Microprocessors and Microsystems 79 (2020) 103285

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Flow-based anomaly intrusion detection using machine learning model with software defined networking for OpenFlow network

N. Satheesh [a], M.V. Rathnamma [b], G. Rajeshkumar [c], P. Vidya Sagar [d], Pankaj Dadheech [e], S. R. Dogiwal [f], Priya Velayutham [g], Sudhakar Sengan [h,*]

[a] Department of Computer Science & Engineering, St. Martin's Engineering College, Dhulapally, Secunderabad, India
[b] Department of Computer Science and Engineering, KSRM College of Engineering (A), YSR Kadapa, Andhra Pradesh, India
[c] Department of Information Technology, Karpagam College of Engineering (Autonomous), Coimbatore-641032, Tamil Nadu, India
[d] Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India
[e] Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan (SKIT), Jaipur, Rajasthan-302017, India
[f] Department of Information Technology, Swami Keshvanand Institute of Technology, Management & Gramothan (SKIT), Jaipur, Rajasthan-302017, India
[g] Department of Computer Science & Engineering, Mahendra Institute of Technology, Namakkal-637503, Tamil Nadu, India
[h] Department of Computer Science and Engineering, Sree Sakthi Engineering College, Coimbatore – 641104, Tamil Nadu, India

ARTICLE INFO

Keywords:
Anomaly intrusion detection
Machine learning model
Multi-layer classification
Network traffic
OpenFlow
Packet data flow
QoS
Software defined networking

ABSTRACT

Moving towards recent technologies, Software Defined Networking (SDN) produces a promising network framework to combine the overall network management system with network programming. It gives a more effective tracking system towards the data center. By centralized system and symmetric controller, it prevents security cracks from creating new threats during OpenFlow packet transmission with vulnerabilities. It creates more interest to the researchers to work towards Flow-based SDN for the priority-driven algorithm in anomaly intruder detection. In this paper, we made a study towards a priority-based model using SDN to control the flow of data packets over the network, gives assurance to the bandwidth enforcement, and reallocation is made through virtual circuits. The network behavior of the system is continuously monitored through the machine learning model for normal and abnormal traffic data transmission to detect anomaly intruders. Flow-based machine learning (ML) model with SDN act as an intelligent system to limits the throughput virtually through the flow of reserved bandwidth and make use of extra bandwidth, which presents more than the utilization bandwidth for priority-based applications with minimal cost while compared with the traditional methods. The proposed work also compared with the schemes available at the network to produce outcomes with fast routing and the fault tolerance of existing networks to overcome the gap open at the security of the SDN architecture to detect and identify vulnerabilities.

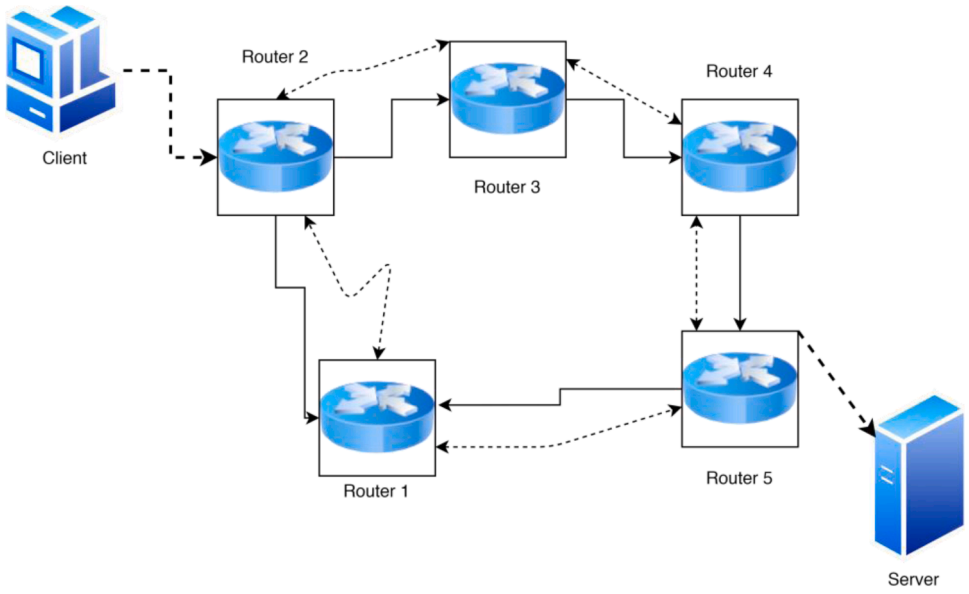
1. Introduction

In place of ossified and opaque network protocols, the Software-Defined Networking (SDN) [1] paradigm proposes that the control plane be cleanly detached through network switches into a centralized server called a controller. The switches simply forward packets in the data plane using controller-send commands. Additionally, the switches can submit controller events concerning the delivery of different packets, flow counters, etc. In response to these events, the controller typically sends commands to switch [2]. This decoupling of the control plane and data plane is the key to produce better management of SDN networks.

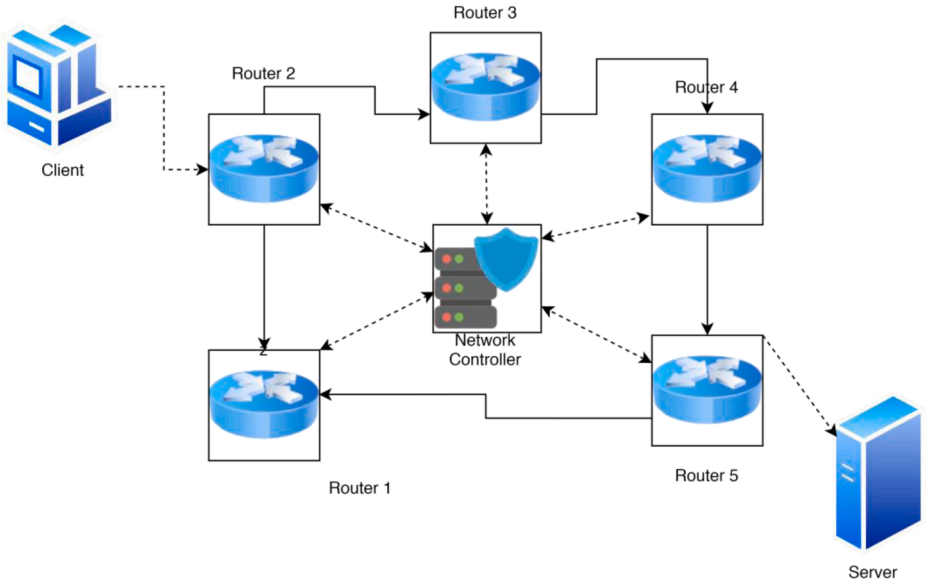
Besides, this platform offers the ability to truly expand data transfer by supplying advanced network management software systems with a consistent specification for external control logic, allowing many different use cases with little effect on any existing technologies. It provides, for example, the feasibility of networking substrates to act dynamically and flexibly over the needs of the storage elements or processing associated with them. This also assures to provide innovative networking support design for campus networks, data centers, and

* Corresponding author.
E-mail addresses: nsatheesh1983@gmail.com (N. Satheesh), rathnamma@ksrmce.ac.in (M.V. Rathnamma), grajesh.grk@gmail.com (G. Rajeshkumar), pvsagar20@gmail.com (P.V. Sagar), pankajdadheech777@gmail.com (P. Dadheech), dogiwal@gmail.com (S.R. Dogiwal), priya.saravanaraja@gmail.com (P. Velayutham), sudhasengan@gmail.com (S. Sengan).

https://doi.org/10.1016/j.micpro.2020.103285
Received 10 August 2020; Received in revised form 11 September 2020; Accepted 24 September 2020
Available online 3 October 2020



(a). Traditional Communication Channel



(b). SDN based Communication Channel

Fig. 1. (a) (b): General framework Traditional and SDN based communicational channel.

possibly carrier networks along to provide dynamic, fast, and application-specific efficient packets forwarding. At the same time, SDN is the primary component of-demand network and intelligent management of existing Open Systems Interconnection (OSI) [3,4] layer protocols on top of the physical substratum.

Fig. 1(a) and (b) represents the general framework traditional and SDN based communicational channel. New Internet protocol (IP) [5,6] networks are highly multi-faceted and more challenging for handling the systems and its trend to be further charged with modern digital technology paradigms like large data systems, virtualized cloud storage, interactive content provision, or data center connectivity. To reverse this situation, dynamic, symmetrical, and manageable network architecture, explicitly software-defined networking (SDN) exemplar, has been proposed as the outcome for building a more reliable network organization for the devices. Thus new policies and their protocols will be carried out through software but without modifying any hardware [7].

Traffic surveillance is an essential service for network management. Attackers always use a botnet or some other malicious software to send large packets that exhaust the resources of the target like bandwidth, memory, CPU, etc. Flow management aims at detecting network traffic issues and minimizing harmful practices like anomaly detection. Deployment of Anomaly Detection Systems (ADS) however, two issues have affected the core network [8,9]: the first one is low detection rates: While some systems can offer high detection rates, they are not usable. In a realistic setting, since they produce many false positives, the second thing is the inability to run ADS algorithms at network core line rates: Packet and Flow sampling is used to minimize this issue, but sampling is performed. It further degrades the accuracy of anomaly detection by distorting essential traffic characteristics.

Detection of irregularities is an essential aspect of traffic management for detecting the suspected disruptive activity in the network [10].

The core of that is the way network traffic is categorized. Machine learning is commonly used for doing this in mainstream networks. These techniques are separated into two primary classes: one is supervised learning, and the other lessons are unsupervised. The supervised learning algorithms, like Supported Vector Machine, Naive Bayes, are suitable for detecting established attacks. Uncontrolled learning algorithms, such as *K*-means, Expectation-Maximization, can be used to identify unknown assaults [11,12]. SDN is a novel area for network management that separates controls and data planes. Whereby the network knowledge and condition can be theoretically clustered, and programs can be abstracted from the actual network technology. What's more, SDN has software-based traffic analysis features, logically unified monitoring, regional network vision, and automated upgrading of forwarding laws, making it easier to spot network traffic anomalies [13].

Subsequently, the anomaly-based method can detect the existing and unexplained attack effectively. When combined with flow-based traffic control, which is usually the anomaly-based intrusion detection method, it would only need to check the packet headers. A clear inference is a need for the flow-based intrusion detection device to manage modest volumes of data. In recent days many research areas available in image processing (IP), information technology (IT), and computer science (CS) like ML, DL, speech recognition (SR), natural language processing (NLP), object, and face detection (FD) are used effectively. Also, many methods of intrusion detection that exploit ML and DL are rapidly attaining high-performance [14,15]. ML strategies are utilized for building NIDS to boost exposure precision and reduce the low false alarm rate. Nowadays, the advanced features except ML and DL methods have been extensively used in tracking anomaly detection [6]. Here, it shows the effects of the flow-based anomaly identification method by using ML and DL methods through SDN as its shows result in flow-based traffic analysis.

Here we provide an analysis of using SDN to track network attacks and argue that SDN is a machine ideal for preventing DDoS assaults, mostly attributable to the usage of standard protocols, utilities, and APIs, thus enabling the implementation of modern alternatives. DDoS [16] and port scan attacks always trigger traffic anomaly on the entire network. The paper proposes a solution focused on flow selection, extraction features, and flow classification to identify DDoS attacks.

The Paper proposes a Flow-based mechanism to obtain information from the network, and uses packet classification information theory, then changes flow rules to block malicious flows. Several factors affect accuracy in the detection of SDN-based intrusion identification methods, which include applicable data set for the volume of data in the dataset, feature selection, with proper ML model superior, learning rate, cross-validation, training time, and so on. Many kinds of literature lack in the identification of all attack groups (DoS, Test, R2L, U2R) within SDN [17]. However, we used both ML and DL approaches for building the network for intrusion identification methods for all kinds of attack categories and calculate the performance by supporting through the NSL-KDD [18] dataset for measuring the comparative weaknesses and strengths of both ways.

This aims to use the controller to gather traffic information to perform lightweight detection. Our method has two contributions compared with anomaly detection in traditional networks and related research:

1 We will not require any additional resources to monitor and measure tools. Some traffic resources, such as network flow, are needed for the identification of traffic anomalies in conventional SDN networks, but we just use the controller to obtain statistical data without raising the burden of network devices.
2 We use some statistical indexes in related papers to describe the traffic anomaly by comparing; our method is more accurate and efficient.
3 Exploration of Service Quality (QoS) [19] in SDN volume, and its extension to the domain of scientific Computing. Our work is one of the first moves in the scientific community to engineering innovations in traffic engineering that SDN architectures render possible.
4 The increasing need for real-time analysis in several science domains that correspondingly increase the demands for both bandwidth and latency guarantees from traffic management frameworks.
5 A comprehensive concept for network management for a typical data center.

The rest of this article is outlined as follows. The design of existing work is studied in a multi-domain with sound is discussed in Section 2. The planned architecture and its capabilities to achieve the required OpenFlow ML-based anomaly intruder detection with SDN are explored in-depth in Section 3. The comprehensive installation of network infrastructure utilizing machine learning models will be seen in Section 4, accompanied by SDN based intrusion detection in Section 5. Next, we show experimentation and results before the conclusion about ML-based SDN will run over and compared with new technologies is compared in Section 6. Finally, we concluded with the OpenFlow ML network-based SDN to produce high security towards intruder detection.

2. Related works

SDN is the network application architecture approach that splits control logic through data plane implementation, or physical devices and operating systems running on those devices. The SDN framework divides the network architecture into three layers: a management layer, a virtualization layer for the network, and a layer for the network operating system (NOS) [20]. At the highest level is the control program that operates on a global network's high-level abstraction, and is essential in driving network policy. Below that is the virtualization layer, where the global network's physical characteristics are reduced to a simplified representation of which the control program may issue commands. The Network OS controls the physical devices which compose the network fabric at the lowest stage. The virtualization layer translates the high-level directives of the control program into a low-level dialog, which the NOS understand.

Classic approaches to traffic engineering include Differentiated Services [21] (or DiffServ), which was initially proposed. As outlined in, a priority is allocated to traffic flows (each packet carries a DSCP code that corresponds to a service class) and sent to a queuing system on the edge router where it reaches the network. Typically, the DiffServ queuing strategy uses a token-bucket algorithm in which overflow packets are dropped, and a weighted round-robin scheduler which controls the subsequent packet transfer. Both are implemented on the switch so that the queuing mechanism enjoys almost instantaneous reaction times to changes in flow rates that cross the switch.

Network security aims to provide the core properties for secured communications, like availability, integrity, and confidentiality [22]. Confidentially, it means, the sense, it seeks to guarantee about the network resources and also the knowledge was available only for authenticated users. In contrast, transparency aims at avoiding unwanted manipulation and deletion of details that traverse the network: accessibility denotes for reliable and assured access by authorized persons to the services and information. A critical technique for applying network protection is cryptography, where the electronic data is secured in such a way that the renders data available for approved entities. Encryption offers confidentiality through preventing unauthorised entities from goes through the messages, where cryptographic checksums or MACs provide authenticity and data integrity. Unfortunately, there is no natural cryptographic approach promising availability, for example, avoiding attacks over Denial of Services (DoS) [23].

An Intrusion Detection System (IDS) [24] actively tracks network traffic flows to recognize any regulation violations and track irregular and unusual network traffic. The detection process may either based on a signature in that packets are associated with a unique name in the

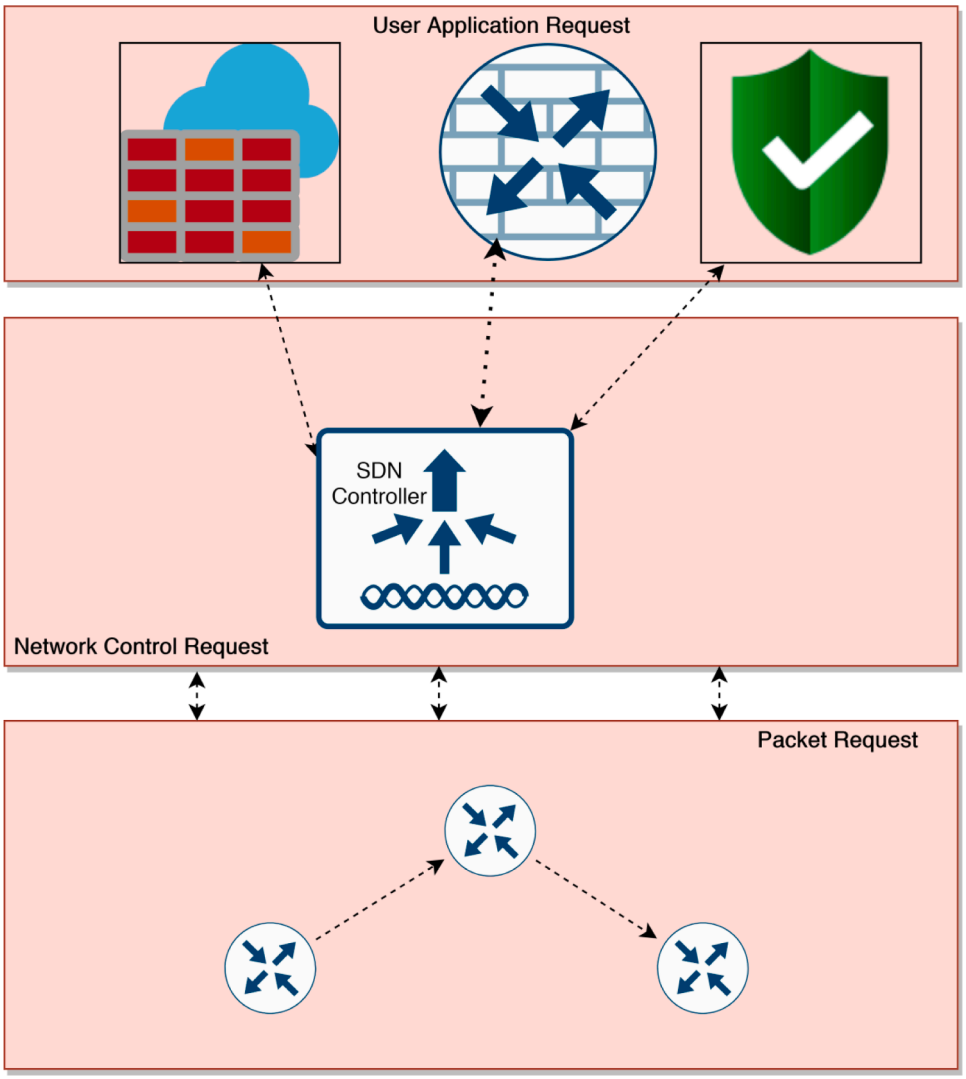


Fig. 2. An underlying SDN architecture.

database, or it will depend on the anomaly where the packets have been compared with a recognized baseline network performance. An ID is also looking towards the packet payload, and it means Deep Packet Inspection (DPI) is opposed to the typical firewalls [25].

Classifications for the detection of flow-based anomalies have been comprehensively investigated in modern society. The Flow-based anomaly identification method can identify harmful and benign flows with high accuracy by utilizing a multi-layer perceptron (MLP) along with the secured layer towards the gravitational search algorithm (GSA) are developed [26]. Here, the authors presented a new idea for inductive NIDS, which utilizes single-class SVM to analyze and train through malicious network data as opposed to other systems that provide fewer false alarms. Nonetheless, numerous researchers have suggested the amount of traffic anomaly identification algorithms by utilizing OF and NOX compatible switches [27]. Thus these methods identify irregularities that are more effective in limited ON, but it doesn't present the ISP. Where, lightweight model for the detection of DDoS attack is performed over the traffic that stream of features, where the data is extracted at low overhead through programmatic interfaces that were provided through the NOX platform. Therefore, the system creates a higher detection rate of using self-organizing maps (SOM) obtained through flow analysis [28, 29].

However, two different methods for feature selection-based intrusion detection systems developed that has been used in recent times to find higher accuracy in intrusion detection using both DL and ML methods. Also, [30] researchers also showed the SDN-based intrusion detection method performance analysis for different ML-based classifiers in addition to various methods for selecting features [31].

The primary real-time applications for OpenFlow from the above network environments are dynamic or constructive switching and routing, multicasting, access control, load balancing, fail-over, and route recovery, QoS policy enforcement, network virtualization, and isolation and monitoring and tooling. Different types of networks may gain more systems, e.g., Controller device monitoring Datacenter and DMZ networks, as well as analysis networks, are praised for capturing the behavior of new flows [32]. This investigated and introduced the conventional DiffServ solution for SDN networks as a plug-in to the Floodlight system. Indeed, most OpenFlow-enabled switches, all the hardware and software switches identified to DiffServ's regular testing team-offer queues. Thus, vulnerability supports Spoofing, manipulation, Repudiation, disclosure of information, denial of service, and Privilege Rising. One may think, for example, Denial of Service (DoS) alternative to an OpenFlow Controller operation, and assess its overall impact Installation. That analysis results in a collection of device components and Pairs in Weakness. At the same time, a multi-layer classifier [33] is highly supportable to identify the vulnerabilities by comparing it with the various existing method to get high accuracy prediction rate in OpenFlow based anomaly detection. The research outlined in the article

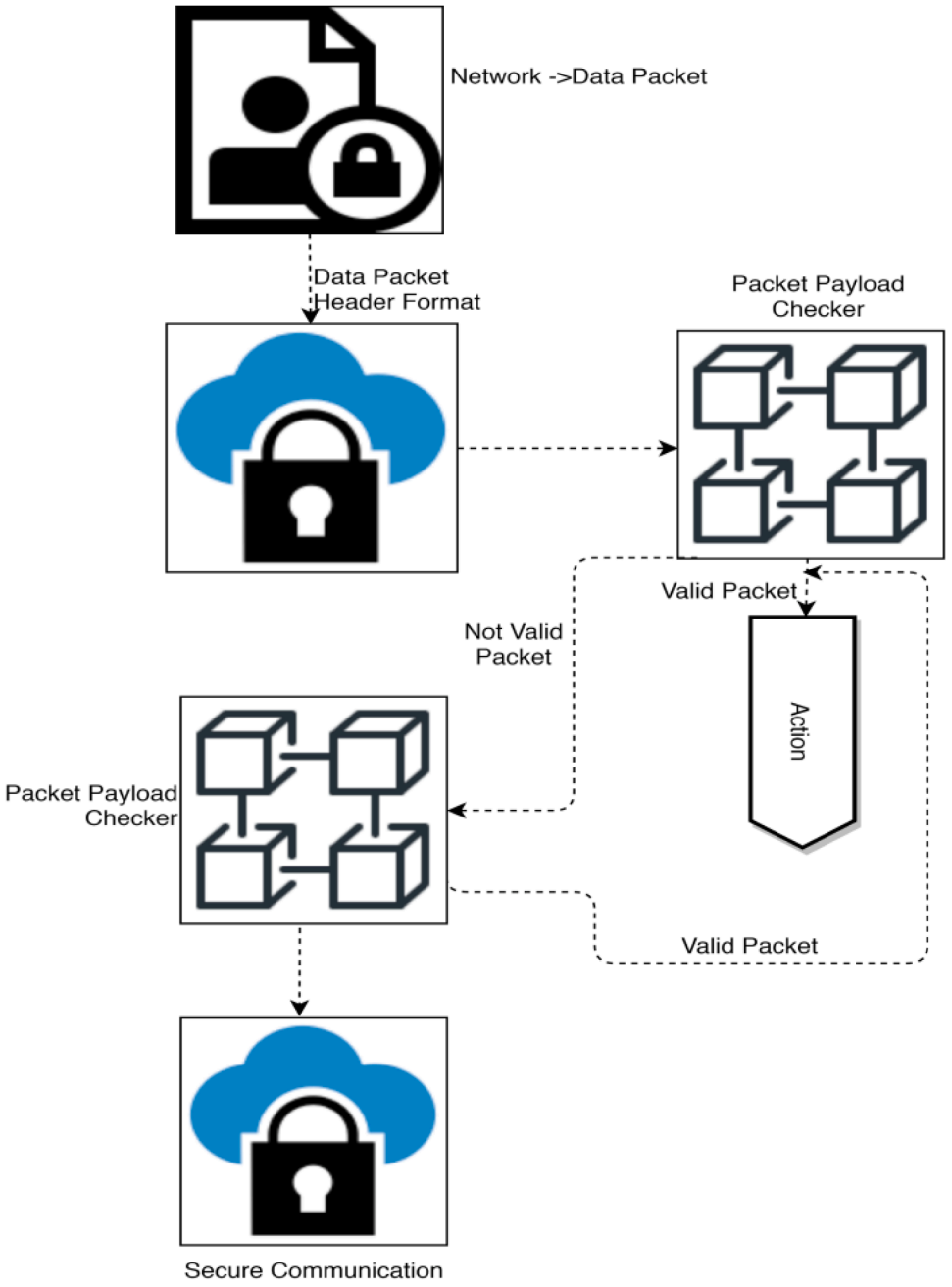


Fig. 3. Packet Flow in an OpenFlow Switch.

introduced a QoS [34,35] implementation on the control layer. Although it presented no advancements that demonstrated any significant benefit over traditional networks to SDN-based traffic engineering, it more than adequately demonstrated how 1) QoS principles could be incorporated into the SDN space, and 2) the northbound API of the controller could be expanded to enable management applications.

3. Proposed methodologies

SDN promotes innovation through the implementation of a centralized, programmable data plane control concept that simplifies the creation of new protocols and network services. The SDN design is made by using the idea by separating the data from control planes (see Fig. 2)

As seen in Fig. 2, each OF activated switches adopts a flow-based decision-making logic provided through the SDN controller [36] that may be responsible for planning each switch's forwarding tables. A pipeline is present inside a typical OF-enabled switch with flow tables consisting of flow accesses comprised of three parts: (a) counters to hold matched flow statistics, (b) matching rules to match incoming packets, and (c) instructions with proactive or reactive configured to be executed on the match. Either software or hardware may implement through forwarding components (i.e., switches which are OF-enabled). Some software switches like Open vs. Switch have to provide data center and virtual network services with great potential. On the other side, specific APIs are introduced for a particular reason (e.g., inter-domain routing and VOIP applications, have not mentioned numerous SDN programming languages like Procera, NetCore, which consist of high-level APIs that may be used for more flexibly and efficiently build different SDN applications [37,38].

Only hosts and switches listed on the controller are allowed to exchange packets in the SDN network. In the Flow Table of the switch, each packet that arrives through OF switch consists of a header that is

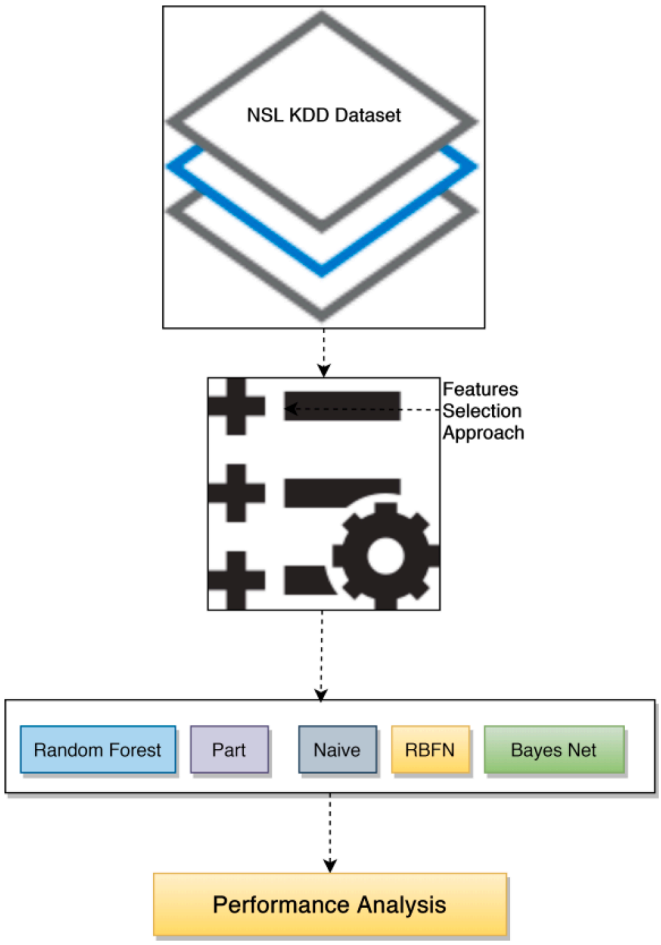


Fig. 4. A multi-layer classification model based on machine learning.

matched towards the flow entries. In this event that some of the flow entry matches the packet header successfully, its statistics have updated (Example, the number of packets and bytes is improved). Otherwise, if they have no flow information (present in the Flow Table of the switch) fits at the header of the packet, instead of this switch must send a message to the controller for requesting the flow of data to process uncertain packets. According to the defined policy, the controller may, in turn, add new flow access of each switch to the Flow Table that is needed for the policy implementation. Thus, traffic may create through all the hosts, which has connected to the existing OF switch, should fill in the switch's Flow Chart. The procedure for processing packets is shown in Fig. 3.

For packet flow, we considered Ethernet source, Ingress port, and destination, VLAN ID, priority, ether type, IP source name, the protocol with ToS, destination, and also considered TCP/UDP host and destination port [39,40].

4. Method of classification using machine learning

The ML-based model is designed and represented here. The architecture of the two-layered hybrid classification is shown in Fig. 4. Due to some influential feature selection, the top layer eliminates an incoherent and unrelated feature, which provides selected functions for the next layer. Later it categorizes the abridged dataset using some productive and symmetrical ML algorithms [41–43]. This model then again trains and test using10fold cross-validation technique. Also, it uses several precise measures to evaluate the performance of the method and is represented in Fig. 4.

According to the basic concept behind much of the ML, the system is that the software train to carry out the function through practicing with an example collection of training data. This training session requires the centralized device and controller framework to conduct specific tasks in which the machine is faced with brand different databases and not encountered before. ML may use a flow-driven anomaly detection framework for creating a predictive technique in automated that is dependent on the training dataset. ML algorithms are needed to address various labeling and forecasting problems [44,45]. Fig. 5 displays a full flow diagram of the anomaly detection system through the OpenFlow controller.

4.1. Collecting flows using Floodlight/OpenFlow

The controller conducts the selection of flow inputs from an OF turn at fixed time intervals (i.e., 5 s). Significant features have been extracted from this collection for classifying traffic as an attack or as standard. As collector collects the samples through all Floodlights authenticated OF switches, and the switch ID is required to support the pinpoint of the classifier where DDoS flooding attacks had been detected at OF switches. Defining the time interval necessary for collecting flow entries is having more importance. After collection gets over during different intervals, later, there may be a delay in detecting an attack and, thus, reduction in time available during possible mitigation. Again, if the group of the time interval is concise, then it will increase in requesting flow packets that result in increases in the overhead for the detection method. Each OpenFlow transfer that is registered with the controller is immediately connected to the detection loop through design. Yet network administrators should limit the sample selection for the virtual switches. As the controller centrally manages the network for switch information, we can monitor all the selected switches and also to analyze their traffic through the DDoS attack point of view. In our method, this analysis is carried out by the controller.

4.2. Selecting features

In this paper, we build five statistical indexes to create a decision tree model by comparing and evaluating it. First, given that N denotes the number of flow inputs in flow array, P_i denotes the number of flow input packets F_i, B_{ij} represents the number of bytes in flow input packet P_i Fi. The meaning of such apps may be dependent on features:

4.2.1. Calculated according to

PAPF (Packet average per Flow entry), EQU. (1) an essential index for tracking network traffic. For example, port scan attacks in DDoS will drastically decrease or increase the average number of packets per Flow entry. In traditional networks, we can directly obtain the value of this index via some network measuring tools, such as NetFlow. But NetFlow is sampling-based; it can't detect the low-frequency attack and will harm device performance [46].

$$PAPF = \sum_{k}^{n} \frac{P_k}{N} \tag{1}$$

ABFE (Average bytes per Flow entry), EQU (2) represents the size of the payload handled in one time period by the flow array. When an attack like DDoS happens, it is often minimal.

$$ABFE = \sum_{k} \sum_{l} \frac{\psi_k^l}{N} \tag{2}$$

PPFE (Percentage of Pair-Flow-Entries), EQU (3) Pair flow-Entries imply two flow-Entries to handle one-way and reverse packets. For example, flow input X = (SIP(source IP) = A, DIP(destination IP) = B) and flow input Y=(SIP = B, DIP = A) are pair flow inputs and PPFE is calculated through:

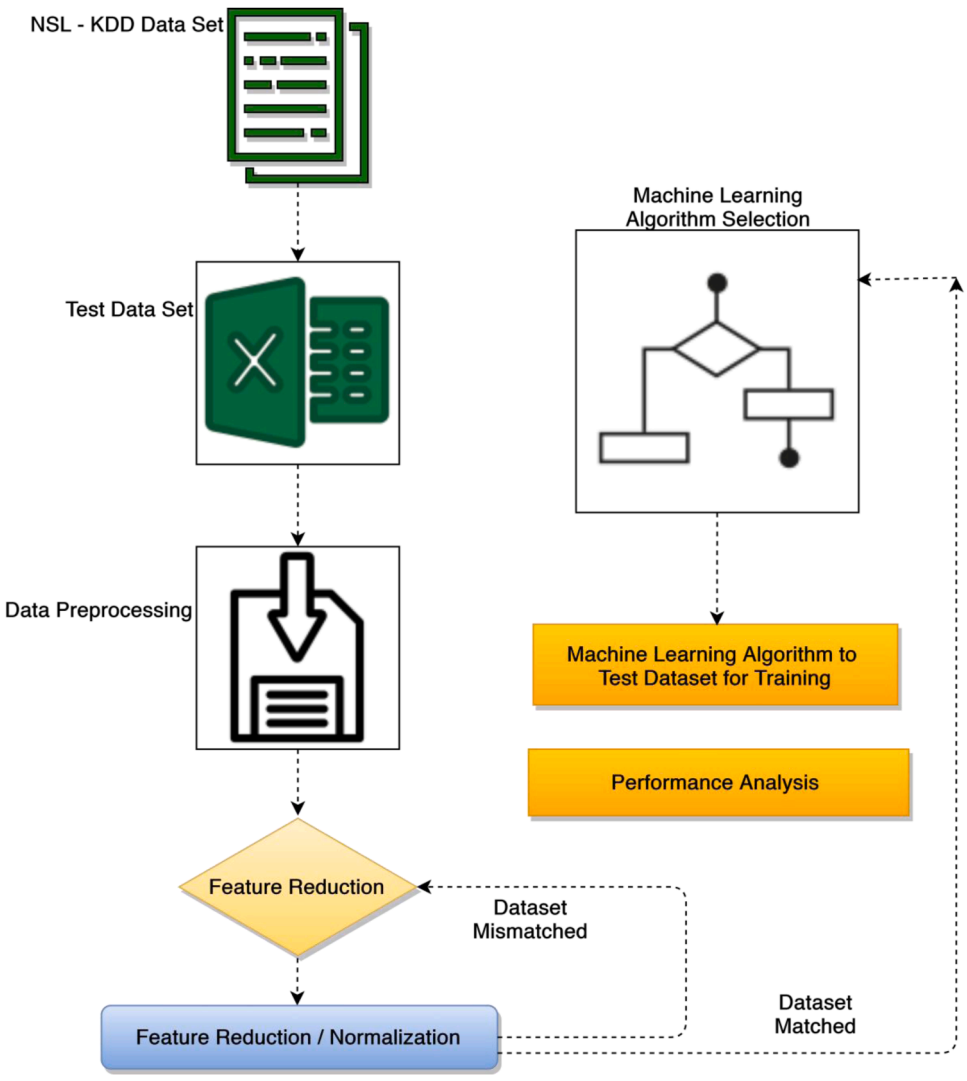


Fig. 5. Flow diagram for a machine learning model to detect an anomaly.

$$PPFE = 2^{*}\frac{FI}{N} \tag{3}$$

FI indicates the number of flow inputs that matched pair Flow input. In the SDN network, PPFE is a special feature. Large non-response packets can contribute to a large rise in PPFE.

SPBF (Source IP number of packets balanced by flow table), EQU (4) represents the variance of source IP number in continuous time unit; this index will quantify the false source IP address in traffic and is prone to a spoofing attack. This is quantifiable as follows:

$$SPBF = \frac{BIPN}{TimeInterval} \tag{4}$$

SIPN is the number of source IPs of packets in the given time interval matched by a single flow table.

ADFE (Average of Duration per Flow entry), EQU (5), Every flow entry has its life-time. In OF switch, if a flow entry in the given time slot doesn't have packets matched, the system will delete it. As mentioned in the paper[4], this function will reduce the number of false positives if there is a limited number of the packets that are swapped between applications.

$$ADFE = \sum_{i=1}^{n} \frac{DFE_i}{N} \tag{5}$$

DFE indicates the duration of the entrance to the i^{th} flow. Abnormal traffic often contributes to ADFE drop.

5. SDN-based anomaly detection framework

The SDN module usually controls OpenFlow transitions of the device component. The SDN controller may request all network data if needed. Then it applied the suggested portion of intrusion detection in the SDN controller for both ML and DL approaches, as outlined in Fig. 6. The following algorithm sums up our proposed solution for the ML-based classification method.

5.1. Algorithm 1: flow-based entry of SDN with ML-based anomaly class detector

Input: Flow-based anomaly class feature extraction from intrusion detector
Output: Identification intruder and allow the standard class
Step 1: START
Step 2: Preprocessing the dataset NSL-KDD
Step 3: Selecting features from the dataset
Step 4: Training the model using the class OpenFlow
Step 5: For check the availability of data to use
Step 6: IF Intruder enters into the class
Step7: Control flow-based ML with SDN model during Flow

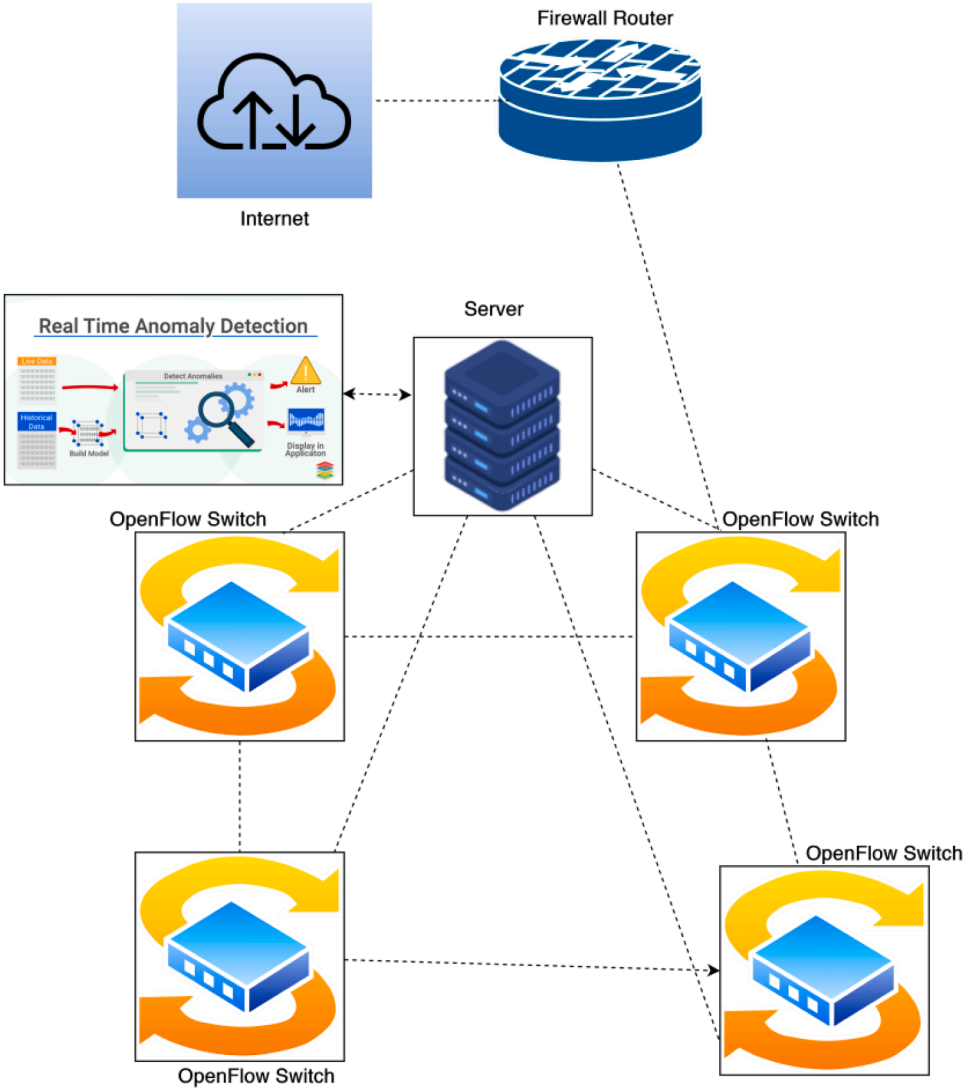


Fig. 6. Flow-based anomaly detection framework over SDN.

Step 8: Update Flow-based entry through SDN with ML model
Step 9: Calculate the type of attack and block the user entry
Step 10: ELSE
Step 11: Permit the entry to use the class through Flow-based SDN-ML
Step 12: END IF
Step 13: END FOR
Step 14: STOP

A request message to OpenFlow stats has been transmitted from the controller to every OpenFlow switches to submit network information. For all available statistics, the OpenFlow reply message, along with every existing data, is again sent back towards the controller through the OpenFlow [47] switch as like controller demand. Fig. 7 noticeably describes how the OpenFlow switch carries the received packet, and how it returns rendering with the presence of data within the flow table, through the OpenFlow protocol. The observable behaviors of the SDN are that the centralized controller may take association opportunities and full Network feedback assessment. Therefore, the OpenFlow protocol will efficiently alleviate the intrusion through the flow table will be adjusted when network anomaly is recognized and discovered. Fig. 8 outlines our proposed approach to the classification method based on transcendent experience.

6. Experimental results

6.1. Test bed

We set up an experimental framework focused on MININET for evaluation of the proposed model, the controller is Floodlight, and the server has 64 GB memory and 32 Core CPU. Next, we construct a network to represent actual traffic on the network, the topology. We employed MININET and OpenVSwitch (OVS) for our experiments. There are three modern SDN controllers and evaluated, for example, Ryu, Floodlight, and ONOS. Within such controls, we used the usual forwarding programs that have simple switch 13 within the Ryu, Floodlight forwarding, and fwd in the ONOS. Fig. 9 represents the architecture of the experimental SDN network topology [48].

The Experiments were executed on a Dell server running on Linux Ubuntu17.04 with the kernel version of 3.16.0 (Power Edge R320, 12-Core Xeon E5-2400 CPU with 32 GB RAM) along with phase (OVS-Switch, Handler, Packet Injection) were allotted for different CPU core for avoiding interference.

6.2. Experimental outcomes of ml approach

We used the WEKA tool and NSL-KDD dataset to carry out experiments. The system contains 6 GB processor, Intel(R) Core(TM) CPU(s)

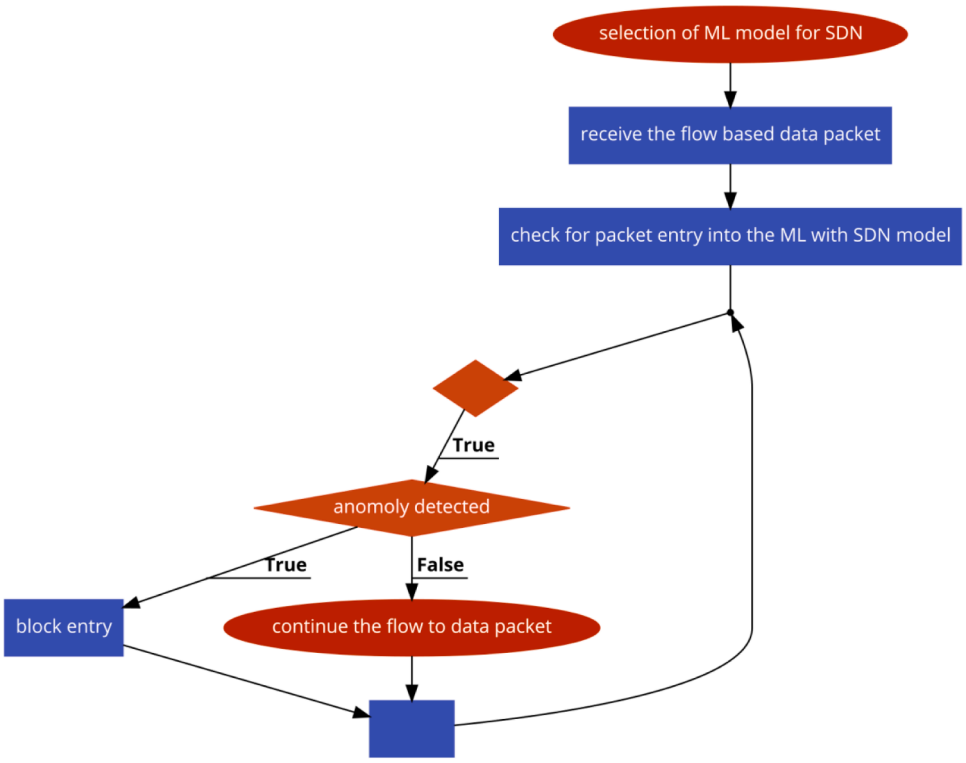


Fig. 7. SDN attacks detecting method based on the ML-based anomaly class detector.

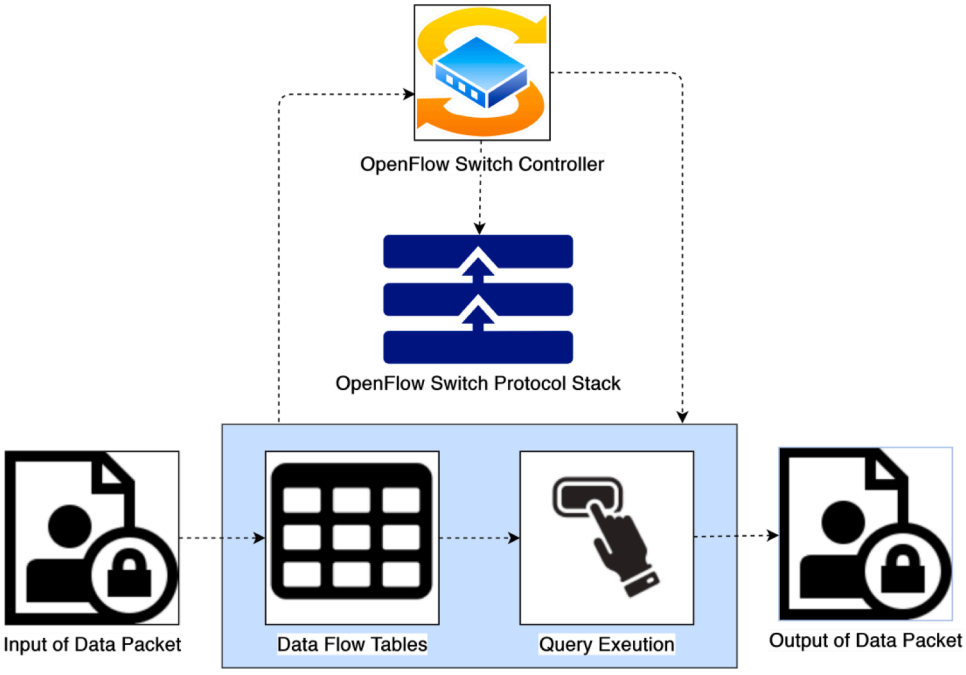


Fig. 8. Flow diagram for managing incoming packets in the OpenFlow switch.

*i*7-2410 M CPU at 2301 MHz, 2.30 GHz, Four Logical, and Dual Core(s) Processors. WEKA has expanded the trouble-free heap size for loading and evaluating the dataset. NSL-KDD datasets are used for training and evaluating each of the 285 functionalities which were selected—this experiment uses a 10-fold cross-validation method for conducting experiments successfully. By dividing the train set into ten subsets, and it will test every subset on the other nine subsets when the model was trained. However, every subset is treated only once, as the test data. Therefore, the procedure repeats ten times. The outcomes of the higher precision classifier are attained through different methods of feature selection had specified in Table 1.

Table 1 shows the results of a whole classifier with various methods of selecting features. In the basis of the study, multiple outcomes of True Positive Rate (TPR), Accuracy (AC), False-Positive Rate (FPR), Recall (R), PRECISION (P), False Alarm Rate (FAR), Mean Absolute Error (MAE), and F-Score are seen. This table utilizes various colors for

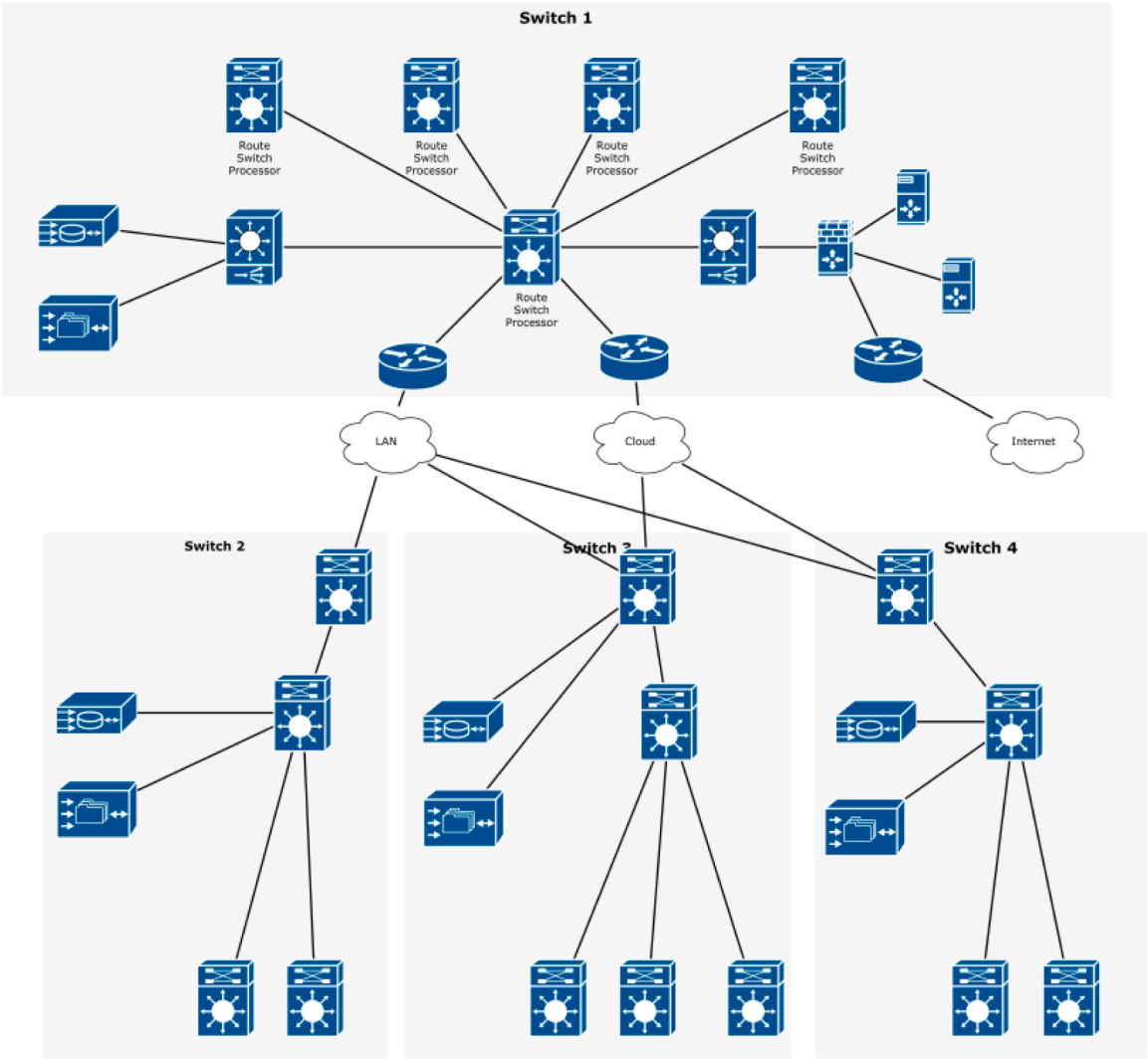


Fig. 9. Experimental SDN network topology.

Table. 1
Flow based ML with SDN classifier shows more accuracy by gain ratio for the feature selection.

Feature Selection Method	Classifiers	Accuracy	Evaluation Precision			F-Score Recall	MAE FAR
Data Gain	RF	79%	0.78	0.72	34%	0.74	0.21
CFS Subset	PART	74%	0.82	0.68	33%	0.76	0.26
Gain Ratio	ML with SDN	82%	0.91	0.79	20%	0.82	0.23

variations to consider the maximum accuracy of common classifier strategies through multiple models for choosing items. To compare, the list of apps reveals just the best scores received by various classifiers. Each colored classifier technique shows the best output consistency for every single feature selection process. By using this higher precision and the lower false alarm rate, the subsequent colored relationship of IG-RF, GR-RF, CST-RF, CFS-PART, and SU-RF indicates the best classifier technique. However, the initial goal is to attain high accuracy, MCC, and recall with low false alarms frequency. It is successfully proficient in the model. By using this processing data, RF with the gain ratio feature selection technique that displays 79.91% as the highest accuracy, which is shown from experimental results. In Figs. 10 and 11, we have plotted the accuracy performance of ML-based models. Upon evaluating these findings, it found that with the RF classifier classifies, the data benefit function selection method provided a higher accuracy of 74.16% while with CFS subset evaluator PART showed an accuracy of 79.19%. Nevertheless, for RF and PART, respectively, the majority of the function selection approaches display an accuracy value of more than 80%. Among them all, the highest accuracy of nearly 82.28% was generated by RF with a gain-ratio selection system.

6.3. *Overlay assessment for end-to-end QoS*

The outcomes have depends on simple testing of the overlay link being implemented and confirmation of the bandwidth limits concerned. Using the MININET v. 2.0.0 framework, the SLA4SDN concept and dynamically performed QoS constraints involved have been tested on OVS dependent OpenFlow network. Where the emulated MININET network's overall performance capacities are explicitly connected to underlying device hardware, the presents the performance

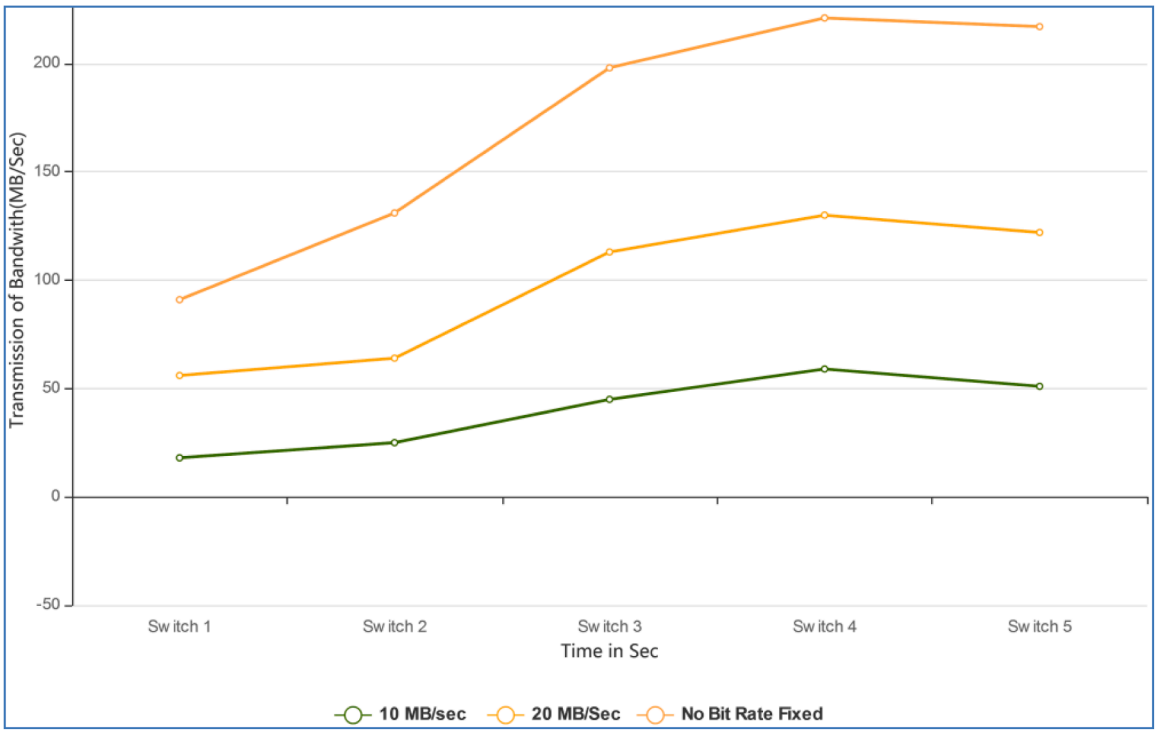


Fig. 10. Measurement of throughput - Queues are configured into two flows.

Fig. 11. Measurement of Throughput - Queues are configured for every flow.

measurements had processed in the laptop should well be equipped along with AMD E-350 1.6 GHz CPU, and UBUNTU Linux MININET VM was exclusively assigned one of the two physical available cores using Virtual Boxlike hypervisor. As the emulated network, it consists of six hosts, who are written off along with SOR1 to SOR6 and two more switches, namely SWT1 and SWT2. Let the hosts with three hosts will be aggregated within two groups, and it is connected to aggregate the switch over either side. The aggregation switches have been linked directly with each other, where evaluation has focused on the shared connection behavior among switches and their ability to deliver specified throughput.

This is processed within two circumstances in two flows SOR-1, SOR-2 and SOR-3, SOR-4, along with the narrow-band of Throughput (T), which offers pretend through frontend, and it is imposed with related

queue configuration, as defined in EQU (6) (7). Where the third Flow (HO5 HO6) was utilized for emulating background traffic with maximum capacity utilization of the network. In the first example, flow-entries process the context of traffic using the standard "OUTPUT" operation of OpenFlow. In comparison, in the second case, the context traffic is often routed through an OpenFlow "ENQUEUE" dependent upon an external queue. This third queue, as described in EQU (7), with generously configured the upper and lower rate limiter, is required instead of the default queue and proceeds through its function.

$$SOR_1 \leftrightarrow SOR_2 = T^{Minimun=16mbps}_{Maximum=20mbps} \quad (6)$$

$$SOR_3 \leftrightarrow SOR_4 = T^{Minimun=30mbps}_{Maximum=38mbps} \quad (7)$$

These three links are collectively calculated to ensure that the required bandwidth is not contravened. However, for each joining connection, the transmission is in progress with offset for exploring effects, and it influences in an overlay with shared the part with aggregation. Where bandwidth among each host-pair is thus evaluated through TCP mode with IPERF. And as stated earlier, two connections at first that have been deployed at a specific bandwidth rate, where the third one has utilized for traffic occurs at the background, is stimulated, and it uses the network that too near extreme capacity. It ensures the third transmission will be affected when the other communications start without an accurate guarantee of the throughput. Thus all the samples are taken in 5 min. Bandwidth fixed streams are paired with a difference of one to two minutes for a minimum period of two minutes of transmission.

The findings of flow analyses are seen in Fig. 10 and Fig. 11. For the first scenario, Fig. 10represents the evaluated throughput values for all three connections. In this situation, two queues, along with rate limiters, have been utilized for implementing QoS to relate SOR-1 SOR-2 (Orange) and SOR-3 SOR-4 (Yellow). Once again, the last link IS between SOR-5 SOR-6 (Green), by using best-effort forwarding, simulates the previously stated background traffic. This transfer (SOR-5, SOR-6) uses the network with the highest bandwidth available, which is about 120 Mbps. When second transmission (SOR-1, SOR-2) begins after 60 s., the performance of the best-effort link automatically decreases. When the third transmission (SOR-3 SOR-4) starts, the sample shows similar behavior. The second and third transmission QoS bandwidth criteria are reached throughout the entire sampling period.

Conversely, the OVS automatically adjusts the best-effort traffic to confirm allocated bandwidth through the other transmissions. Where second setup tests display some activity along with a significantly smaller variance as opposed to the first scenario's best-effort baseline traffic measurements. Nonetheless, the samples indicate that both scenarios meet the requested bandwidth for the transmissions.

$$SOR_5 \leftrightarrow SOR_6 = T^{Minimun=30mbps}_{Maximum=100mbps} \quad (8)$$

The simulated network comprises 5 OpenFlow switches (SW1-SW5) and 42 hosts (SOR- 1, SOR-2), where SOR-1, SOR-2, serves other hosts as servers. To simulate normal network activity and generate regular TCP traffic, UDP traffic, and ICMP traffic, hosts can randomly access server's SOR 1, SOR 2. Hosts may also use the hping3 tool to initiate server SYN Flood, UDP Flood, and ICMP Flood attacks to simulate abnormal network traffic.

To check the effectiveness of the five indexes above, we collect data in our virtual network to measure the importance of increasing the time. The table indicates the pattern of these five indexes. From these graphs, we can see that these indexes are sensitive to abnormal SYN flood-triggered network traffic, UDP flooding, and some other types of DDoS attacks. OpenFlow has several forms of communication; here, we use controller-to-switch connections primarily. The controller can send OpenFlow read-state messages that are collected for statistics with the flow tables, individual, and ports flow entries, e.g., OFPST FLOW stats requests can be used to obtain own Flow Statistics.

Table. 2
Trend of PAPF, ABFE, PPFE, SPBF, and ADEF in the ordinary and abnormal time in Sec.

Time(s)	PAPF	ABFE	PPFE	SPBF	ADEF
0	1.2	02	1.0	00	0.3
10	1.9	30	0.5	00	3.0
20	1.0	100	0.4	160	0.5
30	0.4	250	1.5	70	2.6
40	0.2	50	0.1	110	1.9
50	0.0	20	0.0	130	2.0

Table. 3
Train and test data set.

	Train	Validate	Sum
Normal	1021	350	1371
Abnormal	622	233	855
Total	1643	583	2226

Table. 4
Experimentation Outcomes.

Model	DR%	FR%	Time Taken(*ms*)
FB ML-SDN	95.16	2.49	0.181
K-NN	85.48	2.77	2.489
SVM	85.16	3.32	7.465
NB	89.03	1.39	0.206

When seen in Table 2, the importance of the five mathematical indexes varies dramatically as the assault begins at 25 s. The above experiment verifies that the indexes selected are useful. To check the discrepancy between the decision tree algorithm and some other machine learning algorithms, we collect 1677 flow entries for training and testing over a regular and abnormal time.

Decision tree, support vector machine, *k*-nearest neighbor, and Naive Bayes algorithms are utilized for training and check data sets. Table 3 displays the identification distance, false alarm rate, and test period of the four algorithms. We can see from the experimental results that the decision tree has the highest detection rate of 95.16%, and the false alarm rate is 2.49%, but it also has the lowest test time of 0.181 *ms*. Although Naive Bayes has the lowest false positive rate of 1.39%, its detection rate is dramatically lower than that of the decision tree. *K*-nearest neighbor's identification rate and false warning rate and help vector machine are both naturally lower than the decision tree, too. Having contrasted the classification effects of the four algorithms, we can see that the decision tree is the right one to identify suspicious network traffic. Table 4 gives the number of data sets utilized for processing Flow-based ML with SDN model.

6.4. QoS analysis

Fig. 12 shows the gains as a function of the unreserved (BE) percentage of QoS bandwidth against the Queues Only method; synthetic traffic gains show an exponential increase, evident in the worst-case comparison (Queues Only) and becoming significantly more pronounced for other ways. In the worst case, taken from the randomized outcomes of the 20% Best Attempt, the QoS algorithm indicates an advantage of 4.38% over DiffServ queues. In the best case, the QoS algorithm yields a 20.34% gain over DiffServ queues, taken from the simulated tests of the 34.2% Maximum Attempt. The situation differs when all flows are considered to have total average throughput; the QoS algorithm performs consistently worse than all other competing methods.

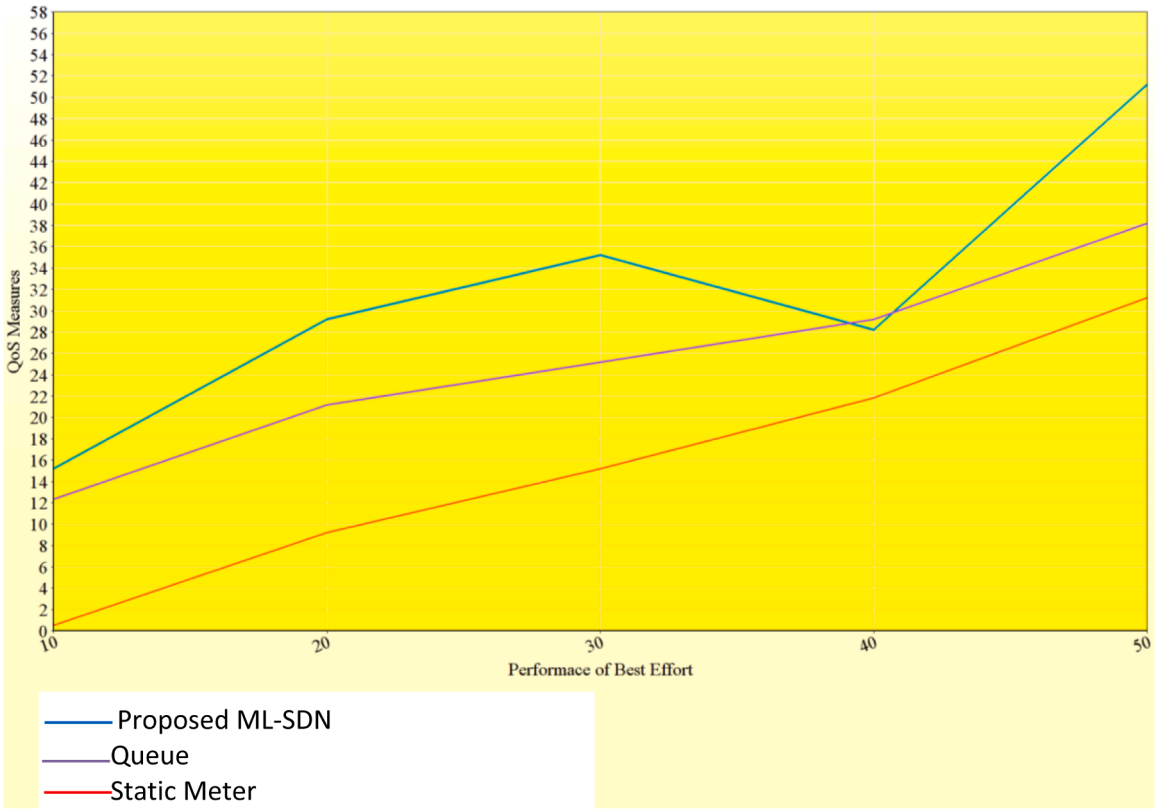


Fig. 12. Performance of QoS rate vs. Best effort queues.

7. Conclusion and future work

SDN is a significant emerging networking model with powerful potential effects and is quickly gaining acceptance. Network protection is utterly crucial, considering the growing amount of cyber threats and their complexity. Although many studies have been done investigating SDN to incorporate a variety of protection enhancements and capabilities, a small amount of work has been done examining the protection of the SDN framework itself. With its theoretically centralized control system, SDN offers a relatively specific approach to network management. As a consequence, SDN defense is probably somewhat different from that of conventional networks, with entirely new vectors of attack.

SDN is more susceptible to threats than conventional networks concerning a unified design. In this paper, we present a lightweight traffic anomaly intruder detection approach based on the decision tree in the SDN network; it can detect the anomaly induced by high-efficiency network assaults. SDN is still in the early stages, as very few people are being educated. It will become much more common as more and more workers are being trained about it. As one of the technologies to be developed for networking power, SDN's future depends on its current situation. SDN is still an evolving technology, and it's got to resolve the hurdles, so someone doesn't hack it. We'll be running it in an existing network in the future and train the model with large-scale traffic info. It can be seen in the control and analysis of complex networks. In this research, we have introduced new methods in software-defined networking to forecast the flow-based anomaly. The machine learning-based Random Forest (RF) model was considered to detect network interference within SDN. Although approaches yield limited experimental effects relative to other works, methods have rendered some vital contribution to SDN usage in the area of intrusion detection.

It is the evidence towards the observational data, and now the ML approach creates slightly better performance than the ML approach. Thus it is crucial to use the ML-SDN model designed for the flow-based anomaly identification to speed up the intrusion detection process and reach optimal accuracy in SDN. However, the proposed method is to improve network traffic in a real SDN environment in the future. The proposed QoS forward approach is to employ global for end-to-end overlay link among hosts that are used for enqueue mechanism for creating a network behavior which is similar to Ethernet in real-time.

Declaration of Competing Interest

There is 'No Conflict of Interest' for submitting this article.

References

[1] M. Kobayashi, S. Seetharaman, G. Parulkar, Maturing of OpenFlow and software-defined networking through deployments, Computer Netw. (61) (2014) 151–175.
[2] Tang, T., Zaidi, S.A.R., McLernon, D., Mhamdi, L., Ghogho, M., Deep recurrent neural network for intrusion detection in SDN-based networks. In 2018 IEEE International Conference on Network Softwarization (NetSoft 2018), Montreal, Canada, Jun 2018.
[3] Z. Fan, Y. Xiao, A. Nayak, C. Tan, An improved network security situation assessment approach in software-defined networks, Peer-to-Peer Netw. Appl. (2017).
[4] Q. Niyaz, W. Sun, A.Y. Javaid, A deep learning-based DDoS detection system in software-defined networking (SDN), EAI Endorsed Trans. Secur. Saf. 4 (12) (2017) 1–12.
[5] R. Wallner, R. Cannistra, An SDN approach: quality of service using big switches floodlight open-source controller, Proc. Asia-Pac. Adv. Netw. 35 (2013) 14–19.
[6] J.H. Cox, J. Chung, S. Donovan, et al., Advancing software-defined networks: a survey, IEEE Access (2017) 1.
[7] A.I. Moustapha, R.R. Selmic, Wireless sensor network modeling using modified recurrent neural networks: application to fault detection, IEEE Trans. Instrum. Meas. 57 (5) (2008) 981–988.
[8] OpenFlow.org, "Openflow switch specification v1.0.0", 2009. Available: http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf.
[9] S.T.V. Pasca, S.S.P. Kodali, K. Kataoka, AMPS: application-aware multipath flow routing using machine learning in SDN, in: Proc. of IEEE Twenty-third National Conference on Communications (NCC), Chennai, India, 2017, pp. 1–6. Mar.
[10] I. Foster, A. Roy, V. Sander, A quality of service architecture that combines resource reservation and application adaptation, in: Quality of Service, 2000. IWQOS. 2000 Eighth International Workshop on, IWQOS2000, (Pittsburgh, PA, USA), IEEE, 2000, pp. 181–188.

[11] S. Sengan, L Arokia Jesu Prabhu, V Ramachandran, V Priya, L. Ravi, V Subramaniyaswamy, Images super-resolution by optimal deep AlexNet architecture for medical application: a novel DOCALN, IoS Press, J. Intell. Fuzzy Syst. (2020) 1–14, https://doi.org/10.3233/JIFS-189146.
[12] V. Vijaya Kumar, M. Devi, P. Vishnu Raja, P. Kanmani, V. Priya, S. Sudhakar, K. Sujatha, Design of peer-to-peer protocol with sensible and secure IoT communication for future internet architecture, in: Microprocessors and Microsystems, 78, Elsevier, 2020, https://doi.org/10.1016/j.micpro.2020.103216. October.
[13] F. Hu, Q. Hao, K Bao, A survey on software-defined network and OpenFlow: from concept to implementation, in: Communications Surveys & Tutorials, 16, IEEE, 2014, pp. 2181–2206.
[14] Sudhakar Sengan, Chenthur Pandian S, A trust and co-operative nodes with affects of malicious attacks and measure the performance degradation on geographic aided routing in mobile ad hoc network, Life Sci. J. 10 (2013) 158–163, 4s.
[15] S. Sengan, Chenthur Pandian S., An efficient agent-based intrusion detection system for detecting malicious nodes in MANET routing, Int.Rev. Comput. Softw. (I.RE.CO.S.) Vol. 7 (6) (2012) 3037–3304.
[16] A. Sivaraman, M. Budiu, A. Cheung, C. Kim, S. Licking, G. Varghese, H. Balakrishnan, M. Alizadeh, N. McKeown, Packet transactions: a programming model for data-plane algorithms at hardware speed, CoRR (2015) abs/1512.05023.
[17] D. Zats, T. Das, P. Mohan, D. Borthakur, R. Katz, Detail: reducing the flow completion time tail in datacenter networks. SIGCOMM, ACM, 2012, pp. 139–150.
[18] V.K. Veerabathiran, D. Mani, S. Kuppusamy, B. Subramaniam, P. Velayutham, S. Sengan, S. Krishnamoorthy, Improving secured ID-based authentication for cloud computing through novel hybrid fuzzy-based homomorphic proxy re-encryption. Soft Computing, Springer, 2020, https://doi.org/10.1007/s00500-020-05119-9.
[19] M. Karthikeyan, K. Sharmilee, P.M. Balasubramaniam, N.B. Prakash, M. Rajesh Babu, V. Subramaniyaswamy, Sudhakar Sengan, Design and implementation of ANN-based SAPF approach for current harmonics mitigation in industrial power systems, Microprocess. Microsyst. (2020), https://doi.org/10.1016/j.micpro.2020.103194.
[20] S. Sengan, V. Subramaniyaswamy, S.K. Nair, V. Indragandhi, J. Manikandan, L. Ravi, Enhancing cyber-physical systems with hybrid smart city cyber security architecture for secure public data-smart network, Future Gener. Computer Syst. (2020), https://doi.org/10.1016/j.future.2020.06.028.
[21] N. Sultana, N. Chilamkurti, W. Peng, R. Alhadad, Survey on SDN based network intrusion detection system using machine learning approaches, Peer-to-Peer Network. Appl. 12 (2019) 493.
[22] Niyaz, Q.; Sun, W.; Javaid, A.Y. A deep learning-based DDoS detection system in software-defined networking (SDN). arXiv 2016, arXiv:1611.07400.
[23] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, Deep recurrent neural network for intrusion detection in SDN-based networks. Proceedings of the 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, 2018, pp. 202–206, 25–29 June.
[24] Elsayed, M.S.; Le-Khac, N.A.; Dev, S.; Jurcut, A.D. Machine-learning techniques for detecting attacks in SDN. arXiv 2019, arXiv:1910.00817.
[25] S. Sengan, C.P. S, Trustworthy position based routing to mitigate against the malicious attacks to signifies secured data packet using geographic routing protocol in MANET, WSEAS Trans. Commun. Vol.12 (11) (2013) 584–2013.
[26] S. Sengan, C.P. S, 'Secure packet encryption and key exchange system in mobile ad hoc network, J. Computer Sci. 6 (2012) 908–912.
[27] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: Proceedings of the Fourteenth International Conference on Machine Learning, (ICML '97), Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1997, pp. 412–420. Nashville, TN, USA, 8 July.
[28] K. Ganesh Kumar, S. Sengan, Improved network traffic by attacking denial of service to protect resource using Z-Test Based 4-Tier Geomark Traceback (Z4TGT), Wirel. Pers. Commun. (2020), https://doi.org/10.1007/s11277-020-07546-1.
[29] Y. Zhai, H. Xu, H. Wang, Z. Meng, H. Huang, Joint routing and sketch configuration in software-defined networking, IEEE/ACM Trans. Network. (2020) 1–14, https://doi.org/10.1109/TNET.2020.3002783.
[30] R.D. Corin, M. Gerola, R. Riggio, F. De Pellegrini, E. Salvadori, Vertigo: network virtualization and beyond. Software Defined Networking (EWSDN), 2012, pp. 24–29, 2012 European Workshop onpagesOct.
[31] E Punarselvam, M. Yacin Sikkandar, M. Bakouri, N.B Prakash, T Jayasankar, S Sudhakar, Different loading condition and angle measurement of human lumbar spine MRI image using ANSYS, Springer, J. Ambient Intell. Humaniz. Comput. (2020) 11, https://doi.org/10.1007/s12652-020-01939-7.
[32] R Vasanthi, R Jayavadivel, K Prasadh, J Vellingiri, G Akilarasu, S Sudhakar, P. M Balasubramaniam, A Novel User Interaction Middleware Component System For Ubiquitous Soft Computing Environment By Using Fuzzy Agent Computing System, Springer, 2020, https://doi.org/10.1007/s12652-020-01893-4. Journal of Ambient Intelligence and Humanized Computing.
[33] S Sudhakar, V Vijayakumar, C SathiyaKumar, V Priya, L. Ravi, V Subramaniyaswamy, Unmanned aerial vehicle (UAV) based forest fire detection and monitoring for reducing false alarms in forest-fires, Elsevier-Computer Commun. 149 (2020) 1–16, https://doi.org/10.1016/j.comcom.2019.10.007.
[34] S. Sengan, S. Chenthur Pandian, Hybrid cluster based geographical routing protocol to mitigate malicious nodes in mobile ad hoc network, InderScience 21 (4) (2016) 224–236.
[35] M. Koerner, H. Almus, H. Woesner, T. Jungel, Metrics and measurement tools in open and the of Elia testbed. Lecture Notes in Computer Science (LNCS) 7586, 2013, pp. 123–134. Heidelberg.
[36] Y. Qian, W. You, K. Qian, FlowVisor vulnerability analysis. Proc. of IFIP Symposium on Integrated Network and Service Management (IM), IEEE, 2017, pp. 867–868.
[37] H. Yamanaka, E. Kawai, S. Shimojo, AutoVFlow: virtualization of large-scale wide-area OpenFlow networks, Comput. Commun. 102 (2017) 28–46.
[38] X. Pan, S. Tang, S. Liu, J. Kong, X. Zhang, D. Hu, J. Qi, Z. Zhu, Privacy-preserving multilayer in-band network telemetry and data analytics: for safety, please do not report plaintext data, J. Lightwave Technol. IEEE Access (2020), https://doi.org/10.1109/JLT.2020.3007491.
[39] 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," 2016 International Conference On Wireless Networks and Mobile Communications (WINCOM), IEEE, pp. 1–6, 2016.
[40] N. Sophakan, C. Sathitwiriyawong, A secured OpenFlow-based software-defined networking using dynamic Bayesian network, in: 2019 19th International Conference on Control, Automation and Systems (ICCAS), IEEE, 2019. Oct.
[41] P.M Balasubramaniam, S Sudhakar, Sujatha Krishnamoorthy, V.P Sriram, S Dhanaraj, V Subramaniyaswamy, Investigations and strategy of intelligent controller (ACBIC) for DC Link control in SAPF system for industrial power systems a control strategy", J. Discret. Math. Sci. Cryptogr. (2020) https://doi.org/10.1080/09720529.2019.1668145. Taylor & Francis.
[42] P.M. Balasubramaniam, S. Sudhakar, S. Krishnamoorthy, V.P. Sriram, S. Dhanaraj, V. Subramaniyaswamy, T. Rajesh, An efficient control strategy of shunt active power filter for asymmetrical load condition using time domain approach, J. Discret. Math. Sci. Cryptogr. (2020), https://doi.org/10.1080/09720529.2019.1668136. Taylor & Francis.
[43] S. Sengan, S. Chenthur Pandian, Investigation of attribute aided data aggregation over dynamic routing in wireless sensor, J. Eng. Sci. Technol. 10 (11) (2015). School of Engineering, Taylor's University1465 - 1476ISSN: 1823-4690.
[44] A.U. Priyadarshni, S. Sudhakar, Cluster based certificate revocation by cluster head in mobile ad-hoc network, Int. J. Appl. Eng. Res. 10 (20) (2015) 16014–16018. ISSN 0973-4562 Vol.
[45] S. Sengan, S. Chenthur Pandian, Authorized node detection and accuracy in position-based information for MANET, Eur. J. Sci. Res. 70 (2) (2012) 253–265.
[46] N. Satheesh, D. Sudha, D. Suganthi, S. Sudhakar, S. Dhanaraj, V.P. Sriram, V. Priya, Certain improvements to Location aided packet marking and DDoS attacks in internet, J. Eng. Sci. Technol. 15 (1) (2020) 94–107. School of Engineering, Taylor's University.
[47] C. Sathiya Kumar, V. Priya, V.P. Sriram, K. Sankar Ganesh, G. Murugan, M. Devi, S. Sudhakar, An efficient algorithm for quantum key distribution with secure communication, J. Eng. Sci. Technol. 15 (1) (2020) 77–93. School of Engineering, Taylor's University.
[48] J. Gopal, J. Vellingiri, J. Gitanjali, K. Arivuselvan, S. Sudhakar, An improved trusted on-demand multicast routing with QoS for wireless networks, Int. J. Adv. Trends Comput. Sci. Eng. Vol.9 (1) (2020) 261–265, https://doi.org/10.30534/ijatcse/2020/39912020. January–February.

Dr. N. Satheesh is working as a Professor, Department of Computer Science & Engineering in St. Martin's Engineering College, Dhulapally, Secunderabad. Since May 2019. He obtained B.E., in Electronics & Communication Engineering from Sri Balaji Chockalingam Engineering College, Arani, University of Madras in 2004, M.E., in Computer Science & Engineering from Faculty of Engineering & Technology, Annamalai University, Chidambaram in 2008 and Ph.D. in Computer Science & Engineering from Karpagam Academy of Higher Education, Karpagam University in 2018. His area of specialization is Wireless Security, Wireless Sensor Networks, Internet of Things. He has 12+ years of teaching experience and 2+ years of software experience. He has 12: International Journal Publications and 06: International Conference Publications, 06: Patent Publications.

Dr.M.V.Rathnamma obtained his Master's Degree and Ph.D. in Computer Science Engineering from JNTUA, Anantha-purmau. Now she is working as an Associate Professor in the Department of Computer Science Engineering, KSRM College of Engineering (A), YSR Kadapa(Dist), A.P. Her areas of interest include Computer Networks, Mobile Adhoc Networks, Trust management in MANETS and other latest trends in technology. She has more than 11 years of experience in teaching and research in the area of Computer Science and Engineering.

Dr.G.Rajeshkumar received his B.E (Information Technology) in 2003 from Periyar University, Salem with First Class, M.E (Computer Science and Engineering) in 2009 from Anna University Chennai, with First Class with Distinction and Ph.D (Under the Faculty of Information and Communication Engineering) in April-2017 from Anna University Chennai, for the thesis titled ‘‘An Enhanced Route Optimization Technique for Wireless Networks with Improved Quality of Service using Trust Based Adaptive acknowledgment Scheme”. He is currently working as Associate Professor in the Department of Information Technology at Karpagam College of Engineering, Coimbatore, Tamil Nadu, India. He has around 15 years of teaching and research experience. His areas of interests are Wireless Communication, IoT, Computer Networking and Security Systems. He is life Professional Membership of ISTE. He has published research papers in reputed journals and conference proceedings, out of which majority of the papers are indexed in SCOPUS, SCI and Web of Science. He has guided many B. Tech. student projects / thesis and currently guiding B. Tech. students for their project / research /dissertation work.

Dr P.Vidya Sagar is an Indian academician who is serving as an Associate Professor in the Department of Computer Science& Engineering in KL University Vijayawada, Andhra Pradesh, India. He got the Ph.D.(Computer Science & Technology) from Sri Krishnadevaya University, Andhra Pradesh, India, in 2016. M.Tech. (Computer Science & Engineering) from Acharya Nagarjuna University, Andhra Pradesh, India, 2010. The major domain/specialization of doctorate Is Software Engineering application with Deep Learning, Image processing, Data Mining and Networking. I had around 10 yrs of IT industrial experience with major MNC’s & I am currently acting as reviewer/editorial member if international journals and organize member for international conferences.

Dr.Pankaj Dadheech received his Ph.D. degree in Computer Science & Engineering from Suresh Gyan Vihar University, Jaipur, Rajasthan, India. He received his M.Tech. Degree in Computer Science & Engineering from Rajasthan Technical University, Kota, and he has received his B.E. in Computer Science & Engineering from the University of Rajasthan, Jaipur. He has more than 15 years of experience in teaching. He is currently working as an Associate Professor in the Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan (SKIT), Jaipur, Rajasthan, India. He has published 10 Patents at Intellectual Property India, Office of the Controller General of Patents, Design and Trade Marks, Department of Industrial Policy and Promotion, Ministry of Commerce and Industry, Government of India. He has presented 45 papers in various National & International conferences. He has 30 publications in various International & National Journals. He has published 2 Books & several Book Chapters. He is a member of many Professional Organizations like the IEEE Computer Society, CSI, ACM, IAENG& ISTE. He has also guided various M.Tech. Research Scholars. He has Chaired Technical Sessions in various International Conferences & Contributed as Resource Person in various FDP’s, Workshops, STTP’s, etc. He is also acting as a guest editor of the various reputed journal publishing houses and Bentham Ambassador of Bentham Science Publisher. His area of interest includes High-Performance Computing, Cloud Computing, Cyber Security, Big Data Analytics, and the Internet of Things.

Dr.S.R.Dogiwal is working as Associate Professor, Department of Information technology at Swami keshvanand Institute of Technology Management, Management & Gramothan, Jagatpura, Jaipur. He has awarded Ph.D., in Computer Engineering (Image Processing). He has completed MCA from the Department of Computer Science, University of Rajasthan, Jaipur. His research Area in Image Processing. He has published multiple papers in different journals related to Image processing. His articles are published in 18 International and National Journals. His work has been profiled broadly, such as in Information Security, Image Processing, Neural Network, and Network. He is a Reviewer and Editor of many reputed Journal. He has supervised12 postgraduate dissertation(M.Tech.).

Dr.Priya Velayutham received her Ph.D. degree in Information and Communication Engineering in 2017 at Anna University, Chennai. Currently, she is working as an Associate Professor in Computer Science and Engineering at Mahendra Institute of Technology, Namakkal, Tamil Nadu. She has more than 12 years of experience in Teaching and Research. Her research interests are in the areas of Artificial Intelligence, Machine Learning, Deep Learning, Cloud computing, Image processing, Data Science, Internet-of-Things, and Bioinformatics. She published her research articles in reputed international journals, which is having a high impact factor. She received Best Faculty Award in Junior/ Department of CSE in 2019 from Shri P.K Das Memorial Best Faculty Award and Best Researcher Award in 2020 for her publications. She is a Life Member of the Indian Society for Technical.

Dr.Sudhakar Sengan is presently working as Professor, Department of Computer Science and Engineering at Sree Sakthi Engineering College, Coimbatore, Tamil Nadu, India. He has 20 Years of Experience in Teaching / Research / Industry. He received his ME degree in the faculty of Computer Science and Engineering from Anna University, Chennai, Tamil Nadu, India. And he received a Ph.D. Degree in Information and Communication Engineering from Anna University, Chennai, India. He has published papers in 75 International Journals, 20 International Conferences, and 10 National Conferences. His research interest includes Network Security, Information Security, IoT, MANET, and Cloud Computing. He is a member of various professional bodies like MISTE, MIEEE, MIAENG, MIACSIT, MICST, MIE, and MIEDRC. He is the Recognized Research Supervisor at Anna University under the faculty of Information and Communication Engineering. He received an award of Honorary Doctorate (Doctor of Letters-D.LITT.) from International Economics University, SAARC Countries in the field of Education and Students Empowerment in April 2017. He has published 15 India Patent in the various fields. He has published a Textbook for Anna University syllabus: Title: "Digital Principles and System Design" Thakur Publications Pvt. Ltd, Chennai. ISBN: 978-93-87880-77-1., Title: "Problem Solving & Python Programming," Charulatha Publication, Chennai. Title: Operating Systems", Thakur Publications Pvt. Ltd, Chennai. ISBN-978-93-88809-15-3.