A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks

Haider, Shahzeb; Akhunzada, Adnan; Mustafa, Iqra; Patel, Tanil Bharat; Fernandez, Amanda; Choo, Kim Kwang Raymond; Iqbal, Javed

Published in:
IEEE Access

Link to article, DOI:
10.1109/ACCESS.2020.2976908

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 31, 2020, accepted February 16, 2020, date of publication February 27, 2020, date of current version March 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976908

A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks

SHAHZEB HAIDER[1], ADNAN AKHUNZADA[1,2], IQRA MUSTAFA[3], TANIL BHARAT PATEL[3], AMANDA FERNANDEZ[4], KIM-KWANG RAYMOND CHOO[5], (Senior Member, IEEE), AND JAVED IQBAL[1]

[1]Applied Security Engineering Research Group, Department of Computer Science, COMSATS University Islamabad, Islamabad 45550, Pakistan
[2]Technical University of Denmark, DTU Compute, 2800 Copenhagen, Denmark
[3]Department of Computer Science, Cork Institute of Technology, T12 P928 Cork, Ireland
[4]Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249, USA
[5]Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249, USA

Corresponding author: Adnan Akhunzada (adnak@dtu.dk)

ABSTRACT As novel technologies continue to reshape the digital era, cyberattacks are also increasingly becoming more commonplace and sophisticated. Distributed denial of service (DDoS) attacks are, perhaps, the most prevalent and exponentially-growing attack, targeting the varied and emerging computational network infrastructures across the globe. This necessitates the design of an efficient and early detection of large-scale sophisticated DDoS attacks. Software defined networks (SDN) point to a promising solution, as a network paradigm which decouples the centralized control intelligence from the forwarding logic. In this work, a deep convolutional neural network (CNN) ensemble framework for efficient DDoS attack detection in SDNs is proposed. The proposed framework is evaluated on a current state-of-the-art Flow-based dataset under established benchmarks. Improved accuracy is demonstrated against existing related detection approaches.

INDEX TERMS Software defined network (SDN), anomaly detection, distributed denial of service (DDoS), deep learning, deep convolutional neural network (CNN).

I. INTRODUCTION

The importance of emerging information and communication technology (ICT) solutions and their role in our social and economic lives is undeniable in our current society. The advancement in technology directly affects the economic growth of a country. On the contrary, the proliferation of technology lends information systems vulnerable to varied cyber threats and attacks. Moreover, novel ICT solutions create new security concerns. The importance of cyber security is also evidenced by the inclusion of cyber as the fifth domain of (warfare) operations and the elevation of United States Cyber Command to a unified combatant command. To keep the legitimacy of work and to paralyze adversarial cyber warfare, it is of extreme importance to create a holistic plan to secure the emerging digital landscape. An adaptive, scalable and cost-effective solution for varied cyber security threats and vulnerabilities is of paramount concern. In this regard, cyber security experts and researchers from all over the world endeavor to create a safe and secure cyberspace in the era of exponentially growing digitization.

Distributed denial of service (DDoS) attacks are currently the most prevalent and sophisticated threat for organizations, and are increasingly difficult to prevent [1]–[3]. In 2018, for example, GitHub was hit with one of the largest DDoS attacks ever [4]. This impactful attack comes in one of the most highlighted cyberattacks of the current cyber age, shaking the ground basis of one of the pillars (availability) of the CIA security triad. Attackers use thousands of dump terminals, machines, and botnets to concurrently launch DDoS attacks that subsequently exhaust the target system main resources, making the entire services unavailable. There are a potentially extreme number of legitimate and powerful tools available, which can be abused to launch DDoS attacks on large and small scales accordingly. In another recent DDoS attack [4], attackers misused the legitimate Memcached tool,

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

TABLE 1. Representation of our proposed ensembles and hybrid ensembles.

Deep Learning Architectures	*Description*
RNN (RNN+ RNN)	Ensemble Recurrent Neural Networks
LSTM (LSTM+LSTM)	Ensemble Long-Short Term Memory networks
RL (RNN+LSTM)	Ensemble-Hybrid RL
CNN (CNN+CNN)	Ensemble Convolutional Neural Networks

whose primary purpose is to reduce strain over the underlying network resources. The attacker abused Memcached objects and spoofed IP addresses, allowing Memcached responses to be directed to the target addresses with 126.9 million packets/second to largely consume target resources. Moreover, the use of spoofed IPs makes the trace-back next to impossible [5] in DDoS attacks. Therefore, the efficient and early detection, mitigation, and prevention of DDoS attacks remain a challenging task. However, strong novel measures can be taken towards timely detection, to allow subsequent countermeasures to prevent or mitigate sophisticated DDoS attacks [6]. There have been interest in utilizing artificial learning approaches (e.g., machine learning and deep learning techniques) to prevent or mitigate sophisticated DDoS attacks [7]–[9], although designing efficient and effective DDoS mitigation strategies remain an ongoing challenge. In this work, a deep CNN framework is proposed for efficient and early detection of DDoS attacks in SDNs, and a deep CNN ensemble mechanism is designed to detect varied Flow-based DDoS attacks. In comparison to this solution, related state-of-the-art deep learning (DL) based ensembles and hybrid approaches (i.e., RNN, LSTM, RL) are applied to the same tasks for verification. Table 1 provides a snapshot of the given ensembles and hybrid approaches leveraged in this work.

Each deep learning architecture described in this table is evaluated against benchmark dataset CICIDS2017 [10], formally known as the ISCX Dataset. CICIDS2017 is purely a Flow-based state-of-the-art SDN dataset (additional details about the dataset are presented in Section III-A). A majority of existing datasets used for network intrusion detections systems (NIDS) are mainly comprised of IP traffic, which does not carry purely Flow-based features and attributes, a requirement for SDNs. Finally, the comparison of the proposed technique with current state-of-the-art ensembles and hybrid approaches is evaluated using standard metrics/parameters, namely accuracy, precision, recall and f1-score. Our proposed approach outperforms with 99.45% detection accuracy and minimal computational complexity.

This work proposes a novel approach to utilize DL-based ensemble and hybrid approaches to detect large-scale DDoS attacks within a Flow-based benchmark dataset, which purely represents software defined networks [11], [12]. The primary contributions of this work are as follows.

1) A novel deep learning framework for the detection of DDoS attacks in SDNs. This proposed framework leverages novel ensemble CNN models for improved detection on Flow-based data.
2) Evaluation of the proposed framework with a state-of-the-art Flow-based dataset CICIDS2017 [10].
3) Verification of the proposed mechanism against current state of the art deep ensembles and hybrid approaches for DDoS attack detection in SDNs. Demonstration of the scalability and cost-effectiveness of the proposed SDN controller (i.e., control plane) based ensemble framework.
4) Demonstration of performance of the proposed ensemble compared to the benchmark algorithms, with high detection accuracy (i.e., 99.45%) and minimal computational complexity.

The remainder of this paper is structured as follows. Section II briefly reviews related work. In Section III, we describe our proposed approach, prior to describing the evaluation setup and findings in Section IV. The last section concludes the paper.

II. RELATED WORK

Commonly, intrusion detection systems (IDS) are placed to monitor threats and malevolent activities. Their basic functionality includes collection of logs and events from the network resources and analysis for potential threats [13], [14]. The existing IDS systems are mainly comprised of two main approaches: signature-based and anomaly-based. These two approaches are widely used with corresponding subcategories such as host, network, and application [6], [15]. The essential components of modern intrustion detection systems are well-established [16], and the emphasis of recent research is therefore on improving their accuracy against evolving attacks.

A significant focus of recent research is on anomaly-based IDS, as these approaches outperform signature and rule-based detection approaches for unknown intrusions [17]–[19]. Therefore, in [20], an anomaly-based deep learning technique, DNN, is used for Flow-based anomaly detection in an SDN environment. NSL-KDD is set as a benchmark dataset for test and experiments. This deep learning algorithm achieves 75.75% accuracy in anomaly detection under some given conditions. Similarly, [21] constructs an IDS with DL-approach LSTM to an RNN network to train the network. This model achieves 96.93% accuracy on the dataset KDD Cup 1999, using an Intel i7 with GPU (GTX Titan X) acceleration in experimental setup. Similarly, [22] proposes DL-based RNN approach for binary and multiclass anomaly detection, achieving 83.28% accuracy on NSL-KDD dataset in 5,516 seconds. DL algorithm CNN for intrusion detection is used in [23], obtaining 97.7% accuracy on KDD Cup 1999 datase with GPU acceleration GTX 1080, 32GB RAM with OS Ubuntu 14.04 in experimental setup. 98.1% recall is achieved in [24] using Bidirectional LSTM in Keras, python with GPU (NVidia GeForce GTX 860M) acceleration in experimental setup. However, training time BLSTM requires is 4,800 minutes.

A lightweight DDoS detection methodology utilizing SOM in SDNs environment is proffered in [25]. Their system works by the extraction of features of interest at certain intervals in order to convert the system in lightweight mode. Precision of the presented system is 99.11% with the false positive rate 0.46 in data plane comprising high training time ranging 553-716 hours. A system is presented comprising a combination of OpenFlow and sFlow with the collection of OpenFlow statistics for anomaly detection and reducing the overhead in b/w OF controllers and switches [26]. However, false positive rate was 40-50%, pretty much higher in detection of attack with the CPU usage 39-58%. An SDN based scheme for DDoS detection and blocking called DBA is proffered in [27] which works over an SDN controller utilizing the OpenFlow interface. Mininet emulator is used for experimental purpose in HTTP flood attack detection. An entropy-based approach is used to measure the randomness of new incoming packets by defining a threshold, implemented over an SDN controller in [28]. Drop in threshold values gives indication of intrusion into the network. However, a limitation here is in the constant threshold value, which can vary accordingly to the conditions in the network. Attack. This work attains 96% attack detection accuracy in control place. Similarly, an entropy-based DDoS detection solution, implanted in OF switch to make statistics and doing analysis on the receiving network traffic, indirectly reducing the overhead of communication b/w controller and switches [29]. Detected false positive rate is 25% in the proffered solution in attack detection. Work on suspicious flow detection is performed in [30] using DL-based hybrid approach comprising RBM with SVM in an SDN environment. The author uses RBM for dimensional reduction of the collected flow statistics then utilize SVM for classification in b/w regular and anomaly traffic. However, KDD'99 dataset is used for experimental purpose which is not a flow-based dataset. The author uses mininet emulator in [31] for flow generation and performs DDoS detection through the proposed DPTCM-KNN algorithm. Two parameters, Strangeness and independence are taken as the judgemental methods by obtaining two p-values. However, the obtained FPR value is ranging from 2-6% which is consider a bit high when compare to the contemporary research in DDoS detection.

DDoS detection and defence architecture is proposed in [32] through the DL-based model. Detection model comprise multiple deep learning layers such as CNN, LSTM and bidirectional RNN layer. Dataset used is ISCX2012 for experimental purposes. The author in [33] uses mininet emulator to create SDN environment while utilize hping3 for DDoS attack generation. Author performs flow state collection, feature extraction and then classification using SVM algorithm. However, obtained classification accuracy is considerably low. Similarly, a multi-vector DDoS detection system is proposed in [34] using DL-based SAE. Author uses TCFI module for traffic collection and feature extraction from traffic flow however, it is difficult to when it comes to detection of low rate attacks from the proffered technique as it alike the regular/genuine traffic from victim end. Reference [35] presents a hypothetical main smart controller placement over other controllers in SDNs for continuous service. However, extra controllers placement will surely slow the SDN network as it is one of the main challenges that SDN networks are facing especially at enterprise level, will rise the scalability issues in SDN environment as well. Reference [36] Presents a lightweight DDoS detection approach using DL-based GRU-RNN with minimum number of features in SDN environment. Nevertheless, detection accuracy obtained by the model is considerably low with high FPR value while the dataset used is NSL-KDD. Similarly, [37] presents DPMM clustering algorithm for DDoS attack detection in SDN environment. Real traces of data are used for experimental purpose. However, author is unable to obtain satisfactory results when it comes to attack detection as it comprise very low detection accuracy with high error rate, also the model is not suitable for large scale attacks.

To the best of our knowledge and including above comprehensive literature work, the author yields that ensemble deep learning approaches for anomaly detection in SDNs are largely ignored and not available yet. Current era needs innovative solutions to fight with the contemporary appalling cyber threats. In this regard, we orchestrate multiple DL-based ensemble and hybrid approaches for DDoS detection (i.e., ensemble LSTM, RNN, CNN and Hybrid RL). The evidently bestowed results from the author unlocks the untapped potential of DL-architectures in detection of any kind of intrusion in the network. Moreover, proffered approaches does not burden the control plane and controller continues to perform its activities as usual. Integration of a low cost, fast performance IDS on the controller can drastically improve its abilities and eradicates the complex modelling of anomaly structural design. The assessment of models has done with proper and almost from all possible standard evaluation parameters, displayed in section 5, experimental results and analysis. Ensemble RNN, LSTM and hybrid RL achieves accuracy more than 98% in minimum time with low computational complexity. However, our proposed ensemble CNN obtains 99.45% highest detection accuracy with excellent training and classification time comprising lower consumption of resources (CPU usage %). Purpose behind our DL-based approaches is not to outperform the existing mainstream classifiers but to reveal the unexploited potential of DL architectures from every phase and keeps the hype of AI in future anomaly detection applications.

III. PROPOSED DEEP CNN ENSEMBLE SDN ARCHITECTURE

In this section, the architecture of our proposed CNN ensemble framework for SDNs is discussed. Software Define Networks (SDNs) is a prevalent networking paradigm which decouples the control logic from the forwarding logic. The centralized control intelligence and the programmability aspects of SDNs provide a platform to implement various functions at the control plane. Various networking functions

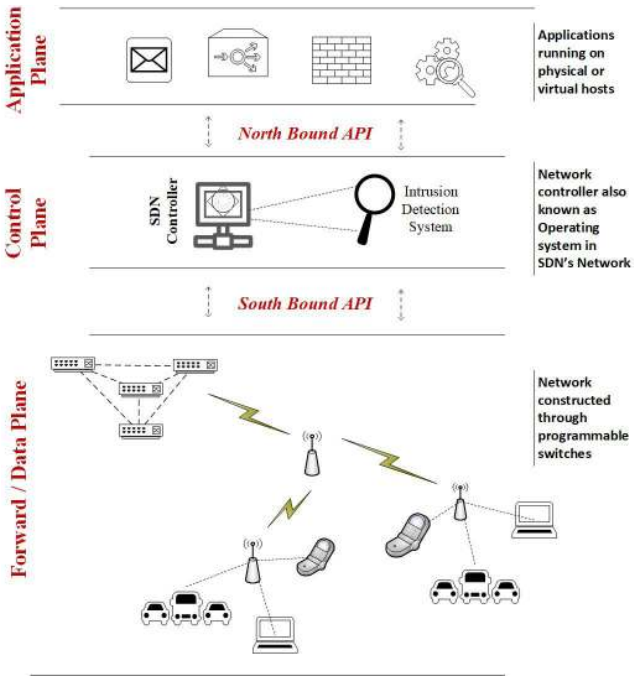


FIGURE 1. IDS placement on SDN control plane.

and modules can be implemented at any commercial SDN controller such as Floodlight, ONOS etc [12], and [11], [38], [39]. Likewise, our proposed framework is an integral part of the control plane which is highly scalable and cost-effective. A soft copy of our proposed deep CNN ensemble framework can be placed on any commercial SDN controller and the network abstraction, and centralized management at the control plane makes it cost-effective. The SDN architecture is comprised of three planes: application plane, control plane, and data plane. These planes and their corresponding southbound and northbound APIs are shown in Figure 1. The application plane is responsible for carrying various applications that simply instruct the controller to perform changes in accordance with the requirements and its northbound API. The control plane is the decision maker and controls the centralized intelligence through the SDN controller. The soft SDN controller is responsible for all of the underlying network management. The control logic is built in the soft SDN controller that directly communicates with the data plane through its southbound API, having programmable soft SDN switches connecting various underlying SDN agents. The southbound API connects and communicates between the control plane and data plane through SDN-enabled switches. Finally, the data plane is comprised of dump terminals, various SDN agents whose responsibility is forwarding according to the control logic, retained at the control plane. Figure 1 depicts our proposed framework at the control plane.

We will now introduce the underlying components of our proposed model. To clarify its use and application, the dataset leveraged to evaluate our model is first discussed. With an understanding of the data distribution and scaling, the following sections detail the construction of our proposed deep ensembles and hybrid architectures, as well as the standard metrics used for the evaluation.

TABLE 2. Feature selection for DDoS attack detection.

	Features
1	Backward Packet Length (B.Packet Len) Std
2	Avg Packet Size
3	Flow Duration
4	Flow Inter Arrival Time (IAT)

A. DATASET DESCRIPTION

Most of the existing datasets used for network anomaly detection consist of IP-based traffic such as DARPA98 [40] KDD99, NSL-KDD. Since 1998, there are almost eleven IP-based datasets that are refined with time series data for IDS evaluation. However, Flow-based attributes and features are critical for SDN. To this end, only the ISCX-2012 and CICIDS-2017 [10] datasets provide sufficient benchmark. CICIDS-2017 is the most refined version of ISCX-12 and represents the current state of the art Flow-based dataset. The public dataset is fully labeled, comprised of at least 80 features of network traffic which includes both benign and multiple types of attack traffic.

Extraction and calculation of features in [10] is accomplished through CICFlowMeter (i.e., formerly known as ISCXFlowMeter) software for all benign and intrusive flows. However, for best feature selection out of 80 extracted features, the authors used RandomForestRegressor (Pedregosa *et al.*, 2011). The importance of each feature was calculated by multiplication of the average standardized mean value of each feature with the corresponding features' importance value. From the best selected features, 4 features Backward Packet Length (B. packet Len) Std, Flow Duration, Avg Packet Size, and Flow inter arrival time (IAT) Std, were selected for DDoS attack detection listed in [10]. Flow IAT related features such as Min, Mean, Max and Flow Duration are one of the best-selected features for DoS detection.

B. DATASET DISTRIBUTION

A total of 140,000 samples are loaded for classification, including distribution of 60:40 for benign and DDoS traffic, as shown in Figure 2. Normal traffic is labeled as 0 while DDoS attack type traffic as 1. Train_test_split method is used using the scikit learn library to split the dataset into 80% for training and 20% for testing purpose with test_size = 0.2. After splitting the dataset, 112,000 samples are set for training and 28,000 samples are set for testing purpose. As provided in the dataset, the features themselves are incomparable for learning networks, given that they may be in a varied range and may contain continuous or discrete values. The following section outlines the steps needed to normalize and scale these features prior to training and testing.

C. PACKETS FEATURE SCALING

Feature scaling in data preparation and pre-processing plays a vital role in building an efficient deep learning model.

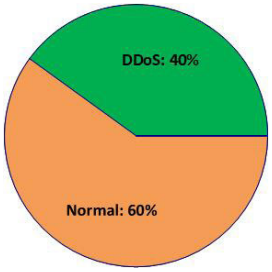


FIGURE 2. Data ratio.

This step will additionally assist with the many computations, highly computationally extensive calculations, and parallel computations, by reducing redundancies in the data. In this work, we perform feature scaling on the data packets with the help of standardization (Z-score normalization) before they are fed into the deep learning models. Features are re-scaled and comprised of Std properties with $\mu = 0$ and $\sigma = 1$, where μ is average/mean in which σ represents the mean.

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

Standardization makes the features centered around 0, and std value of 1. Standardization is not only important when we compare measurements having unlike units, however; it is also required generally for many ML algorithms to reduce different distributions effect [34]. For feature scaling, we used StandardScaler, a scikit learn library which transform our independent variables in to a form that its distribution have a mean value equals to 0 and Std value equals to 1. The main idea behind feature scaling is to normalize or standardize each feature/variable or column that they will have mean = 0 and Std = 1, before we put them to the machine learning models. To calculate mean and standard deviation, formals used are:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(xi) \tag{2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(xi - \mu)^2} \tag{3}$$

After Z score-normalization, input packets dimensional reconstruction is done in which two-dimensional feature matrix $x \in IR[HxW]$ is transformed in to a higher order matrices called tensor, an order 3 tensor as $x \in IR[HxWxD]$. Dimensional reconstruction is done to fulfill the input requirements of deep learning model. The four feature fields are selected as independent variables from input packets m as given in the dataset of ISCX 2017 for DDoS. Boolean fields are given as dependent variables for labeling the benign and malicious traffic.

D. PROPOSED DL MODEL'S ARCHITECTURES

To demonstrate the potential of DL-algorithms, we propose four DL-based architectures (Ensemble RNN, LSTM, CNN and Hybrid RL). These models are constructed using the simple deep learning models (RNN, LSTM and CNN). Two similar DL models are combined to build up a new ensemble model while two complimentary models (RNN+LSTM)

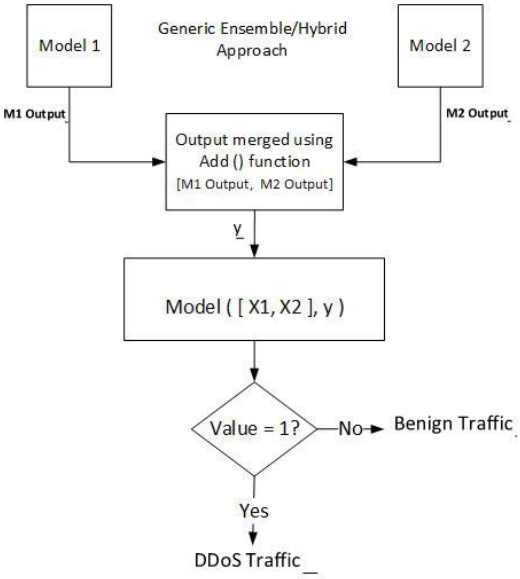


FIGURE 3. A general illustration of the proposed hybrid NN architectures.

for hybrid model construction to demonstrate potential of DL-algorithms. Figure 3 represents the generic representation of all of the proposed DL ensemble and hybrid architectures: In Figure 3, two models M1 and M2 are combined to build a hybrid model. The output of M1 and M2 is joined using the Add () function at the merger layer in Keras which serves as an input to the merger layer. Add () function takes M1 and M2 outcome as a list of input tensors, all of the same shape and returns a single output tensor y. This tensor value y acts as an output in the new model Model3. New Model3 takes 3 elements: X1, X2, and y. Two input values X1 and X2, are inputs from M1 and M2 while the 3rd tensor element y is the merged output of two models M1 and M2. Here new model M3 acting as a hybrid model which performs its operations and predict the binary outcome either 1 or 0. 1 identifies DDoS attack while 0 predicts for normal/benign traffic. Figure 4 shows the graphical presentation of ensemble RNN, LSTM and hybrid RL, and Figure 8 similarly displays for ensemble CNN.

In Figure 4, Model1 can be a RNN or LSTM model. In a similar way, we can replace Model2 by RNN/LSTM depending on our requirements to construct: ensemble RNN, LSTM or ensemble and hybrid RL. Similar procedure we follow for the ensemble CNN in Figure 5.

Information about the single DL-models and their parameters setting used in formation of ensembles and hybrid model is given in table 3.

As shown in Table 3, each model (M1 or M2) in ensemble RNN, LSTM and hybrid RL uses 4 fully connected (FC) layers with 256, 128, 64, 32 number of neurons respectively. Activation function (AF) used is Relu and Sigmoid in the hidden and output layer respectively of each single DL-Model for layer.

A similar method is adopted in Ensemble CNN construction. Each model (M1 and M2) contains

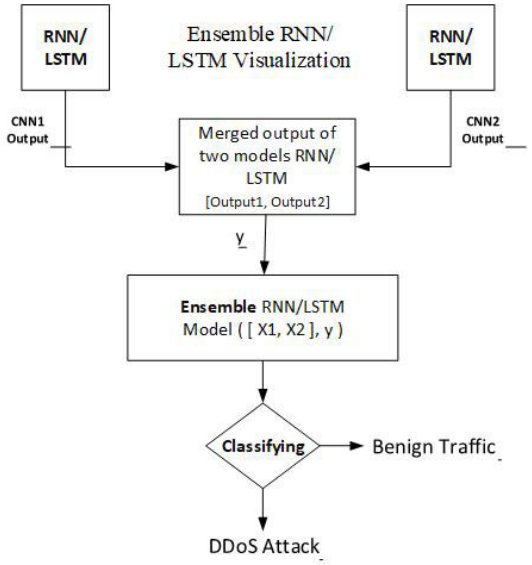


FIGURE 4. Ensemble RNN/LSTM.

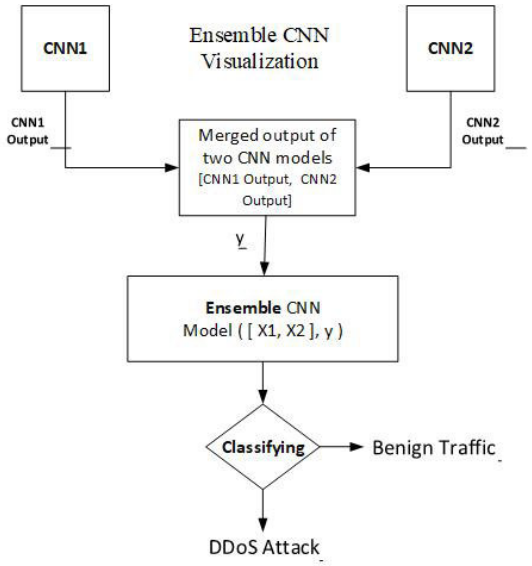


FIGURE 5. Ensemble CNN.

TABLE 3. Parameters selection for each single DL-model.

DL-Parameters	RNN	LSTM	CNN
Layers	4 FC	4 FC	3 Conv2D, 2 MaxPool, 1 Flatten, 3 FC
Neurons	256, 128, 64, 32	256, 128, 64, 32	-
Activation Function	Relu, Sigmoid	Relu, Sigmoid	Relu, Sigmoid
No of epochs set	100	100	100

3 two-dimensional convolutional layers, 2 max pooling layers, 1 layer to flatten, and 2 dense fully-connected layers. Rectified Linear Unit (RELU) serves as activation function in hidden layers while Sigmoid at the output layers or each model in Ensemble CNN, given its demonstrated performance for

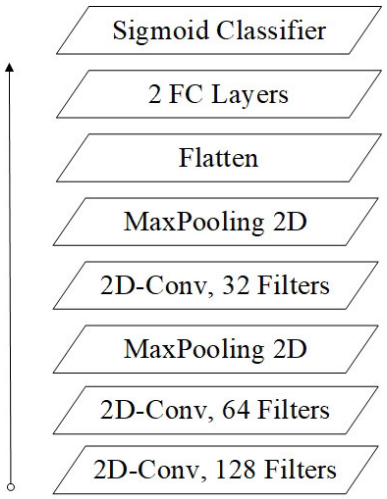


FIGURE 6. CNN layers flow.

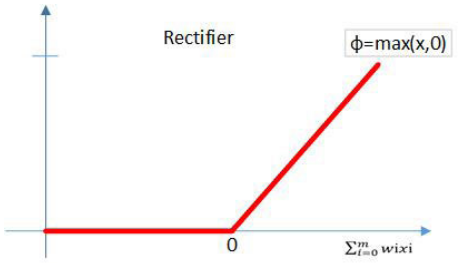


FIGURE 7. Rectifier AF.

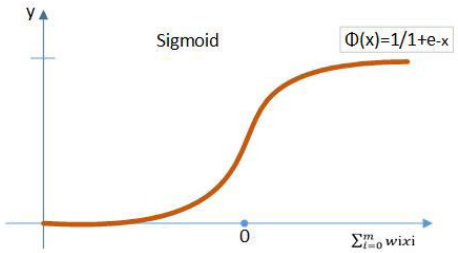


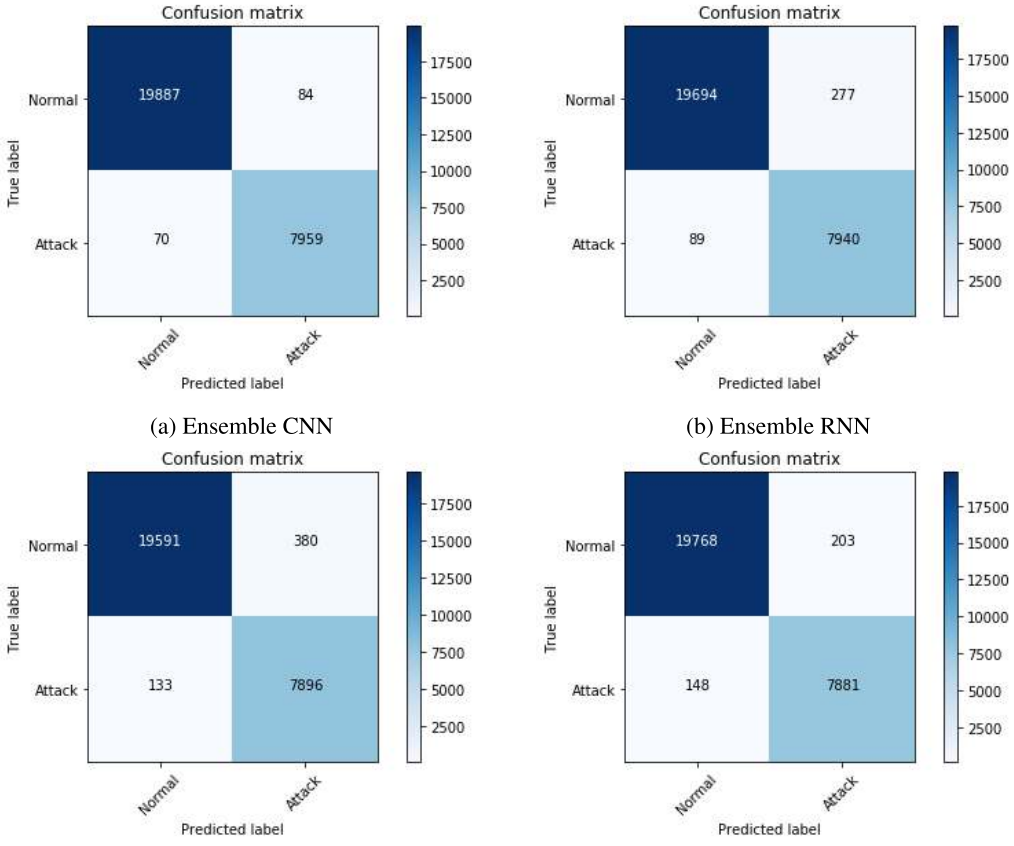
FIGURE 8. Sigmoid AF.

binary results. Graphical illustration of the sequence of layers used in Ensemble CNN has given in Figure 6.

In ensemble CNN, a single CNN model uses 3 two-dimensional convolutional layers with 128, 64, & 64 filters respectively. Its fourth layer is a maximum pooling layer. Its fifth layer is a two-dimensional convolutional layer with 32 filters, followed by another max pooling layer. A flatten layer follows in order to structure the input to the 2 subsequent dense layers, and a final output layer with a sigmoid classifier.

Activation functions play the fundamental role in activating neurons in the neural network. Based on the experiments, we choose activation functions RELU and sigmoid for the hidden layers and output layers in each of the proposed ensemble and hybrid model.

Figures 7 and 8 are the graphical representation of the activation function RELU and sigmoid. In Figure 7, RELU has sharp twist or curve still it acts as an ultimate activation



(a) Ensemble CNN (b) Ensemble RNN

(c) Ensemble LSTM (d) Hybrid RL

FIGURE 9. Confusion matrix of the proposed DL-Models.

TABLE 4. Systems specifications.

System Manufacturers	Lenovo
Processor	Intel Core i7-6700 CPU with 3.4 GHz Processor
Memory	8GB
Operating System	Microsoft Windows 10

function in neural networks. Similarly, Figure 8 is the illustration of activation function Sigmoid, which has smooth curve unlike threshold and RELU.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section briefly explains the experimental setup and discusses the classification results of our proposed framework. Results are analyzed based upon the detection accuracies, precision, recall, and f-measure, as well as the training & testing times and memory consumption of each of the proposed hybrid model.

A. EXPERIMENTAL SETUP

In our proposed approaches, we use Keras Library [41] with Tensorflow backend [42]. Table 4 illustrates the hardware and software specifications used in the experimental described in this section.

Often, training deep learning models requires significant resources, akin to the 100+GB RAM and GPU's acceleration levereaged in [12], [24], [43] and [41]. However, our proposed models demonstrate comparable results on less significant systems.

B. RESULTS AND ANALYSIS

Derivation of results and performance evaluation of the proposed DL architectures is made underlying the dataset [15]. Presented models are assessed via standard metrics: detection accuracy with the precision, recall, f1-measure (Pr, Rc F1), and receiver operator characteristics curve (ROC) graph. Values of all of the previously defined evaluation parameters are demonstrated in the confusion matrix. Finally, training and testing time comprised of system memory consumption (CPU usage %) of each of the proposed DL model is also presented for results and analysis.

$$Accuracy(A) = \frac{Accurately\ classified\ records}{Total\ Samples} * 100 \quad (4)$$

$$Precision(Pr) = \frac{True\ Positive}{True\ Positive\ +\ False\ Positive} * 100 \quad (5)$$

$$Recall(Rc) = \frac{TruePositive}{TruePositive + FalseNegative} * 100 \quad (6)$$

$$F1.measure = \frac{2.Rc.Pr}{Rc\ +\ Pr} * 100 \quad (7)$$

The confusion matrices of each of the proposed DL-models (RNN, LSTM, RL and CNN) in this work are presented in Figure 9. The description of actual and predicted classes

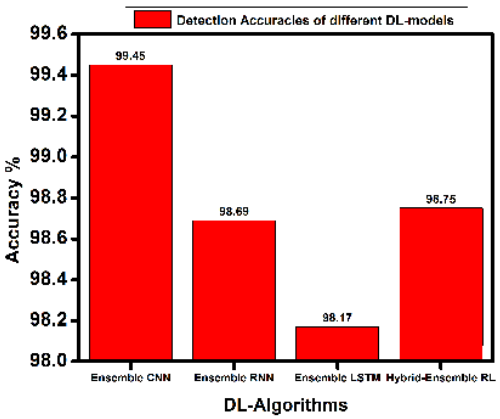


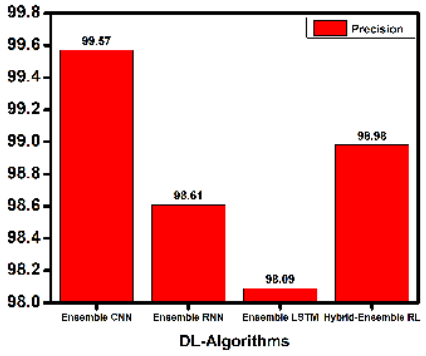
FIGURE 10. Detection accuracies.

named as True label and Predicted label. This parameter is considered, to be the complete performance descriptor for the simulation models and aids in determining the remaining evaluation metrics, for a more complete review of the proposed models. It yields true positives, true negatives, false positives and false negatives on which rest of the evaluation metrics. Detection accuracy of four of the proposed models is determined by taking the main diagonal values in confusion matrix over the total number of samples, shown in Figure 10.

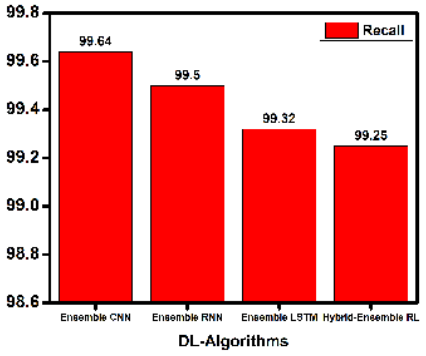
Figure 10 displays a graphical view of detection accuracies of all the proposed Hybrid RL model presented in this paper. High detection accuracy is demonstrated in Figure 13, indicates that a combination of two single CNN models (Ensemble CNN) unlocks the untapped potential of the mainstream CNN and marks the highest accuracy (99.45%) in terms of anomaly detection. From the results and graphs, it is to be noted that other DL-based ensemble and hybrid approaches also imparted an excellent role in anomaly detection: accuracy up to 98.75%. While the primary applications of CNN architectures are in computer vision and media processing, this work demonstrates the applicability and strength of ensemble CNNs for improved and efficient intrusion detection.

Figure 11 illustrates the comparison of precision, recall and F1-measure of the proposed algorithms, demonstrating significant results for the ensemble, hybrid approaches.

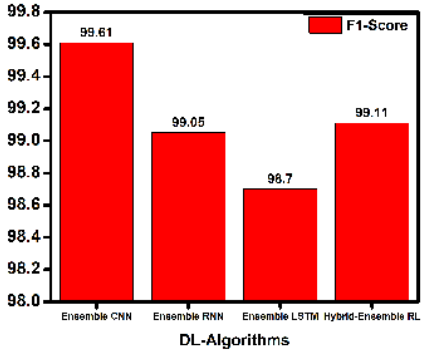
Precision in Figure 11a indicates the correct positive predicted values (PPV) over all the predicted values for a specific class by the classifier. Ensemble CNN attains highest percentage in prediction of the PPV among the other ensemble and hybrid approaches. Similarly, recall or true positive rate (TPR) known for the correct recognition from the relevant samples, of all the models has given in Figure 11b. Ensemble CNN achieves highest value in predicting the values or records correctly. Then f1-measure or f1-score, which utilizes the values of the precision and recall to orchestrate the holistic plan for intimate evaluation of the



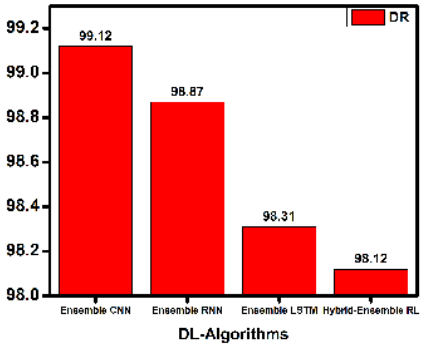
(a) Ensemble CNN



(b) Ensemble RNN

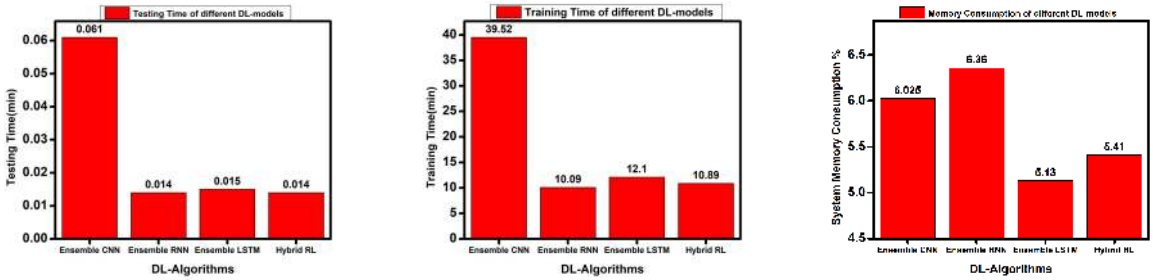


(c) Ensemble LSTM



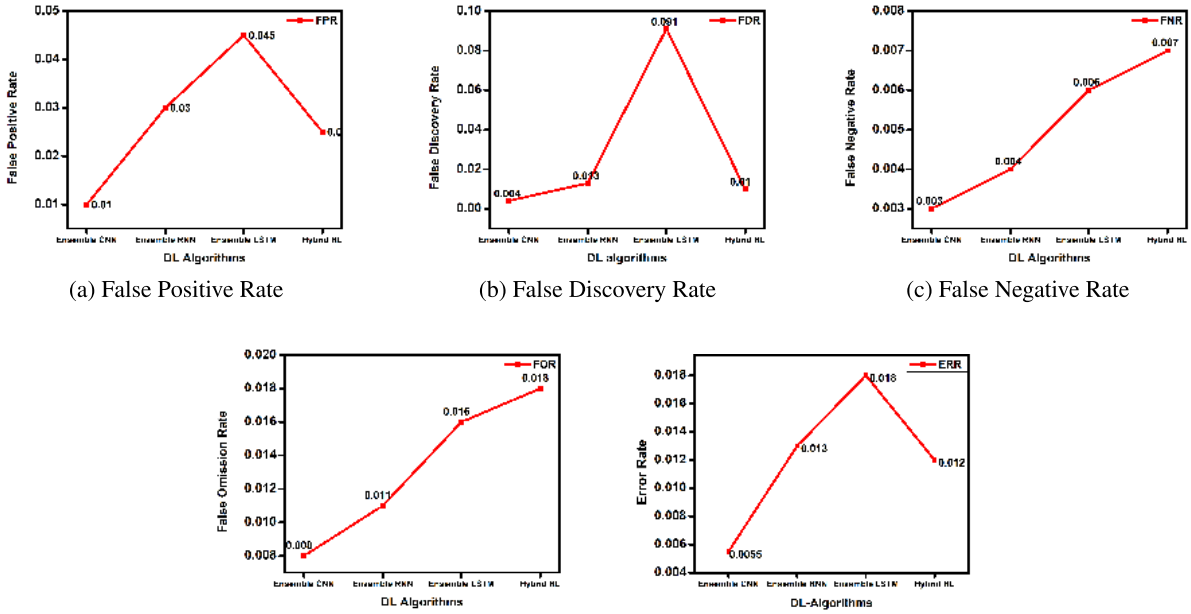
(d) Hybrid RL

FIGURE 11. Comparison of ensemble deep learning models using standard metrics.



(a) Testing Time (b) Training Time (c) CPU Usage (%)

FIGURE 12. Testing time, Training time and CPU usage (%) of each of the proposed DL-Model.



(a) False Positive Rate (b) False Discovery Rate (c) False Negative Rate

(d) False Ommission Rate (e) Error Rate

FIGURE 13. FPR, FDR, FNR, FOR and ERR.

TABLE 5. Comparison of the proposed network-based intrusion detection system (NIDS) with current state-of-the-art.

Paper	Accuracy	Pr.	Rc.	F1	Test Time (min)	Train Time (min)	CPU Usage	Algorithm
Our Contribution	99.45%	99.57	99.64	99.61	0.061	39.52	6.02%	Ensemble CNN
[28]	99.98%	99.03	99.04	99.50	N/A	N/A	N/A	Hybrid (RBM+SVM)
[31]	95.24%	N/A	N/A	N/A	N/A	N/A	N/A	SVM
[4]	83.28%	96	72	82.82	N/A	91.93	N/A	RNN
[22]	N/A	88.9	98.1	93.27	N/A	4800	N/A	BLSTM

specific model, is also calculated and presented in Figure 11c while Figure 11d indicates the detection rate of the proposed ensemble and hybrid models. Area under curve (AUC) and Receiver operating characteristics (ROC) also called AUROC curve is the widely used metric for performance evaluation in classification problems. This metric utilize sensitivity and specificity (TPR and FPR respectively). More AUROC value, better the classifier is in making decision b/w regular and irregular traffic. Figures 4 and 5 illustrates the graphical view of the decision made by the proffered ensemble and hybrid classifiers in anomaly detection. Testing and training time is an important factor in determining the performance of all the proposed models. Figure 12 shows the time taken in minutes, in classification and training by the projected DL-algorithms and the CPU usage from each of the DL-Model.

Ensemble CNN outperforms the other three proposed DL-approaches in almost all the evaluation metrics results however; there is a tradeoff between their training and testing time. Training and testing time of Ensemble CNN is almost 3 times than the other proposed approaches. However, this training and testing time (of ensemble CNN) is consider to be conducive and better than the previous proposed DL-based algorithms in normal and DDoS traffic classification in the existing research work [22], [23] and [24]. For further evaluation of the models, we also concluded the multiple parameters derived from confusion matrix comprising false positive rate (FPR), false discover rate (FDR), false omission rate (FOR), false negative rate (FNR) and error rate (ERR) shown in Figure 13.

CPU usage is almost equal of all of the proposed DL models. This capability of the algorithms promises the reliable and innovative development of future applications through ensemble and hybrid approaches in DL-world and keeps the hype of the AI. Moreover, our proposed technique demonstrates improvements when compared with the existing state of the art DL algorithms (i.e., hybrid and ensemble). The proposed CNN ensemble overcomes the other three approaches (i.e., ensembles RNN, LSTM, RL and Hybrid RL) in terms of performance and exhibits high detection accuracy (99.45%). Despite high detection accuracy, it also outperforms the other state of the art three approaches with reasonable testing and training time. Finally, Table 5 presents that our proposed approach outperforms relatively the extant mainstream DL-based algorithms used for DDoS detection.

Table 5 demonstrates performance of our proposed ensemble CNN approach in efficient DDoS detection in comparison with existing competing approaches presented in [6], [24], [33]. While the approach in [30] shows a high detection accuracy, its corresponding high false positive rate is less desirable in the critical context of detecting intrusions on a network. Unfortunately, some basic evaluation parameters such as model testing and memory consumption are not available in most of the extant research work for rigorous verification.

V. CONCLUSION

Contemporary innovative research and novel cyber security solutions are indispensable to properly secure the new era of digitization. We proposed an efficient and scalable deep CNN ensemble framework to address the issue of the most prevalent and sophisticated DDoS attack detection in SDNs. We evaluated our proposed framework with benchmark deep learning ensembles and hybrid state-of-the-art algorithms on a flow-based SDN dataset. The proposed algorithm demonstrates improvements both in detection accuracy and computational complexity. Finally, We endorse varied deep learning ensemble based detection and prevention mechanisms for the emerging large-scale distributed networks.

REFERENCES

[1] K. S. Sahoo, S. K. Panda, S. Sahoo, B. Sahoo, and R. Dash, "Toward secure software-defined networks against distributed denial of service attack," *J. Supercomput.*, vol. 75, no. 8, pp. 4829–4874, Feb. 2019.

[2] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Commun. Surveys Tutr.*, vol. 15, no. 4, pp. 2046–2069, 4th Quart., 2013.

[3] M. Conti and A. Gangwal, "Blocking intrusions at border using software defined-Internet exchange point (SD-IXP)," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 1–6.

[4] S. Kottler, "February 28th DDoS incident report," Tech. Rep., Mar. 2018.

[5] J. Schreier, "How ddos attacks work, and why they're so hard to stop," Tech. Rep., Dec. 2014. [Online]. Available: https//:kotaku.com/how-ddos-attacks-work-and-why-theyre-so-hard-to-stop-1676445620.

[6] R. Latif, H. Abbas, and S. Assar, "Distributed denial of service (DDoS) attack in Cloud- assisted wireless body area networks: A systematic literature review," *J. Med. Syst.*, vol. 38, no. 11, p. 128, Sep. 2014, doi: 10.1007/s10916-014-0128-8.

[7] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, and L. Gong, "Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN," *Int. J. Commun. Syst.*, vol. 31, no. 5, p. e3497, 2018.

[8] N. Shone, T. Nguyen Ngoc, V. Dinh Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[9] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 29–35.

[10] 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.

[11] A. Akhunzada, A. Gani, N. B. Anuar, A. Abdelaziz, M. K. Khan, A. Hayat, and S. U. Khan, "Secure and dependable software defined networks," *J. Netw. Comput. Appl.*, vol. 61, pp. 199–221, Feb. 2016.

[12] A. Akhunzada, E. Ahmed, A. Gani, M. K. Khan, M. Imran, and S. Guizani, "Securing software defined networks: Taxonomy, requirements, and open issues," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 36–44, Apr. 2015.

[13] P. Mell, "Understanding intrusion detection systems," *EDPACS*, vol. 29, no. 5, pp. 1–10, Dec. 2006.

[14] (Dec. 2018). *What it is Network Intrusion Detection System?*. COMBOFIX. Accessed: Dec. 10, 2018. [Online]. Available: https://combofix.org/what-it-is-network-intrusion-detection-system.php

[15] R. Z. A. Mohd, M. F. Zuhairi, A. Z. A. Shadil, and H. Dao, "Anomaly-based NIDS: A review of machine learning methods on malware detection," in *Proc. Int. Conf. Inf. Commun. Technol. (ICICTM)*, 2016, pp. 266–270.

[16] 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

[17] D. Gold, "Is signature-and rule-based intrusion detection sufficient?" Tech. Rep., Mar. 2017. [Online]. Available: https://www.csoonline.com/article/3181279/security/is-signature-andrule- based-intrusion-detection-sucient.html.

[18] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, nos. 1–2, pp. 18–28, Feb. 2009.

[19] J. Singh and M. J. Nene, "A survey on machine learning techniques for intrusion detection systems," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 2, no. 11, pp. 4349–4355, Nov. 2013.

[20] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2016, pp. 258–263.

[21] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Feb. 2016, pp. 1–5.

[22] 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.

[23] Y. Liu, S. Liu, and X. Zhao, "Intrusion detection algorithm based on convolutional neural network," in *Proc. 4th Int. Conf. Eng. Technol. Appl.*, 2017, pp. 9–13.

[24] A. Elsherif, "Automatic intrusion detection system using deep recurrent neural network paradigm," *J. Inf. Secur. Cybercrimes Res.*, vol. 1, no. 1, 2018.

[25] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE Local Comput. Netw. Conf.*, Oct. 2010, pp. 408–415.
[26] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Comput. Netw.*, vol. 62, pp. 122–136, Apr. 2014.
[27] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "A SDN-oriented DDoS blocking scheme for botnet-based attacks," in *Proc. 6th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2014, pp. 63–68.
[28] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 77–81.
[29] R. Wang, Z. Jia, and L. Ju, "An entropy-based distributed DDoS detection mechanism in software-defined networking," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2015, pp. 310–317.
[30] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues, "Hybrid Deep-Learning-Based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 566–578, Mar. 2019.
[31] H. Peng, Z. Sun, X. Zhao, S. Tan, and Z. Sun, "A detection method for anomaly flow in software defined network," *IEEE Access*, vol. 6, pp. 27809–27817, 2018.
[32] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, and L. Gong, "Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN," *Int. J. Commun. Syst.*, vol. 31, no. 5, p. e3497, Jan. 2018.
[33] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS attack detection method based on SVM in software defined network," *Secur. Commun. Netw.*, vol. 2018, pp. 1–8, Apr. 2018.
[34] A. Abdelaziz, T. F. Ang, M. Sookhak, S. Khan, A. Vasilakos, C. S. Liew, and A. Akhunzada, " Survey on network virtualization using OpenFlow: Taxonomy, opportunities, and open issues," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 10, Oct. 2016.
[35] M. R. Haque, S. C. Tan, Z. Yusoff, C. K. Lee, and R. Kaspin, "DDoS attack monitoring using smart controller placement in software defined networking architecture," in *Computational Science and Technology*, vol. 481. Aug. 2018, pp. 195–203.
[36] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep recurrent neural network for intrusion detection in SDN-based networks," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 202–206.
[37] M. E. Ahmed, H. Kim, and M. Park, "Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 11–16.
[38] A. Akhunzada and M. K. Khan, "Toward secure software defined vehicular networks: Taxonomy, requirements, and open issues," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 110–118, Jul. 2017.
[39] A. Abdelaziz, A. T. Fong, A. Gani, U. Garba, S. Khan, A. Akhunzada, H. Talebian, and K.-K.-R. Choo, "Distributed controller clustering in software defined networks," *PLoS ONE*, vol. 12, no. 4, Apr. 2017, Art. no. e0174715.
[40] M. Conti, A. Gangwal, and M. S. Gaur, "A comprehensive and effective mechanism for DDoS detection in SDN," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2017, pp. 1–8.
[41] "Keras: Deep learning for humans," Tech. Rep., 2018. [Online]. Available: https://github.com/keras-team/keras.
[42] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: http://arxiv.org/abs/1603.04467
[43] S. Bastke, M. Deml, and S. Schmidt, "Combining statistical network data, probabilistic neural networks and the computational power of GPUs for anomaly detection in computer networks," Tech. Rep., 2009, pp. 1–6.

SHAHZEB HAIDER received the bachelor's degree from Bahauddin Zakariya University, Multan, Pakistan, in 2016, and the master's degree in information security from COMSATS University Islamabad, Pakistan, in 2020. He is currently pursuing a career in information security, as a Cyber Security Expert, with one of the largest financial institution Muslim Commercial Bank (MCB) Bank Ltd., Pakistan.

ADNAN AKHUNZADA is an enthusiastic and dedicated professional with extensive 12 years of R&D experience both in ICT industry and academia, with demonstrated history and a proven track record of high impact published research, i.e., patents, journals, transactions, book chapters, reputable megazines, conferences, and conference procedings. His experience as an Educator & Researcher is diverse. It includes his work as a Lecturer, a Senior Lecturer, a Year Tutor, an Assistant Professor at the COMSATS Institute of Information Technology (CIIT), a Senior Researcher at RISE SICs Vasteras AB, Sweden, and as an Associate Professor and the Deputy Director of the Namibia University of Science and Technology, Windhoek, Namibia, having mentorship of graduate students, and supervision of academic and R&D projects both at UG and PG level. He is rigorously been involved in international accreditation such as Accreditation Board for Engineering and Technology (ABET), and curriculum development according to the guidelines of ACM/IEEE. He is also working as an incharge of BS Software Engineering (SE) and Telecommunication Networks (TN) Programme with Comsats University, Islamabad, Pakistan. He is currently involved in various EU and Swedish funded projects of cyber security on industrial controls systems (ICS), process automation, cyber physical systems (CPS), the Internet of Things (IoT), large scale distributed systems, and modeling and design secure software-defined networks. He is a member of technical programme committee of varied reputable conferences and editorial boards. He is currently serving as an Associate Editor for IEEE Access.

IQRA MUSTAFA received the bachelor's degree in software engineering from the International Islamic University Islamabad, in 2014, and the M.Sc. degree in information security from COMSATS University Islamabad, Pakistan, in 2019. She is currently pursuing the Ph.D. degree in cyber security with the Cork Institute of Technology, Nimbus Research Center, Ireland, on fully funded scholarship. She also worked as a Software Developer with the SKANS School of Accountancy, Islamabad, from 2015 to 2016. She also worked as a Research Assistant in the domain of cryptography and machine learning at the COMSATS University Islamabad, from 2017 to 2019. Her major areas of interests are cryptography and network security. She holds a U.S. Patent Post Quantum Cryptographic Communication Protocol. She has command on different languages and tools such as ASP.NET, Python, Sql server and Metasploit, as well as Wireshark.

TANIL BHARAT PATEL was born in Mumbai, India. He received the bachelor's degree (with distinction) in the field of electronics and telecommunication from Mumbai University, in 2015, and the master's degree (Hons.) in cybersecurity from the Cork Institute of Technology, Cork, U.K., in 2019. After completing the degree, he has taken training to complete the professional courses of computer networking like Cisco Certified Network Associate (CCNA) and Cisco Certified Network Professional (CCNP) in routing and switching. He has depth knowledge for networking components and Cisco device configurations. During the course of time, he has done several internships like Doordarshan and Western Railways, both of them are good Indian Government firms. During his career, he has also come across several languages such as C++, Java, and Python and has been constantly working to upgrade himself.

AMANDA FERNANDEZ is currently with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA.

KIM-KWANG RAYMOND CHOO (Senior Member, IEEE) holds the Cloud Technology Endowed Professorship with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio. He is also an Adjunct Associate Professor with the University of South, and a Fellow of the Australian Computer Society. He was a recipient of various awards, including the ESORICS 2015 Best Paper Award, the Winning Team of the Germany's University of Erlangen-Nuremberg Digital Forensics Research Challenge 2015, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the Fulbright Scholarship, in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award, in 2008. He serves on the Editorial Board for *Cluster Computing*, *Digital Investigation*, the *IEEE Cloud Computing, Future Generation Computer Systems*, the *Journal of Network and Computer Applications*, and *PLoS ONE*. He also serves as the Special Issue Guest Editor for *ACM Transactions on Embedded Computing Systems* (2017; DOI: 10.1145/3015662), *ACM Transactions on Internet Technology* (2016; DOI: 10.1145/3013520), *Digital Investigation* (2016; DOI: 10.1016/j.diin.2016.08.003), *Future Generation Computer Systems* (2016; DOI: 10.1016/j.future.2016.04.017), the *IEEE Cloud Computing* (2015; DOI: 10.1109/MCC.2015.84), the IEEE NETWORK (2016; DOI: 10.1109/MNET.2016.7764272), the *Journal of Computer and System Sciences* (2017; DOI: 10.1016/j.jcss.2016.09.001), *Multimedia Tools and Applications* (2017; DOI: 10.1007/s11042-016-4081-z), and *Pervasive and Mobile Computing* (2016; DOI: 10.1016/j.pmcj.2016.10.003).

JAVED IQBAL, photograph and biography not available at the time of publication.