IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 27, Issue 1, Ser. 3 (Jan. – Feb. 2025), PP 33-45
www.iosrjournals.org

SDN Based Machine Learning Model For Intrusion Detection With WEKA Environment

Ahmad Ajiya Ahmad[1], Prof. Souley Boukari[2], Abdullahi Musa Bello[3], Sa'adatu Gimba[4], Abubakar M. Bichi[5], Mustapha A. Muhammad[6]

[1, 3, 4, 5,](Department Of Computer Science, Faculty Of Science, Federal University Gashua, Yobe State, Nigeria)
[2, 6](Mathematical Science Department, Abubakar Tafawa Balewa University, Bauchi Sate, Nigeria)

Abstract:
The proliferation of telecommunication technologies nowadays brings in more sophisticated and more dynamic systems that can no longer be control efficiently by traditional network architecture. Cyberspace has become important for securing information systems and data sharing. Nevertheless, Data encryption, authentication and firewall are not sufficient for internet security as attackers constantly produces advanced techniques and tools for cyber-attacks. To handle the deficit of traditional network design, a dynamic and robust networking technology, known as Software-Defined Networking architecture (SDN) has developed. However, the emerging technology also produced several security threat and malicious activities. SDN take advantage of an Application programming Interface (API) and separate the architectures' control plane from the data plane and offer better feature to handle and manage network security vulnerabilities. This creates an opportunities for the implementation of Intrusion Detection systems (IDSs). This paper presents an SDN-based intrusion detection model that integrates machine learning (ML) algorithms, implemented using the WEKA environment, to enhance network security. A prominent dataset CSE-CIC-IDS2018 is used to verify the performance of the machine learning model. The model applies classification techniques to detect and mitigate potential security threats in real time. The performance of the machine learning model is measured and evaluated in WEKA application. The experimental result shows that for anomaly detection the proposed scheme performs better with Naive Bayes, as it produced high accuracy, precision, recall and F-measure compare to other classification algorithms. This demonstrates that the model can effectively identify a variety of network attacks, achieving high accuracy with minimal computational overhead.

Key Word*: Cyberspace; Data encryption; firewall; Software-Defined Networking architecture (SDN); Application programming Interface (API); Intrusion Detection systems (IDSs);; CSE-CIC-IDS2018; WEKA application; Naive Bayes;.*

Date of Submission: 29-01-2025 Date of Acceptance: 09-02-2025

I. Introduction

The rapid growth and scalability of traditional network infrastructures have led to significant challenges in managing and mitigating malicious activities [8], [7]. Cyber-attacks have evolved into critical threats for industries and enterprises, often targeting sensitive data through unauthorized access, theft, modification, and damage. Software-Defined Networking (SDN) architecture, while offering centralized control and programmability, introduces new vulnerabilities to its controllers and OpenFlow networks [1]. These vulnerabilities have drawn the attention of researchers, who aim to leverage SDN's architecture to alleviate intrusion attacks. However, attackers can still exploit SDN networks, leading to research efforts to mitigate these security challenges and provide innovative solutions [17].

The SDN architecture separates the network into infrastructure and control layers, offering centralized management and dynamic programmability. This separation allows for improved resource management and reduced cybercrime risks by enabling real-time updates to policies and configurations. However, the dynamic nature of SDN also introduces novel security challenges, such as Denial-of-Service (DoS) threats, which have garnered significant attention from both academia and the IT industry [2], [17]. Researchers have proposed solutions to address these issues, ranging from intrusion detection systems to advanced authentication mechanisms. Despite these efforts, SDN security remains an active area of research, with unresolved problems such as high false positive rates and inefficient detection methods [14], [15].

Various studies have focused on mitigating DoS attacks in SDN environments. For instance, [20] proposed using statistical approaches, such as confidence intervals and mean throughput, to detect anomalies in SDN controllers, improving accuracy and reducing overhead. Similarly, [25] suggested statistical approach involving mean entropy and adjustable window sizes to prevent packet overload at the controller. Although

these methods show promise in addressing specific attack types, they often fail to generalize across diverse threats, making them less effective for large-scale SDN networks. Researchers have emphasized the importance of exploring multiple attack types using comprehensive datasets to improve detection accuracy and reduce false alerts [20].

The lack of realistic and comprehensive datasets for training machine learning algorithms in SDN environments poses a significant challenge. Existing public datasets, such as KDD Cup 99 and NSL-KDD, are often outdated or tailored to traditional network architectures, making them unsuitable for SDN-specific anomaly detection [6], [9]. Additionally, these datasets typically focus on specific attack types, such as DoS, rather than encompassing a broader range of threats [23], [13]. Addressing this gap requires generating high-quality SDN-specific datasets and applying feature selection techniques to minimize redundancy and irrelevance [27]. Machine learning-based network intrusion detection systems (NIDS) offer a promising solution to enhance detection accuracy and reduce false alarm rates, addressing the persistent challenges in SDN security [17.

This paper proposes an SDN-based intrusion detection model that leverages ML to improve the detection and mitigation of network threats. Using WEKA, a widely used tool for machine learning and data mining, we integrate various classifiers, such as Decision Trees, Naive Bayes, k-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM), into the proposed system. This work aims to fill the gap by providing an adaptive, scalable, and efficient intrusion detection system capable of identifying complex attack patterns within SDN environments. The objective is to enhance network security without imposing high computational costs or requiring significant manual configuration.

II. Objectives

1. To design and implement an SDN-based intrusion detection system (IDS) integrated with machine learning algorithms.
2. To evaluate the performance of various machine learning classifiers in detecting network intrusions.
3. To develop an adaptive and scalable intrusion detection model for SDN environments.

III. Software-Defined Networking (SDN) Architecture

Software-Defined Networking (SDN) revolutionizes traditional network architectures by decoupling the control plane from the data plane [15], [22]. This separation fundamentally changes how networks are designed, managed, and operated. In traditional networking, control functions, which determine how data is routed and managed, are embedded within individual network devices like switches and routers. SDN introduces a more centralized and software-driven approach by moving the control logic to a centralized software controller, resulting in greater flexibility, programmability, and dynamic network management. This paradigm shift facilitates centralized control, dynamic reconfiguration, and enhanced visibility into network operations. Figure No. 1 shows overview of SDN Architecture.

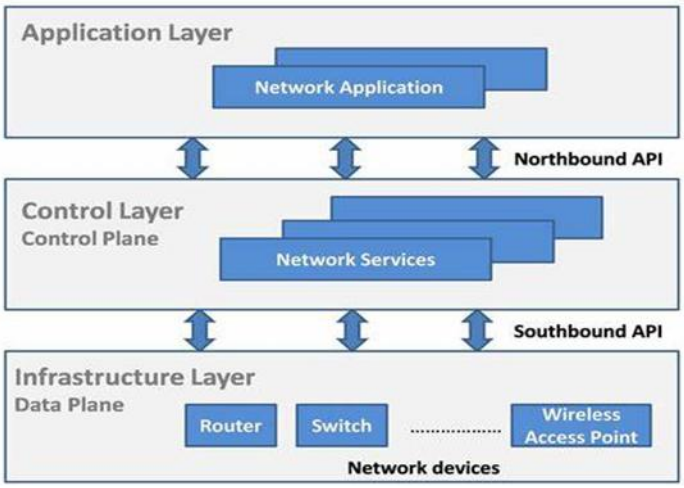

Figure No. 1: Overview of SDN Architecture

SDN Architecture Layers

The SDN architecture comprises three primary layers, namely, Infrastructure Layer (Data plane), Control Layer (Control Plane) and Application Layer. The control plane centralized SDN controllers and is responsible for making decisions about network behavior, such as routing, traffic prioritization, and security policies. The data plane, on the other hand, consists of network devices like switches and routers and handles the

actual forwarding of data packets. The key distinction is that the control plane is abstracted from the hardware devices and managed by the SDN controller. Application Layer hosts various applications that interact with the controller to implement network functionalities.

SDN Architecture Attacking Targets of Security Threats

All the components of the SDN architecture can be targeted by various security threats. Though, the controller and control layer bandwidth are the most considerate aim points for attacks [3].

SDN Switch: SDN switches are responsible for forwarding and processing of fresh incoming traffics. The immense security concern in SDN switch is that, they have inadequate flow table size [6].

SDN Controller: The controller is the brain power of SDN network that executes vital actions for SDN, any irregularity in it can hinder the whole performance of the network. The whole functionality of a SDN network rest on the controller, conversely, the attackers are very attractive to it as target point [17].

Route between SDN Switches: The intruder may easily intercept these packets when the route between switches is wireless [3].

Route between Two Controllers: The route between two controllers has potential for attackers to easily gain access to essential information, it is important to protect the communication between the controllers to be protected and authentic [7].

Route between Controller and Switch: In the situation where a packet reaches at a switch and a switch is incapable to treat it, then the packet is forwarded to the controller for additional processing. Subsequently, fresh packet forwarding rules are attached to the flow table of the switches [3]. These data packets can be delayed and altered with new malicious rule by an intruder to change the direction of the traffic dataflow [16].

Applications: Most of the applications implemented in an SDN network are developed by third parties that are not attentive for the security necessities. This could lead to a security threat such as unauthorized access and information compromising [1]. These applications can turn out to be the easiest target point for hindering service of controllers [17].

SDN's ability to provide real-time insights and control over network behavior makes it a promising candidate for intrusion detection [31]. Intrusion Detection Systems (IDS) play a pivotal role in ensuring the security of modern networks by identifying malicious activities or policy violations [18]. By examining the literature, this review delineates various methodologies employed in intrusion detection, with a special emphasis on Machine Learning (ML) techniques and their integration with Software-Defined Networking (SDN) for improved detection mechanisms.

IV. Machine Learning And Intrusion Detection

Machine Learning (ML) has emerged as a transformative tool in the domain of intrusion detection, offering capabilities to discern patterns and anomalies within network traffic. Both supervised and unsupervised ML algorithms are extensively utilized to enhance the accuracy and efficiency of intrusion detection systems.

Supervised Learning

Supervised ML techniques, such as Decision Trees, Random Forest, and Support Vector Machines (SVM), classify network traffic based on labeled training data. Decision Trees and Random Forest, in particular, have gained prominence for their interpretability and robustness. Neural Networks, leveraging their deep learning capabilities, are also increasingly adopted for complex pattern recognition tasks [3].

Unsupervised Learning

Unsupervised ML algorithms, including Isolation Forest and clustering techniques, excel in anomaly detection by identifying deviations from established network behavior. These methods are especially useful in detecting previously unknown attack patterns.

Despite their advantages, ML techniques face limitations such as computational overhead and susceptibility to adversarial attacks. Consequently, this review underscores the importance of integrating ML with SDN to address these challenges.

Integration of SDN and ML for Intrusion Detection

The convergence of SDN and ML presents a synergistic approach to intrusion detection [10]. By leveraging SDN's centralized control and ML's pattern recognition capabilities, this integration enables proactive and adaptive security mechanisms. Previous studies have demonstrated the efficacy of combining these paradigms to detect and mitigate a wide range of cyber threats.

SDN Network Attack Cases

The unique architecture of SDN introduces novel vulnerabilities, leading to distinct classes of attacks compared to traditional networks and some of these cyber threats are discussed below.

Denial-of-Service (DoS) Attacks: DoS attacks are prevalent in SDN environments, targeting the controller's resources and rendering the network inaccessible to legitimate users [18]. These attacks can involve flooding the controller with "packet-in" messages, overwhelming its capacity to process legitimate traffic [5].
Distributed Denial-of-Service (DDoS) Attacks: A DDoS attack involves a large number of compromised hosts flooding the target system with attacks and rendering it inoperable and unmanageable [21]. These attacks often exploit vulnerabilities in network protocols such as ICMP, UDP, and TCP [18].
Probe Attacks: Probe attacks involve scanning the target network to gather information about its structure, operating systems, and open ports [20].
Man-in-the-Middle (MITM) Attacks: In MITM attacks, attackers intercept and manipulate communication between a client and server, potentially stealing sensitive information or injecting malicious traffic [5].
SQL Injection Attacks: SQL injection attacks exploit vulnerabilities in database-driven applications, allowing attackers to execute unauthorized SQL commands and access sensitive data [21], [8].
Eavesdropping Attacks: Eavesdropping attacks involve intercepting network traffic to steal confidential information. These attacks can be passive or active, depending on the attacker's involvement [31.
Malware Attacks: Malware attacks introduce malicious software into systems to compromise their functionality. Common types of malware include viruses, worms, Trojans, and macro viruses [8].

Machine Learning Classifiers

The timely identification of intrusions is critical to network security. This study evaluates the performance of various ML classifiers, including J48, Random Forest, Isolation Forest, AdaBoost.M1, and Naive Bayes, using the CSE-CIC-IDS-2018 dataset.
J48 (C4.5 Decision Tree): J48 is a Decision Tree classifier implemented in WEKA, designed to create decision trees based on input datasets. Its inherent structure makes it a reliable choice for network traffic analysis.
Random Forest: Random Forest employs multiple decision trees on different data subsets, enhancing prediction accuracy through an ensemble approach.
Isolation Forest: Isolation Forest specializes in anomaly detection by isolating rare instances within the data. Its performance in detecting deviations from normal traffic patterns makes it a valuable tool for intrusion detection.
AdaBoost.M1: AdaBoost.M1 is a boosting algorithm that iteratively combines weaker models to create a robust classifier. Its application in intrusion detection remains underexplored, offering a promising avenue for research.
Naive Bayes Algorithm: Naive Bayes is a probabilistic classifier that assumes feature independence. Its simplicity and computational efficiency make it a popular choice for predictive modeling in intrusion detection.

V. Related Work

There have been a various researches proposed models on intrusion detection with machine learning in SDN infrastructure, some of those researches are discussed in this section. [3] Has reported that, SDN lacked a mechanism to detect abnormal traffic behaviour and to separate legitimate traffic from attack traffic while improving and maintaining system performance. This remains as a big issue in an SDN network [12]. Therefore, this study works to improve robustness and security by proposing machine learning based NIDS model for detecting attacks and separating intrusive attack from normal attacks in the SDN network. It involves monitoring the network traffic behaviour for abnormality.

Furthermore, According to [27] High Quality training datasets is required because reducing the large number of false alerts and increasing accuracy during the process of detecting unknown attack patterns remains unresolved problem. Unfortunately, the dataset available have deficiencies, correlated dataset and applicable to only DoS attack [25]. Therefore, it's important to reduce data attributes and select the most significant features or attributes to improve dataset efficiency. In this paper, different feature selection methods are used to reduce the number of attributes in our dataset. [19] Developed a system that can efficiently detect and mitigate network intrusions in an SDN infrastructure. However, the author explored weakness, that is, limited to DoS attack and cannot manage complex SDN design.

[30] Proposes an enhanced SDN Intrusion Detection System for SDN that used Machine Learning (ML) classification classifiers for detecting DDoS attacks. The research noted that, its limited to only DDoS attack and data sharing is not relevant to this research as no available datasets were collected or analyzed in the study. Thus, this research cannot efficiently detect intrusion on real time in SDN. [32] Proposed an intrusion detection model that applied machine learning classifiers to classify infections in an SDN networks. This model has limitations on its application and scalability, as it cannot manage large scale SDN network design.

[17], [18], [25] used one type of attack e.g. DoS, to test the accuracy of intrusion detection using machine learning. Since, the accuracy with other recent attacks is still not known. There is tendency to be low with other attacks [26]. Therefore, there is need to use different categories of attacks for determining the accuracy of machine learning intrusion detection. We don't know how they will behave based on the effectiveness of the machine learning algorithms. This paper will provide the solution to this problem by using the most recent dataset CSE-CIC-IDS2018. It's a qualitative and comprehensive SDN dataset with numerous type of attack for anomaly detection and better performance evaluation in SDN environment. By synthesizing and analyzing existing knowledge, the chapter positions the research within the broader landscape of intrusion detection, highlighting the need for innovative solutions to counter evolving cyber threats. Hence, this chapter sets the stage for the subsequent exploration of the proposed integration of SDN and IDS with ML to improve the security of SDN using Machine Learning classification

VI. Methodology

The proposed SDN-based intrusion detection system leverages the WEKA environment for the implementation and evaluation of machine learning classifiers. According to Svetlana, (2004) WEKA is a machine learning (ML) developed by the University of Waikato in New Zealand that also implements data mining algorithms. It was designed with Java programming language and can be accessed through the Graphical User Interface (GUI). It provides several functions include, preprocessing, visualization, classification, regression, clustering, feature selection, and reduction. WEKA is a state of the art facility that is used in this research for developing the data mining tasks, creating applications to real-world problems and applied straight to a dataset. This section explains the stages follow for building, training and testing the proposed ML model. These stages are data preprocessing, feature reduction, cross validation technique and result analysis. After WEKA installation, the GUI chooser permits the user to choose from the five types of applications as shown in Figure No. 2 below.

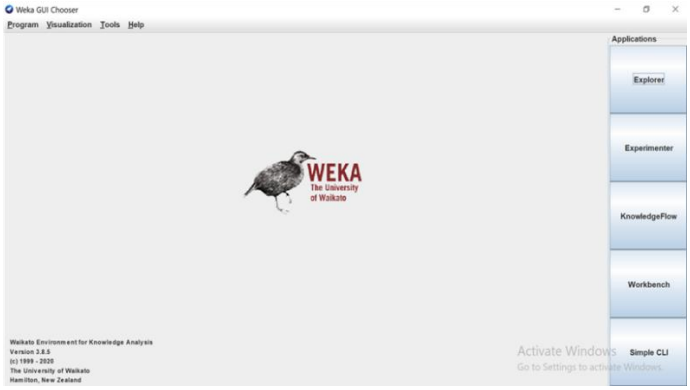

Figure No. 2: WEKA Graphical user interface.

The experiments in this research were performed using the explorer application. At first, only the preprocess tab is enabled which permits users to upload and preprocess the dataset. From a local file directory we Open file tab and uploaded our CSV dataset file. WEKA application supports a different data file formats as shown in Figure No. 3.

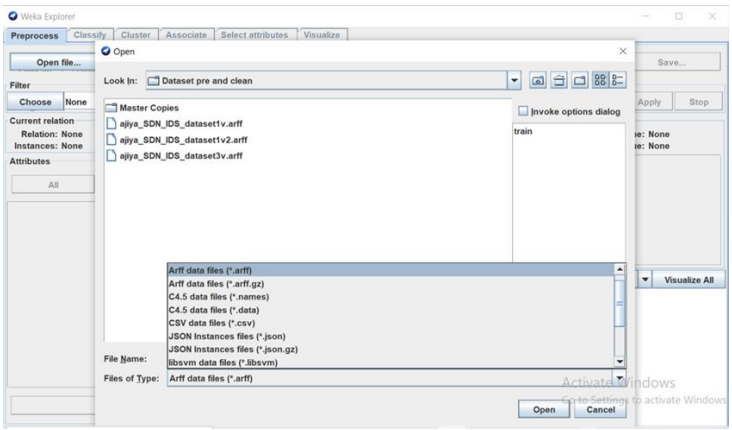

Figure No. 3: Uploading dataset file with various file formats in WEKA

The methodology consists of several stages, including SDN setup, data collection, feature selection, and model training and evaluation.

SDN Setup

To simulate an SDN environment, we use Mininet, a network emulator that allows the creation of realistic SDN topologies. Mininet is used to emulate a network consisting of switches, hosts, and controllers. The SDN controller is configured to monitor and manage network traffic, while flow data is collected from the network to form the dataset used for training the ML models [9].

Data Collection and Feature Extraction

This research utilized machine learning (ML) techniques on a subset of the CSE-CIC-IDS2018 dataset, recognized as a benchmark for predicting different types of network attacks. Developed collaboratively by the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE), the CSE-CIC-IDS2018 dataset is one of the most contemporary resources available for intrusion detection testing [11]. Its primary objective is to provide detailed, low-level data regarding network traffic, encompassing both normal (non-malicious) and malicious activities. The dataset's distribution includes 13,484,708 instances of legitimate traffic (83.32%) and 2,249,612 instances of malicious traffic (16.68%), as outlined in Table No. 1.

Table No. 1: CSE-CIC-IDS2018 dataset Class Distribution

Class	Frequency	Percentage
Number of legitimate Traffic	13,484,708	83.32%
Number of Malicious Traffic	2,249,612	16.68%

The dataset is formatted as comma-separated values (CSV), making it compatible with the WEKA package and other data analysis tools. This format allows researchers to apply a variety of programming languages and execute diverse classifiers with ease and flexibility. The data within the CSE-CIC-IDS2018 dataset is generated from simulated network attacks in a controlled environment [29]. It provides network features extracted from both benign and attack traffic, encompassing 14 distinct attack types. These include Distributed Denial of Service (DDoS), PortScan, Infiltration, SSH-BruteForce, FTP-BruteForce, Brute Force-Web, Brute Force-XSS, among others.

For this study, the dataset was restructured into a binary classification format by consolidating all attack types into a single "anomaly" class. This approach simplifies the classification process, focusing on the differentiation between normal and anomalous network behavior [30]. The rich and diverse nature of the dataset ensures a comprehensive representation of network traffic patterns, making it ideal for ML-based intrusion detection system development. Further details regarding the types of attacks included in the dataset are illustrated in Figure No. 4.

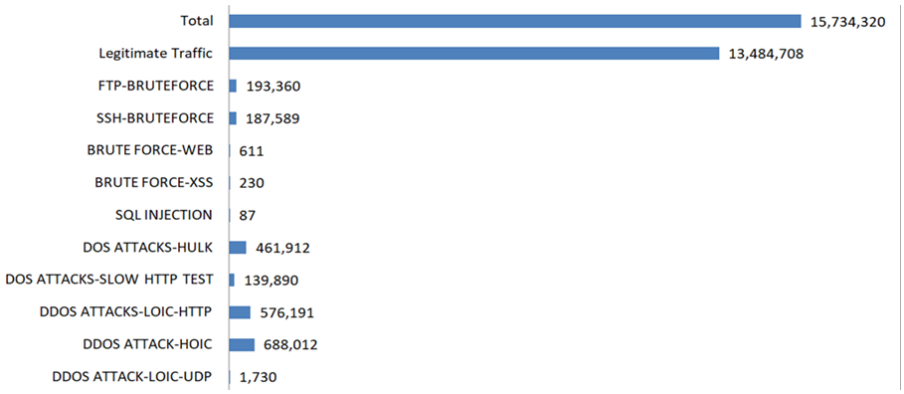

Figure No. 4: Number of CSE-CIC-IDS2018 dataset instances

This comprehensive dataset not only provides a basis for effective ML model training but also enables the analysis of key features critical for anomaly detection. By leveraging these features, this study aims to enhance the performance of intrusion detection systems in Software-Defined Networking (SDN) environments.

Data Processing

Data preprocessing is a critical stage in developing a robust Machine Learning (ML) model. This phase includes essential steps such as data cleaning and standardization to ensure the quality and consistency of the dataset.

During data cleaning, unrelated, incorrect, incomplete, and irrelevant data points are identified and addressed through cleansing, replacement, or deletion. Missing values in the dataset are replaced with mean values, ensuring no gaps in the data. Columns containing infinity values or non-valuable data features are similarly replaced with mean values. Furthermore, columns with zero values are entirely removed to eliminate redundancy. These preprocessing steps significantly reduced the dataset to 2,080,893 instances and 12 attributes, as depicted in Figure No. 5.

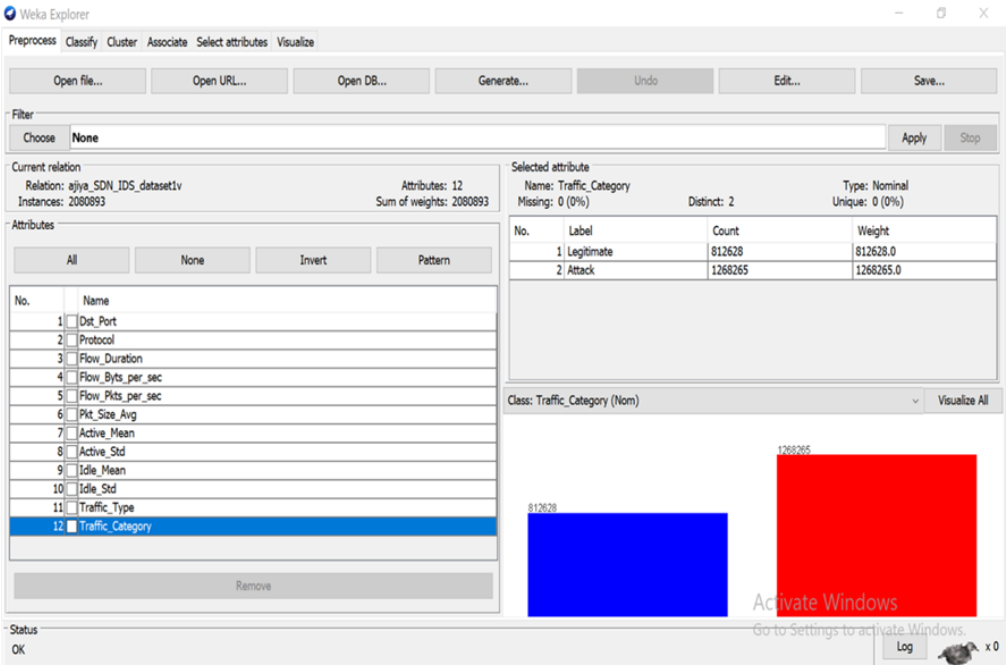

Figure No. 5: Weigh distribution of legitimate and attack traffic in WEKA

Data standardization follows the cleaning process, ensuring uniformity in feature values [28]. This involves transforming the dataset such that the mean value is adjusted to 0, and the standard deviation is scaled to 1. This step is crucial for achieving consistency in the dataset, which enhances the performance of ML algorithms by mitigating biases from disparate feature scales.

The preprocessing phase is supported by the WEKA application, which provides a comprehensive overview of the dataset through the "Preprocess" tab. This interface outlines key dataset characteristics, including the number of instances, attributes, missing features, instance counts, instance weights, and labels. An example of the dataset overview as presented in the WEKA preprocess application is illustrated in Figure No. 6.

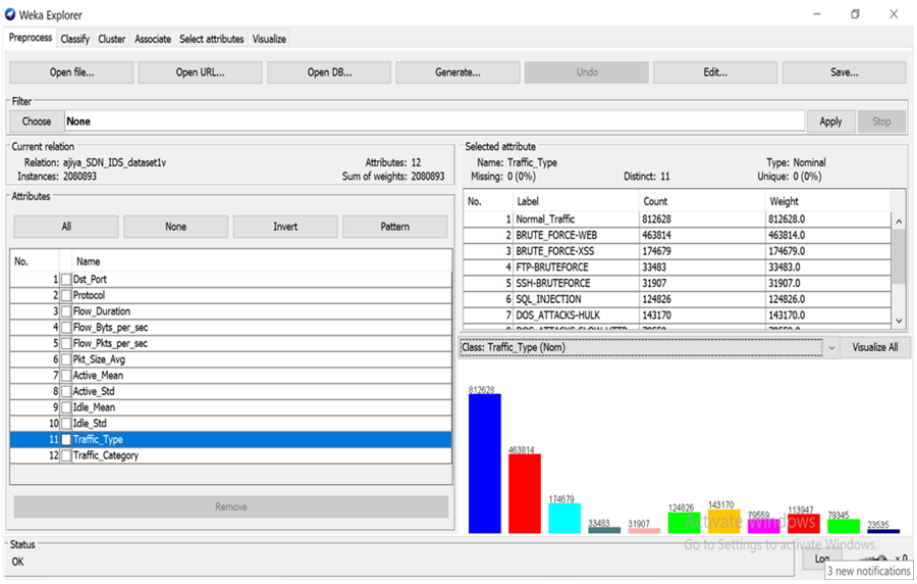

Figure No. 6: The overview of dataset class distribution in WEKA

By executing these preprocessing steps, the dataset is optimized for subsequent ML model development, ensuring improved performance and reliable results.

Feature Selection

Feature selection is an essential step in predictive modeling, involving the selection of relevant variables or attributes from a dataset to improve the accuracy and efficiency of machine learning algorithms. According to [6], this process identifies and retains only the variables that contribute meaningfully to the predictive modeling problem. It also removes redundant or irrelevant features that could otherwise degrade model performance. [17] Highlight that the presence of redundant attributes in intrusion datasets reduces the reliability and consistency of anomaly detection systems.

In this study, Principal Component Analysis (PCA) is employed as a feature selection algorithm to enhance model efficiency. PCA reduces the dataset by transforming a large set of correlated features into a smaller set of uncorrelated features known as principal components [17]. This transformation simplifies the dataset while retaining its most critical information, thus accelerating machine learning processes and improving predictive accuracy.

The PCA process involves multiple stages. First, the dataset is standardized to regularize feature values using the default standardization method available in the WEKA application. Next, a correlation matrix is computed to identify relationships between features, providing insight into feature dependencies. A rank filter is then applied within the PCA framework to order features by importance, with features ranked from highest to lowest based on their contribution to the output.

Features with lower rankings are removed due to their minimal significance. The WEKA application facilitates PCA implementation through its preprocess tab under the unsupervised directory. The application enables users to identify and retain the most relevant features. As a result of applying PCA, this study reduced the dataset to the nine highest-ranked features in WEKA, as shown in Figure No. 7. Additionally, the number of CSE-CIC-IDS2018 dataset instances after PCA feature reduction is illustrated in Figure No. 8, demonstrating the effectiveness of this feature selection approach in optimizing the dataset for machine learning analysis.

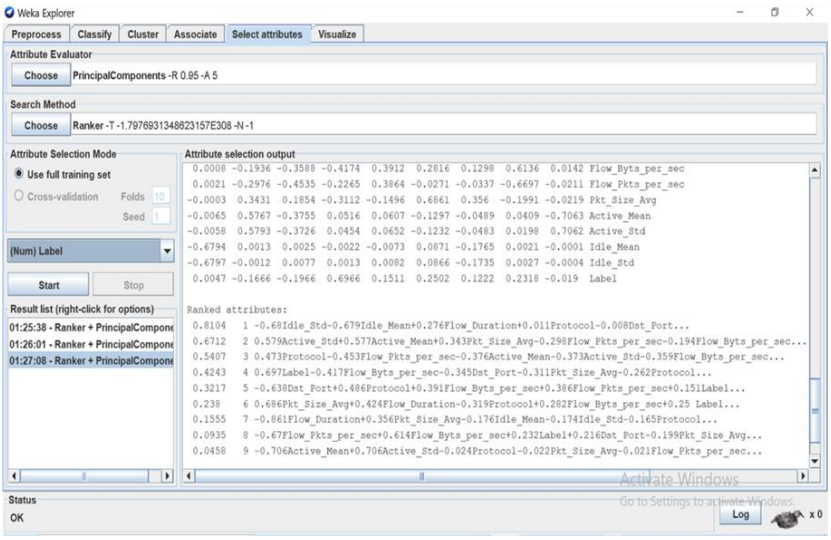

Figure No. 7: PCA feature reduction in WEKA

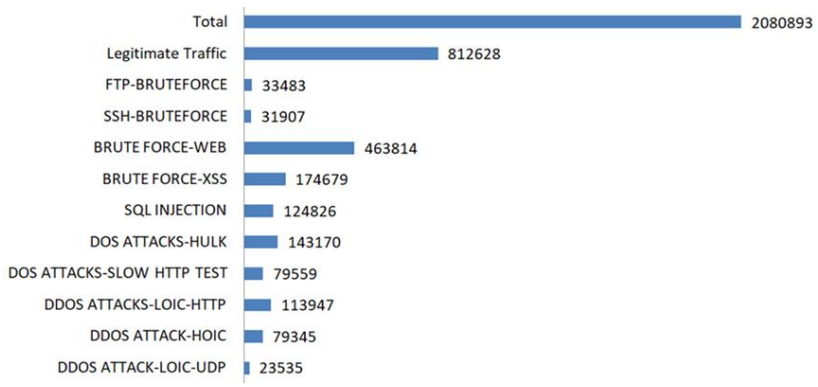

Figure No. 8: Number of CSE-CIC-IDS2018 dataset instances After PCA feature reduction

Machine Learning Classifiers

We test several classifiers in WEKA, including Decision Trees, Naive Bayes, KNN, Random Forest, and SVM. These algorithms are chosen due to their popularity and proven effectiveness in intrusion detection applications. Decision Trees are used for their simplicity and interpretability, while SVM and Random Forest are chosen for their ability to handle complex, high-dimensional data and provide robust performance [4].

Evaluation Metrics

The performance of each classifier is evaluated using standard metrics, including accuracy, precision, recall and F1-score. These metrics are used to assess the model's ability to correctly classify normal and attack traffic while minimizing false positives and false negatives [24].

Accuracy (AC): Accuracy defines the number of true positive assessment and number of negative assessment by number of all assessments.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad Equation \quad (1)$$

Precision (P): Precision shows the proportion of the network intrusion detection system detected intrusions that are real intrusion. High value of precision lowers the false alarm rate.

$$Precision\ (P) = \frac{TP}{TP + FP} \qquad Equation \quad (2)$$

Recall (R): Recall (R) shows the proportion of correctly classified positive examples. We are in search of a high value of R.

$$Recall\ (P) = \frac{TP}{TP + FN} \qquad Equation \quad (3)$$

F-measure (F): By taking the balance among accuracy and recall, it gives a better measure of accuracy. We are in seeking of a high F-measure value.

$$F = 2\,TP/(2\,TP + FP + FN) \qquad Equation \quad (4)$$

Cross Validation Technique

WEKA provides two primary methods for training and testing machine learning models: cross-validation and percentage split. Cross-validation is a robust evaluation technique that partitions the original dataset into training and testing sets, using a specified number of folds. In contrast, the percentage split method divides the dataset into distinct training and testing subsets based on a predefined ratio.

Cross-validation utilizes a key parameter, *k*, which represents the number of folds into which the dataset is divided. Each fold is used as a testing set once while the remaining folds serve as the training set, ensuring that every data point is utilized for both training and testing purposes. This iterative approach enhances the reliability of the evaluation by reducing the impact of variance due to data partitioning [17].

For this research, the k-fold cross-validation technique was employed with $k = 10$. This means the dataset was randomly partitioned into 10 equally sized subsets. Each subset was used as a test set while the remaining nine subsets served as the training set, and the process was repeated for all folds. This comprehensive evaluation ensures that the classifiers' performance is assessed accurately across the entire dataset. Figure No. 9 illustrates the test option selection interface for cross-validation in the WEKA application.

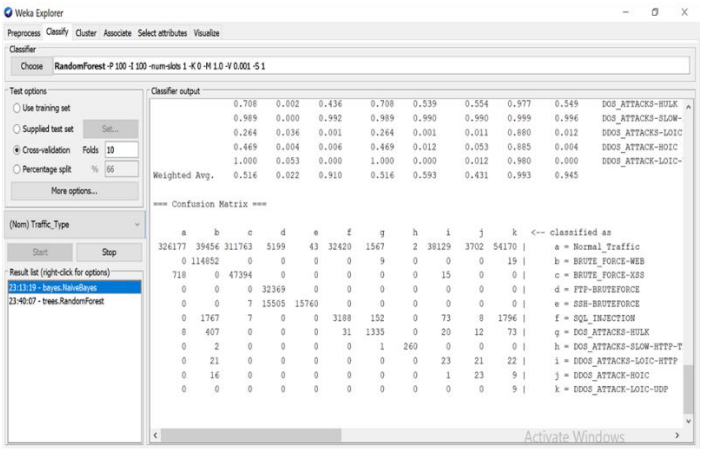

Figure No. 9: Classify tab with confusion matrix and cross-validation selection

VII. Results And Discussion

Performance Evaluation of the Proposed ML Model

This section evaluates the performance of the proposed SDN intrusion detection model using accuracy, precision, recall, and F-measure as performance metrics. The assessment was conducted with the CSE-CIC-IDS2018 dataset in the WEKA application, utilizing Principal Component Analysis (PCA) for feature selection. Machine learning classifiers, including Random Forest, Isolation Forest, J48, AdaBoost.M1, and Naive Bayes, were tested and compared in this research. Figures No. 10 – 13 provide a visual representation of the evaluation outcomes across these metrics.

Accuracy: The accuracy metric determines the fraction of correctly classified packets in the dataset, as defined in equation (1). It measures the classifier's ability to identify legitimate and malicious traffic accurately. Figure No. 10 illustrates the accuracy values for various algorithms. Naive Bayes achieved the highest accuracy at 98.72%, slightly surpassing Random Forest, which recorded 98.69%. Conversely, Isolation Forest demonstrated the lowest accuracy at 84.94%. The accuracy values for AdaBoost.M1 and J48 were 91.63% and 96.31%, respectively, reflecting their moderate performance.

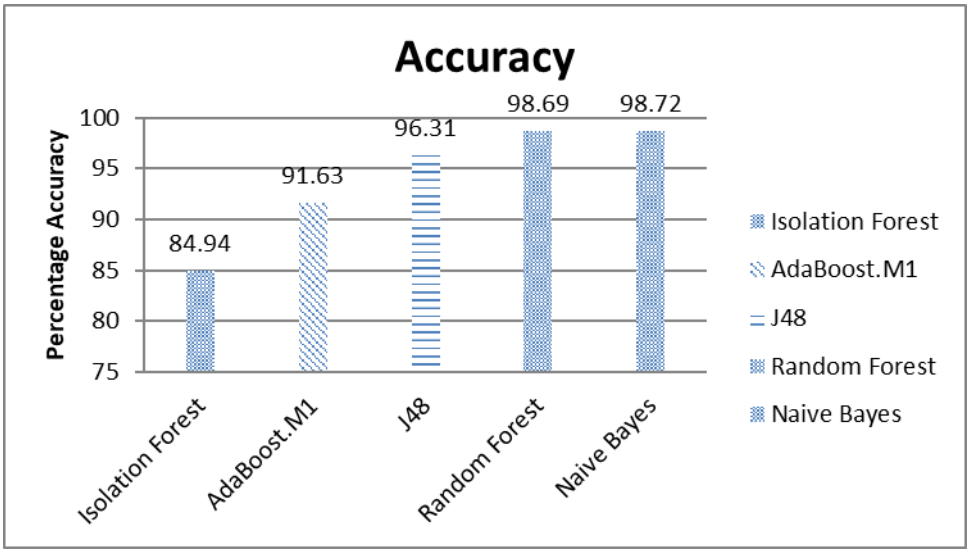

Figure No. 10: Accuracy

Precision: Precision evaluates the reliability of the model when identifying positive instances and is calculated as per equation (2). It represents the proportion of correct positive predictions out of all predicted positives. As shown in Figure No. 11, Naive Bayes achieved the highest precision of 97.45%, followed by Random Forest with 95.53%. AdaBoost.M1 and J48 achieved precision values of 92.39% and 93.35%, respectively, while Isolation Forest displayed the lowest precision at 84.31%.

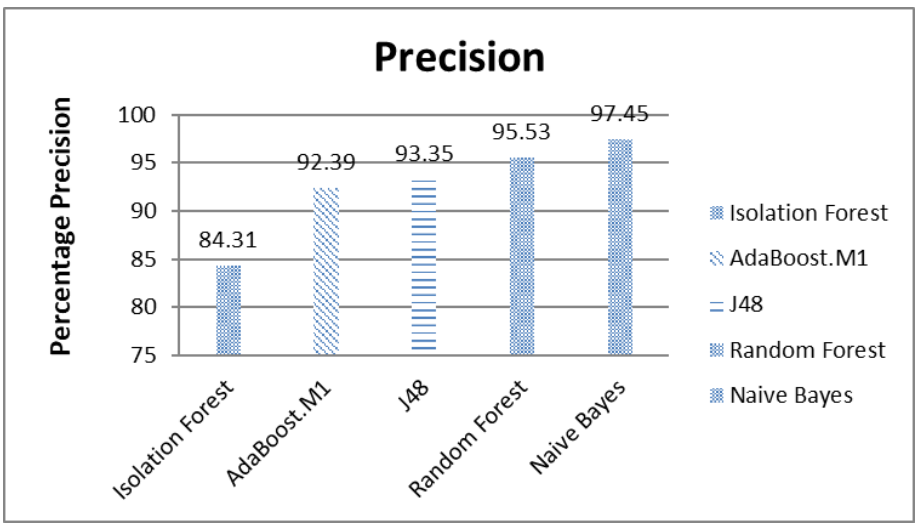

Figure No. 11: Precision

Recall: Recall assesses the model's efficiency in identifying relevant instances in the dataset, as defined in equation (3). Figure No. 12 highlights that Naive Bayes performed the best recall value, achieving a recall of 97.92%. Random Forest followed closely with a recall of 96.71%. AdaBoost.M1 and J48 demonstrated recall values of 93.51% and 94.07%, respectively. Isolation Forest again recorded the lowest recall value at 83.21%, indicating its lower effectiveness in detecting relevant traffic.

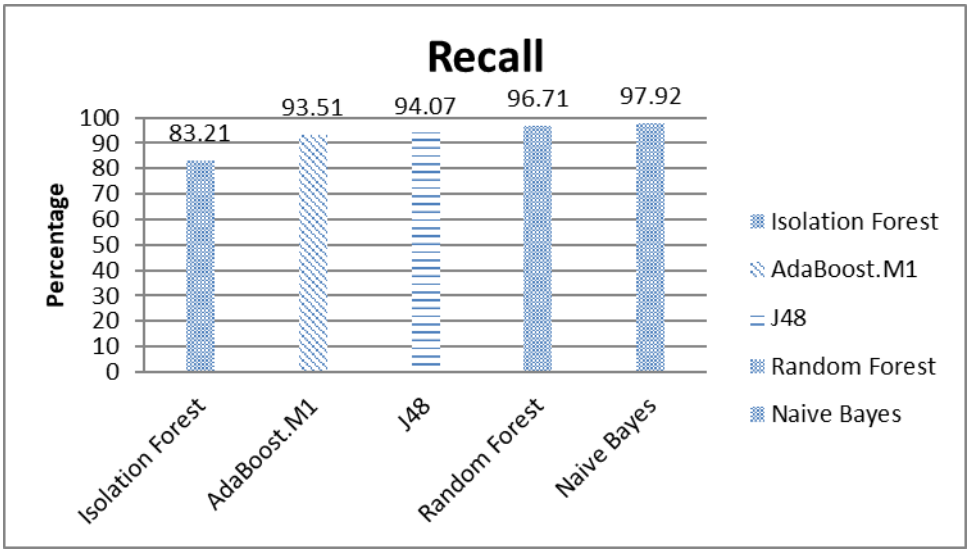

Figure No 12: Recall

F-measure: The F-measure provides a balanced evaluation by combining precision and recall, as described in equation (4). This metric evaluates both the correctness of positive classifications and the classifier's ability to identify all positive instances. Figure No. 13 shows that Naive Bayes and Random Forest tied for the highest F-measure at 98.23%. J48 followed with 96.66%, while AdaBoost.M1 scored 95.42%. Isolation Forest showed the lowest performance with an F-measure of 83.32%.

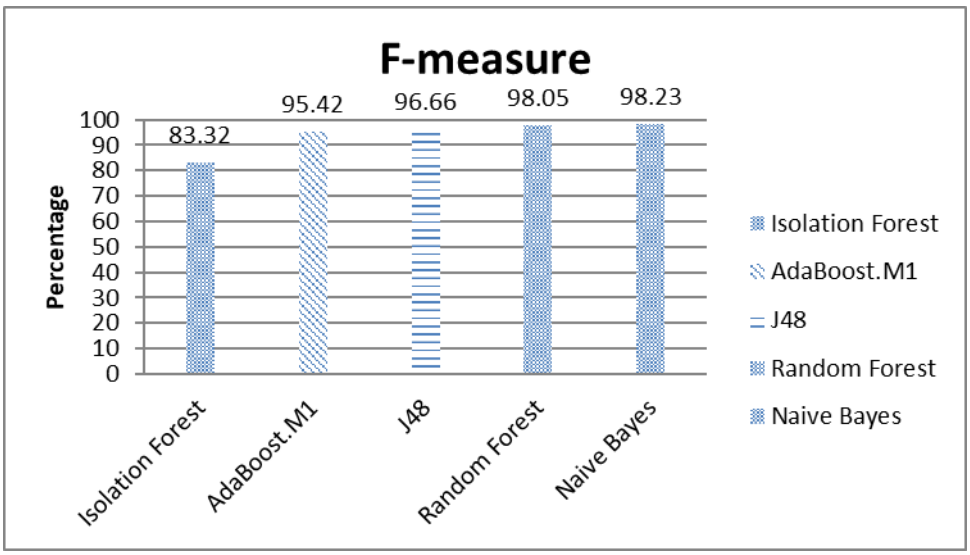

Figure No. 13: F-measure

From the performance results, Naive Bayes and Random Forest emerged as the top-performing classifiers, consistently achieving high values across all metrics, including accuracy, precision, recall, and F-measure. These results highlight their suitability for intrusion detection in SDN environments, while other algorithms demonstrated varying levels of effectiveness.

Performance of Classifiers

This study introduces a machine learning-based classification framework designed for identifying network intrusions in an SDN environment, leveraging the CSE-CIC-IDS2018 dataset. The dataset simulates

real-world network traffic scenarios, encompassing both normal and attack traffic flows. To ensure robust evaluation, we employed a data splitting strategy, allocating 75% of the dataset for training the machine learning models and the remaining 25% for testing. This approach ensures a balanced and comprehensive assessment of the model's capabilities.

To evaluate the proposed model, key performance metrics accuracy, precision, recall, and F-measure were analyzed. These metrics provide a holistic understanding of the model's ability to differentiate between benign and malicious traffic accurately. Among the classifiers tested, Naive Bayes emerged as a strong performer, achieving high accuracy in the detection of network anomalies. Its elevated accuracy underscores its ability to align predictions with real-world outcomes, making it a valuable choice for anomaly detection in SDN environments.

In addition to Naive Bayes, other classifiers were evaluated, with their results presented in Table No. 2 and Figure No. 14. These visualizations highlight the comparative performance of various machine learning algorithms. The findings suggest that employing Naive Bayes can enhance intrusion detection efficiency in SDN multi-controller models, providing both high accuracy and faster detection rates for improved network security.

Table No. 2: Comparison of different machine learning classification algorithms

Algorithm	Accuracy	Precision	Recall	F-measure
Isolation Forest	84.94%	84.31%	83.21%	83.32%
AdaBoost.M1	91.63%	92.39%	93.51%	95.42%
J48	96.31%	93.35%	94.07%	96.66%
Random Forest	98.69%	95.53%	96.71%	98.05%
Naive Bayes	98.72%	97.45%	97.92%	98.23%

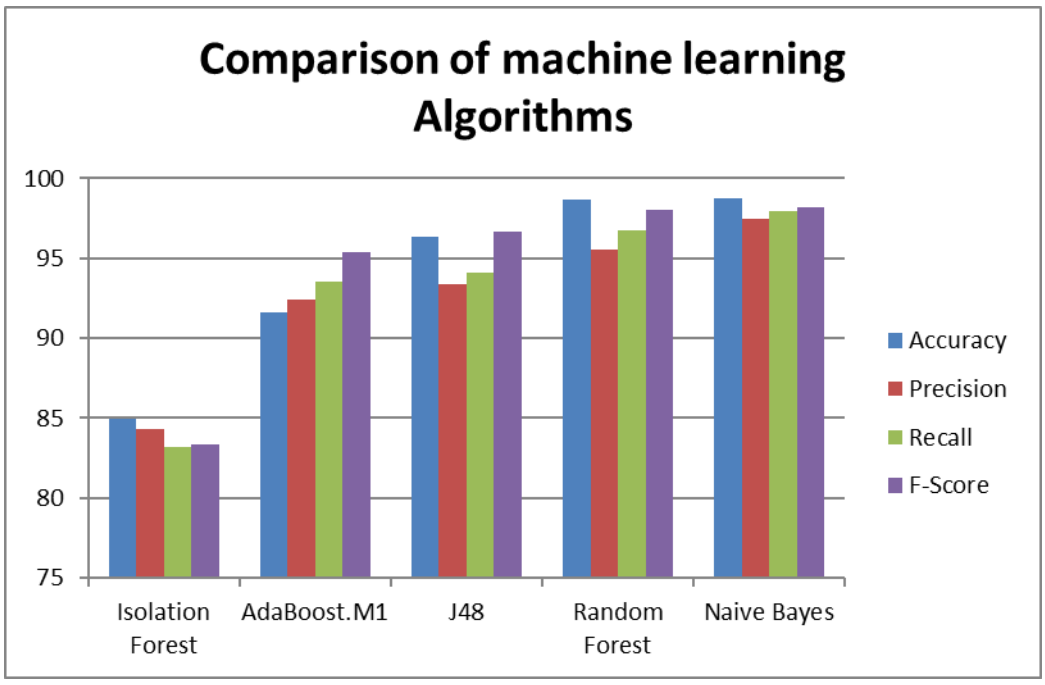

Figure No. 14: Comparison of various machine learning classification algorithms

VIII. Conclusion

This study has presented an SDN-based machine learning model for intrusion detection within Software-Defined Networking (SDN) environments, leveraging the WEKA tool for its implementation. The integration of machine learning algorithms within SDN frameworks offers significant potential in enhancing network security by accurately detecting and mitigating security threats. Through the application of supervised learning algorithms such as Decision Trees, Random Forests, and Support Vector Machines, the model demonstrated promising results in terms of detection accuracy and performance.

The use of the WEKA environment proved to be an efficient and effective approach, allowing for easy manipulation of datasets and quick experimentation with various machine learning models. By evaluating the model’s performance against existing intrusion detection systems, we found that it outperformed traditional methods, offering higher detection rates and lower false positives.

While the results are promising, further research is needed to refine the model by incorporating additional features, exploring deep learning techniques, and optimizing the SDN architecture for scalability and

real-time deployment. Future studies could focus on expanding the dataset to include more diverse network traffic patterns and testing the model in a production environment for real-world applicability.

In conclusion, the research presented contributes to the ongoing development of secure SDN infrastructures and highlights the importance of machine learning in enhancing the reliability and security of modern network systems. The proposed model provides a valuable foundation for future advancements in the field of intrusion detection, offering potential solutions for addressing the growing challenges of network security.

References

[1]. Akhunzada, A., Gani, A., Khan, M. K., Hayat, A., Anuar, N. B., & Shiraz, M. (2019). Securing Software Defined Networks: Taxonomy, Requirements, And Open Issues. Journal Of Network And Computer Applications, 66, 17-35. Https://Doi.Org/10.1016/J.Jnca.2016.11.018

[2]. Alharbi, F., Nguyen, T., & Nguyen, D. T. (2021). A Survey Of Intrusion Detection In SDN: Approaches, Challenges, And Opportunities. Journal Of Network And Computer Applications, 177, 102918.Https://Doi.Org/10.1016/J.Jnca.2021.102918

[3]. Barach, Jay. (2025). Towards Zero Trust Security In SDN: A Multi-Layered Defense Strategy. ICDCN '25: Proceedings Of The 26th International Conference On Distributed Computing And Networking. P. 331 – 339. Https://Doi.Org/10.1145/3700838.37036.

[4]. Bhushan, B., & Jain, A. (2020).A Review Of Machine Learning Methodologies For Network Intrusion Detection Systems In SDN.Journal Of Network And Computer Applications, 174, 102856.Https://Doi.Org/10.1016/J.Jnca.2020.102856

[5]. Birkinshaw, J., Et Al. (2019). Cybersecurity In The SDN Era. Journal Of Network Security, 23(4), 45-62.

[6]. Budugutta, S., &Nithya, V. (2017).A Comparative Study On Intrusion Detection Datasets.International Journal Of Advanced Research In Computer Science, 8(3), 245-251.

[7]. Chai M. C. Y. And Eswaran, S. (2025). A Survey On 5G Wireless Network Intrusion Detection Systems Using Machine Learning Techniques. Digital Forensics In The Age Of AI. P: 24. DOI: 10.4018/979-8-3373-0857-9.Ch008.

[8]. Chellani, R., Gupta, A., &Rana, S. (2016). Survey On Network Intrusion Detection Systems Using Machine Learning.International Journal Of Computer Applications, 143(3), 9-13. Https://Doi.Org/10.5120/Ijca2016909972

[9]. Haider, W., Hu, J., Slay, J., Turnbull, B., &Xie, Y. (2020). Generating Realistic Intrusion Detection System Dataset Based On Fuzzy Qualitative Modeling. Journal Of Network And Computer Applications, 124, 1-21. Https://Doi.Org/10.1016/J.Jnca.2018.10.018

[10]. Hammad M., Nabil H. And Wael E. (2023). Enhancing Network Intrusion Recovery In SDN With Machine Learning: An Innovative Approach. Arab Journal Of Basic And Applied Sciences. 30:1, 561-572, DOI: 10.1080/25765299.2023.2261219.

[11]. Hu, F., Hao, Q., &Bao, K. (2014). A Survey On Software-Defined Network And Openflow: From Concept To Implementation. IEEE Communications Surveys & Tutorials, 16(4), 2181-2206.Https://Doi.Org/10.1109/COMST.2014.2326417

[12]. Jafarian , T., Ghaffari A., Seyfollahi A., Arasteh B. (2025). Detecting And Mitigating Security Anomalies In Software-Defined Networking (SDN) Using Gradient-Boosted Trees And Floodlight Controller Characteristics. Computer Standards & Interfaces. Volume 91, 103871. Https://Doi.Org/10.1016/J.Csi.2024.103871.

[13]. Jamal S. R., Ahmed M. A. And Abdul S. M. K. (2023). Performance Evaluation Of Software-Defined Networking Controllers In Wired And Wireless Networks. Telecommunication Computing Electronics And Control. Vol. 21, No. 1, Pp. 49~59 ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.V21i1.23468.

[14]. Kaur, A., &Prinima. (2018). An Overview Of SDN Security Issues And Solutions. International Journal Of Engineering Research & Technology, 7(4), 112-115.

[15]. Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., &Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. Proceedings Of The IEEE, 103(1), 14-76. Https://Doi.Org/10.1109/JPROC.2014.2371999

[16]. Kulkarni M., Goswami B., Paulose J. And Malakalapalli, L. (2025). Unlocking The Power Of Software-Defined Networking (SDN) In Revolutionizing Network Management. Advanced Cyber Security Techniques For Data, Blockchain, Iot, And Network Protection. P: 28. DOI: 10.4018/979-8-3693-9225-6.Ch012.

[17]. Kumar, P., Kumar, R., & Singh, A. (2020). Statistical Approaches For Ddos Detection In SDN. International Journal Of Computer Networks And Communications, 12(2), 1-16. Https://Doi.Org/10.5121/Ijcnc.2020.12201

[18]. Lakshmanan, R., Et Al. (2018). Mitigating Dos Attacks In SDN.International Journal Of Cybersecurity, 29(1), 33-50.

[19]. Maddu, M. And Narasimha Y. R. (2024). Network Intrusion Detection And Mitigation In SDN Using Deep Learning Models. International Journal Of Information Security. V. 23, P. 849–862.

[20]. Majid, A., Ali, T., & Khan, S. (2021). Machine Learning And Deep Learning Techniques For Intrusion Detection In Networks: A Survey.Journal Of Network And Computer Applications, 172, 102848.Https://Doi.Org/10.1016/J.Jnca.2020.102848

[21]. Melnick, J. (2018). Distributed Denial-Of-Service Attacks In SDN.Computer Security Journal, 34(7), 71-85.

[22]. Mutaz, T., Et Al. (2018).Vulnerabilities In SDN Applications.Journal Of Network Security, 27(3), 45-62.

[23]. Niyaz, Q., Sun, W., &Javaid, A. Y. (2016). A Deep Learning Approach For Network Intrusion Detection System.Future Generation Computer Systems, 86, 24-38. Https://Doi.Org/10.1016/J.Future.2016.11.016

[24]. Pei, X., Ding, J., & Li, Y. (2020).Hybrid Intrusion Detection System For SDN.Journal Of Information Security And Applications, 54, 102523.Https://Doi.Org/10.1016/J.Jisa.2020.102523

[25]. Ramkumar, S., Kumar, S., & Thomas, G. (2020). An Entropy-Based Approach For Detecting Ddos Attacks In SDN Environments. Computers & Security, 92, 101-118. Https://Doi.Org/10.1016/J.Cose.2020.101826

[26]. Rojas, G., Et Al. (2020). Application Layer Dos In SDN. Advanced Networking Review, 15(4), 89-102.

[27]. Said, A., Ahmed, M., & Mohammed, E. (2020). Challenges In Building SDN Datasets For Machine Learning: A Comprehensive Survey. Journal Of Information Security And Applications, 50, 102-513. Https://Doi.Org/10.1016/J.Jisa.2020.102513

[28]. Scott-Hayward, S., O'Callaghan, G., &Sezer, S. (2016). SDN Security: A Survey. Future Networks, 6, 55-.

[29]. Songma, S., Sathuphan, T. And Pamutha T. (2023). Optimizing Intrusion Detection Systems In Three Phases On The CSE-CIC-IDS-2018 Dataset. Computers 2023, 12, 245. Https:// Doi.Org/10.3390/Computers12120245.

[30]. Susan N. S., Muthalagu R. & Mothabhau P. P. (2024). SD-IIDS: Intelligent Intrusion Detection System For Software-Defined Networks. Multimedia Tools And Applications. V. 83, P. 11077–11109.

[31]. Waqas, A., Et Al. (2022). Advanced Intrusion Detection Techniques.Cybersecurity Advances, 18(5), 110-128.

[32]. Zawad, M. M., Rahman S. A., Islam S., Monirujjaman M. K. (2024). SDN Intrusion Detection Using Machine Learning Method. Computer Science - Cryptography And Security. Https://Doi.Org/10.48550/Arxiv.2411.05888