IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 1, 2020, accepted July 12, 2020, date of publication July 16, 2020, date of current version July 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009733

An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks

KSHIRA SAGAR SAHOO[1], (Member, IEEE), BATA KRISHNA TRIPATHY[2], KSHIRASAGAR NAIK[3], (Senior Member, IEEE), SOMULA RAMASUBBAREDDY[1], BALAMURUGAN BALUSAMY[4], MANJU KHARI[5], AND DANIEL BURGOS[6], (Senior Member, IEEE)

[1]Department of Information Technology, VNRVJIET, Hyderabad 500090, India
[2]School of Electrical Sciences, Indian Institute of Technology Bhubaneswar, Bhubaneswar 752050, India
[3]Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada
[4]School of Computing Science and Engineering, Galgotias University, Greater Noida 203201, India
[5]Department of CSE, AIACTR, New Delhi 110031, India
[6]Research Institute for Innovation & Technology in Education (UNIR iTED), Universidad Internacional de La Rioja (UNIR), 26006 Logroño, Spain

Corresponding author: Daniel Burgos (daniel.burgos@unir.net)

ABSTRACT Software-Defined Network (SDN) has become a promising network architecture in current days that provide network operators more control over the network infrastructure. The controller, also called as the operating system of the SDN, is responsible for running various network applications and maintaining several network services and functionalities. Despite all its capabilities, the introduction of various architectural entities of SDN poses many security threats and potential targets. Distributed Denial of Services (DDoS) is a rapidly growing attack that poses a tremendous threat to the Internet. As the control layer is vulnerable to DDoS attacks, the goal of this paper is to detect the attack traffic, by taking the centralized control aspect of SDN. Nowadays, in the field of SDN, various machine learning (ML) techniques are being deployed for detecting malicious traffic. Despite these works, choosing the relevant features and accurate classifiers for attack detection is an open question. For better detection accuracy, in this work, Support Vector Machine (SVM) is assisted by kernel principal component analysis (KPCA) with genetic algorithm (GA). In the proposed SVM model, KPCA is used for reducing the dimension of feature vectors, and GA is used for optimizing different SVM parameters. In order to reduce the noise caused by feature differences, an improved kernel function (N-RBF) is proposed. The experimental results show that compared to single-SVM, the proposed model achieves more accurate classification with better generalization. Moreover, the proposed model can be embedded within the controller to define security rules to prevent possible attacks by the attackers.

INDEX TERMS DDoS attack, GA, KPCA,N-RB, SDN, SVM.

I. INTRODUCTION

RECENTLY the Software Defined Networks (SDN) paradigm has gained significant interest from many researchers. The SDN paradigm offers a greater potential to provide a secure, flexible, and reliable network system [1]–[3]. Separation of the control plane from the underlying infrastructure layer is the main innovation behind SDN. The centralized. controller manages the packet-forwarding devices that need to be configured via a well-designed interface like OpenFlow [4], [16]. In SDN, the network devices like switches have only forwarding logic, whereas the control logic and decision-making ability are softwarized at the controller. This allows the controller to instruct the switches with new network policies, and underlying devices start to follow the policies maintain in the flow table [40]. When a packet arrives at a switch, it checks its flow table, and if the flow matches, it forwards the packet to the destination. If no match founds in the flow table, OpenFlow enabled switch sends control packet to the controller for making an appropriate decision. The controller can handle multiple flow tables maintained by OpenFlow switch, consequently achieving programmability in the control layer of SDN. According to the controller policy, the flow tables can serve as a switch, firewall or router that

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

exhibit similar roles. Despite all these impressive innovations, various architectural components pose additional security threats to SDN. As far as different issues to be addressed, the security of SDN is considered as the highest concern. Among many security threats, one of the critical security issues is Distributed Denial of Service (DDoS). The main aim of this attack is to make computing resources unavailable to the legitimate users. This attack is usually caused by more than one bot, penetrated by software from malicious code. As the initial process is simple, the DDoS attack can quickly spread and cause massive damage to the network, but the defend process is very troublesome. Although the network administrator can identify a possible attack, it may not be reasonable to account for concurrent attacks in real time. Hence, it is essential to impose certain security rules on the controller. Therefore, an efficient detection technique and mitigation rules must be designed for future network architecture like SDN. Since the controller is the central intelligent part of the SDN, several techniques like neural network and machine learning can be used to leverage network security.

For detecting DDoS attack, two different approaches have been followed by Intrusion Detection Systems; such as: signature-based and anomaly-based detection method [5].

In signature-based approach, IDS monitors the packets and then compares these packets against a set of signatures from known malicious threats. For new signature, the IDS takes certain man hours to test and deploy the signature. Hence, it is necessary to have a less human intervention system. The anomaly-based IDS is based on the concept of a baseline for network behaviour. The Machine Learning (ML) approach helps in implementing the network behaviour that can learn from historical data and provide a prediction for the upcoming packets based on the training data. These techniques have shown notable performance in the classification of the attack traffic and legitimate traffic. Moreover, instead of checking the packet payload, ML techniques require a particular set of features of the flows such as a combination of source IP and destination IP addresses, a combination of source and destination port addresses, flows duration etc. [6]. As compared to Deep Packet Inspection (DPI) based techniques, ML technique incurs a lower computational cost [8], [12], [28], [39]. So, in this work ML approach has been chosen for DDoS detection.

When the SDN infrastructure under the DDoS threat, both the controller layer and the forwarding layer suffer from resource depletion. Although previous research efforts have shown tremendous improvements in the control layer anomaly detection, but it lacks a detailed analysis [9], [8], [17]–[19]. Machine Learning based research for Intrusion Detection System (IDS) usually needs a large volume and large dimensional network traffic data in a constantly changing network environment. Besides the relevance of choosing the most suitable features from the dataset, setting the performance parameters of the implemented algorithms with the optimal value is another important factor, which influences to design an efficient detection model [36].

Motivated by this fact, we have designed a DDoS detection framework that utilizes SVM as the learning model. Although SVM is treated as a good classifier in terms of accuracy and generalization capabilities, but the limitation here is the higher training time [22]. Hence, to overcome these, various feature selection techniques have developed which can be integrated with SVM for obtaining a better result with reduced dimensional data. In [26], authors deployed kernel principal component analysis (KPCA) as the feature selection technique and GA deployed for optimizing the parameter of SVM. For better accuracy and less testing time our proposed work follows the SVM model proposed by Kuang *et al.*

The main contribution of the paper is given below:

a. This work utilizes SVM technique as the prime classifier for predicting malicious traffic. An effective solution has been proposed for protecting SDN and has analyzed it through three different SVM variants.
b. The proposed detection approach combines SVM with KPCA and GA. Feature extraction has been carried out by KPCA, and SVM classifier is used for attack classification. Further to lessen the training time an improved radial basis kernel function has functionalized. Additionally, genetic algorithm has utilized for optimizing various parameters of the classifier.
c. The detection module is run over the controller. Validate the proposed DDoS detection framework through a simulated environment that comprises POX controller, OVS, and Mininet emulator.
d. The attack detection results compared with other classifiers which show that the proposed SVM model performs effective and accurate classification than others.

The rest of this paper is organized as follows. Section II describes the related work that uses ML approaches for anomaly detection in the SDN environment. Section III, Section IV provides background details and a detailed explanation of the proposed work. Section V and Section VI discuss the design principle and simulation results, respectively. Finally, in Section VII, we have summarized this paper.

II. RELATED WORK

For DDoS detection, ML is treated as an effective technique which can detect against the control plane attack. In this section we will discuss few previous research works that are made for SDN and utilized ML and DL based techniques are summarized below. The state-of-art detection mechanisms are listed in Table 1. The detection mechanisms are segregated based on feature selection (FS) and parameter optimization (PO).

In [7], authors have conducted their anomaly detection framework over OpenFlow and SFlow simulator. For detection purpose, entropy technique is applied and for traffic collection, sFlow is used for sampling mechanism. However, the entropy technique has a significant shortcoming, i.e., it always follows the normal distribution of the traffic. The COFFEE model utilizes the OpenFlow protocol to

TABLE 1. Existing DDoS attack detection techniques used in SDN.

References	Technique used	Description	FS/PO
[9]	Neural Network,(Self-Organizing Maps)	Utilizes SOM technique for classification. PPF, APF, GDP,GSF like features used.	Y/N
[8]	SVM	Traditional features used for attack traffic detection.	N/N
[10]	Rate limiting and TRW-CB	Collect traffic and implement such techniques	Y/N
[11]	BPNN	Proposed SD-anti-DDoS uses BPNN for packet trace-backing.	N/N
[45]	SPRT	For high rate DDoS Sequential Probability Ratio Test has been used.	N/N
[20]	Radial Basis Function	RBF network is further optimized with GA.	Y/N
[34]	Gradient Boosting (XGBoost)	GA is used over XGBoost for better accuracy.	N/Y
[31]	SVM	GA is used for optimizing parameter.	N/Y
[33]	Deep Learning-based Sparse Autoencoder (SAE)	For traffic accumulation and extraction, TCFI module is used.	Y/N
[30]	PSO-BP neural network	Entropy and Machine Leaning is used.	Y/N
[46]	Advanced SVM	Features like AFP, AB, SI are used. No feature selection technique is used.	Y/N
[25]	SVM	GA is used for feature selection.	Y/Y
[23]	SVM	PSO is used for optimizing parameters.	Y/Y

distinguish the botnet and delete it from the network by extracting the features from the flow [13]. To extract more features the suspected flows send to the controller for extracting more features. Ashraf *et al.* uses various features to utilize ML techniques for handling DDoS attack [14].

The extreme gradient boosting (XGBoost) technique is used as the detection method by Zhou *et al.* [18]. They have validated their work with an SDN simulated environment that builds on Mininet and POX controller. To validate their work, authors have used a data set which is collected by TcpDump packet analyser tool. In [33], Niyaz *et al.* use Deep Learning-based Sparse Autoencoder (SAE) for their malicious defense system. For traffic accumulation and extraction, authors have used TCFI module inside the controller. In a similar context, Garg *et al.* employed Deep Learning-based RBM with the SVM technique [37]. For dimensional reduction, the RBM technique has used. In a flow-based network like SDN, a scalable deep CNN model has employed by the researchers to curb DDoS attack [38]. They have appraised the model with hybrid algorithms on an SDN dataset.

Now a few research articles which have employed nature-inspired algorithm for IDS will be discussed. Zhenpeng *et al.* proposed a DDoS solution that utilizes normal entropy metric and Particle Swarm Optimization (PSO)-BP neural network [30]. In [31], authors have used the ML model for anomaly detection that explores the bio-inspired algorithm for feature selection. Alqahtani *et al.* proposed an ML model for anomaly detection in the wireless sensor network [34]. The ML model is based on bio-inspired techniques like the Genetic Algorithm (GA) and XGBoost classifier. For better classification, later gradient boosting technique has employed. For intrusion detection, in [25], authors have applied GA as a feature selection technique with SVM as the detection classifier. In another work, Srinoy *et al.* proposed a detection model that adopted particle swarm optimization (PSO) technique for extracting attack features [23].

The literature survey concluded that in SDN, the control plane is more vulnerable to DDoS attacks, and most of the authors have preferred ML techniques. Moreover, work on feature selection from a dataset still insignificant. An insignificant and small number of features are not able to detect all types of attacks. Hence, in this paper, by appropriating the SDN controller capabilities, we have adopted SVM as the choice of the classifier with the principal component for better detection accuracy. However, the standard SVM has few limitations. One of the limitations is that the performance of the model depends on its parameters selection. Therefore, in this work feature selection (FS) has been carried out by KPCA, and GA utilized for parameter optimization (PO) of SVM.

III. BACKGROUND

A. DATA FLOW IN SDN

In SDN, the underlying switches only posses the forwarding logic. When a packet arrives at an OF switch, it checks its flow table and if the flow matches, it forwards the packet to the destination. If no match is found in the flow table, it sends the *packet_in* to the controller for taking appropriate decision. Thus by following the above procedure, SDN separates the processing plane and forwarding plane. If a huge volume of spoofed packets is sent together, each time there is a miss-match in the flow table and in turn, large *packet_in*

events are sent to the controller. The limited memory space of the controller causes a delay for the processing the requests. This processing delay creates a chance for the attackers to modify the flow entries, dropping the legitimate traffic, makes overflow the flowtable, etc. This can be expressed as a DDoS attack on the SDN controller.

B. DDoS ATTACK IN SDN

In DDoS attack, rate of arriving incoming packet to the network is high, the collection of legitimate and spoofed packet will collectively bind the network resources hence make the resource exhaust. If this process continues server will be unreachable for the new incoming legitimate packet and the packet will be dropped by making the network unreliable. DDoS attacks can broadly be categorized into three types, such as volumetric attack, protocol-exploitation attack, and application layer attack. The UDP flood and TCP flooding attacks come under volumetric attacks, whereas HTTP flood and DNS flooding categorized as application-layer attacks [44].

In SDN, the control plane is responsible for centralized network intelligence. In single controller architecture, there is a high possibility of single point of failure (SPF). If the attacker gets access to the controller, it can cause massive destruction to the network infrastructure [36]. The controller applications like load balancing, firewall, routing are operated on top of the control plane. For instance, if firewall application get accessed, then a different Access Control List (ACL) can be formed [43]. Though TLS/SSL connection between the controller and OF switch creates a secure connection; in case the loss of TLS connection, it needs a backup controller for the switch. In such a scenario, OF switch can use flow tables as per its choices. A malicious flow rule can be implanted into the flow table which may create DDoS attack onto the controller. Besides this, the flow format of SDN has some important properties. The SDN controller uses the southbound protocol such as OpenFlow to take action against the flow entries. There may be more than one rule for the same flow. The various fields of flow include priority, counter, timeout, action field, etc. Each field is designated for a specific task.

For example, the counter field keeps the information about the received bytes per flow, the timeout field indicates the time needed for a flow to expire since it was placed in the flow table. The instruction field specifies the action needed for a flow entry. The Figure 1 represents the discussed scenario.

IV. DDOS DETECTION USING ML APPROACH

There are three types of machine learning (ML) algorithms; they are supervised learning, unsupervised learning, and reinforcement learning algorithms [41], [42]. In supervised learning algorithms, each input data is associated with a class which is called label. During testing, the machine predicts the class of input data based on the training sample. This is called supervised because the class of training sample is known during the learning phase. In unsupervised cases, we don't

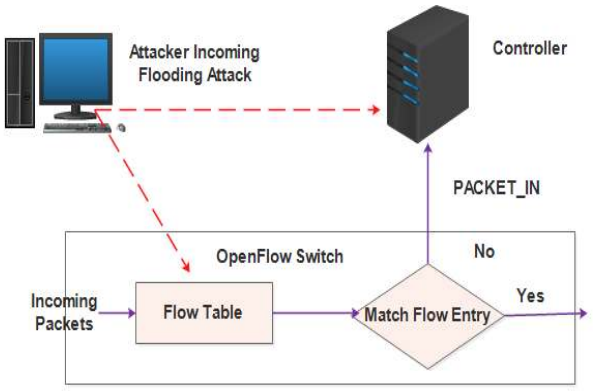


FIGURE 1. Attack to both forwarding and control layer.

have any labelled responses. These methods usually used to cluster the population in various groups. On the other hand, in reinforcement learning method, the machine is continuously trained using the trial and error approach. It learns from previous knowledge and attempts to achieve the best possible knowledge to make the right decisions. In this work, we have studied on supervised learning algorithm such as SVM, because the algorithm has the immense ability to handle high dimension data and much demand.

A. SUPPORT VECTOR MACHINE (SVM)

The basic principle of SVM, is to find an optimal hyperplane that produces a better generalization of the dataset [21]. It develops a model that predicts whether a new sample falls into one of the categories or not. Let's given a training data set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x_i \in R^n$ and y $\{+1,-1\}$.

The x_i represents the transferred input vector and y_i is the target value. SVM is a binary classifier in which the class labels contain only two values $+1$ or -1. From the inputs, SVM draws an optimal hyper-plane H that separates the data into different classes and the hyper-plane H can be defined as:

$$x_i \in R^n : (\vec{w}, \vec{x}) + b = 0, \quad \vec{w} \in R^n, \ b \in R \tag{1}$$

The algorithm is based on finding the hyper-plane which gives the maximum distance of separation between training samples using the following function.

$$f(\vec{x}) = sign(\vec{w}, \vec{x}) + b \tag{2}$$

For the problem of multiclass learning, SVM solved it as a single multi-class problem further it is modified into multiple binary problems. For detecting attacked traffic, two linearly separable data is considered. Hence, the optimal hyper-plane can be combined by the inequality as given in Equation 3.

$$y_i\{(\vec{w}, \vec{x}) + b\} \geq 1, \quad s.t.i = 1, \ldots, n \tag{3}$$

So, the optimization problem can be written as given in Equation 4.

$$\begin{aligned} &minimization \frac{1}{2}(w^T, w) \\ &\text{s.t } y_i(w.x + b) \geq 1 \end{aligned} \tag{4}$$

But, for a non-separable case, the optimization problem can be formalized as follows:

$$\begin{aligned} &minimization \frac{1}{2}\left(w^T, w\right) + C\sum_{i=1}^{n} \xi_i \\ &\text{s.t. } y_i\left(w.x + b\right) + \xi_i \geq 1; \quad \xi_i \geq 0 \end{aligned} \tag{5}$$

where, ξ is the slack variable which helps to select the hyper-plane with less error and cost value (C) is the regularization parameter. The optimal C value can be obtained by an empirical investigation by the user. A large cost value resulting to smaller-margin, in turn it may cause an over-fitting situation.

B. KPCA+SVM CLASSIFICATION MODEL

To get better performance, parameter selection has major significance. Using Radial Basis Function (RBF) in the training process of a model produces a large number of hyper plane which takes a long period of time for training the model. To solve, such problem this model combines SVM with Kernel Principal Component Analysis (KPCA) to reduce the dimensions of features and at the same time reduces the training time. In the proposed model, KPCA maps the high dimensional input features into a new lower dimensional eigen space. Further, it extracts the principal features from the training data-set for classifying the attack. For dimensional reduction and feature selection Principal Component Analysis (PCA) is a widely used technique. The selection of a subset of features from a large feature set is based on the highest co-relation with the principal component. It can have the ability to extract the linear structure information but fails to extract non-linear information. But, K-PCA transfer input data into higher dimensional space in which PCA is carried out.

Let, $\{a_1, a_2, \ldots, a_n\}$ be the set of n training samples. The j^{th} transferred feature y_j value can be obtained by using Equation 6. By using this, the Kernel-PCA, transformed the feature vector to a new sample vector.

$$y_j = \frac{1}{\lambda_j}\gamma_j^T \left[k(a_1, a_n'), \ldots, (a_n, a_n')\right]^T \tag{6}$$

where, a_n', is the new a's value and γ_j is the orthogonal eigen vector to the q largest eigen value i.e. $\lambda_1 \geq \lambda_2, \ldots, \geq \lambda_q$.

C. KERNEL FUNCTION USED IN SVM

It is not possible to find a linear decision boundary for some classification problems. If data points projected into a higher dimension space from the original space, a hyper-plane in the projected dimension helps to classify the data points. To deal with such problem, a kernel function is used to transfer the data set to a higher dimensional space. In general, the Computational cost increases, if the dimension of the data increases. The dot product of two vectors of the same dimensional produces a single number. Hence, the kernel function can utilize this property in a different space without even visiting the space. The standard method of calculating the dot product requires $o(n^2)$ time, whereas kernel requires with $o(n)$ time.

In SVM there are some well-known kernel functions are used such as RBF, polynomial, sigmoid, etc. Since, RBF kernel function requires fewer parameters set, in most of the classification problem, SVM performs well in this kernel function. However, in a networking scenario, network flows contain several attributes, which may vary from protocol to protocol. Therefore, when the differences between the attribute sets are very large, RBF kernel may create a sizable number of support vectors (SV). A large number of SVs may increases the training period of the model. To lower the training period and to improve the overall performance, an improved kernel function called N-RBF is developed. Further, to normalize the attribute values, the NRBF can be expressed as:

$$K\left(x_i, x_j\right) = exp\left(\frac{-\left|\frac{x_i - mv}{ms} - \frac{x_j - mv}{ms}\right|^2}{\sigma^2}\right) \tag{7}$$

where, K,mv,ms represents the dimension of the sample vector, mean value, and the mean squared deviation of the features respectively. Further, mv_i and ms_j can be described as follows:

$$mv_j = \frac{1}{n}\sum_{i=1}^{n} P_{ij} \tag{8}$$

$$ms_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(P_{ij} - mv_j\right)^2} \tag{9}$$

where, n represents training samples and p_{ij} represents the j^{th} attribute of the i^{th} sample. N-RBF is a positive kernel function.

Further, the selection of C and σ plays an important role in the performance of SVM model. There are several disciplined approaches that have been utilized to get the optimal parameters. Technique like GA, simulated annealing (SA), and Particle Swarm Optimization like meta-heuristic algorithms can be employed for finding the optimal parameters.

D. OPTIMIZING PARAMETERS WITH STANDARD GA

Genetic algorithms (GA) is a search technique based on the ideas of natural selection and genetics. This technique is primarily used to generate high-quality solutions for optimization and search problems. The algorithm simulates on the basis of "survival of the fittest" type scenario, where each generation of the algorithm attempts to improve upon the previous generation. It operates on the finite population of chromosomes and each chromosome is a possible solution. The best possible solution using GA can be obtained by setting various generic operators such as crossover, mutation, stopping criteria, etc. The process of selection, evaluation, re-combinations form one generation in the execution of the genetic algorithm. Our objective function (Mean Absolute Error (MAE)) is a minimization problem which has given in Equation 10 and it searches the best possible combination

of C and σ.

$$MAE = \frac{1}{T}\sum_{i=1}^{T} |\frac{AL_i - PR_i}{AL_i}| \tag{10}$$

where, T, AL_i and PR_i represent classification period, actual values and predicted values respectively.

The selection process of optimized SVM parameters using GA has been illustrated in Algorithm 1. In the algorithm, roulette wheel method has used for selecting new population.

Algorithm 1 Optimized SVM Parameter Using GA

Input:
1. Population size
2. crossover probability
3. mutation probability

// Chromosome represents C and σ value in binary form
// Bit 1 represents selection of corresponding feature and vice versa (bit 0).

Output: Obtain optimal parameters σ and C

Optimization Loop :

1. **for** it = 1 to maxIt **do** //maxIt-maximum iteration
2. **for** i = 1 to nPop **do** //nPop- Total population
3. Calculate Fitness value using Eq 10.
4. Select new population using roulette wheel method
5. Select individuals with crossover probability - apply two pint cross over
6. Select individuals with mutation probability
7. **end for**
8. **end for**

Obtain optimal parameters σ and C

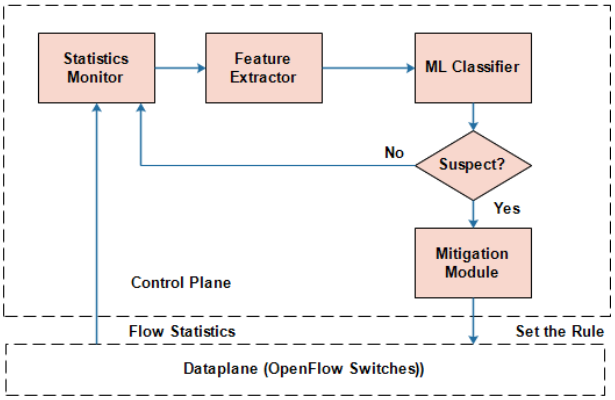


FIGURE 2. The proposed DDoS detection framework for SDN.

Algorithm 2 Proposed DDoS Detection Procedure

Input: Set ΔT

Output: Attack classification

1. **for** Each active OF switch in the network during ΔT **do**
2. OF Switches ← Controller(*Flow_Start_Request*)
3. Collect the Flow Statistics
4. Extract Feature set (F^0) using KPCA
5. **for** Test (F^0) with trained K-PCA+GA+SVM model **do**
6. DDoS Detection Process Start
7. **if** (The Classifier predict the attack) **then**
8. Action *Delete_Flow_Entry*
9. **else**
10. Allow the flow to access the host
11. **end if**
12. **end for**
13. **end for**

V. DESIGN PRINCIPLE OF THE PROPOSED DDOS FRAMEWORK

A. PROPSED DETECTION FRAMEWORK

Our DDoS attack detection framework monitors the OpenFlow (OF) switches during predetermined time intervals ΔT. During such intervals, the controller sends *flow_stat_request* to each switch present in the network. In turn, the controller receives the flow statistics and then the statistics is fed to the statistics monitor module to extract the features discussed in the above section. After feature selection, the proposed ML classifier, classifies the traffic whether it is normal or malicious traffic.

Figure 2 describe an overall proposed detection framework. The Algorithm 2 summarizes the proposed approach.

Detail description about each module is given below.

Statistics Monitor : The module sends F*low_start_request* message to the OF switches and in turn, it receives the flow statistics information.

Feature extractor : Feature extractor module is meant for extracting the features that are essential for attack detection. For feature extraction, the proposed work utilizes KPCA technique. After feature selection, all extracted feature is inputed to the ML classifier.

ML Classifier: This module is responsible for classifying the traffic as per the training model. In this approach, SVM is considered as the ML classifier. Any learning method can be used as per the requirement. We have used KPCA+SVM+GA model for DDoS attack detection because this model takes least training/testing time with much better accuracy than single SVM.

Mitigation Module: For DDoS mitigation, a separate module is designed inside the controller. After DDoS detection, immediately mitigation module sets a flow rule which drops all the packets coming from the underlying switch. This rule prevents the flows to a particular $IP_{destination}$ address with a specific $IP_{protocol}$. The rest of the flows communicate in the network ordinarily.

Since the considered data-set comprises five different types of traffic, hence multi-SVM classifier is applicable for DDoS detection. There are two popular techniques that are used

for SVM multiclass classification such as: ''One-against-all'' and ''Binary tree''.

B. PROPOSED SVM MODEL FOR DDOS DETECTION IN SDN

The ''Binary tree'' technique requires only *(n-1)* two-class classifiers for a case of n class problem. Whereas, 'One-against-all' approach requires n number of two-class SVM classifiers. In this approach, each class is trained with all the samples. Due to less number of classes required for the training process, 'Binary Tree' classifier has been considered for constructing the model. Based on the characteristics of the traffic, four SVM classifiers are developed to identify the five different classes. The basic principle of proposed SDN based DDoS detection framework which is the combination of SVM along with KPCA and GA is shown in Figure 3. For the proposed model, all SVMs use N-RBF as the kernel function. Moreover, the two important parameters of SVM i.e. C and σ, are optimized with the GA technique which has been discussed earlier. Then, with the help of these optimal parameters the SVM model is trained. The proposed detection model comprises of two stages. In the first stage, KPCA is employed for achieving the principal component and SVM is used as the classifier. The second stage utilizes the feature subset for the training and testing of SVM. The N-RBF kernel is adopted by KPCA as well as SVM classifier. The parameter selection of SVM-GA technique has been illustrated in Figure 4.

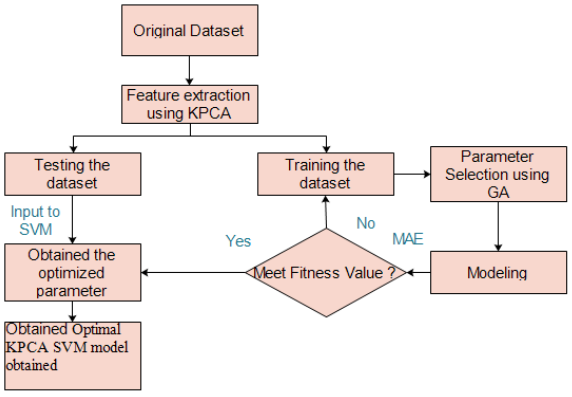


FIGURE 3. Proposed SVM model for DDoS detection.

VI. SIMULATION AND RESULTS

In this section, we discuss a comparative study of the proposed approach with other approaches such as PCA-GASVM, and traditional SVM in terms of accuracy rate, false alarm rate (FAR), and training time, etc. For experimentation the following data set and simulation environment is considered.

A. DATASET SELECTION

For training and testing purpose, a modern DDoS dataset has considered [24]. This dataset consists of 27 features and 21,60,668 records.

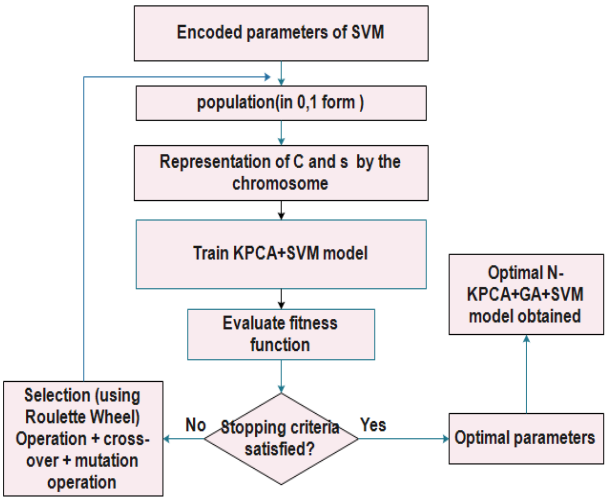


FIGURE 4. Process of optimizing SVM parameters using Genetic Algorithm.

TABLE 2. Traffic distribution.

UDPFlood	201344
HTTPFlood	4110
Smurf	12590
SiDDoS	6665
Normal	1935879

The distribution of the records in the dataset has given in TABLE 2. In order to verify the effectiveness of the proposed model, another dataset called NSL-KDD is used. It contains 41 features, and randomly 1,08,400 records have considered for the simulation [35]. It is the more refined version of KDD'99 dataset. NSL-KDD contains different attacks such as Probe, DoS, R2L, U2R etc. In both the dataset, the redundant records are not present; hence, ML classifiers will not be biased towards more common instances. The selected records in each group from NSL-KDD is inversely proportional to the percentage of instances in the initial KDD data set. As a result, different ML algorithms can perform efficiently and evaluate accurately [47]. The proposed algorithm evaluated over two dataset separately. For experimenting, we named the previous data set as ''Data set-I'' and NSL-KDD is named as ''Data set-II''. The feature set involved in the Dataset-I and Dataset-II has listed in TABLE 3 and TABLE 4, respectively.

B. SIMULATION ENVIRONMENT

In order to do simulation for SDN network, it is important to select a controller. We have chosen POX controller for the experiment [27]. It is considered as a fast, and a customized controller. Mininet is a standard network emulator tool that can be used for SDN [15]. It can make a prototype of the network on a laptop or PC. The network topology of any size can be tested on it and the developed code can be used for

TABLE 3. Feature set (Dataset-I).

No.	Feature Name	No.	Feature Name
1	Destination_Address	15	Packet_in
2	Source_Address	16	Packet_out
3	Packet_ID	17	pktr
4	From_Node	18	Packet_delay_node
5	To_Node	19	Packet_rate
6	Packet_size	20	Pakt_average_size
7	Packet_type	21	Byte_rate
8	FID	22	Utilization
9	Flags	23	Packet_delay
10	Sequence_No	24	Packet_sent_time
11	No_of_Packets	25	Packet_rec_time
12	No_of_Bytes	26	First_pkt_sent
13	Node_name_from	27	Last_pkt_reserved
14	Node_name_to		

TABLE 4. Feature set (Dataset-II).

No.	Feature Name	No.	Feature Name
1	Protocol	15	Duration
2	Source_Byte	16	Dest_Byte
3	Count	17	Guest_login
4	Logged_in	18	Dst_host_count
5	rerror_rate	19	Dst_host_srv_count
6	serror_rate	20	Diff_srv_rate
7	Wrong_fragment	21	Srv_diff_host_rate
8	urgent	22	Dst_host_same_srv_count
9	Flags	23	Dst_host_dif_srv_count
10	Land	24	Num_failed_login
11	dst_host_srv_rerror_rate	25	Num_Compromised
12	dst_host_rerror_rate	26	Root_shell
13	dst_host_srv_serror_rate	27	Num_root
14	Num_shell	28	Num_access_file
29	Is_host_login	30	Num_outbound_cmds
31	dst_host_serror_rate	32	
33	dst_host_srv_serror_rate	34	dst_host_same_src_port_rate
35	Num_accessFiles	36	Num_file_creations
37	Same_srv_rate	38	Su_attempted
39	Diss_srv_rate	40	Srv_count
41	Dst_host_same_srv_rate		

real network. Hence, Mininet 2.0.0 emulator is considered for this experiment.

The above-described classification algorithms were conducted on the machine having core i5 processor, 8 GB RAM, 64-bit operating system and clock speed of 2.30 GHz. Mininet 2.0.0 has installed on the Virtual Box that supports OpenFlow version 1.3. Using Mininet a topology has created which contains 15 switches and 64 hosts. In the experiment, a single host tries to attack the other hosts whose IP is 10.0.0.1 with IP spoofing.

C. PERFORMANCE METRICS

Once the model is trained, the next step is to identify the type of attack and attacked hosts in the testing phase. An ML model is accurate if it correctly predicts the attack type during the attack. The performance of the model was evaluated based on the confusion matrix. The test outcome can be termed as positive or negative, for which the following terms have used. Further, the performance of the detection model is measured using the following metrics given in Equation 11, Equation12, and Equation13.

$$Accuracy = \frac{TP + FN}{TP + FN + FP + TN} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$Precision\,(in\%) = \frac{TP}{TP + FP} \tag{13}$$

where,

- True Positives (TP) Rate: Attack traffic classified as attack traffic.
- False Negative (FE) Rate: Attack traffic classified as legitimate traffic
- False Positive (FP) Rate: Legitimate traffic classified as attack traffic.
- True Negative (TN) Rate: Legitimate traffic classified as legitimate traffic.

D. PARAMETER SETTINGS

Experiment in ML, is usually split into training and testing part. Then the model has to fit into the train data, in order to predict the test data. The following experiments have been carried out to verify the effectiveness of the proposed SVM model on SDN environment. After 50 times simulations, the optimal parameters have been determined for SVM, which are tabulated in Table 5.

TABLE 5. Optimal parameters for various SVM model.

C	93.119
α	0.09
Population Size	40
Number of Iteration	100
Cross-over probability	0.8
Mutation Probability	0.06

E. RESULT DISCUSSION

1) CLASSIFICATION

In our experiments, a 5-fold SCV technique is used to make the classifier stable and more generalized for independent datasets. The two datasets, Dataset-I and Dataset-II contain a different number of samples for each traffic category. The stratified technique splits each fold in such a way that

TABLE 6. Detection accuracy (in %) and total time taken by different SVM variants, KNN, and random forest.

ML methods	70:30		80:20		90:10		70:30		80:20		90:10	
	Accuracy	Time (sec.)	Accuracy	Time (sec.)	Accuracy	Time (sec.)	Accuracy	Time (sec.)	Accuracy	Time (sec.)	Accuracy	Time (sec.)
	Data set- I						Data set- II					
N-KPCA+GA+SVM	98.553	1120.10	98.713	1202.55	98.907	1229.84	92.381	2120.10	92.843	2212.54	92.917	2421.11
KPCA+GA+SVM	98.033	1205.58	98.164	1321.28	98.226	1394.16	92.131	2205.96	92.382	2335.96	92.842	2425.16
PCA+GA+SVM	97.041	1362.24	97.516	1399.42	97.831	1412.78	90.849	2261.54	90.161	2395.17	90.245	2401.50
Single-SVM	94.417	1562.75	95.055	1910.67	95.542	2046.61	89.721	2272.44	89.755	2395.96	89.993	2495.96
KNN	91.300	9.091	91.649	10.091	92.025	10.451	90.412	15.66	91.001	22.091	91.372	29.451
Random Forest	93.31.2	1002.67	94.02	1191.282	94.337	1319.39	91.841	2602.67	91.892	3295.96	91.903	3375.26

each fold contains an equal proportion of samples from each class.

2) COMPARISON WITH OTHER CLASSIFIER

The experiments were conducted to verify the effective-ness of the novel KPCA-GA+SVM model. During the experiment, the model runs for 50 times with various combinations of training and testing set such as 70:30, 80:20, and 90:10. Each set contains both normal and attack class, and randomly records have chooses in phase. We evaluated the proposed model by comparing with PCA+GA+SVM, single SVM, KNN and RF method in terms of accuracy, precession, and recall. The accuracy percentage and total time taken by various classifiers are given in Table 6. The table comprises the result of different SVM variant, KNN, and Random Forest classifier.

From the result set, it can be observed that, dimensional reduction approach can enhance the overall performance and running time of the model in both the data sets. Moreover, the accuracy of the N-KPCA+GA model is 98.907% which is better than the rest of the model. The reason is obvious, employing kernel function to PCA, more number of principal components can be deduced than general PCA, which eventually shows better performance. It can be noted that compared to KPCA+GA+SVM, proposed SVM model is more effective in terms of accuracy and false rate. Single-SVM takes more training time, due to its trial-judging concept. Whereas, the training time of others is in the acceptable range. In terms of testing time KNN takes less time compared to other classifiers. From this analysis it is inferred that more the training/testing data, classifiers takes more time.

3) COMPARISION WITH ATTACK CLASS

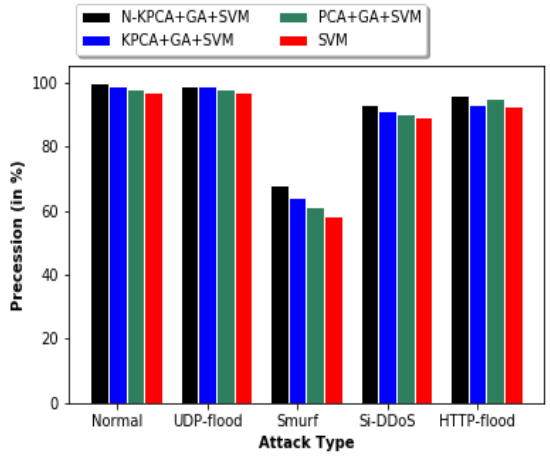
For class wise comparison, the confusion matrix of N-KPCA+GA+SVM has been demonstrated in Table 5. Here, the objective is to observe the classification of normal traffic and other attack traffic. The accuracy measure of training/test data has already shown in Table 7. The confusion matrix contains the result of 10%, 20%, and 30% test data of Data set-I. The dataset contains five different types of traffic such as: Normal, smurf, UDP-flood, Si-DDoS, and HTTP-Flood. From the confusion matrix it is inferred that, using PCA and K-PCA, enhance the accuracy level of SVM than the single SVM which does not follow any feature extraction mechanism.

In another experiment, we measured the precession and recall value of the proposed model with the considered state of art algorithms. As far as the precision and recall value is concerned, all the models achieved higher precision and recall value for both ''normal'' and ''UDP-flood'' class.

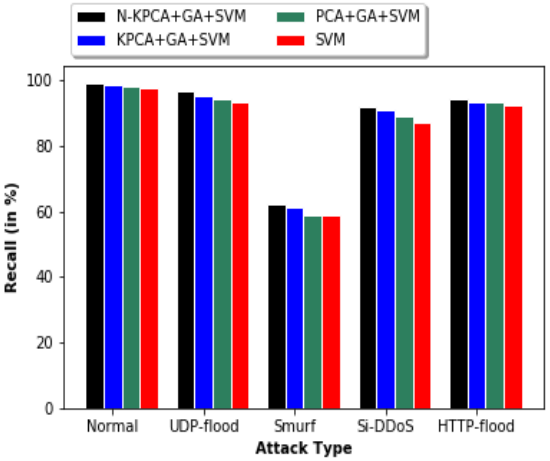
From Figure 5a and Figure 5b, it is also noted that classifying the ''smurf class'' is the most challenging task for all models. However, from the previous work [24], the proposed model improves the ''smurf class'' detection result. In ''smurf class'' a large volume of ICMP echo messages are being forwarded, which is difficult to classify as benign or attack traffic.

4) ON-LINE TESTING

For the on-line testing of the proposed detection model, a tree topology having 15 switches with a POX controller has been created. Then the result outcomes are examined using an analytic tool called sFlow-rt [29]. The proposed ML model has written inside the controller. With a partic-



(a) Precision results of different models



(b) Recall results of different models

FIGURE 5. Precision and Recall result of different SVM model.

ular period, the statistics monitor module collects the traffic statistics from the OpenFlow switch. The pre-trained K-PCA+GA+SVM model observes the traffic patterns of the respective switch and then takes decision about the traffic. If the classifier detects the traffic as malicious, the mitigation process starts. The mitigation module present inside the controller takes the Data Path ID (DPID) of the OF switch on which attack is traced, and further, it sends a flow rule to block the incoming flows for certain time period (for this case 10 seconds). The flows will be blocked which contains the victims' destination IP address. After a pre-specified time period, the flow rule is removed from the flow table.

VII. CONCLUSION AND FUTURE WORK

The paper proposed a novel DDoS detection and mitigation framework for an SDN system. For detection purposes, the multi-layer Support Vector Machine (SVM) has used as the classifier. For better accuracy and to lessen the testing time, KPCA with GA has been employed in this model. The KPCA technique is used to extract the principal features from the DDoS dataset; GA is used for selecting suitable

TABLE 7. Confusion matrix of test data using NKPCA+ GA+SVM.

Actual Class	Classified class				
	HTTP-Flood	Normal	SiDDoS	Smurf	UDP-Flood
10% Test data					
HTTP-Flood	412	7	17	0	0
Normal	7	193493	16	0	0
SiDDoS	0	37	637	0	0
Smurf	3	400	37	811	0
UDP-Flood	0	2050	0	0	19201
20% Test data					
HTTP-Flood	720	8	19	0	0
Normal	9	386951	43	0	0
SiDDoS	0	88	1303	0	0
Smurf	3	1885	61	1602	0
UDP-Flood	0	3929	0	0	36509
30% Test data					
HTTP-Flood	1271	7	42	0	0
Normal	20	580761	47	0	0
SiDDoS	0	104	1932	0	0
Smurf	10	1239	102	2384	0
UDP-Flood	0	5945	0	0	54337

parameters for SVM classifier. Moreover, N-RBF is used to lessen the training period. Furthermore, the experimental outcome exhibits that on DDoS dataset, KPCA performs effectively than PCA. The accuracy of the proposed model is 98.907%, which is better than the rest of the model. By employing kernel function to PCA, more number of principal components can be reduced than general PCA, which eventually shows better performance.

Developing more interesting algorithms that combine kernel functions with some other classification methods is the future scope of this work. Additionally, more focus will be on higher detection accuracy for "smurf class" and "SiDDoS" class traffic in a real SDN testbed. Although the model performs well in detecting the attack traffic in a single controller environment, it may fail to identify the attack traffic in a multi-controller environment. As future work, we can improve our model to determine the attack in such a multi-controller context.

REFERENCES

[1] Y. Zhang, L. Cui, W. Wang, and Y. Zhang, "A survey on software defined networking with multiple controllers," *J. Netw. Comput. Appl.*, vol. 103, pp. 101–118, Feb. 2018.

[2] P. Visu, L. Lakshmanan, V. Murugananthan, and M. V. Cruz, "Software-defined forensic framework for malware disaster management in Internet of Thing devices for extreme surveillance," *Comput. Commun.*, vol. 147, pp. 14–20, Nov. 2019.

[3] E. Molina and E. Jacob, "Software-defined networking in cyber-physical systems: A survey," *Comput. Electr. Eng.*, vol. 66, pp. 407–419, Feb. 2018.

[4] A. Mondal, S. Misra, and I. Maity, "AMOPE: Performance analysis of OpenFlow systems in software-defined networks," *IEEE Syst. J.*, vol. 14, no. 1, pp. 124–131, Mar. 2020.

[5] M. Conti, C. Lal, R. Mohammadi, and U. Rawat, "Lightweight solutions to counter DDoS attacks in software defined networking," *Wireless Netw.*, vol. 25, no. 5, pp. 2751–2768, Jul. 2019.
[6] C. B. Zerbini, L. F. Carvalho, T. Abrao, and M. L. Proença, "Wavelet against random forest for anomaly mitigation in software-defined networking," *Appl. Soft Comput.*, vol. 80, pp. 138–153, Jul. 2019.
[7] 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.
[8] R. T. Kokila, S. T. Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in *Proc. 6th Int. Conf. Adv. Comput. (ICoAC)*, Dec. 2014, pp. 205–210.
[9] 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.
[10] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Berlin, Germany: Springer, 2011, pp. 161–180.
[11] Y. Cui, L. Yan, S. Li, H. Xing, W. Pan, J. Zhu, and X. Zheng, "SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks," *J. Netw. Comput. Appl.*, vol. 68, pp. 65–79, Jun. 2016.
[12] H. Wang, L. Xu, and G. Gu, "FloodGuard: A DoS attack prevention extension in software-defined networks," in *Proc. 45th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2015, pp. 239–250.
[13] L. Schehlmann and H. Baier, "COFFEE: A concept based on OpenFlow to filter and erase events of botnet activity at high-speed nodes," in *Proc. INFORMATIK-Informatik Angepasst Mensch, Organisation Und Umwelt*, 2013, pp. 1–15.
[14] J. Ashraf and S. Latif, "Handling intrusion and DDoS attacks in software defined networks using machine learning techniques," in *Proc. Nat. Softw. Eng. Conf.*, Nov. 2014, pp. 55–60.
[15] F. Keti and S. Askar, "Emulation of software defined networks using mininet in different simulation environments," in *Proc. 6th Int. Conf. Intell. Syst., Modeling Simulation*, Feb. 2015, pp. 205–210.
[16] S. J. Vaughan-Nichols, "OpenFlow: The next generation of the network?" *Computer*, vol. 44, no. 8, pp. 13–15, Aug. 2011.
[17] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2013, pp. 413–424.
[18] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 251–256.
[19] M. Latah and L. Toker, "Towards an efficient anomaly-based intrusion detection for software-defined networks," *IET Netw.*, vol. 7, no. 6, pp. 453–459, Nov. 2018.
[20] N. Dayal and S. Srivastava, "An RBF-PSO based approach for early detection of DDoS attacks in SDN," in *Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2018, pp. 17–24.
[21] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, Oct. 2017.
[22] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Appl. Soft Comput.*, vol. 18, pp. 178–184, May 2014.
[23] S. Srinoy, "Intrusion detection model based on particle swarm optimization and support vector machine," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Apr. 2007, pp. 186–192.
[24] M. Alkasassbeh, G. Al-Naymat, A. Hassanat, and M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 1, pp. 436–445, 2016.
[25] T. Shon, Y. Kim, C. Lee, and J. Moon, "A machine learning framework for network anomaly detection using SVM and GA," in *Proc. 6th Annu. IEEE SMC Inf. Assurance Workshop*, 2005, pp. 176–183.
[26] F. Kuang, W. Xu, S. Zhang, Y. Wang, and K. Liu, "A novel approach of KPCA and SVM for intrusion detection," *J. Comput. Inf. Syst.*, vol. 8, no. 8, pp. 3237–3244, 2012.
[27] L. R. Prete, A. A. Shinoda, C. M. Schweitzer, and R. L. S. de Oliveira, "Simulation in an SDN network scenario using the POX controller," in *Proc. IEEE Colombian Conf. Commun. Comput. (COLCOM)*, Jun. 2014, pp. 1–6.
[28] K. S. Sahoo, A. Iqbal, P. Maiti, and B. Sahoo, "A machine learning approach for predicting DDoS traffic in software defined networks," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2018, pp. 199–203.
[29] *Sflow-RT*. Accessed: Feb. 15, 2020. [Online]. Available: http://sflow-rt.com/index.php
[30] Z. Liu, Y. He, W. Wang, and B. Zhang, "DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN," *China Commun.*, vol. 16, no. 7, pp. 144–155, Jul. 2019.
[31] X. Li, D. Yuan, H. Hu, J. Ran, and S. Li, "DDoS detection in SDN switches using support vector machine classifier," in *Proc. Joint Int. Mech., Electron. Inf. Technol. Conf.* Chongqing, China: Atlantis Press, 2015, pp. 1–5. [Online]. Available: https://www.atlantis-press.com/proceedings/jimet-15
[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, Mar. 2018.
[33] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based DDoS detection system in software-defined networking (SDN)," 2016, *arXiv:1611.07400*. [Online]. Available: http://arxiv.org/abs/1611.07400
[34] M. Alqahtani, A. Gumaei, H. Mathkour, and M. Ben Ismail, "A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks," *Sensors*, vol. 19, no. 20, p. 4383, Oct. 2019.
[35] M. R. Parsaei, S. M. Rostami, and R. Javidan, "A hybrid data mining approach for intrusion detection on imbalanced NSL-KDD dataset," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 6, pp. 20–25, 2016.
[36] 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, Aug. 2019.
[37] 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.
[38] S. Haider, A. Akhunzada, I. Mustafa, T. B. Patel, A. Fernandez, K.-K.-R. Choo, and J. Iqbal, "A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 53972–53983, 2020.
[39] N. Meti, D. G. Narayan, and V. P. Baligar, "Detection of distributed denial of service attacks using machine learning algorithms in software defined networks," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1366–1371.
[40] K. S. Sahoo, P. Mishra, M. Tiwary, S. Ramasubbareddy, B. Balusamy, and A. H. Gandomi, "Improving end-users utility in software-defined wide area network systems," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 696–707, Jun. 2020.
[41] 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.
[42] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for Internet of Things data analysis: A survey," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 161–175, Aug. 2018.
[43] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-Peer Netw. Appl.*, vol. 12, no. 2, pp. 493–501, Mar. 2019.
[44] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions," *Comput. Commun.*, vol. 107, pp. 30–48, Jul. 2017.
[45] P. Dong, X. Du, H. Zhang, and T. Xu, "A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
[46] M. M. Oo, S. Kamolphiwong, T. Kamolphiwong, and S. Vasupongayya, "Advanced support vector machine- (ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN)," *J. Comput. Netw. Commun.*, vol. 2019, pp. 1–12, Mar. 2019.
[47] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," in *Proc. 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Dec. 2014, pp. 1–6.

KSHIRA SAGAR SAHOO (Member, IEEE) received the M.Tech. degree in information and communication technology from IIT Kharagpur, Kharagpur, India, in 2014, and the Ph.D. degree in computer science and engineering from the National Institute of Technology Rourkela, Rourkela, India, in 2019. He is currently working as an Assistant Professor with the Department of IT, VNRVJIET, Hyderabad, India. He has more than 50 research articles in various top international journals and conferences. His research interests include future generation network infrastructure, such as SDN, edge computing, the IoT, and so on. He is a member of IEEE Computer Society and an Associate Member of Institute of Engineers (IE), India.

BATA KRISHNA TRIPATHY received the M.Tech. degree in information and communication technology from IIT Kharagpur, Kharagpur, India, in 2014, and the Ph.D. degree from the School of Electrical, Indian Institute of Technology Bhubaneswar, Bhubaneswar, India, in 2019. His research interest include SDN, the IoT, WSN, edge computing, SD IoT, and so on.

KSHIRASAGAR NAIK (Senior Member, IEEE) is currently a Full Professor with the Department of Electrical and Computer Engineering, University of Waterloo. Previously, he held faculty positions at Carleton University, Ottawa, and the University of Aizu, Japan. He worked as a Software Developer for three years in Wipro, Bangalore – now one of the largest software consultancy companies in the world. He is a coauthor of two widely used textbooks, namely, *Software Testing and Quality Assurance: Theory and Practice* (Wiley, 2008) and *Software Evolution and Maintenance: A Practitioner's Approach* (Wiley, 2014). His research interests include vehicular networks, delay tolerant networks, energy performance testing of mobile applications, detection of anomalous behavior of wireless devices and physical systems, energy harvesting IoT (Internet of Things) devices for sustainable monitoring of physical systems, communication security, and communication protocols for smart power grids. Designing mathematical models and building prototype sensor networks for performing real-life, controlled experiments lie at the core of his research. He has served on the editorial boards of many journals, including the *Journal of Peer-to-Peer Networking and Applications*, the *International Journal of Parallel, Emergent and Distributed Systems*, the *Journal of Circuits, Systems, and Computers*, and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He was a co-guest editor of four special issues of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and the IEEE TRANSACTIONS ON CLOUD COMPUTING.

SOMULA RAMASUBBAREDDY received the master's degree in computer science and engineering, in 2015. He is currently working as an Assistant Professor with the Department of IT, VNRVJIET, Hyderabad, India. His areas of interest are mobile cloud computing and big data analytics.

BALAMURUGAN BALUSAMY received the B.E. degree in computer science and engineering from Bharathidasan University, Tiruchirappalli, India, in 2001, the M.E. degree in computer science and engineering from Anna University, Chennai, India, in 2005, and the Ph.D. degree in computer science and engineering from VIT University, Vellore, India, in 2015. He is currently a Professor with the School of Computing Science and Engineering, Galgotias University, Greater Noida, India. His current research interests include big data, network security, and cloud computing. He is a Pioneer Researcher in the areas of big data and the IoT and has published over 70 articles in various top international journals.

MANJU KHARI received the master's degree in information security from the Ambedkar Institute of Advanced Communication Technology and Research, India, and the Ph.D. degree in computer science and engineering from the National Institute of Technology Patna, Patna. She is currently an Assistant Professor with the Ambedkar Institute of Advanced Communication Technology and Research, under Government of NCT Delhi, affiliated with Guru Gobind Singh Indraprastha University, India. Her research interests include software testing, software quality, software metrics, information security, and nature-inspired algorithms.

DANIEL BURGOS (Senior Member, IEEE) received a postgraduate in artificial intelligence & machine learning from MIT, and the Ph.D. degree in communication, the Dr.Ing. degree in computer science, the Ph.D. degree in education, the Ph.D. degree in anthropology, and the D.B.A. degree in business administration. He is currently as a Full Professor of technologies for education & communication and the Vice-Rector for International Research, the UNESCO Chair of eLearning, and the ICDE Chair of open educational resources with the Universidad Internacional de La Rioja. He is also the Director of the Research Institute for Innovation & Technology in Education (UNIR iTED). Further, he is also a Professor with An-Najah National University, Palestine, an Adjunct Professor with the Universidad Nacional de Colombia (UNAL), Colombia, an Extraordinary Professor with North-West University, South Africa, a Visiting Professor with Coventry University, U.K., and the Universidad de las Fuerzas Armadas (ESPE), Ecuador. He has published over 150 scientific articles, 20 books, and 15 special issues on indexed journals. He has developed +55 European and Worldwide Research and Development projects. His research interests include adaptive, personalised and informal eLearning, learning analytics, open education and open science, eGames, and eLearning specifications.