) were compared. The algorithms NB, SVM, RF, and MLP used in this work are widely applied in related works for detecting web attacks [4], [20], [33]–[35].

Concerning evaluation of the model proposed in this work, the results were compared with those obtained in [23] using the same CICIDS2017 dataset [21], which explores the same types of security attacks. However, our work aims at achieving results superior to the accuracy values obtained by the DBN system [23] in the detection of the following types of attacks: DDoS, Infiltration, Brute Force, and Web attack.

Additionally, the model is tested by using other two network intrusion scenarios; an actual scenario of a university campus and a medium-sized company.

The experimental tests presented an average accuracy of 0.98, demonstrating that the Tree-CNN algorithm with the SRS activation function achieves high attack detection results.

The main contributions presented in this paper are summarized as follows:

1) Validation of the use of the Tree-CNN algorithm to classify detection of DDoS, Infiltration, Brute Force, and Web attacks in the CICIDS2017 dataset and in two actual network scenarios. Currently, Tree-CNN is applied in different approaches, such as image classification, but to the best of our knowledge, it has not been implemented in IDS.
2) Reduced training time for the Tree-CNN algorithm for IDS, applying the SRS activation function. The Softmax and ReLU activation functions were also used to obtain performance comparison results.
3) Performance validation of the proposed solution in real environments. Thus, it was possible to build a new dataset containing actual network data.

The rest of this article is structured as follows: Section II presents the related works and the main definitions of concept, and Section III introduces the proposed Tree-CNN using different activation functions. In Section IV, experimental results are provided and discussed. Finally, conclusions are drawn in Section V.

II. RELATED WORK

IDSs have been implemented in several studies [36]–[39], as they are fundamental in protecting computer networks against virtual attacks.

Traditional IDS are based on fixed or dynamic rules [7], [40], [41] to identify attacks on the network. However, attackers employ various techniques to camouflage their attacks and confuse the defense systems of the target. Furthermore, detection is performed based on pre-existing signatures or anomalous traffic. Signature detection is a failure for zero-day attacks [42], [43]; as attacks of this type are unknown, atypical traffic detection can be executed by machine learning algorithms [6], comparing regular patterns with patterns attacks, and classifying traffic.

There are several challenges for adjusting and correctly configuring an IDS [16], as is the case with zero-day attacks, in which IDS has no prior knowledge of the vulnerability. As stated in [16], current and realistic datasets for training machine learning algorithms are critical for machine learning-based IDS. Optimization of these datasets is also necessary because with fewer attributes, the classifications can be faster.

In this context, a study [44] was carried out on the datasets DARPA98 [44], [45], KDD99 [44]–[47], ISCX2012 [44], and ADFA13 [44], [48]. In this study, the authors point out that the main problem in research is the propagation of errors contained in problematic datasets, as is the case with KDD99 [10], [49], which has a large number of redundant records. According to [9], about 78% of training records and 75% of test records repeat in the KDD99 set. The comparison of efficiency between several traditional (shallow) machine learning and deep learning algorithms is the subject of studies in [6], [17], [20], [22].

In assessing the performance of Machine Learning (ML) algorithms in the NSL-KDD dataset, in [17], the SVM, Naive Bayes, Decision Tree, and Artificial Neural Network (ANN) with MLP have been studied for detecting attacks of the following types: DoS, Probe, Remote to Local, and User to Root [47], [50]. In [17], an accuracy of 97.72% was shown for SVM and 97.82% for ANN. According to the authors, this was possible by adjusting the *c* and *gamma* parameters for SVM, while for ANN, the best result was achieved with four layers. Despite not showing a significant difference between the results of the two algorithms that obtained a better performance, the authors pointed out that the classification speed was much higher after applying the PCA technique in reducing the dimensionality of the NSL-KDD dataset [9]. Thus, the need for optimizing the dataset is confirmed according to [16], [44], [51]–[53].

ML algorithms for network intrusion detection were also evaluated in [18], in which the results of the Logistic Regression, K-Nearest Neighbor (KNN), Gaussian Naive Bayes, Random Forest, Linear SVM, and Linear algorithms Discriminant Analysis are presented for detecting DDoS attacks. The authors used the CICIDS2017 dataset [21] as a source for testing and training the algorithms. The algorithm with the highest efficiency in the classification was Random Forest, reaching 96.2% accuracy. The authors attribute the positive result to the cross-validation technique used by the algorithms. This technique eliminates the problem known as underfitting, which refers to the creation of a model with little data, making this model not correctly represent a real environment [54]. The studies also reinforce the need to reduce the dimensionality of the dataset [18]; the SelectPercentile method is used for reducing the number of attributes from 85 to 12, in addition to replacing the Not a Number (NaN) values with the median of the values in the same column.

In [23], the authors created a method for detecting network intrusion based on DBN. The method consists of several layers of Restricted Boltzman Machines (RBM), using a greedy algorithm in the unsupervised training phase. According to [23], the algorithm optimizes each training layer, applying fine adjustments of parameters of the network layers. For training and testing of the model, the CICIDS2017 dataset [21] was used. The study achieved an accuracy of 97.71 % for Brute Force attacks, 96.67 % for Dos/DDoS, 96.37 % for Infiltration, and 98.37 % for Web attacks. The model proposed in [23] showed superior results when compared with others, such as SVM, Recurrent Neural Network (RNN), Spiking Neural Network (SNN), and Feedforward Neural Network (FNN). For this reason, our work was compared with the DBN method [23].

III. TREE-CNN AND ACTIVATION FUNCTIONS

This section presents the Tree-CNN algorithm and the Soft-Root-Sign (SRS) activation function. Both have previously been studied mainly in image classification [32], [55]. However, in this research, the Tree-CNN algorithm and the SRS are applied to the classification of network traffic.

A. TREE-CNN

Traditional CNN models are designed to be sequential and generate detection at the top [56] of the network, that is, without branches. In hierarchical models, branches can refine the classifications performed by the network trunk. The initial layers of CNN learn generic resources [57], and this feature is used to transfer learning data [58]. In a hierarchical CNN, such as Tree-CNN, the upper nodes generally classify the classes using simple resources [59], decreasing the complexity in the training phase. The hierarchical models of CNN demonstrate better performance than the Deep Convolutional Neural Network (DCNNs) models [59].

A Tree-CNN starts as a single root node and, subsequently, new hierarchies are created to accommodate new classes. A similar topology is applied in [60], where the new classes are added to the old class, divided into two superclasses using an error-based model.

Tree-CNN is used in [32]. However, the topology is applied in a different scenario, a car image testing, where the precision results are greater than 85%. In this work, the contribution is to adapt and test the Tree-CNN topology in a scenario of computer network traffic classification for attack detection.

The steps of a Tree-CNN are usually:

- Initially, the neural network is trained to classify data into N classes. The data belonging to a new class are presented to the neural network, and then, the network grows to accommodate the new class;
- The neural network grows by adding branches or leaf nodes to the current structure;
- The objective of reducing the training effort is composed of two components: the number of updated weights and the number of examples, old or new, required for training;
- Finally, the updates are located in a sector of the tree.

In a Tree-CNN, $P(x_t, Tr)$ represents the probability that the input data will be correctly classified in a category by the neural network trunk net. There are two subnets, one to encode the input function in a fixed number of classes, which can be a leaf node or a branch, and the other to encode the locations for the output functions, the neural network trunk net.

The function $P(C_i, Tr)$ represents the probability that the input data will be correctly classified in the *i*-th category,

whereas $P(x_t, Br)$ is the probability that the input data is correctly classified in the i-th category by a branch of the network. In the branch, $P(x_t, Br) > P(x_t, Tr)$, as the branch of the network is responsible for distinguishing the i-th category from other similar categories, where it is different from the trunk net of the network. Assuming that $P(C_i, Tr)$ is close to 1, the probability of Tree-CNN $P(x_t, Br)P(C_i, Tr)$ correctly classifying the category in branch A is higher than the probability of $P(x_t, Tr)$.

B. ACTIVATION FUNCTIONS

Activation functions play a crucial role in the structure of artificial neural networks to solve complex problems. In this work, the activation function "Soft-Root-Sign" (SRS) [26], [61]–[63] will be used. This function can be adjusted adaptively to the output through a pair of independent trainable parameters, with a better generalization performance and a faster learning speed.

The SRS activation function is defined by:

$$SRS(p) = \frac{p}{\frac{p}{\alpha} + e^{-\frac{p}{\beta}}} \tag{1}$$

where α and β variables are a pair of trainable positive parameters. The SRS presents a nonmonotonic region in which $p < 0$ provides the property with zero mean. When $p > 0$, it avoids and rectifies the output distribution. The SRS derivative is defined as follows:

$$SRS'(p) = \frac{(1 + \frac{p}{\beta})e^{-\frac{p}{\beta}}}{(\frac{p}{\alpha} + e^{-\frac{p}{\beta}})^2} \tag{2}$$

The output of an SRS is limited by the range $\frac{\alpha\beta}{\beta - \alpha e}, \alpha$.

In our experimental tests, other activation functions were used, such as Softmax and ReLU. These activation functions were used for comparison purposes.

IV. METHODOLOGY

In this section, the main steps to obtain the proposed IDS model for classifying attacks will be presented, using the Tree-CNN (SRS).

The Tree-CNN uses a maximum depth of 4 with a maximum number of child nodes for a branch node of 10. The model works with 11 layers, containing four convolutional blocks, each block having two sets of 3×3 convolutional kernels with a momentum of 0.9. A weight decay of 0.01 is used, in which all CNNs are trained for 300 epochs.

Fig. 1 illustrates the methodology followed to analyze the network traffic flow as well as the application of the deep learning algorithm for detecting malicious traffic. Firstly, the network traffic was captured from a university campus to build a dataset. This dataset was used for the training and initial testing phase. Later, a PCA technique was applied, its output feeding the ML model by using the Tree-CNN and the SRS activation function. Finally, the classification into anomalous or not anomalous traffic was performed, as well its validation. It is important to note that the two datasets are used for performance evaluation of the proposed IDS, one of them being an existing public dataset named CICIDS2017 [21], and the second one built using the network traffic of a medium-sized company.

The blocks introduced in Fig.1 are described below.

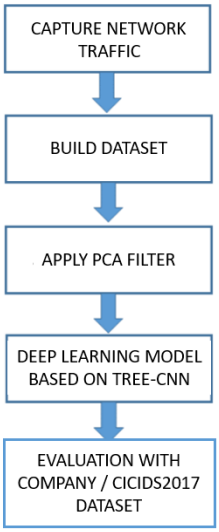


FIGURE 1. Methodology for the network traffic analysis using the deep learning model.

A. CAPTURING THE NETWORK TRAFFIC

Firstly, the network traffic is captured to be analyzed for the proposed system.

The traffic is collected by using the tcpdump tool and converted into a readable format by the Waikato Environment for the Knowledge Analysis (WEKA) software and the TensorFlow library [64].

The network traffic, extracted from a university campus, is captured to form the dataset to be built. It is important to note that the classification of the data to form the dataset is performed by three specialists, which classify the attacks into DDoS, Infiltration, Brute Force, and Web attack. The specialists analyze the main characteristics, such as flow number, source flow, and address variation. According to these features, the labeling is performed to determine whether it was an attack or not, and which kind of attack it was.

B. DATASET

In this work, three datasets are used, one for training and testing the proposed model, in which data are extracted from a university campus, and two more datasets for performance evaluation, extracted from a medium-sized company and the CICIDS2017 [21].

1) DATASET BUILT IN THE STUDY

In this work, a new dataset is created for the training and perform initial tests of the algorithm. The data contain benign traffic and attacks. The new dataset is obtained in a common

LAN network topology of a university campus, with eight subnets installed, divided into Dep1 to Dep8 and server room. For all the departments, different operating systems are used, such as Microsoft Windows 7, Windows 8.1, and Windows 10, and the Linux Ubuntu distribution. Different Microsoft Windows servers, such as 2012 and 2016 are used for the server room.

Table 1 shows the number of computers in each department and server room.

TABLE 1. Distribution of machines between departments in the university campus scenario.

Department	Number of computers
Dep1	50
Dep2	30
Dep3	45
Dep4	40
Dep5	40
Dep6	15
Dep7	45
Dep8	45
Server Room	15

A graphical representation of the network topology is presented in Fig. 2.

A Linux distribution called Kali Linux is used for executing the attacks. This distribution focuses on auditing and security of computers and networks. It is widely used in the security area, as it has a set of preinstalled tools that facilitate the audit process.

The dataset of the scenario was built with the types of attacks studied in this study; DDoS, Infiltration, Brute Force, and Web, as well as benign traffic, thus forming a heterogeneous basis.

All traffic was captured from the attack period with the tcpdump tool and stored, thereby creating records of benign (regular) traffic and malicious traffic produced by the attack.

After the network traffic was captured, the data were treated and the dataset was built.

Table 2 presents a brief survey of the characteristics of the main public datasets, where Dataset1 refers to the dataset built of the university campus and Dataset2 refers to the medium-sized company scenario. In the comparison with our datasets, the evolution of datasets is observed, mainly in the zero-day type of attack, in which there is no knowledge of the attack, only traffic patterns that suggest an attack in progress. Additionally, the majority of the datasets do not have characteristics of new devices like IoTs, which is a missing characteristic in present days. On the other hand, our datasets have more actual characteristics with more up-to-date common attacks.

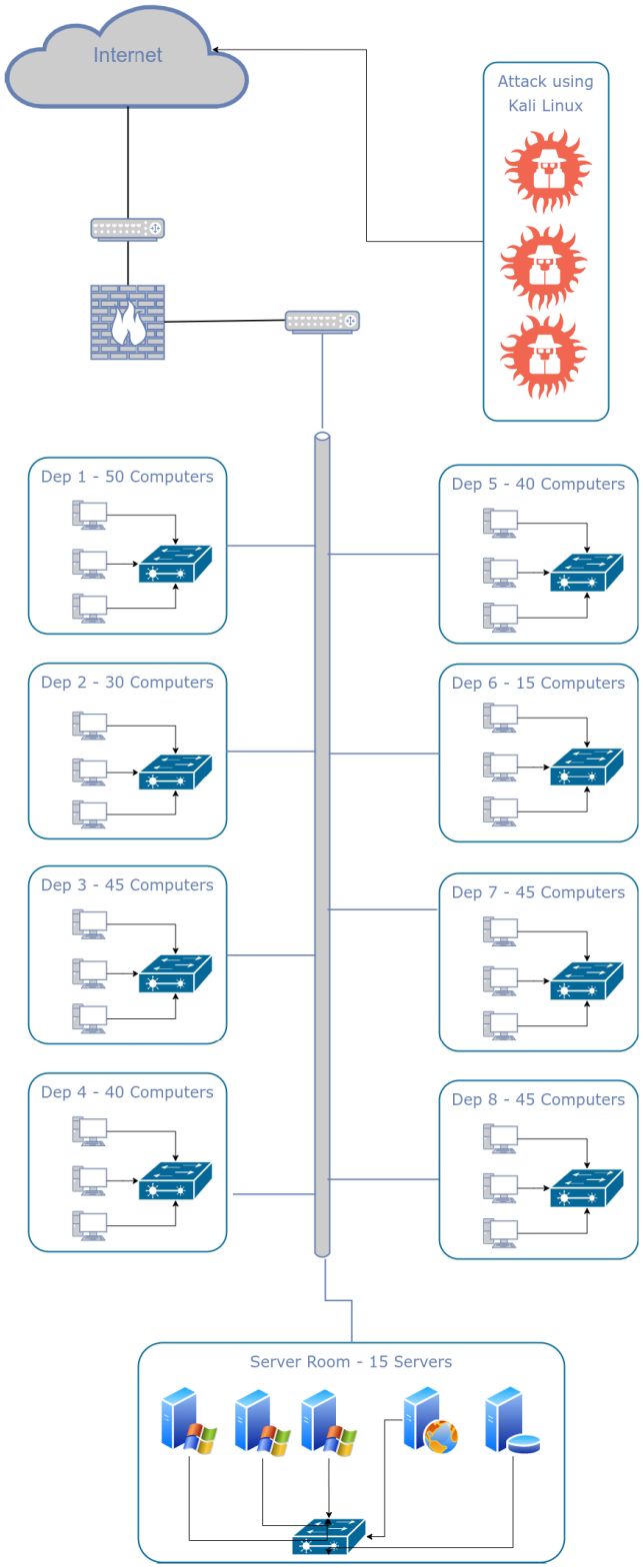


FIGURE 2. Diagram of the network setting used to form the dataset built for the University campus scenario.

2) EVALUATION DATASETS

The proposed system was evaluated with the CICIDS2017 dataset [21] used in [23]. The CICIDS2017 dataset [21] presents benign and common attacks, for instance, DDoS, Infiltration, Brute Force, and Web attack, which resembles the true real-world data. For this dataset, abstract behavior of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols was built.

TABLE 2. Brief survey of the characteristics of the main public datasets.

Dataset	Real traffic	IoT traces	Zero-day	year
DARPA 98	no	no	no	1998
KDDCUP 99	no	no	no	1999
ADFA13	yes	no	yes	2013
CICIDS2017	yes	no	yes	2017
Dataset1	yes	no	yes	2020
Dataset2	yes	yes	yes	2020

The CICIDS2017 [21] data were captured from the period starting at 9 a.m., Monday, July 8th, 2019, and ending at 5 p.m. on Friday, July 12th, 2019, for a total of five days. Monday represents the typical day and only includes benign traffic. The attacks were executed both in the morning and afternoon on Tuesday, Wednesday, Thursday, and Friday.

Additionally, the proposed system applies a real scenario of a medium-sized company. This scenario consists of eight departments, with 1,126 interconnected devices, including workstations, printers and copiers, telephones with Voice over IP (VoIP) technology, mobile devices via a wireless network, and servers. The departments are interconnected by routers, and within the departments, the devices are interconnected by a network wired to the switch, in addition to wireless network devices. All devices are connected to the Internet.

Fig. 3 shows the model using the Tree-CNN algorithm and the SRS activation function applied in an authentic environment in a medium-sized company. This scenario was chosen because it is one of the main targets of attacks, as it is the typical network environment in companies, where successful attacks can produce financial gains to attackers.

The activities on the network are composed of access to the Internet, namely visits to web pages, sending and receiving e-mails, and telephone calls using VoIP.

C. APPLICATION OF THE PRINCIPAL COMPONENT ANALYSIS FILTER

In the preliminary tests, the Principal Component Analysis (PCA) technique used to choose the characteristics and the effectiveness of selecting the main attributes in the analysis of attacks were verified.

In this work, the PCA was used to reduce the dimensionality of the data. The Java programming language was applied to the implementation. Initially, 41 features were extracted from the dataset, which are the traditional features. After the PCA application, a reduction occurred for 12 features, which are related to date, time, time period, source address, network congestion flow, keywords, normal flow, current number of users, old users, new users, access ports, and protocol. Initial experimental tests were performed, and the results showed that the feature reduction did not affect the accuracy of the different kinds of attacks.

After collecting and classifying data, a model with such data was generated, and this model was then used to classify new input data.

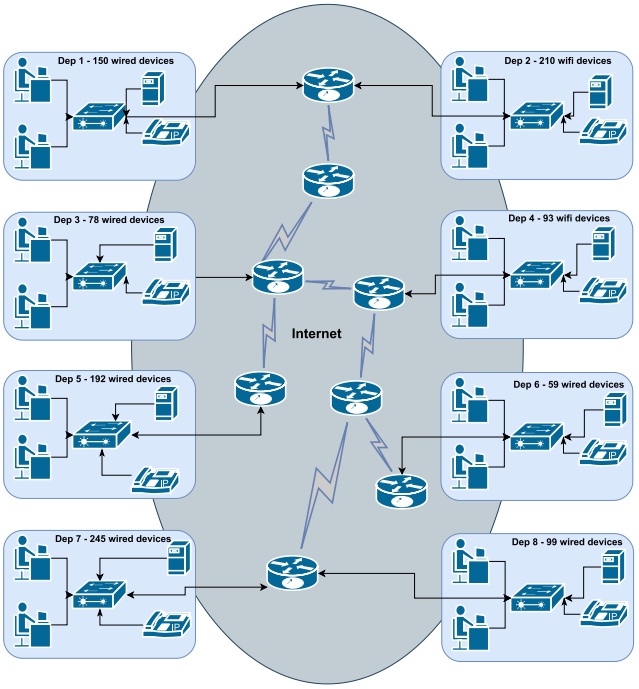


FIGURE 3. Computer network environment diagram used in a medium-sized company.

D. DEEP LEARNING MODEL BASED ON TREE-CNN

In this work, a hierarchical model was used, that is, with several CNNs. This topology showed better results than the single-layer models, in which the CNN acts as a root node with several leaf nodes. In the Tree-CNN, a new learning task is defined to identify attacks in this context, and a sample containing the benign network traffic and attacks is inserted in the root node.

Then, a dimensional matrix is obtained from the output layer with the number of children of the root node. The SRS probability function was used for this topology.

Fig. 4 illustrates a representation of the Tree-CNN. In the figure, the Tree-CNN is represented by the input node (ROOT), the branch nodes are the intermediaries having a father and two or more children, and the leaf node represents the last level of the tree. The samples represent the captured traffic that is classified by the Tree-CNN model.

To obtain the model, the dataset was divided into 60% for the training phase, 20% for validation, and 20% for testing. Twelve attributes of the captured traffic were extracted. These attributes were labeled by the classifier after going through Tree-CNN, which generated an estimated value for each traffic sample.

As stated before, algorithms such as NB, SVM, RF, MLP were used for a comparison with the proposed model. These algorithms have been studied in several papers, e.g., [17], [18], [35], [65], [66] to detect computer network intrusion.

The WEKA software was used for the single-layer ML algorithms, while the implementation was performed in Python 3.5 for the Deep Learning algorithms, with the TensorFlow, Numpy, Pandas, and Jupyter libraries.

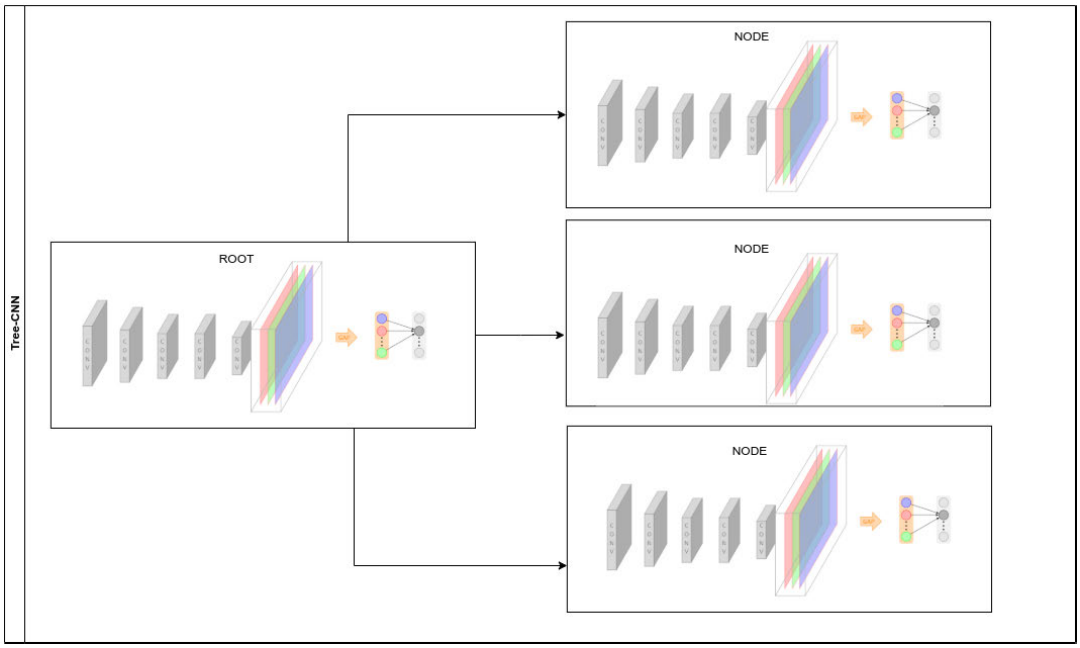


FIGURE 4. Tree-CNN topology developed in this work for classifying network traffic.

The Tree-CNN algorithm used a batch size of 10, a momentum of 0.8, a learning rate of 0.01, training 50 epochs, and a dropout rate of 0.5 was used. The values were chosen based on experimental tests.

E. EVALUATION OF THE PROPOSED MODEL

Two scenarios were used for the evaluation of the proposed algorithm; the dataset of a medium-sized company and the CICIDS2017 [21] dataset.

Additionally, the DBN system [23] was chosen as the model for the comparative assessment of the proposed attack detection system. It is important to note that the DBN system [23] was selected for comparison because it detects the same kinds of attacks as the ones studied in our work. Further, for comparison, the DBN system [23] was also tested in the same dataset that was used in our work, viz. the medium-sized company.

Furthermore, and for comparison purposes, our detection system based on the Tree-CNN was also implemented with the Softmax and ReLu activation functions. Thus, the use of the SRS activation function can be properly justified.

F. PERFORMANCE MEASURES

The evaluation of the effectiveness of each classifier in ML is measured based on the results of accuracy, recall, precision, and F-measure.

Measurement calculations are expressed by the following equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$F\text{-}measure = \frac{2 \times (precision \times recall)}{precision + recall} \tag{6}$$

The values for the variables True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are determined from the confusion matrix.

V. RESULTS AND DISCUSSION

This section describes the results of experiments related to the performance evaluation of the Tree-CNN algorithm with the SRS activation function.

A. PERFORMANCE OF THE TREE-CNN ALGORITHM AND THE SRS ACTIVATION FUNCTION IN THE DATASET BUILT IN THIS STUDY

According to Tables 3, 4, 5, and 6 the Tree-CNN and the SRS activation function achieved the highest average accuracy (0.99/0.98/0.99/0.98), sensitivity (0.98/0.98/0.98/0.99), precision (0.97/0.99/0.96/0.99), and F-measure (0.98/0.98/0.98/0.98) for classifying the DDoS, Infiltration, Web, and Brute force attack in the university campus scenario.

It is worth noting that the Tree-CNN in conjunction with other activation functions, such as Softmax and ReLU, presented better results than the other algorithms.

The results of the machine learning algorithms in the training phase are presented in Table 3 for the DDoS attack,

TABLE 3. Results of the machine learning algorithms for classifying the DDoS attack in the training phase for the dataset built.

Model	Accuracy	Recall	Precision	F-Measure
NB	0.79	0.83	0.80	0.80
SVM	0.82	0.84	0.83	0.82
RF	0.85	0.86	0.85	0.85
MLP	0.89	0.89	0.88	0.89
Tree-CNN (ReLU)	0.93	0.94	0.95	0.93
Tree-CNN (Softmax)	0.95	0.94	0.94	0.94
Tree-CNN (SRS)	0.99	0.98	0.97	0.98

TABLE 4. Results of the machine learning algorithms to classify the Infiltration attack in the training phase for the dataset built.

Model	Accuracy	Recall	Precision	F-Measure
NB	0.78	0.83	0.79	0.79
SVM	0.82	0.84	0.81	0.82
RF	0.85	0.86	0.83	0.85
MLP	0.89	0.89	0.87	0.89
Tree-CNN (ReLU)	0.94	0.93	0.94	0.93
Tree-CNN (Softmax)	0.96	0.94	0.93	0.94
Tree-CNN (SRS)	0.98	0.98	0.99	0.98

TABLE 5. Results of the machine learning algorithms to classify the Web attack in the training phase for the dataset built.

Model	Accuracy	Recall	Precision	F-Measure
NB	0.77	0.82	0.80	0.79
SVM	0.81	0.82	0.83	0.81
RF	0.84	0.86	0.84	0.84
MLP	0.88	0.89	0.88	0.88
Tree-CNN (ReLU)	0.91	0.94	0.95	0.92
Tree-CNN (Softmax)	0.94	0.94	0.94	0.94
Tree-CNN (SRS)	0.99	0.98	0.96	0.98

TABLE 6. Results of the machine learning algorithms to classify the Brute Force attack in the training phase for the dataset built.

Model	Accuracy	Recall	Precision	F-Measure
NB	0.80	0.82	0.80	0.80
SVM	0.83	0.88	0.84	0.84
RF	0.85	0.89	0.85	0.86
MLP	0.89	0.92	0.87	0.87
Tree-CNN (ReLU)	0.94	0.96	0.94	0.94
Tree-CNN (Softmax)	0.95	0.94	0.94	0.91
Tree-CNN (SRS)	0.98	0.99	0.99	0.98

TABLE 7. Results of accuracy and F-measure of the attacks classified through the Tree-CNN (SRS) in the testing phase for the dataset built.

Attack	Acccuracy	Recall	Precision	F-Measure
DDoS	0.98	0.98	0.96	0.98
Infiltration	0.97	0.97	0.98	0.97
Web attack	0.98	0.97	0.95	0.97
Brute force	0.97	0.98	0.98	0.97

in Table 4 for the infiltration attack, in Table 5 for the web attack, and in Table 6 for the brute force attack.

The Tree-CNN with SRS obtained the highest performance compared with the other models in the training phase, and thus, this configuration is used in the following steps. Table 7 shows the values of accuracy, recall, precision, and F-measure of the proposed algorithm for the attacks studied in the testing phase using our dataset built for the study.

TABLE 8. Results of accuracy and F-measure of the attacks classified through the Tree-CNN (SRS) in the testing phase in the medium-sized company.

Model	Accuracy	Recall	Precision	F-measure
DDoS	0.97	0.96	0.97	0.96
Infiltration	0.98	0.97	0.97	0.97
Brute Force	0.98	0.97	0.98	0.97
Web	0.99	0.98	0.98	0.98

B. EVALUATION OF THE TREE-CNN ALGORITHM AND THE SRS ACTIVATION FUNCTION

The model proposed in this work was evaluated according to real-time network traffic in a medium-sized company, the scenario of which was shown in Fig. 3.

Table 8 shows the accuracy results obtained by the proposed Tree-CNN (SRS) model in the testing phase in the medium-sized company.

C. EVALUATION OF THE PROPOSED SYSTEM WITH THE DBN MODEL

The performance of the proposed system was compared with the solution presented in [23] based on Deep Belief Network (DBN) in terms of their capability of detecting intrusions. To make the comparison with the proposed system, the DBN system [23] was tested in the dataset used.

The results of accuracy, recall, precision, and F-measure are shown in Fig. 5. Scen 1 refers to the dataset built of the university campus and Scen 2 refers to the medium-sized company scenario.

According to the results shown in Fig. 5, the model proposed in this work, the Tree-CNN with the SRS activation function, achieved an accuracy value of 0.98 in the detection of Infiltration attacks in the two evaluation scenarios, while the DBN system [23] achieved a value of 0.93 when using our dataset and 0.92 in the medium-sized company scenario.

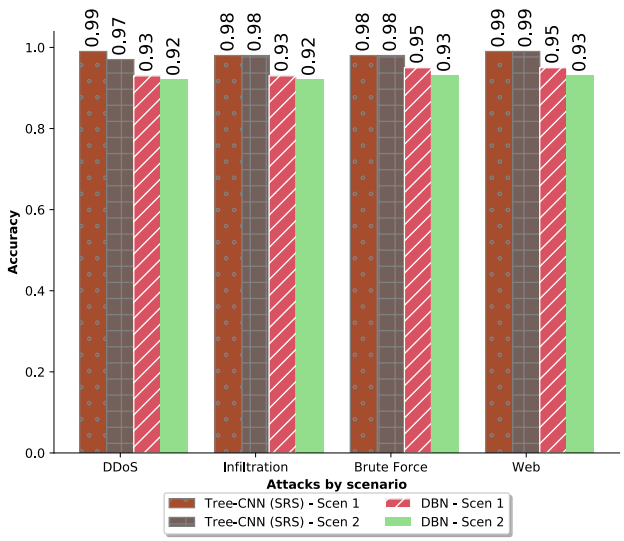


FIGURE 5. Comparison of the accuracy of the proposed detection system with the DBN model in the different attacks evaluated.

TABLE 9. Results of the accuracy of the attacks classified by the DBN system [23] in the testing phase, using the CICIDS2017 [21] dataset.

	DDoS	Infiltration	Brute Force	Web
DBN [23]	0.96	0.96	0.97	0.98
Tree-CNN (SRS)	0.99	0.98	0.98	0.99

TABLE 10. Results of the execution time for the classification of attacks by the DBN system [23] and the proposed algorithm in the testing phase, using the CICIDS2017 [21] dataset.

	Execution Time (sec.)
DBN [23]	1903
Tree-CNN (SRS)	1201

In addition to the accuracy values, our model presented better results in terms of the F-Measure. The best result achieved by the DBN system was 0.95 for detecting Brute Force and Web attacks, whereas our model achieved an F-Measure of 0.98 in the other attacks.

In addition, the proposed attack detection system was also compared with the CICIDS2017 dataset [21] for performance analysis. The results are shown in Table 9.

There is a slight difference in the results; however, the proposed system achieved a faster detection, decreasing the detection speed of the attacks by 36% on average, as can be seen in Table 10.

Table 10 also shows the computational complexity of the proposed method in relation to the execution time. Additionally, in relation to the use of the Random Access Memory, the proposed algorithm had a decrease of 10% in the execution time compared to the DBN method.

VI. CONCLUSION

The activation function plays a critical role in deep neural networks; therefore, the use of a more effective activation function was investigated. In this work, SRS presented good results for accuracy, sensitivity, precision, and F-measure. Based on the results obtained in the tests in different environments, the model proposed in this work was better for the DBN system, and the detection speed of the attacks decreased by 36% on average for testing the model compared with the DBN system. This result is very significant, especially in a critical environment, where the early detection of attacks is essential to prevent damage. The time complexity when using the SRS activation function was low compared with the other algorithms because the SRS has a better generalization performance as well as a faster learning speed for generating the model through batch normalization, thereby accelerating the deep network training. Furthermore, the SRS presents the fastest convergence rate, thus decreasing the time complexity in the Tree-CNN. This work validated the use of the SRS activation function in the Tree-CNN algorithm. The experiments showed significantly higher learning rates and proved the properties of the SRS, that is, smoothness, nonmonotonicity, and delimitation. Moreover, the bounded property of the SRS activation function differs from the other activation functions.

The assertiveness rate of Tree-CNN and SRS presented higher values concerning other metrics, validating the proposal to use this algorithm solution.

In a future work, the target is to test other activation functions and the combination of ReLU and Softmax functions.

VII. ACKNOWLEDGMENT

The authors would like to thank Hanna Niemelä for helping to proofread this paper.

REFERENCES

[1] R. T. Geraldi, S. Reinehr, and A. Malucelli, "Software product line applied to the Internet of Things: A systematic literature review," *Inf. Softw. Technol.*, vol. 124, Aug. 2020, Art. no. 106293.
[2] M. J. Baucas and P. Spachos, "Using cloud and fog computing for large scale IoT-based urban sound classification," *Simul. Model. Pract. Theory*, vol. 101, May 2020, Art. no. 102013.
[3] D. Rodriguez, J. Abrahao, D. Begazo, R. Rosa, and G. Bressan, "Quality metric to assess video streaming service over TCP considering temporal location of pauses," *IEEE Trans. Consum. Electron.*, vol. 58, no. 3, pp. 985–992, Aug. 2012.
[4] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018.
[5] D. Zegarra Rodriguez, R. Lopes Rosa, E. Costa Alfaia, J. Issy Abrahao, and G. Bressan, "Video quality metric for streaming service using DASH standard," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 628–639, Sep. 2016.
[6] P. Nandhini, M. Senthil, and S. Darsniya, "A network intrusion detection system for IoT using machine learning and deep learning approaches," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 3, pp. 1017–1023, Mar. 2020. [Online]. Available: http://sersc.org/journals/index.php/IJAST/article/view/5933
[7] D. Bolzoni, S. Etalle, P. Hartel, and E. Zambon, "POSEIDON: A 2-tier anomaly-based network intrusion detection system," in *Proc. 4th IEEE Int. Workshop Inf. Assurance (IWIA)*, Apr. 2006, p. 10.
[8] A. K. Saxena, S. Sinha, and P. Shukla, "General study of intrusion detection system and survey of agent based intrusion detection system," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 421–471.
[9] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.
[10] J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, 2000.
[11] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Tech. Rep. 99, 2000, pp. 1–15.
[12] A. Warzynski and G. Kolaczek, "Intrusion detection systems vulnerability on adversarial examples," in *Proc. Innov. Intell. Syst. Appl. (INISTA)*, Jul. 2018, pp. 1–4.
[13] A. Sagala, "Automatic SNORT IDS rule generation based on honeypot log," in *Proc. 7th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2015, pp. 576–580.
[14] I. M. M. Matin and B. Rahardjo, "Malware detection using honeypot and machine learning," in *Proc. 7th Int. Conf. Cyber IT Service Manage. (CITSM)*, Nov. 2019, pp. 1–4.
[15] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
[16] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, Dec. 2019.
[17] W.-L. Chu, C.-J. Lin, and K.-N. Chang, "Detection and classification of advanced persistent threats and attacks using the support vector machine," *Appl. Sci.*, vol. 9, no. 21, p. 4579, Oct. 2019.
[18] N. Bindra and M. Sood, "Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset," *Autom. Control Comput. Sci.*, vol. 53, no. 5, pp. 419–428, Sep. 2019.
[19] R. G. Guimaraes, R. L. Rosa, D. De Gaetano, D. Z. Rodriguez, and G. Bressan, "Age groups classification in social network using deep learning," *IEEE Access*, vol. 5, pp. 10805–10816, 2017.
[20] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019.

[21] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[22] D. Aksu and M. Ali Aydin, "Detecting port scan attempts with comparative analysis of deep learning and support vector machine algorithms," in *Proc. Int. Congr. Big Data, Deep Learn. Fighting Cyber Terrorism (IBIGDELFT)*, Dec. 2018, pp. 77–80.

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

[24] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," 2017, *arXiv:1710.00811*. [Online]. Available: http://arxiv.org/abs/1710.00811

[25] R. Carvalho Barbosa, M. Shoaib Ayub, R. Lopes Rosa, D. Z. Rodríguez, and L. Wuttisittikulkij, "Lightweight PVIDNet: A priority vehicles detection network model based on deep learning for intelligent traffic lights," *Sensors*, vol. 20, no. 21, p. 6218, Oct. 2020.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.* Madison, WI, USA: Omnipress, 2010, pp. 807–814.

[28] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 4939–4948.

[29] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1319–1327.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[31] Y. Zhou, D. Li, S. Huo, and S.-Y. Kung, "Soft-root-sign activation function," 2020, *arXiv:2003.00547*. [Online]. Available: http://arxiv.org/abs/2003.00547

[32] D. Roy, P. Panda, and K. Roy, "Tree-CNN: A hierarchical deep convolutional neural network for incremental learning," *Neural Netw.*, vol. 121, pp. 148–160, Jan. 2020.

[33] M. K. Sharma, D. Sheet, and P. K. Biswas, "Abnormality detecting deep belief network," in *Proc. Int. Conf. Adv. Inf. Commun. Technol. Comput. (AICTC)*, 2016, pp. 1–6.

[34] Z. Liang, G. Zhang, J. X. Huang, and Q. V. Hu, "Deep learning for healthcare decision making with EMRs," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2014, pp. 556–559.

[35] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," 2017, *arXiv:1701.02145*. [Online]. Available: http://arxiv.org/abs/1701.02145

[36] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowl.-Based Syst.*, vol. 78, pp. 13–21, Apr. 2015.

[37] S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, and F. Herrera, "On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 193–202, Jan. 2015.

[38] O. Can and O. K. Sahingoz, "A survey of intrusion detection systems in wireless sensor networks," in *Proc. 6th Int. Conf. Modeling, Simulation, Appl. Optim. (ICMSAO)*, May 2015, pp. 1–6.

[39] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[40] A. Razzaq, K. Latif, H. F. Ahmad, A. Hur, Z. Anwar, and P. C. Bloodsworth, "Semantic security against Web application attacks," *Inf. Sci.*, vol. 254, pp. 19–38, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025513005677

[41] G. M. L. II, "Security analytics: Using deep learning to detect cyber attacks," M.S. thesis, Univ. North Florida, Jacksonville, FL, USA, 2017.

[42] G. Creech, "Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks," Ph.D. dissertation, Univ. New South Wales, Canberra, NSW, Australia, 2014.

[43] M. K. Mishra and R. Dash, "A comparative study of chebyshev functional link artificial neural network, multi-layer perceptron and decision tree for credit card fraud detection," in *Proc. Int. Conf. Inf. Technol.*, Dec. 2014, pp. 228–233.

[44] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset," in *Proc. Int. Conf. Inf. Sci. Secur. (ICISS)*, Dec. 2016, pp. 1–6.

[45] N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems," in *Proc. 4th Int. Workshop Building Anal. Datasets Gathering Exper. Returns for Secur. (BADGERS)*, Nov. 2015, pp. 25–31.

[46] M. I. O. T. LINCOLN LABORATORY. (1999). *1999 DARPA Intrusion Detection Evaluation Dataset*. [Online]. Available: http://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset

[47] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation," in *Proc. DARPA Inf. Survivability Conf. Expo. (DISCEX)*, vol. 2, 2000, pp. 12–26.

[48] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns," *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 807–819, Apr. 2014.

[49] A. Özgür and H. Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ*, vol. 4, no. Nov. 2016.

[50] A. Khanna, D. Gupta, S. Bhattacharyya, V. Snasel, J. Platos, and A. Hassanien, *Int. Conf. Innov. Comput. Commun. (ICICC)* (Advances in Intelligent Systems and Computing), vol. 2. Singapore: Springer, 2019.

[51] E. W. Tavares Ferreira and A. Akira Shinoda, "The development and evaluation of a dataset for testing of IDS for wireless networks," *IEEE Latin Amer. Trans.*, vol. 14, no. 1, pp. 404–410, Jan. 2016.

[52] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs," *Comput. Secur.*, vol. 73, pp. 411–424, Mar. 2018, doi: 10.1016/j.cose.2017.11.004.

[53] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Comput. Sci.*, vol. 167, pp. 636–645, 2020.

[54] M. Kirk, *Thoughtful Machine Learning: A Test-Driven Approach*. Newton, MA, USA: O'Reilly Media, 2014.

[55] S. Jiang, T. Xu, J. Guo, and J. Zhang, "Tree-CNN: From generalization to specialization," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, pp. 1–12, Dec. 2018.

[56] X. Zhu and M. Bain, "B-CNN: Branch convolutional neural network for hierarchical classification," 2017, *arXiv:1709.09890*. [Online]. Available: http://arxiv.org/abs/1709.09890

[57] S. S. Sarwar, P. Panda, and K. Roy, "Gabor filter assisted energy efficient fast learning convolutional neural networks," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2017, pp. 1–6, doi: 10.1109/islped.2017.8009202.

[58] S. S. Sarwar, A. Ankit, and K. Roy, "Incremental learning in deep convolutional neural networks using partial network sharing," *IEEE Access*, vol. 8, pp. 4615–4628, 2020.

[59] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu, "HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition," in *Proc. 15th IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2740–2748.

[60] T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang, "Error-driven incremental learning in deep convolutional neural network for large-scale image classification," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 177–186.

[61] D. B. Mehta, P. A. Barot, and S. G. Langhnoja, "Effect of different activation functions on EEG signal classification based on neural networks," in *Proc. 4th Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2020, pp. 132–135.

[62] S. Jang and Y. Son, "Empirical evaluation of activation functions and kernel initializers on deep reinforcement learning," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2019, pp. 1140–1142.

[63] R. Zaheer and H. Shaziya, "GPU-based empirical evaluation of activation functions in convolutional neural networks," in *Proc. 2nd Int. Conf. Inventive Syst. Control (ICISC)*, Jan. 2018, pp. 769–773.

[64] N. Shukla, *Machine learning With TensorFlow*. Shelter Island, NY, USA: Manning Publications, 2018.

[65] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[66] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1222–1228.

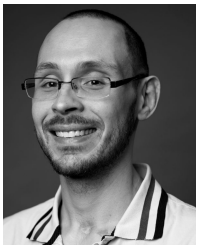
ROBSON V. MENDONÇA received the B.Sc. degree in information systems, in 2017. He is currently pursuing the master's degree in computer science with the Federal University of Lavras, Brazil. He is also an IT Technologist at IFSULDEMINAS. He has professional experience in systems development, networks infrastructure, and servers. His research interests include information security, artificial intelligence, the Internet of Things, and automation.

ARTHUR A. M. TEODORO was born in Formiga, Brazil, in 1997. He received the bachelor's degree in computer science from the Federal Institute of Minas Gerais, in 2018. He is currently pursuing the master's degree with the Department of Computer Science, Federal University of Lavras. His research interests include reconfigurable devices, artificial intelligence, optimization, and systems security.

RENATA L. ROSA received the M.S. degree from the University of São Paulo, in 2009, and the Ph.D. degree from the Polytechnic School of the University of São Paulo (EPUSP), in 2015. She is currently an Adjunct Professor with the Department of Computer Science, Federal University of Lavras, Brazil. She has a solid knowledge in computer science based on more than ten years of professional experience. Her current research interests include computer networks, telecommunication systems, machine learning, quality of experience of multimedia service, social networks, and recommendation systems.

MUHAMMAD SAADI received the B.Sc. degree from the National University of Computer and Emerging Sciences, Pakistan, in 2007, the M.Sc. degree from the National University of Malaysia, Malaysia, in 2009, and the Ph.D. degree in electrical engineering from Chulalongkorn University, Bangkok, Thailand. He was with the National Electronics and Computer Technology Center, Aimagin Ltd., Thailand. He is currently an Assistant Professor with the Department of Electrical Engineering, University of Central Punjab, Pakistan. His current research interests include visible light communication, indoor localization, and next generation networks. During his Ph.D., he has received the Best Paper Award twice in ITC-CSCC 2014 and ITC-CSCC 2015. He is also a regular reviewer of leading journals in the area of wireless communication.

DICK CARRILLO MELGAREJO (Member, IEEE) received the B.Eng. degree (Hons.) in electronics and electrical engineering from the National University of San Marcos, Lima, Perú, in 2004, and the M.Sc. degree in electrical engineering from Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil, in 2008. He is currently pursuing the Ph.D. degree in electrical engineering with the Lappeenranta—Lahti University of Technology. From 2008 to 2010, he contributed to WiMAX (IEEE 802.16m) standardization. From 2010 to 2018, he worked with the design and implementation of cognitive radio networks and the projects based on 3GPP technologies. Since 2018, he has been a Researcher with the Lappeenranta—Lahti University of Technology. His research interests include mobile technologies beyond 5G, energy harvesting, intelligent meta-surfaces, and cell-free mMIMO.



PEDRO H. J. NARDELLI (Senior Member, IEEE) received the B.S. and M.Sc. degrees in electrical engineering from the State University of Campinas, Brazil, in 2006 and 2008, respectively, and the Ph.D. degree from University of Oulu, Finland, and State University of Campinas, following a dual degree agreement, in 2013. He is currently an Associate Professor (tenure track) of the IoT in energy systems with LUT University, Finland, and holds a position of an academy of Finland research fellow with a project called Building the Energy Internet as a large-scale IoT-based cyber-physical system that manages the energy inventory of distribution grids as discretized packets via machine-type communications (EnergyNet). He leads the Cyber-Physical Systems Group, LUT, and the Project Coordinator of the CHIST-ERA European consortium Framework for the Identification of Rare Events via Machine Learning and IoT Networks (FIREMAN), and the project Swarming Technology for Reliable and Energy-aware Aerial Missions (STREAM) supported by Jane and Aatos Erkko Foundation. He is also a Docent with the University of Oulu in the topic of Communications Strategies and Information Processing in Energy Systems. His research interest includes wireless communications particularly applied in industrial automation and energy systems.

DEMÓSTENES Z. RODRÍGUEZ (Senior Member, IEEE) received the B.S. degree in electronic engineering from the Pontifical Catholic University of Peru, the M.S. and Ph.D. degrees from the University of São Paulo, in 2009 and 2013, respectively. He is currently an Adjunct Professor with the Department of Computer Science, Federal University of Lavras, Brazil. He has a solid knowledge in telecommunication systems and computer science based on 15 years of professional experience in major companies. His research interests include QoS and QoE in multimedia services and architect solutions in telecommunication systems. He is a member of the Brazilian Telecommunications Society.