Computers, Materials & Continua
DOI:10.32604/cmc.2021.014594
Article

Tech Science Press

Machine Learning Empowered Security Management and Quality of Service Provision in SDN-NFV Environment

Shumaila Shahzadi[1], Fahad Ahmad[1,*], Asma Basharat[1], Madallah Alruwaili[2], Saad Alanazi[2], Mamoona Humayun[2], Muhammad Rizwan[1] and Shahid Naseem[3]

[1]Department of Computer Sciences, Kinnaird College for Women, Lahore, 54000, Pakistan
[2]College of Computer and Information Sciences, Jouf University, Sakaka, 72341, Saudi Arabia
[3]Division of Computer Science & Information Technology, University of Education, Lahore, 54000, Pakistan
*Corresponding Author: Fahad Ahmad. Email: drfahadahmadmian@gmail.com

Received: 01 October 2020; Accepted: 18 October 2020

Abstract: With the rising demand for data access, network service providers face the challenge of growing their capital and operating costs while at the same time enhancing network capacity and meeting the increased demand for access. To increase efficacy of Software Defined Network (SDN) and Network Function Virtualization (NFV) framework, we need to eradicate network security configuration errors that may create vulnerabilities to affect overall efficiency, reduce network performance, and increase maintenance cost. The existing frameworks lack in security, and computer systems face few abnormalities, which prompts the need for different recognition and mitigation methods to keep the system in the operational state proactively. The fundamental concept behind SDN-NFV is the encroachment from specific resource execution to the programming-based structure. This research is around the combination of SDN and NFV for rational decision making to control and monitor traffic in the virtualized environment. The combination is often seen as an extra burden in terms of resources usage in a heterogeneous network environment, but as well as it provides the solution for critical problems specially regarding massive network traffic issues. The attacks have been expanding step by step; therefore, it is hard to recognize and protect by conventional methods. To overcome these issues, there must be an autonomous system to recognize and characterize the network traffic's abnormal conduct if there is any. Only four types of assaults, including HTTP Flood, UDP Flood, Smurf Flood, and SiDDoS Flood, are considered in the identified dataset, to optimize the stability of the SDN-NFV environment and security management, through several machine learning based characterization techniques like Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR) and Isolation Forest (IF). Python is used for simulation purposes, including several valuable utilities like the mine package, the open-source Python ML libraries Scikit-learn, NumPy, SciPy, Matplotlib. Few Flood assaults and Structured Query Language (SQL) injections anomalies are validated and effectively-identified through the anticipated procedure. The classification results are promising and show that overall accuracy lies between 87% to 95% for SVM, LR, KNN, and IF classifiers in the scrutiny of traffic, whether the network traffic is normal or anomalous in the SDN-NFV environment.

Keywords: Software defined network; network function virtualization; machine learning; support vector machine; K-nearest neighbors; logistic regression; isolation forest; anomaly detection; attacks

1 Introduction

In this research, the proposed framework may help develop a comprehensive response for insufficiency, odd acknowledgment, shirking, and augmentation to delineate availability and capability under any conditions.

1.1 Anomaly Types

The anomaly is an undefined property of a standard sample. To understand the anomaly, different algorithms are required. The anomaly is analyzed under three headings:

- If a particular sample of data looks different from the properties it carries from the entire dataset; it is called a point anomaly.
- A data sample's out-of-pattern behavior depends on or occurs under specific conditions known as a contextual anomaly.
- According to standard data, if a data load consisting of similar data has anomalous properties, it is called the collective anomaly.

1.2 Network Attacks

The network security principle tries to secure the network from malicious data using confidentiality, integrity, and availability. All network characteristics and their relationship with different malicious attacks are shown in Fig. 1.

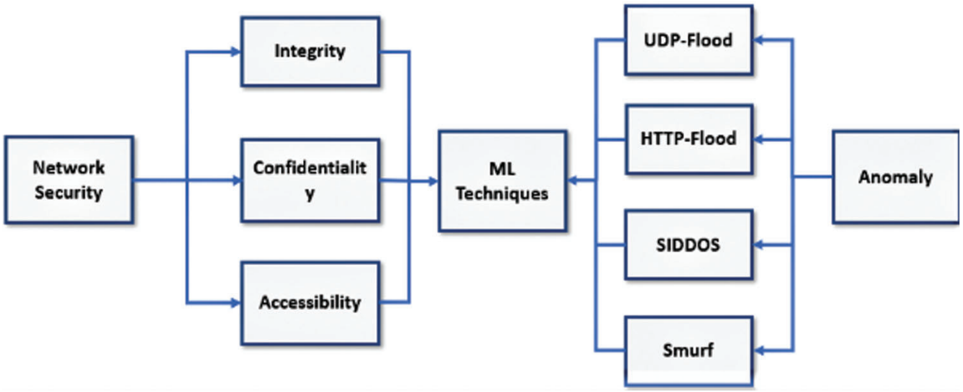


Figure 1: Security management through anomaly detection

- Information should be accessible only to authentic users, and unauthorized users would be barred named as confidentiality [1].
- Adding, modifying, and deleting information can only be done by the authentic user. Unauthorized users would not be able to modify information is called integrity [2].
- The system should always provide accessibility to the authentic user.

1.3 Network Attacks

The major types of network attacks include:

- HTTP-Flood is a Distributed Denial of Service (DDoS) attack flooded over a web application using port 80, through HTTP GET or POST request. In a web application, HTTP might not be used for ill-formed packets, bluffing, or replication.
- UDP-Flood Attack is a Denial of Service (DoS) attack processed over the User Datagram Protocol (UDP) using port 53 as a target port to an IP address. DoS attack is flooded in the form of an application-specific irregular UDP.
- To accomplish DDoS, a web site using SQL injection, SQL Injection is an application layer attack that takes advantage of security vulnerabilities in websites and applications, and when executed, gives the hacker access to an underlying database is known as SiDDoS attack.
- In the Smurf attack, Internet Control Message Protocol (ICMP) can be used to broadcast a request from a target node, transmit network traffic, and slow down its transmission speed [3].

1.4 Software Defined Network

SDN's primary function is to manage the system, control, and vesting the system's capabilities to turn out simple programs and detach the system applications and system administration. It is legitimately programmable and permits to progressively arrange a comprehensive traffic stream to address evolving issues utilizing SDN programs. The system is legitimately brought together in a programmed SDN controller that keeps up an overall impression of the integrated system, which shows up as one single intelligent switch, as depicted in Fig. 2 SDN might be considered as a controller that combines the framework coordinating rationally, and decoupling the data plane (send/receive packets) from the control plane (direct) [4].

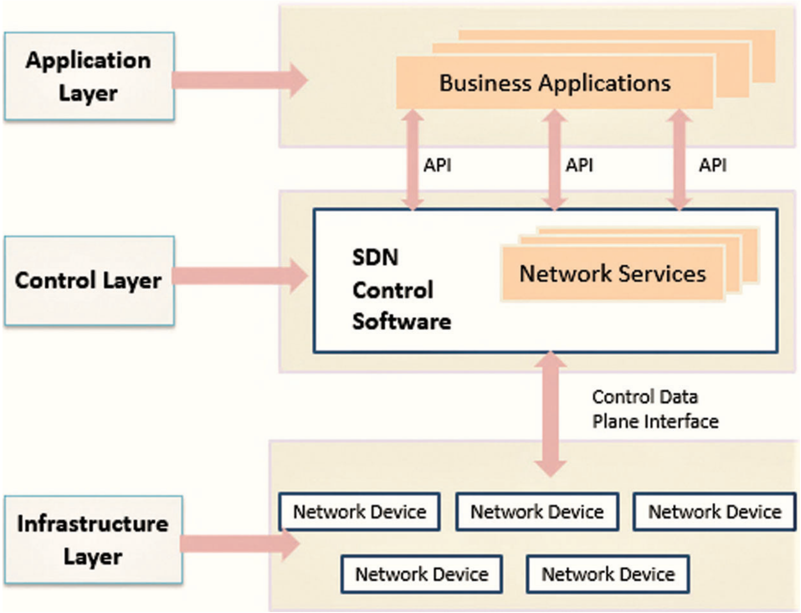


Figure 2: Software defined network architecture

An essential segment in the control plane is the controller, which has an overall view of the network and can subsequently propel the organization of streams in a versatile and adaptable manner. The Open Flow scheme is the communication standard between both data and control planes [5]. The scheme represents a channel between these two planes, making it possible to incorporate or remove entries in the switch stream tables. Besides this, programmable frameworks can shatter the interruptions to new and propelling advancements in computer networks.

1.5 Network Functions Virtualization

Network Function Virtualization (NFV) refers to adopting virtualization technologies in a network environment that separates system capacities from restrictive equipment and runs them as programming in virtual machines (VMs). The European Telecommunications Standards Institute (ETSI) has been leading the standardization of this approach. For example, the various capacities, firewalls, traffic control, and virtual directing are called Virtual Network Functions (VNFs). NFV uses virtualized networking components to support an infrastructure fully independent of hardware. Multiple VMs can run on a single server and scale to consume the remaining free resources. It means that resources are less often sitting idle, and data centers with virtualized infrastructure can be more effectively used. Within the data center and the outside networks, the data plane and control plane can also be virtualized with NFV [6].

There are three principal parts in the NFV design: framework, administration, organization, and the executives presented in Fig. 3. The infrastructure, known as NFVI (NFV Infrastructure), covers equipment and programming assets, just as the virtualization condition. The Service part covers the virtualized organize capacities, or VNFs, which are chosen from an archive. They are executed in virtualized conditions rather than devoted equipment (otherwise called center boxes). Finally, Management and Orchestration in NFV, known as NFVMANO, centers around increasingly particular errands. For example, if there is a firewall and an Intrusion Detection System (IDS) executed as VNFs, the NFV-MANO decides the request wherein packets of a stream should cross these VNFs [7].

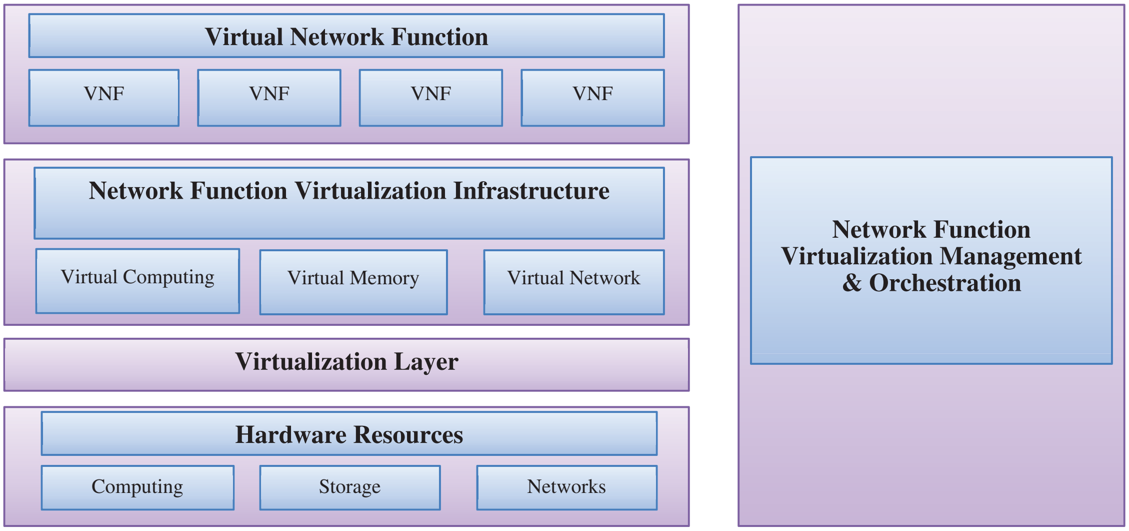


Figure 3: Network function virtualization

1.6 Hybrid SDN-NFV Controller

Network software is a growing optimistic phenomenon aimed at fundamentally advancing telecommunications industries by adopting network infrastructure, cloud computing technology, and software models [8]. SDN and NFV are the main forces behind this transition. By decoupling the control and data plane, SDN provides a new level of network modifiability. A centralized controller supervises the network's state and establishes instructions for the network components to appropriately handle the traffic flows. On the other hand, NFV enhances virtualization frameworks to implement network components as software instances, ensuring an increased degree of versatility and elasticity in services' lifecycle management. By replacing dedicated expensive hardware with commodity servers to host software-based network appliances, NFV will allow a remarkable reduction in CAPEX/OPEX cost [9]. Although SDN and NFV are two different frameworks, their everyday use will further improve the network's potential security services and meet the broad array of those requirements imposed by new applications [10]. The enormous number of resources and tactile internet applications are significant representations of challenging scenarios that reveal various new technology and vulnerabilities. Taking advantage of the versatility and scalability provided by SDN and NFV integration, the telecommunication operators should be able to effectively implement the necessary security policies in the designated environment [11].

SDN and NFV combination can be used to handle control challenges in an SDN controller that can be virtualized throughout the cloud of servers rapidly. This method also delivers extra compensations such as reduced hardware maintenance idle-time and lessened recovery-time if disaster or failover occurs.

NFVI delivers the infrastructure that involves all the hardware, and software means vital in organizing VNFs [12]. The hardware means involve computing, storage, and network essentials that deliver the processing, storage, and connectivity competences to VNFs over a virtualization layer. The virtualization layer delivers an abstraction to the hardware and empowers the software to utilize the virtualized infrastructure as a substitute [13].

1.7 Machine Learning Approach

Machine learning enables systems to learn from the patterns given by it in the form of instructions. In machine learning, computers can be trained through instruction and assign different patterns and analyze their behavior in solving complex problems. Machine learning is used to interpret extensive and complicated data to make the existing systems autonomous by using some learning algorithms required to reason and process the raw data by incorporating supervised, unsupervised, and reinforcement learning.

- Supervised learning techniques, has expertly grouped, and marked the training information.
- Unsupervised learning separates the information into bunches as per different properties and watches their relationship.
- Reinforcement learning is required to labeled or unlabeled a tiny portion of the data.

This learning is also combined the high performance of supervised learning with the low performance of unsupervised learning.

1.8 Machine Learning Classifiers

Machine Learning empowered classifiers are used for classification purposes in different scenarios. Identification of different kinds of attacks in the SDN-NFV environment through different ML classifiers can also be planned, and accuracy can be used to measure each classifier [14]. The detector can use multiple classification models like Support Vector Machine (SVM), Isolation Forest (IS), K-Nearest Neighbors (KNN), and Logistic Regression (LR). Different implementations' performance rate shows that the KNN classifier is much better than other competing classifiers in diverse kinds of attack detection. The improved performance rate of detecting attacks in the SDN-NFV environment will reduce the delay in processing time with an extensive load [15]. The purpose of traffic quality improvement is to ensure the network services and end-user quality of experience in the NFV environment [16].

- Support Vector Machine (SVM) could be applied to distinguish unusual conduct and stretch out to identify a few sorts of assaults from traffic. SVM is an unsupervised ML algorithm that can be used for anomaly detection. The main goal of SVM is to classify each packet of traffic in two distinct classes. 1 is used to represent "Normal," and +1 is used for "Anomaly/Attack," and then C-SVC is used to solve the following optimization problem.

$$a = \sum_{i=0}^{N} \lambda_i y_i \overrightarrow{x_i} \tag{1}$$

$$\sum_{i=0}^{N} \lambda_i y_i = 0 \tag{2}$$

The extreme hyperplane helps to characterize which inclination of all the conceivable isolating lines is the best classifier. SVM chooses the most considerable edge for isolating hyperplane and augments the capacity to incorporate any concealed quality. A straight condition can define the new hyperplane.

$$f\ (x) = ax\ +\ b \tag{3}$$

The distance between the hyperplane and the data points can be calculated by:

$$M1 = \frac{|f(x)|}{||a||} = \frac{1}{||a||} \tag{4}$$

- Isolation Forest unequivocally distinguishes abnormalities instead of describing specific information. IF, similar to any tree ensemble strategy, is based on the choice of trees. It assembles a decent performing model with few trees utilizing little subtests of fixed size, paying little heed to an informational index's size. An abnormality score is required for dynamics on account of IF. It is characterized as:

$$Anomaly\ Score\ (S) = 2\frac{-E(h(k,m,N))}{c(n)} \tag{5}$$

$$c(n) = 2(\ln(n-1) + 0.5772156649) - 2 \tag{6}$$

$$E(h(k,m,N)) = \frac{\sum_{i=1}^{N} \begin{cases} if\ k == 1, \sum_{j}^{M} 1 \\ else, \sum_{j=1}^{M} 1 + c(k) \end{cases}}{N} \tag{7}$$

where N is a total number of trees, M is a total number of binary splits, k is a total number of data points in the final node, and n is the total number of data points in a sample. Each observation is given an anomaly score, and the decision can be made accordingly if Score = 1 indicates anomalies, if Score < 0.5 indicates standard observations, and otherwise if Score = 0.5 indicates isolation.

- The K-Nearest Neighbor's algorithm (k-NN) is a nonparametric approach utilized in classification; the yield is a class association. Traffic is classified by a plurality vote of its neighbors, with the traffic being allotted to the anomaly class famous among its k nearest neighbors. If k = 1, then the traffic is simply allocated to that single nearest neighbor's anomaly class.
- Logistic Regression or logistic model is used to model the probability of an anomaly class that occurs. It can be extended to model several classes with different features consideration.

1.9 Problem Statement

The standing frameworks are deficient in security, and computer systems face few irregularities, which stimulates the necessity for an innovative recognition and mitigation approaches to keep the system in the operative state. To cope with the Software Defined Network (SDN) and Network Function Virtualization (NFV) inadequacies, it is essential to eradicate network security configuration errors that can create susceptibilities that affect overall efficiency, lessen network performance, increase maintenance cost, and compromise the quality of service.

1.10 Objectives

This research will contribute by achieving the following objectives in the SDN-NFV environment:

- Real-time recognition of assaults quickly and viable by considering system oddity with ML strategies in the SDN-NFV environment.
- To enhance security management of the SDN-NFV environment.
- Augmentation of QoS and user experience optimization in the SDN-NFV environment.

2 Literature Review

Software Defined Network (SDN) entails unique security concerns, especially where its controller is defenseless against attacks by Distributed Denial of Service (DDoS). If DDoS attacks occur against the SDN server, the server's operation and contact capability would be overwhelmed and detected DDoS attacks in SDN using an ML-based model through feature selection methods. To simplify these models, feature selection methods were selected whose training time is comparatively shorter [3].

Open stack-based private cloud is used to detect DDoS attacks that directly target the bandwidth flooding, connection flooding, and the server infrastructure over the internet using network protocols and standards [17]. DDoS attacks threaten the cloud's network layer, set up with invalid requests, and refuse legitimate requests. The proposed framework with embedded OpenStack firewall and raw socket programming focused on controlling the network traffic. machine learning techniques such as Decision Tree (DT), K-Nearest Neighbor (KNN), Naive Bayes, and Deep Neural Network (DNN) algorithms have been compared against the trained model based on the dataset created during the management of DDoS attack. From the reported outcomes, it is depicted that DNN has higher precision and accuracy than other classifiers.

Internet of Things (IoT) systems are vulnerable to various security threats ranging from DoS to intrusion into the network and data storage [16,18]. A novel security architecture based on machine leaning has been proposed, which leverages both SDN and NFV, enables them to mitigate various threats [19]. The projected intelligent platform incorporates the monitoring agent and reaction agent that uses machine learning models in IoT to differentiate network traffic patterns. The rate of anomalies' identification was promising [20,21].

Artificial Intelligence (AI) based defense mechanisms proposed a novel approach known as Multi-Layered Intrusion Detection and Prevention (ML-IDP) to identify intrusion in the SDN-NFV enabled cloud of 5G networks. The proposed approach defends against security attacks using AI [22]. The proposed ML-IDP approach is tested using NS3.26 for various security threats, including IP spoofing, overloading the flow table, DDoS, control plane saturation, and hijacking of host location. From the results of the experiment, it is proved that the ML-IDP effectively detects and avoids attacks [23]. The artificial intelligence has much more application in the security management regime that guides to look into human-like cognitive abilities to attain more consistent and operative defense capacities [24].

Classification models Support Vector Machine (SVM), Naive Bayes (NB), Artificial Neural Network (ANN), and K-Nearest Neighbors (KNN) have been trained and tested with the identified dataset [25]. The test results showed that using the wrapper feature selection with a KNN classifier achieved the maximum accuracy rating in DDoS attack detection. The outcome shows that ML-based selection algorithms can produce improved results through detecting DDoS attacks in the SDN environment with decent loads and reduce processing time [26,27].

The Mouseworld, an SDN-NFV based safety traffic analysis, described a novel experimental system, integrating SDN and NFV to establish an ecosystem capable of mixing, transmitting real and synthetic traffic, storing, and marking useable this traffic for training and validation purposes [28]. ML algorithms are used to identify cybersecurity threats [29]. The Mouseworld architecture comprises a series of modules for traffic generation, selection, analytics, algorithm training, and visualization. The results presented security threat detection to confirm the feasibility and validity of the proposed system [30].

Machine Learning (ML) assisted planning, and provisioning for SDN-NFV enabled Metropolitan Area Networks (MANs) to support the control plane in making strategic decisions by using intelligent optimization algorithms [31]. The suggested architecture ensures equal behavior against the past, present, and future applications, for instance, network resource allocation decisions using ML approaches as an executive. The Net2Plan platform, Python platform support ML algorithms, libraries in SDN-NFV-enabled MANs and express favorable outcomes [32,33].

3 Materials and Methods

The development of new traffic monitoring and identification practices to distinguish and characterize unusual traffic is a fundamental concern to secure the system against noxious assaults.

3.1 Incoming Network Traffic

Capturing traffic is the initial phase for monitoring of network traffic in the NFV component. Interruptions can be either inside or outside; therefore, approaching traffic and nearby traffic should be checked. Traffic can be observed and caught at various levels, which not just influences the data accessible for investigation, yet additionally, the exactness of the performance rate for identification.

3.2 Mitigation Agent

After capturing traffic, it is the point to recognize the presence of anomalous cases in traffic flow. For the most part, traffic is preprocessed before being passed to the specialized layer. This layer grasped attacks by identifying the altered part of the traffic. In this way, attacks are distinguished from regular traffic by the mitigation process. Fig. 4 represents the process of identification, and there are some specific features which are essential to identify anomaly or attack:

3.2.1 Volume

The size and complexity of network traffic have increased more rapidly. Thus, traditional time-consuming methods of traffic anomaly detection can no longer meet the timeliness requirement.

3.2.2 Assortment

Network traffic is generated from diverse sources. Novel attacks can overlook the detection systems based on signatures and cause significant damage.

3.2.3 Worth

With the low-value density, some traffic data features do not play a role in identifying anomaly or even hinder the process.

3.2.4 Speed

Velocity detection is critical. Anomalies are ultimate to be detected in real-time.

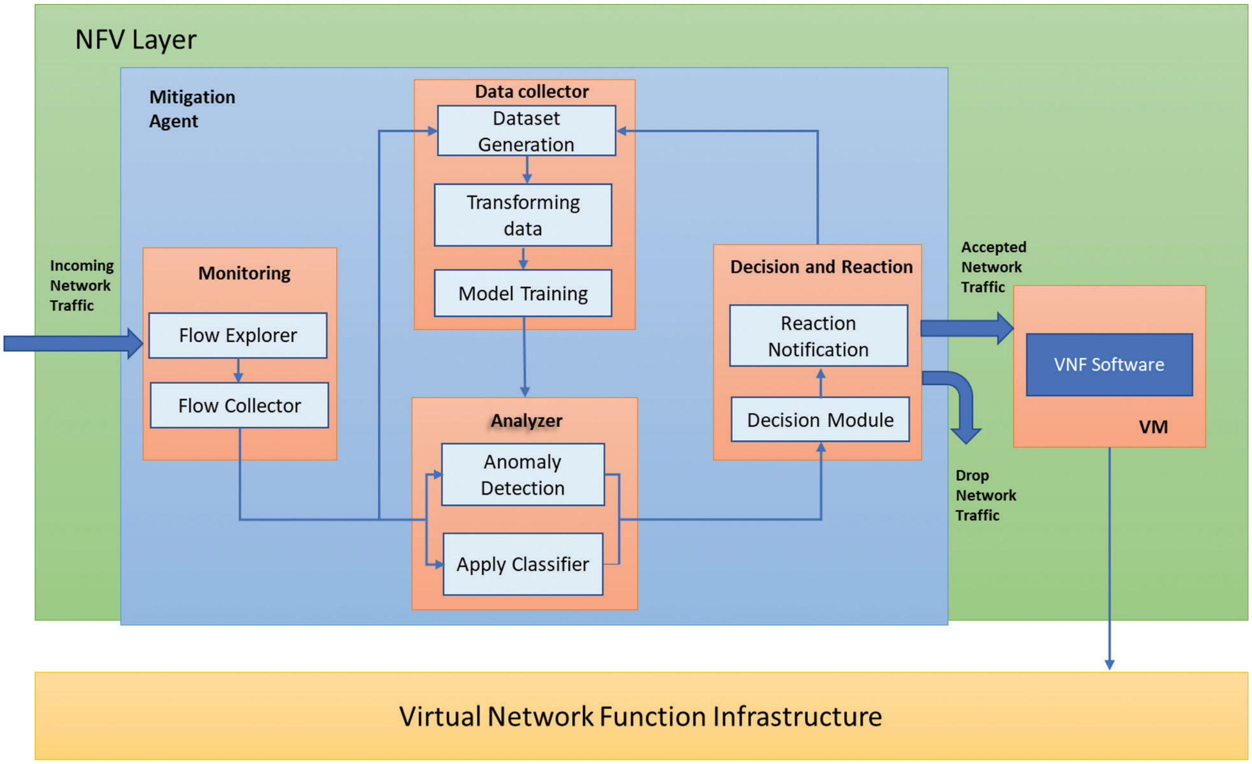


Figure 4: Attack detection in network function virtualization layer

3.3 Monitoring

The initial phase of anomaly identification is to gather features of traffic streams. This refers to the controller investigating the flow of data for authorized access before sending it to the NFV component's various resources.

3.3.1 Flow Explorer

It is essential to pick a reasonable performance level of measuring attacks that differentiate different classes of anomaly cases and normal cases in the dataset and find the occurrence at which layer or resources.

3.3.2 Flow Collector

After setting a measurement scale of flow level, the identification process is happening to check if the flow level is excessively high at resources or layer, then data is passed to the analyzer phase. Then again, if the flow level is excessively low at the controller, missing data are gathered, in light of which the irregularity resource may neglect to distinguish certain anomalous cases.

3.4 Data Controller

In this process, the data controller is versatile and intelligently handles the SDN and NFV based environment, and as per some fundamental processes, an analyzer can progressively determine the levels of complexity for different layers. For example, the identifier may demand a better precision for the flow of data, which causes disturbing behavior or a smaller irregularity in the flow of data traffic. The data controller controls and makes optimized data handling by passing data to multiple modules represent in the flowchart shown in Fig. 5.

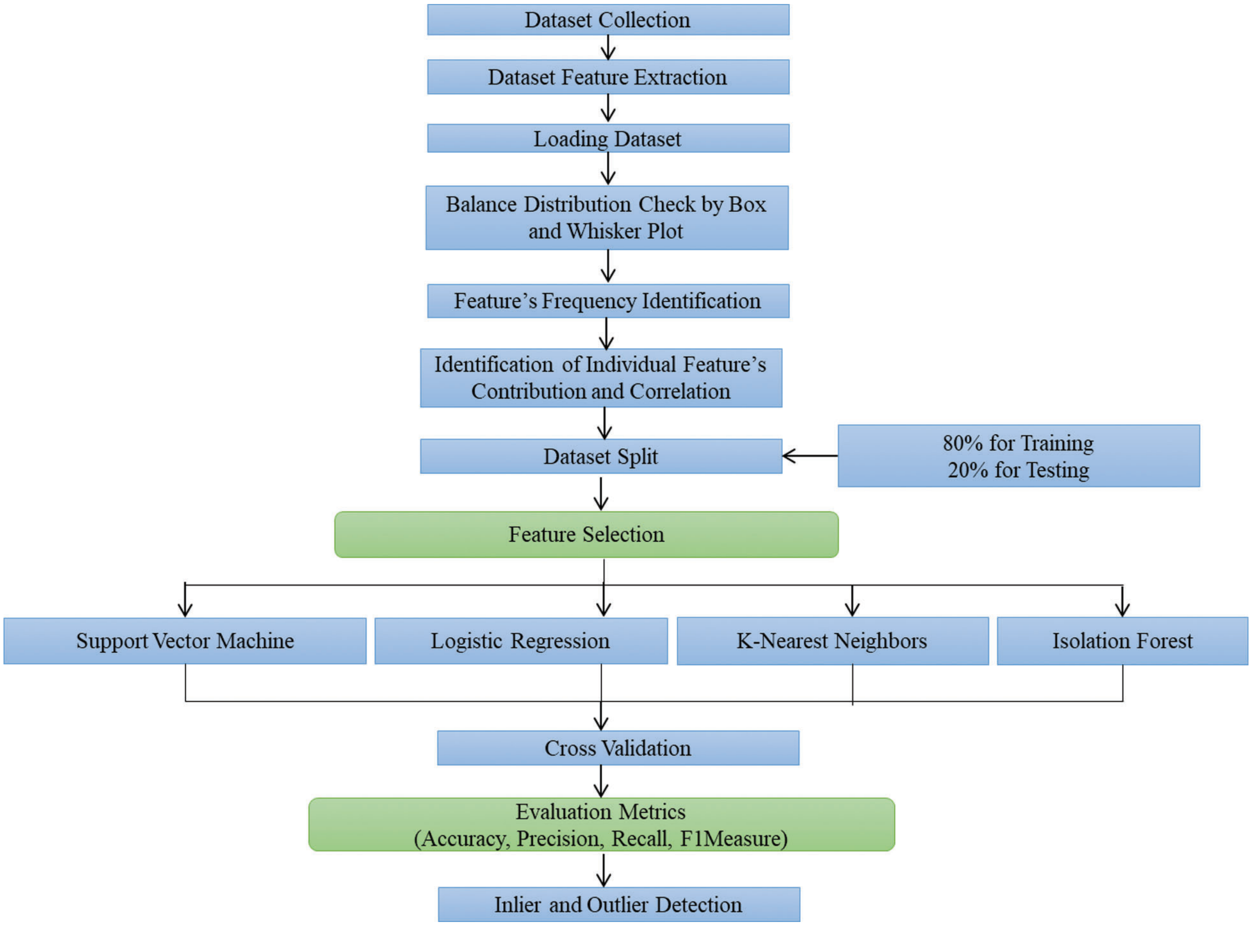

Dataset Collection
Dataset Feature Extraction
Loading Dataset
Balance Distribution Check by Box and Whisker Plot
Feature's Frequency Identification
Identification of Individual Feature's Contribution and Correlation
Dataset Split
80% for Training
20% for Testing
Feature Selection
Support Vector Machine
Logistic Regression
K-Nearest Neighbors
Isolation Forest
Cross Validation
Evaluation Metrics
(Accuracy, Precision, Recall, F1Measure)
Inlier and Outlier Detection

Figure 5: Identification scheme for inliers & outliers

3.4.1 Data Generation

To understand the upcoming issues, framework engineering recognizes both high-and low-volume attacks level. In the proposed framework, the controller can be identified as an incoming attack presence within the system and can classify online traffic by utilizing an ML-based procedure. The ML-based procedure can help make derivations using periodic traffic tests to arrange modules through data stream scrutiny. The proposed approach is good with the framework and does not require any more memory consumption. Besides this, client information security is ensured at all phases of framework activity.

3.4.2 Transforming Data

Procedures, such as separating, changing, and naming the database cases, are performed in this module. The data flow traffic is separated from the recognized files and passes to the next module. Each transformed multiclass data is based on the descriptor database containing all factors. The multiclass data with their data type is transformed using the lable_encoder command to make efficiently accessible data flows for various ML classifiers.

3.4.3 Model Training

A way to deal with programmed highlight choices has been created utilizing the cross-approval method for models that meet specific classification quality measures. This methodology is utilized to define the data points received by training data.

3.5 Analyzer

Enhancement in performance is a significant breakthrough in the identification mechanism, which involves identifying the least possible set of inputs capable of effectively representing different components. A few strategies for variable choice are accessible in writing and actualized in programming libraries like Scikit-learn. The choice of features is performed. Some AI calculations are broadly utilized, i.e., IF, LR, KNN, and SVM.

3.5.1 Classification

The dataset covers machine learning methods capable of detecting intrusions of all ordinary examples; exceptional execution in real conditions is not guaranteed regardless of whether the models achieve high precision on test sets. Features will be identified from the dataset, and further, classifiers will be utilized for classification into two disjoint classes normal or anomalous stream [34].

3.5.2 Anomaly Detection

Network attacks are affected by traffic instances designed to undermine a structured framework's functionality, legitimacy, designation, trustworthiness, and other vital assets. The anomaly detection phase involves many attributes in the collected audit data. Various derived statistical measurements and the classification results will decide whether it is an attack.

3.6 Decision and Reaction

The global decision performs in a better way rather than local decisions to moderate the overload conditions. This process will be performed to analyze the performance quality of all VNFs in the NFV by considering an overloading condition.

3.6.1 Reaction Notification

The VNF-level segments will be responsible for analyzing the excess of network traffic, and instead of blocking the overall traffic, notify the component to reduce its traffic for malicious attack avoidance caused by users. The local decisions will be made by the VNF-level segments that can quickly lessen the overload state, whereas global decisions will be made by the network-level agents to regulate if the VNF-level agents are in overload state. If an overload state exists for a short time, the network agents will be allowed to elicit worldwide decisions. If the network controller requires more time to trigger than the VNF-level agents, network-level overload control will require different parameters to react against network traffic blockage. At this point, the VNF and NFV will be ceaselessly tuned.

3.6.2 Reliable Incoming Traffic

The reliable incoming traffic will be forwarded to the VNF component.

3.7 Drop Network Traffic

The VNF-level traffic will be blocked if network traffic will be flooded; instead, the network-level traffic will be blocked, and dismiss notifications will be sent to the VNF-level individually to each VNFs.

Algorithm: Anomalous traffic identification

```
Input: Received packet from the SDN-NFV network;
  Adata: Set of active data with selected features;
    Ldata: Labels assigned to dataset;
      Output: Frequency calculated for selected data features;
          For (i = 0 to N)
            do
```

(Continued)

(continued).

```
                Ldata: Test_train _split;
                ML_Classifier: Executed;
        Checking: If
                  (non-anomalous_caes)
                Then
                  (Ldata Passed)
End
                Else
                  (Ldata: Blocked)
        Cross_validation: (checking_performance_rate)
End
        Return {<performance_rate>};
End
```

4 Validation and Results

For the detection of traffic anomaly or attack in the SDN-NFV environment, this research will work in the different phases as follows:

4.1 Dataset Collection

For efficient learning, a dataset has to be preprocessed. The dataset is accessible to the public and used in our research. It presents knowledge collected in a detailed state using the Network Simulator 'NS2', which has four vicious attack categories: HTTP Flood, UDP Flood, DDoS Using SQL Infusion (SiDDoS), and Smurf, as shown in Tab. 3. The dataset provides authoritative instances of a normal and malicious traffic attack. We present the analysis of the dataset's features by using the observable investigation and techniques outlined in Tab. 1. The dataset has twenty-eight features, five classes (four assault classes and one normal traffic class), and 21647 records. Tabs. 2 and 3 represent the frequency records of packets types (ACK, CBR, PING, TCP) and normal/attack types.

Table 1: Overview of the dataset

	General information				Nature of the data			Data volume		Recording environment				Evaluation	
	Attack type	**Publicly available**	**Normal traffic**	**Attack traffic**	**Metadata**	**Format**	**Anonymity**	**Count**	**Duration**	**Kind of traffic**	**Type of network**	**Complete network**	**Predefined splits**	**Balanced**	**Label**
Dataset	HTTP Flood, UDP Flood, SiDDoS, Smurf.	Yes	Yes	Yes	No	Packet	Yes (IPs)	11261 packets	NS.	Synthetic	NS.	N. S	No	No	Yes

4.2 Simulation Platform

Python, an open-source, object-oriented programming language, is considered an attractive language due to its essential grammar and dynamic structure. In Python, it is straightforward to compose and investigate the code for errors. Despite each of these points of interest, it works with numerous libraries, where "AI" applications might be possible. In this unique context, Python was chosen because of its substantial interests. Sklearn (Scikit-learn) is an AI library that can be used with the programming language for Python. Sklearn offers a broad scope of choice to the user with its various AI applications. In Python, after loading the dataset, it described all the features of the dataset explained in Tab. 4. Pandas is a fantastic information investigation library running on Python. When working with an enormous dataset, Pandas permit to handily perform numerous tasks, for example, sifting, mass section/column erasure, expansion, and substitution. Matplotlib is a library that is utilized to make diagrams for the investigation of results, and NumPy, a Python library that permits to perform scientific and coherent tasks rapidly and effectively. Here we take out the class name on the testing information by changing the ARFF document into the CSV record position. We have a sum of 28 attributes, including the class name.

Table 2: Records of the packet type

Packet type	No. of records
ACK	7909
CBR	5657
PING	70
TCP	8011

Table 3: Records of anomaly attack

Normal/Attacks	No. of records
Normal	19428
UDP Flood	1998
Smurf	121
SiDDoS	58
HTTP Flood	42

Table 4: Dataset feature

Feature no.	Feature name	Description	Type
0	SRC_ADD	Source IP address	Continuous
1	DES_ADD	Destination IP address	Continuous
2	PKT_ID	Packet identity	Continuous
3	FROM_NODE	Low layer identity	Continuous
4	TO_NODE	High layer identity	Continuous
5	PKT_TYPE	Type of packet	Continuous
6	PKT_SIZE	Packet size	Continuous
7	FLAGS	Flags (SYN, ACK, FIN…) of packet	Symbolic
8	FID	Identity of transfer layer	Continuous
9	SEQ_NUMBER	Sequence number	Continuous

(Continued)

Table 4 (continued).

Feature no.	Feature name	Description	Type
10	NUMBER_OF_PKT	Number of received packets	Continuous
11	NUMBER_OF_BYTE	Number of received bytes	Continuous
12	NODE_NAME_FROM	Name of lower layer	Symbolic
13	NODE_NAME_TO	Name of higher layer	Symbolic
14	PKT_IN	Input packet or not	Continuous
15	PKT_OUT	Output packet or not	Continuous
16	PKT_R	Routing packet or not	Continuous
17	PKT_DELAY_NODE	Delay occurred or not	Continuous
18	PKT_RATE	Number of received packet per second	Continuous
19	BYTE_RATE	Number of received bytes per second	Continuous
20	PKT_AVG_SIZE	Average received packet size	Continuous
21	UTILIZATION	Packet used or not	Continuous
22	PKT_DELAY	The rate for packet delay	Continuous
23	PKT_SEND_TIME	Time to send packet	Continuous
24	PKT_RESEVED_TIME	Time of reserved packet	Continuous
25	FIRST_PKT_SENT	Rate to sent first packet	Continuous
26	LAST_PKT_RESEVED	Rate of last reserved packet	Continuous
27	PKT_CLASS	Packet class (Normal, HTTP flood, UDP flood, SiDDoS, Smurf)	Symbolic

4.2.1 Box and Whisker Graphics

When reviewing these box and fraction graphics, reasonable utilization occurs for practically all types of attacks related to the identified features. The box plot (otherwise known as a box-and-whisker plot) is an integrated figure outlining a selected dataset's general properties in an underlying representation. The lower limit of the container explains the main quartile Q1, such as the limit that indicates 25% of the estimations. The middle Q2 exists and represents the distance between the upper and lower quartiles, dIQR, is the interquartile range (IQR), and contains 50% of the data, and the top limit of the case states the Q3 quartile, that is 75% of the estimations. The midpoint of the overall situation, concerning Q1 and Q3, can demonstrate whether the rectangle is balanced or slanted.

$$IQR = Q3 - Q1 \tag{8}$$

$$Lowerwhisker = Q1 - k*IQR \tag{9}$$

$$Upperwhisker = Q3 + k*IQR \tag{10}$$

It provides a graphical depiction of the considered performance metrics through the boxplot diagrams shown in Fig. 6, representing the distribution of each feature's results. Each boxplot is representing the results for each distinct feature from Packet_ID to Packet_Class, respectively. Values that exceed 1.5dIQR above the upper quartile or below the lower quartile are considered minor outliers; values that exceed 3dIQR above the upper quartile or below the lower quartile are considered significant outliers. These four limits form the fences of the boxplot. Outliers are defined as values beyond the whiskers and marked with circle sign.

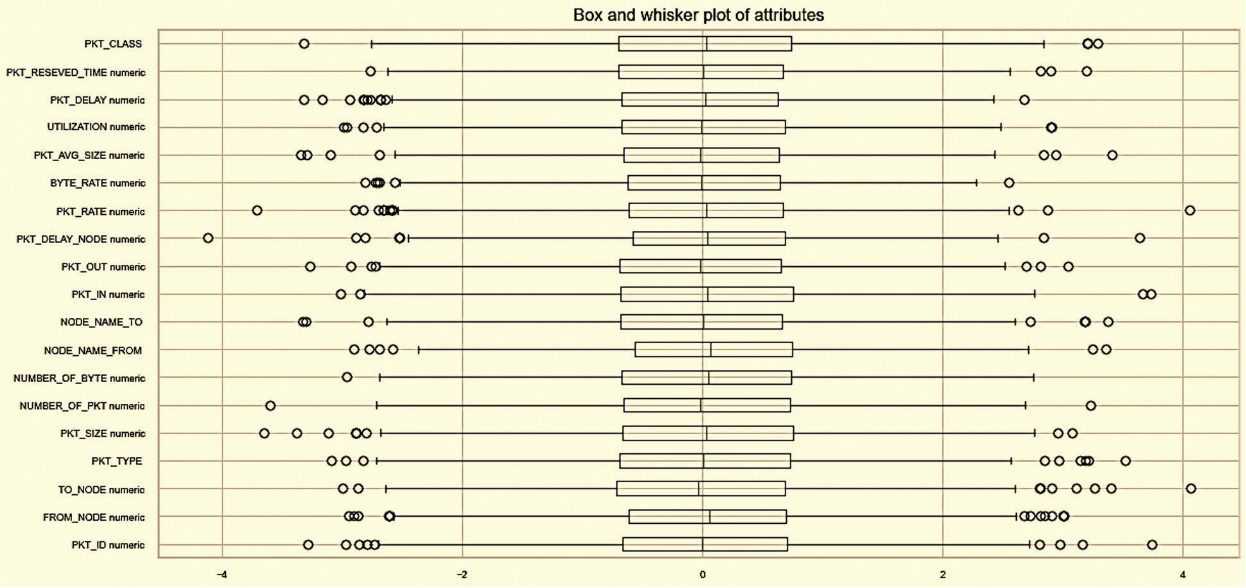


Figure 6: Box and whisker plot for each attribute

4.2.2 Frequency of Selected Dataset Features

The most suggested method used to find all occurrences is to determine the frequency rate; this gets all element distribution and could also print single element frequency if required. Figs. 7–9 represent the frequency ranges of Packer type (CBR, TCP, ACK, PING), Packet class (Normal, SiDDoS, UDP-Flood, HTTP-Flood, and Smurf), and Packet received (client 1, client 2, client 3,.., server1, etc.) from each node.

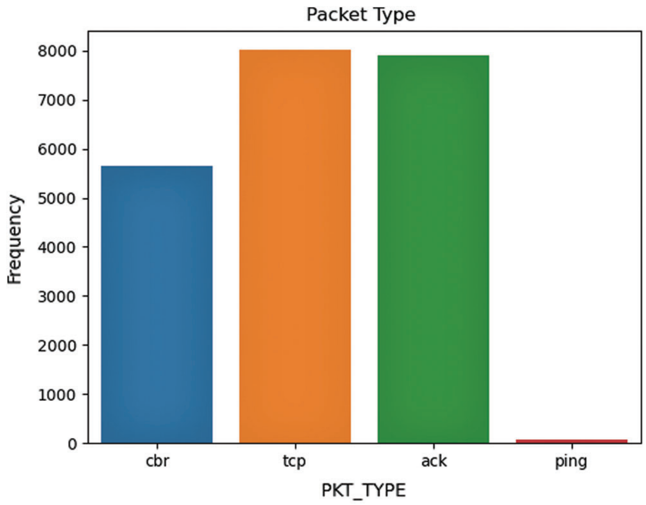


Figure 7: Frequency of each packet type

4.2.3 Histogram Plot for Feature's Values

Fig. 10 shows the distribution of different network traffic features and indicate the unusual changes in features' values as anomaly/attack in the form of different peaks. In the following representation, the statistical dispersal of different featured values is highly skewed due to the difference in each feature's values.

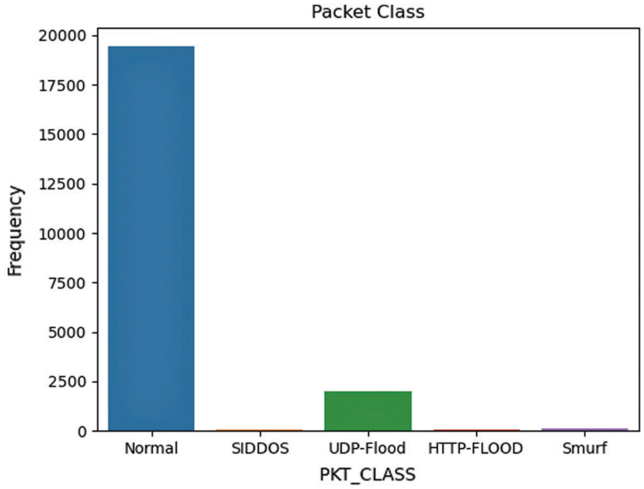


Figure 8: Frequency of each packet class

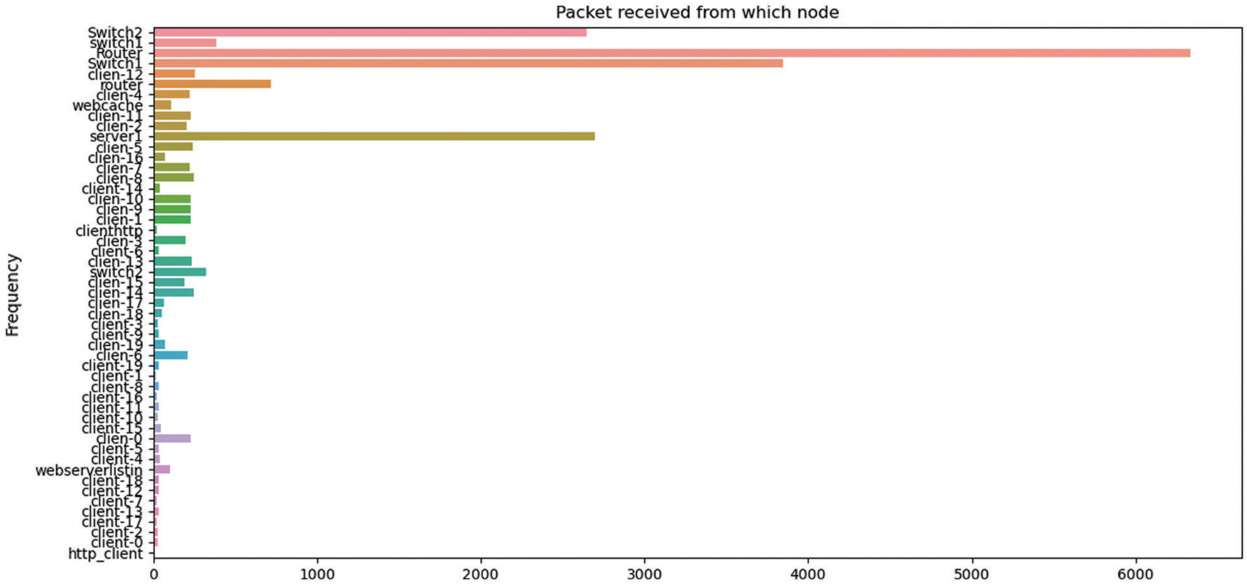


Figure 9: Packet reception frequency from each node

4.2.4 Training and Testing Dataset

Once a refined dataset has obtained about traffic characteristics and features that have been selected from the dataset for building the model, the next step is to generate the train and test dataset for anomaly/attack detection. The dataset will be divided into two different datasets, namely training and testing datasets. The model will be built using the training set, and then we will test it on the testing set to evaluate how the model performs for anomaly/attack. Sklearn based, train_test_split, is utilized that isolates the information into two sections at the sizes determined by the user. Moreover, the primarily suggested ratio is 80% training, 20% testing dataset. The train_test_split order makes a choice asymmetrical while making a decision. This procedure is known as cross-validation. To guarantee that the outcomes acquired during the application are healthy.

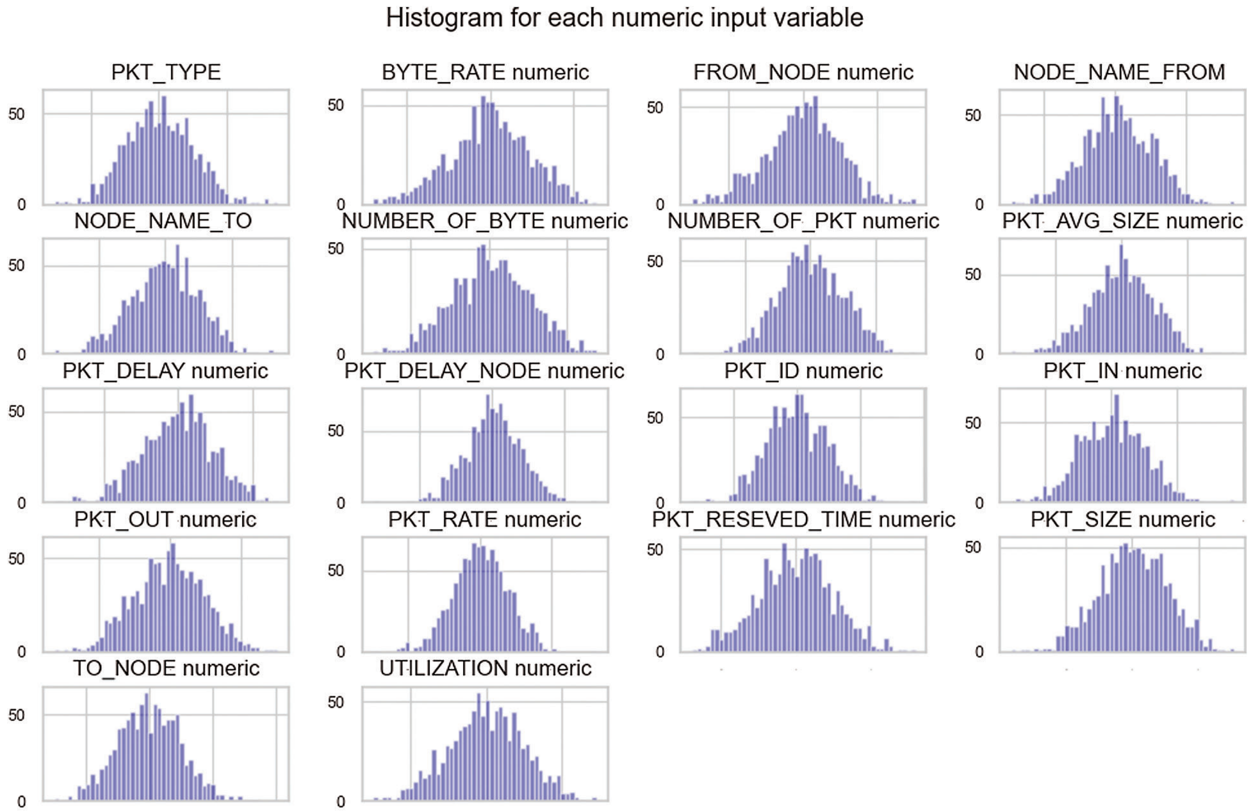


Figure 10: Histogram plot for statistically distributed feature's values

4.2.5 Normalization of Training Dataset

Normalization is needed because the representation provided by its input vectors influence the model. It is a little bit like translating data, and the same scaling can be applied to training results. That means saving the scale and offset and used it again for the training results. Error is measured by comparing the training dataset results before and after normalization separately, which is visualized in Figs. 11–13 represent the dataset's flow that also normalized the confusion matrix at an appropriate scale.

Fig. 14 describes that the model will be programmed with consistent data, and all information would need to be converted to the training scale before it is fed into the model. The algorithm must return a weighted prediction and compare those predictions to achieve the performance metrics for comparison instances [35]. The classification report is a Scikit-Learn built-in metric developed especially for classification problems. The report also returns average accuracy, precision, recall, and F1-score.

4.2.6 Correlation Matrix

The dataset contains records of the incoming/outgoing traffic in an SDN-NFV network. Each record captures some information about traffic like SRC_ADD, DES_ADD,…, LAST_PKT_RESERVED, etc. There is also a Boolean field −1 to 1 determining if that particular feature is positively correlated, negatively correlated, and neutral. The correlation matrix with relevant features represents the relationship among identified features, as shown in Fig. 15.

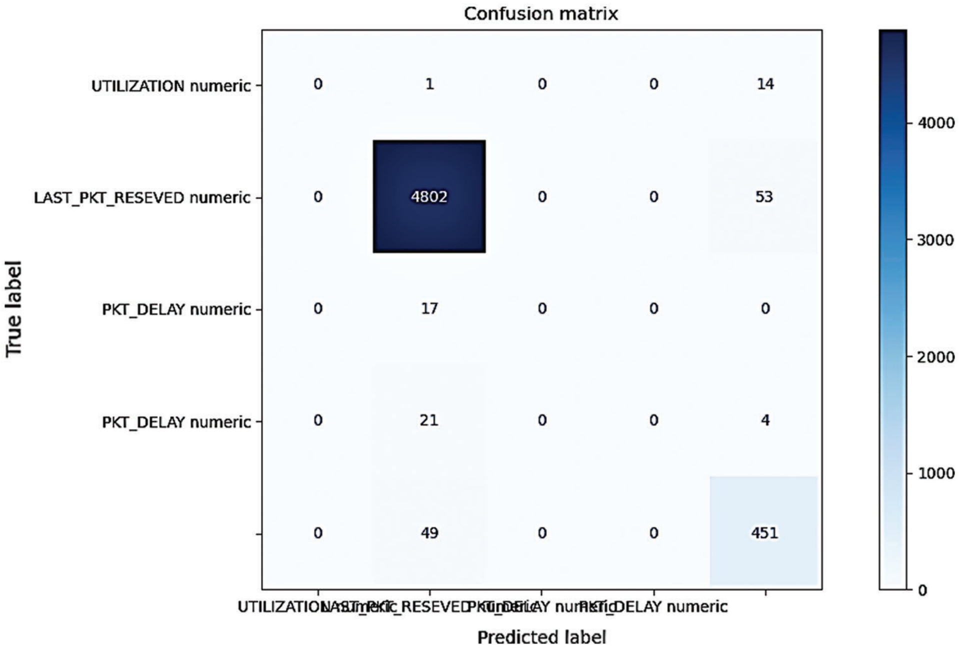


Figure 11: Confusion matrix for training dataset without normalization

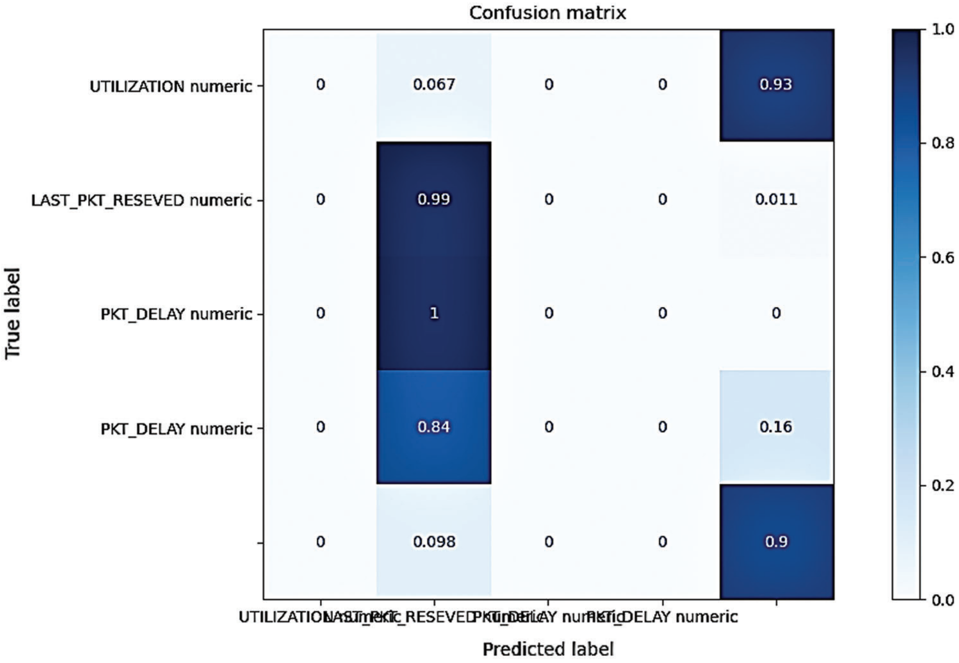


Figure 12: Confusion matrix for training dataset with normalization

4.2.7 Evaluation Metrics

TP, FP, TN, and FN are true positive, false positive, true negative, and false negative. Here, an attack is a positive class, and normal is a negative class. In calculating these four items, the four values summarized below are used:

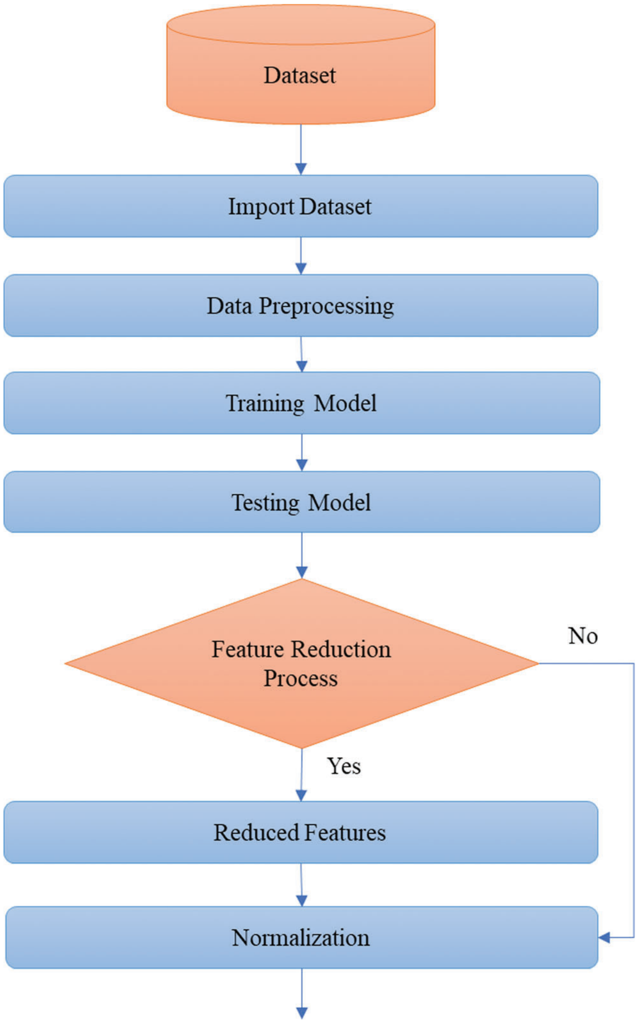
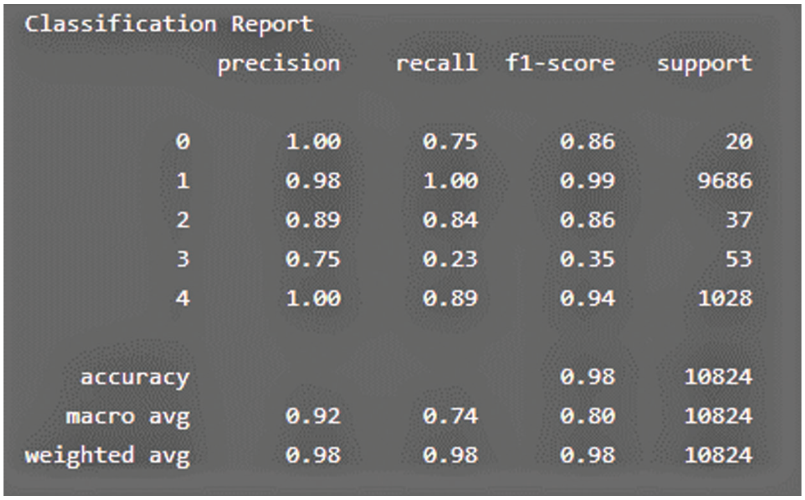


Figure 13: Normalization of features



```
Classification Report
              precision    recall  f1-score   support

           0       1.00      0.75      0.86        20
           1       0.98      1.00      0.99      9686
           2       0.89      0.84      0.86        37
           3       0.75      0.23      0.35        53
           4       1.00      0.89      0.94      1028

    accuracy                           0.98     10824
   macro avg       0.92      0.74      0.80     10824
weighted avg       0.98      0.98      0.98     10824
```

Figure 14: Classification report

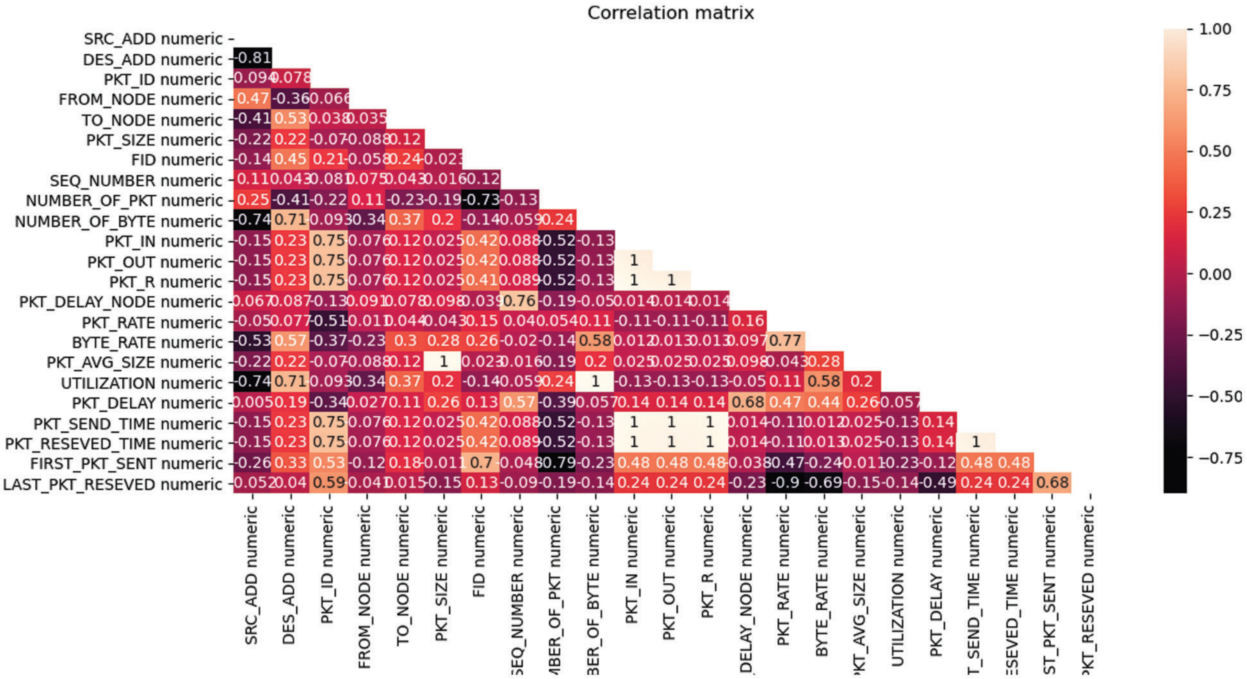


Figure 15: Correlation matrix to analyze anomaly attributes

- TP: True Positive (Correct Detection); the standard traffic is classified as normal.
- FP: False Positive (Type-1 Error); the standard traffic is classified as an anomaly/attack.
- FN: False Negative (Type-2 Error); malicious traffic is classified as normal.
- TN: True Negative (Correct Rejection), the malicious traffic is classified as anomaly/attack.

This distribution is presented in Tab. 5. by visualizing the Confusion matrix.

Table 5: Performance metric

		Predicted	
		Positive	Negative
True	Positive	TP	FN
	Negative	FP	TN

4.2.8 Performance Measures

Various measures are used to determine the execution of ML methods. Their proposed strategies have various attributes and give various outcomes for recognizing anomaly/attack. A few exhibition measures, such as accuracy, precision, recall, and false alarm rate, are calculated for identifying traffic as standard or malicious shown in Tab. 6.

$$Acc_{tr} = \frac{1}{N_1}\sum_{j=1}^{N1}\left(y_j\,,\ tr - \ y_j\,,\ pred\right) \tag{11}$$

Table 6: Number and ratio of normal and anomaly cases

Cases	Count
Anomaly	2219
Normal	19428
Ratio of Anomaly	0.092
Ratio of Normal	0.897

$$Acc_{te} = \frac{1}{N_2}\sum_{j=2}^{N2}\left(y_j\ ,\ te -\ y_j\ ,\ pred\right) \tag{12}$$

Accuracy measures the rate of accurately classified anomaly/attack instances in a class, and it can be called as:

Accuracy = *Number of correct detections/Total number of detections*

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{13}$$

Precision is the ratio between the received relevant anomaly/attacks to the total number of relevant and irrelevant anomaly/attacks is called precision or positive predictive, and it can be calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

A recall is the ratio between the received relevant anomaly/attacks to the total number of relevant anomalous/attacks is called recall or positive sensitivity that can be calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

F1 Measure or F-score can be calculated by using the Harmonic Mean. It arranges the information to correctly classify the data, classify the observations, and recalls the characteristic achieved when a significant number of instances are not missed.

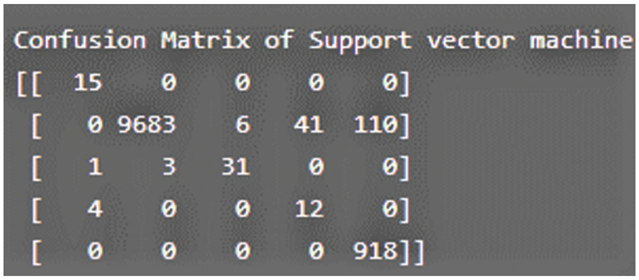
$$F1 = 2*\frac{1}{Precision} + \frac{1}{Recall} \tag{16}$$

Using the confusion matrix, it is easy to calculate the accuracy, precision, recall, and other performance rates of different ML classifiers. Figs. 16–19 signify different ML classifiers' performance through the Confusion matrix.

Now statistical values about accuracy and anticipated characterizations are completed by the efficient models. Execution measurements of such models are ordinarily assessed utilizing the data in the lattice presented in Tabs. 6 and 7. The outlier fraction is 0.1134. Fig. 20 epitomizes the selected dataset's anomaly/attack patterns according to the appropriate traffic type by a Confusion matrix.

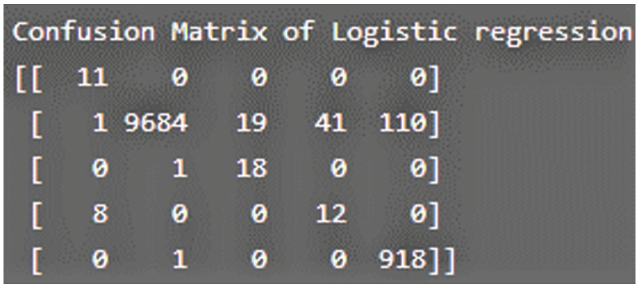
4.2.9 Detection of Anomaly by Inlier and Outliers

This method attempts to distinguish incoming/outgoing traffic in the SDN-NFV environment as normal or anomaly/attack. In the SDN-NFV environment, an inlier pattern represents normal network behavior and outliers as possible anomalous attempts [36]. Fig. 21 shows the inliers and outliers for anomaly detection using machine learning techniques.



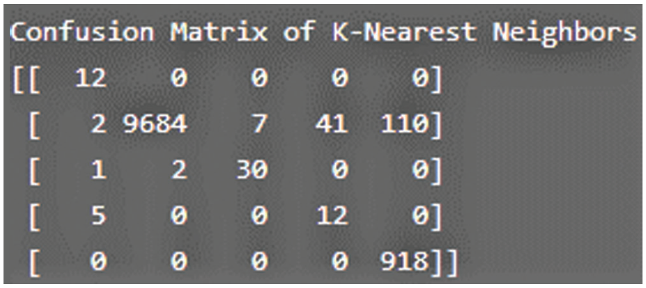
```
Confusion Matrix of Support vector machine
[[  15    0    0    0    0]
 [   0 9683    6   41  110]
 [   1    3   31    0    0]
 [   4    0    0   12    0]
 [   0    0    0    0  918]]
```

Figure 16: Confusion matrix for support vector machine



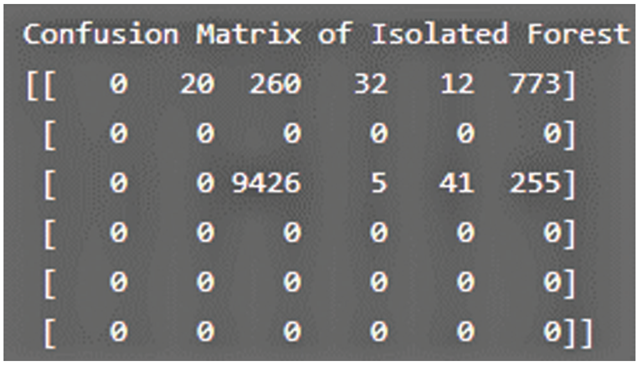
```
Confusion Matrix of Logistic regression
[[  11    0    0    0    0]
 [   1 9684   19   41  110]
 [   0    1   18    0    0]
 [   8    0    0   12    0]
 [   0    1    0    0  918]]
```

Figure 17: Confusion matrix for logistic regression



```
Confusion Matrix of K-Nearest Neighbors
[[  12    0    0    0    0]
 [   2 9684    7   41  110]
 [   1    2   30    0    0]
 [   5    0    0   12    0]
 [   0    0    0    0  918]]
```

Figure 18: Confusion matrix for K-nearest neighbors



```
Confusion Matrix of Isolated Forest
[[   0   20  260   32   12  773]
 [   0    0    0    0    0    0]
 [   0    0 9426    5   41  255]
 [   0    0    0    0    0    0]
 [   0    0    0    0    0    0]
 [   0    0    0    0    0    0]]
```

Figure 19: Confusion matrix for isolation forest

Table 7: Inliers and outliers detection

Machine learning algorithm	Inliers	Outliers
Isolation forest	990	10
Cluster-based local outlier factor	990	10
K nearest neighbors	991	9

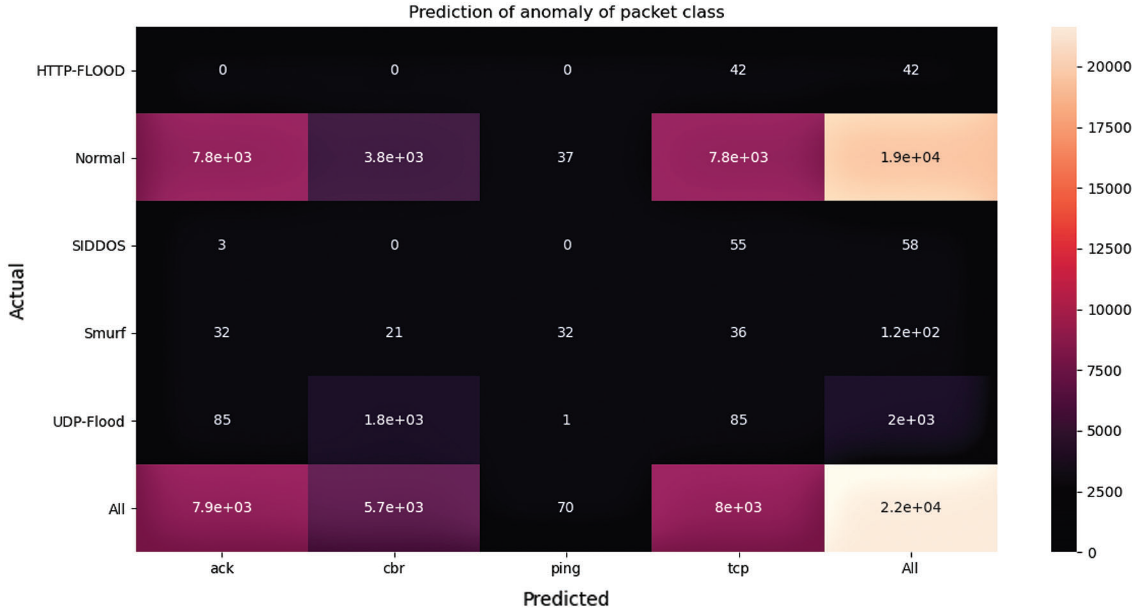


Figure 20: Confusion matrix for anomalous classes prediction

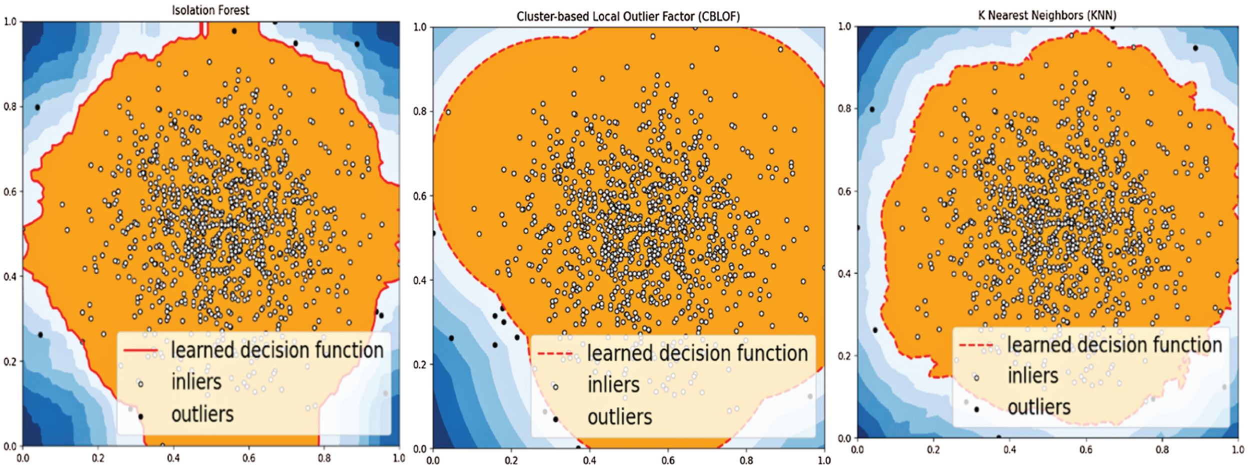


Figure 21: Inlier and outliers detection by IF, CBLOF, and KNN

5 Discussion

This section sightsees outcomes of the proposed simulated framework. After accuracy measurement, precision is identified as actual and in-depth, correct classification measurement, i.e., how many are normal from all the instances classified. The values obtained show how effective the machine learning algorithms are when the incoming/outgoing traffic is classified as normal. It is also said to be a positive predictive factor. The average precision for SVM, LR, KNN, and IF are correspondingly in the range of 0.984, 0.982, and 0.983, 0.969 can be visualized in Tab. 8. The classifiers are evaluated using the confusion matrix based on the measurements, as reported in Figs. 16–19 for SVM, LR, KNN, and IF.

Table 8: Performance results

Algorithms	Accuracy	Precision	Recall	F1 score
Support vector machine	0.984	0.984	0.983	0.981
Logistic regression	0.983	0.982	0.983	0.983
K-nearest neighbors	0.984	0.983	0.984	0.984
Isolation forest	0.870	0.969	0.973	0.919

We have computed the accuracy, precision, and recall of our model from the above-reported confusion matrices. The overall accuracy is 87% to 95% for SVM, LR, KNN, and IF classifiers shown in Tab. 8. In any case, it is necessary to visualize the precision limit, especially when the information is comprehensive, as in our model, where the number of cases in the normal class is higher than in the abnormal. The quality and analysis of every information are calculated in this manner. It is observed that all classifiers are accurately executed and checked for the identification of irregularities. The recall is described as being from all common instances; how many are categorized correctly. The recall results of each classifier were also represented in Tab. 8. The algorithm with higher recall values shows minimal false negatives represent ML achieved high accuracy. F1 Score is the weighted average of precision and recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if the dataset has uneven class distribution.

In optimization, performance can be compared in different settings through True Positive Rate (TPR) and False Positive Rate (FPR) values. The higher TPR values show a higher positive rate, which represents optimized performance, and the higher FPR values show a higher negative rate, which is not suitable for performance. In the Tab. 9. SVM shows the highest TPR value, and IF shows the lowest FPR value.

To compare our findings with similar studies, we created a comparison table. A study contains the NSL-KDD and CAIDA dataset with their machine learning classifier are given below [37]:

Table 9: Performance rate for TPR and FPR

	TPR	FPR
SVM	0.98476	0.70027
Logistic regression	0.98328	0.00418
K-nearest neighbors	0.98448	0.00388
Isolation forest	0.87084	0.02583

It can be seen from Tab. 10. that the proposed machine learning models using identified approaches in our study results are better in comparison with the results from the NSL-KDD and CAIDA dataset [27,38]. It can be observed from the literature that similar studies used different models; therefore, it is difficult to compare results exactly. Finally, results show that the network services of the SDN-NFV environment proved successful in terms of detecting anomalies/attacks with machine learning techniques. The performance rate represented that anomalous cases in the dataset can be detected and optimized efficiently by using ML techniques. The algorithms proposed to work are implemented using Python. Several valuable utilities like the MINE package, the open-source Python ML libraries Scikit-learn, NumPy, SciPy, Matplotlib are also used during the simulation process.

Table 10: Performance analysis of different classifier on NSL-KDD and CAIDA datasets

Training dataset	Models	Accuracy (%)	Precision (%)	Recall (%)
NSL-KDD	Dynamic neural network	91.43	97.60	91.43
	Long short-term memory recurrent neural network	93.28	97.60	93.87
	Deep belief networks	95.00	97.42	96.24
	K-nearest neighbors	94.35	93.55	–
	LG	85.00	85.00	87.00
	Decision tree	97.00	88.00	91.00
	Artificial neural network	99.00	86.00	–
CAIDA	Support vector machine	92.4	–	–
	K-nearest neighbors	97.15	–	
	Artificial neural network			
	Naive bayes			

In conclusion, the proposed framework SDN-NFV environment is deployed for monitoring of anomalous cases in network traffic to improve the network services and user quality of service time by using classification models based on machine learning techniques like SVM, LR, KNN, and IF, respectively to minimize or even eliminate human interference. Dataset has been selected that incorporates current attacks that have been infrequently used in the past for other studies. The dataset contains 28 features and five classes. The python platform is utilized in this work because Python can be utilized with high certainty because of its capability of creating legitimate outcomes that mirror a certain domain of intelligence. The gathered information has been recorded for various sorts of attacks and will notify the anomalous cases to the SDN-NFV controller, whereas the non-anomalous cases will be moved forward to the NFV layer for network services with higher accuracy rate. The ML algorithms applied to the identified dataset for grouping various kinds of attacks precisely: HTTP-Flood, UDP-Flood, Smurf, and SiDDoS. The selected ML classifiers accomplished the particular task with accuracy and precision, as discussed above. It will help to characterize the normal network traffic to ensure the network services of the SDN-NFV environment.

Due to the research scope, some common types of attacks and security parameters have been selected, but in the future more categories of innovative attacks and features in SDN/NFV environment can be incorporated to better understand and cope with the identified issues. Furthermore, innovative algorithms can also be designed or existing can be modified to detect a broader range of attacks than already discussed to get far better efficiency and accuracy.

Acknowledgement: Thanks to our families and colleagues who supported us morally.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. K. Das, "European Union's general data protection regulation, 2018: A brief overview," *Annals of Library and Information Studies*, vol. 65, no. 2, pp. 139–140, 2018.

[2] A. Shabbir, M. Shabbir, M. Rizwan and F. Ahmad, "Ensuring the confidentiality of nuclear information at cloud using modular encryption standard," *Security and Communication Networks*, vol. 2019, no. 4, pp. 1–17, 2019.

[3] S. Shah and S. P. Bendale, "An intuitive study: Intrusion detection systems and anomalies, how AI can be used as a tool to enable the majority, in 5G era," in *5th Int. Conf. On Computing, Communication, Control And Automation*, Pune, India, pp. 1–8, 2019.

[4] A. Ahmad, "Evaluation of machine learning techniques for intrusion detection in software defined networking," Faculty of Information Technology and Electrical Engineering, University of Oulu, 2020.

[5] X. Shi, Y. Li, H. Xie, T. Yang, L. Zhang *et al.,* "An openflow-based load balancing strategy in SDN," *Computers, Materials & Continua*, vol. 62, no. 1, pp. 385–398, 2020.

[6] M. S. Kumar, "Analysis of network function virtualization and software defined virtualization," *International Journal on Informatics Visualization*, vol. 1, no. 4, pp. 122–126, 2017.

[7] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[8] G. Allen and T. Chan, *Artificial intelligence and national security.* Belfer Center for Science and International Affairs, Harvard Kennedy School, USA, 2017.

[9] J. Ali, G. M. Lee, B. H. Roh, D. K. Ryu and G. Park, "Software-defined networking approaches for link failure recovery: A survey," *Sustainability*, vol. 12, no. 10, pp. 4255–4282, 2020.

[10] S. Shahzadi, B. Khaliq, M. Rizwan and F. Ahmad, "Security of cloud computing using adaptive neural fuzzy inference system," *Security and Communication Networks*, vol. 2020, no. 8, pp. 1–15, 2020.

[11] A. Rehman, R. L. Aguiar and J. P. Barraca, "Network functions virtualization: The long road to commercial deployments," *IEEE Access*, vol. 7, pp. 60439–60464, 2019.

[12] C. Benzaid and T. Taleb, "ZSM security: Threat surface and best practices," *IEEE Network*, vol. 34, no. 3, pp. 124–133, 2020.

[13] C. T. Yang, S. T. Chen, J. C. Liu, Y. Y. Yang, K. Mitra *et al.,* "Implementation of a real-time network traffic monitoring service with network functions virtualization," *Future Generation Computer Systems*, vol. 93, pp. 687–701, 2019.

[14] I. Abdulqadder, D. Zou, I. Aziz, B. Yuan and W. Dai, "Deployment of robust security scheme in SDN based 5G network over NFV enabled cloud environment," *IEEE Transactions on Emerging Topics in Computing*, pp. 1, 2018.

[15] V. G. Nguyen, A. Brunstrom, K. J. Grinnemo and J. Taheri, "SDN/NFV-based mobile packet core network architectures: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1567–1602, 2017.

[16] M. Bagaa, T. Taleb, J. B. Bernabe and A. Skarmeta, "A machine learning security framework for IoT systems," *IEEE Access*, vol. 8, pp. 114066–114077, 2020.

[17] B. Karan, D. Narayan and P. Hiremath, "Detection of DDoS attacks in software defined networks," in *3rd Int Conf. on Computational Systems and Information Technology for Sustainable Solutions*, Bengaluru, India, pp. 265–270, 2018.

[18] N. Dilawar, M. Rizwan, F. Ahmad and S. Akram, "Blockchain: Securing internet of medical things (IoMT)," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 1, pp. 82–89, 2019.

[19] I. Farris, J. B. Bernabé, N. Toumi, D. Garcia-Carrillo, T. Taleb *et al.,* "Towards provisioning of SDN/NFV-based security enablers for integrated protection of IoT systems," in *IEEE Conf. on Standards for Communications and Networking*, Helsinki, Finland, pp. 169–174, 2017.

[20] D. Behnke, M. Muller, P. B. Bok, S. Schneider, M. Peuster *et al.*, "NFV-driven intrusion detection for smart manufacturing," in *2019 IEEE Conf. on Network Function Virtualization and Software Defined Networks*, Dallas, TX, USA, pp. 1–6, 2019.

[21] A. M. Zarca, J. B. Bernabe, A. Skarmeta and J. M. A. Calero, "Virtual IoT honeynets to mitigate cyberattacks in SDN/NFV-enabled IoT networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1262–1277, 2020.

[22] I. H. Abdulqadder, S. Zhou, D. Zou, I. T. Aziz and S. M. A. Akber, "Multi-layered intrusion detection and prevention in the SDN/NFV enabled cloud of 5G networks using AI-based defense mechanisms," *Computer Networks*, vol. 179, 107364, 2020.

[23] C. Cath, S. Wachter, B. Mittelstadt, M. Taddeo and L. Floridi, "Artificial intelligence and the 'good society': The US, EU, and UK approach," *Science and Engineering Ethics*, vol. 24, no. 2, pp. 505–528, 2018.

[24] F. Ahmad and K. Ahmed, "Holographic interface management in the age of artificial intelligence," *International Journal of Computer Science and Network Security*, vol. 17, no. 3, pp. 77–86, 2017.

[25] A. Shabbir, M. Shabbir, M. Rizwan and F. Ahmad, "Neuro-biological emotionally intelligent model for human inspired empathetic agents," *Journal of Cognitive Systems*, vol. 4, no. 1, pp. 1–11, 2018.

[26] S. Ayoubi, N. Limam, M. A. Salahuddin, N. Shahriar, R. Boutaba *et al.,* "Machine learning for cognitive network management," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 158–165, 2018.

[27] D. S. Wei, K. Xue, R. Bruschi and S. Schmid, "Guest editorial leveraging machine learning in SDN/NFV-based networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 245–247, 2020.

[28] A. Pastor, A. Mozo, D. R. Lopez, J. Folgueira and A. Kapodistria, "The mouseworld, a security traffic analysis lab based on NFV/SDN," in *Proc. of the 13th Int. Conf. on Availability, Reliability and Security*, Hamburg, Germany, pp. 1–6, 2018.

[29] Z. Khalid, M. Rizwan, A. Shabbir, M. Shabbir, F. Ahmad *et al.,* "Cloud server security using bio-cryptography," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, pp. 166–172, 2019.

[30] S. Lee, C. Yoon, C. Lee, S. Shin, V. Yegneswaran, "DELTA: A security assessment framework for software-defined networks," in *Network Distributed System Security*, San Diego, California, USA, 2017.

[31] S. Nadeem, M. Rizwan, F. Ahmad and J. Manzoor, "Securing cognitive radio vehicular *ad hoc* network with fog node based distributed blockchain cloud architecture," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 1, pp. 288–295, 2019.

[32] M. S. Bonfim, K. L. Dias and S. F. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," *ACM Computing Surveys*, vol. 51, no. 6, pp. 1–39, 2019.

[33] P. Krishnan, S. Duttagupta and K. Achuthan, "SDN/NFV security framework for fog-to-things computing infrastructure," *Software: Practice and Experience*, vol. 50, no. 5, pp. 757–800, 2020.

[34] A. Khraisat, I. Gondal, P. Vamplew and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 2–22, 2019.

[35] C. Liu, G. Zhang, B. Li, R. Ma, D. Jiang *et al.,* "A SDN-based intelligent prediction approach to power traffic identification and monitoring for smart network access," *Wireless Networks*, vol. 67, no. 4, pp. 1–12, 2020.

[36] C. M. Mathas, O. E. Segou, G. Xylouris, D. Christinakis, M. A. Kourtis *et al.*, "Evaluation of Apache spot's machine learning capabilities in an SDN/NFV enabled environment," in *Proc. of the 13th Int. Conf. on Availability, Reliability and Security*, Hamburg, Germany, pp. 1–10, 2018.

[37] I. P. Prasanna and M. Suguna, "Detection of distributed denial of service attack using NSL-KDD dataset: A survey," in *Int. Conf. on Computer Networks, Big Data and IoT*, India, Springer, pp. 866–875, 2019.

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