Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 235 (2024) 1851–1859

Procedia
Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering (ICMLDE 2023)

Machine Learning Approach to Intrusion Detection: Performance Evaluation

Vishal Giraddi*, Shantala Giraddi, Narayan D G, Anupama Bidaragaddi, Suvarna G Kanakareddi

School of Computer Science and Engineering , KLE Technological University, Hubli, India

Abstract

Software-Defined Networking (SDN) simplifies network administration as well as reduces the need for manual configuration on individual devices. SDN is expected to remain a crucial technology in shaping the future of network infrastructure. However due to key vulnerabilities like Open Flow Protocol Weaknesses, Flow Table Poisoning, Weak authentication and authorization mechanisms, it is prone to intrusion attacks. In this paper, authors aim to build framework based on various machine learning models (ML) to capture anomaly packets. Four models Decision tree, Support Vector Classifier, Naïve Bayes Classifier and artificial neural networks are designed and trained to capture anomaly packets. While SVC achieved an impressive accuracy of 99%, Artificial Neural Networks (ANN) demonstrated satisfactory speed and efficiency, despite their accuracy being slightly lower than SVM's. ANN is then deployed in SDN simulated using Mininet. The model is able to detect all DOS traffic. Other types of attacks SQL Injection,U2R Attack, Buffer Overflow Attack, Probe Attack, Brute Force Attack could not be tested as there is provision to simulated these using Mininet.

Keywords: *Intrusion detection, SDN, ANN, Machine learning, Mininet, InSDN dataset.*

*Vishal Giraddi. Tel.: +91 8310389734.
E-mail address: giraddivishal2000@gmail.com

10.1016/j.procs.2024.04.176

1. Introduction

SDN is an approach to network topology which intends to simplify and centralized network management. In traditional networking, network devices have their control plane and data plane tightly coupled, meaning the control logic and forwarding decisions are made within the devices themselves. In an SDN, the control plane is detached from the data plane [1]. Also the network control logic is moved to a centralized controller. There is greater programmability, configurability and flexibility [2] in managing the network since controller communicates with the network devices through an open protocol such as Open Flow. As a result the operating expenses are markedly lesser than those of conventional networks [3]. This has led to a widespread adoption of SDN in data centre network environments.

While SDN can help reduce software complexity in network management, it's important to note that SDN itself introduces new software components, such as the controller and associated software stack. These components need to be managed and maintained, and their complexity should be taken into account when adopting SDN. However, overall, SDN's centralization, abstraction, programmability, and automation capabilities contribute to simplifying network management and reducing software complexity in the context of network administration. However Centralization, Insecure Controller Communication and Lack of Proper Authorization and Authentication [4] have made SDN vulnerable to attacks.

Targeting the SDN controller or injecting malicious flow rules can have severe consequences, potentially disrupting the entire network and causing service outages. By compromising the controller, an attacker gains centralized control over the network, allowing them to manipulate traffic, redirect it to unintended destinations, or block critical services.

Ensuring security for Software-Defined Networking (SDN) is of paramount importance. IDS which can monitor real-time traffic, detect and also recognize the class of attack would significantly help in striking this issue. Consequently, incorporating IDSs into the network architecture plays a vital role in monitoring and detecting malicious activities.

Our study aims to achieve the following objectives:

- Analyzing the InSDN dataset [8].
- Developing and implementing an IDS utilizing machine learning (ML) techniques.
- Assessing the efficacy of various ML algorithms in detecting network intrusion.

The structure of the paper is outlined as follows. Section 2 deliberate related works. In section 3, suggested methodology is summarized and also outcomes obtained are highlighted. In section 4, the paper is concluded

2. Related Works

In the work [5], authors present a system that suggests two rules, namely the Entropy based rule and the Correlation based rule, for detecting Distributed Denial of Service (DDoS) attacks targeted at SDN controllers. The system utilizes discuss about Renyijoint entropy, which measures two packet header features, namely the IP address of the source and the IP address of destination, as random variables. By employing statistical methods, specifically entropy, the system strives to uncover DDoS attacks. The authors specifically focus on DDoS attacks in their research.

The authors [6] introduce an Intrusion Detection System (IDS) module that utilizes Long Short-Term Memory (LSTM) algorithm, to address the scalability challenges associated with fully connected neural networks. The study focuses on various types of attacks, including DDoS, Botnet, Brute-Force, L2L, L2R, R2L, and R2R attacks [6]. To conduct their research, the authors employ the ISCX IDS 2012 dataset.

The author [7] proposes an SDN module that utilizes convolutional neural networks for effective, adaptable, and early disclosure of sceptical DDoS attacks. The study specifically focuses on analyzing DDoS attacks. For conducting the research, the author employs the CICIDS 2017 dataset.

The author [8] discusses various datasets used in the study and highlights the limitations of utilizing datasets other than the InSDN dataset. The InSDN dataset, generated by the author themselves, is introduced. Furthermore, the author provides an explanation of different attack phases and the various types of attacks that can be identified using this dataset. The attacks identified include DoS, DDoS, password guessing attacks, botnet attacks, web attacks, probes, U2R attacks, and merge attacks.

SDNs share weakness with traditional networks while also having new ones due to their unique architecture. This work [9] presents a lightweight IDS for SDNs. The IDS gathers statistical flow data from OpenFlow switches and examines traffic using feature extraction and aggregation. Results show high accuracy (0.98 F1 score) in detection with a very low false alarm rate.

SDN's growth and DDoS concerns inspire a statistical-based approach [10] for accurately detecting attacks on vulnerable SDN controllers. The method entails packet feature analysis to minimize false results and complexity, offering efficient defence against DDoS attacks.

Study on IDS with stacking ensemble models [11] carried out with self-attention mechanism and CNN. The developed model is evaluated on 5G network for binary classification.

This study focuses on rapid intrusion detection using real-time flow features with less number of packets. Results show that a basic Random Forest model surpasses complex deep learning in detecting intrusions in SDN.

The Authors in [13] conduct a comprehensive analysis of the NSL-KDD dataset. The approach encompasses clustering the dataset into normal and attack classes employing five distinct clustering algorithms and subsequently employing DL for an IDS.

The authors in [14] tackle SDN security risks and proposes ML-SDNIDS, a solution using auto encoder and one-class SVM for control plane IDS, while utilizing P4 programming and ML for real-time data plane intrusion detection.

The authors in [15] combine CNN and LSTM to build a hybrid IDS for detecting network attacks, enhancing accuracy to 96% using the InSDN dataset. Regularization is used improve CNN's performance.

The goal of our study is to develop an IDS to identify different categories of attacks on Software Defined Networks. Four models GNB, Decision tree, SVC and ANN are designed and trained and validated. The proposed system is designed using InSDN dataset.

3. Proposed Methodology

The structure of the suggested approach is shown in fig 1. Building IDS comprises of three phases.

1. Building Intrusion Detection Model (IDM) and assess their performance.
2. Deployment of model in the SDN environment
3. Generate the normal, attack traffic and evaluate model

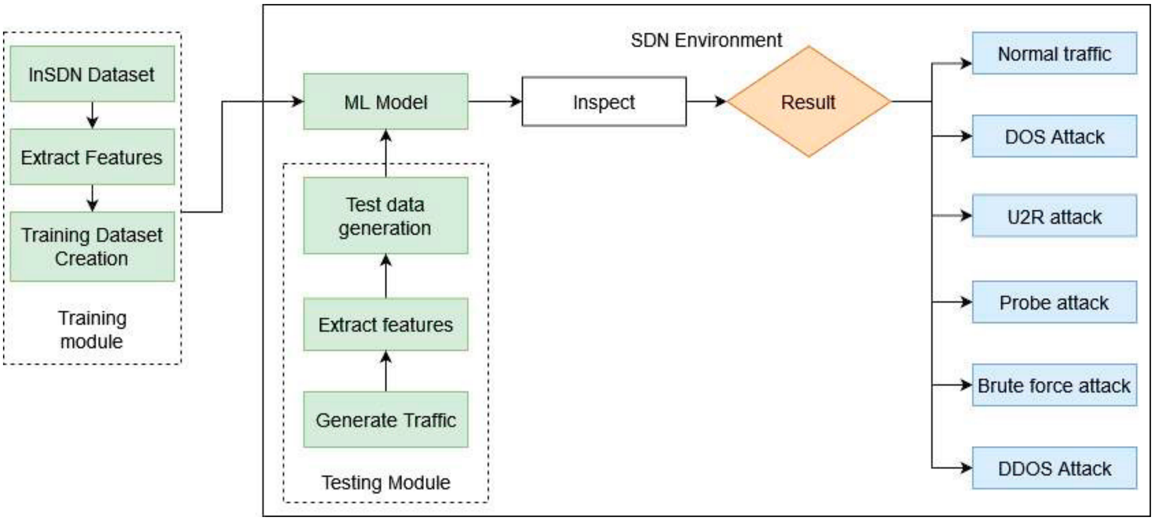


Figure 1. Flow diagram of proposed IDS

3.1 Building Int

In this section, the process of developing a model is explained. Our ML models have developed and tested on the InSDN dataset. The models explored are Decision tree, XG Boost, Support vector machine and Artificial Neural

Networks.

- Dataset description: INSDN [8], Elsayed, Mahmoud] dataset is used to build the Intrusion detection model. Dataset consists of more than 1 lakh tuples and 84 attributes. Kali Linux is used to create six types of attacks DOS (SYN attack), U2R attack, Buffer overflow attack, Probe attack, Brute force attack, DDos attack, was created by emulating the environment under virtual machines [8].
- Pre-processing: It is a vital stage in the overall process to enable machine learning models to effectively utilize the data. Ensuring data quality, consistency, accuracy, and usefulness is essential for the accuracy and effectiveness of the final model. The network traffic data is pre-processed, in the following steps.

The categorical variable label is converted to numeric variable, other categorical variable like source ip address and timestamp are deleted. With the remaining attributes correlation heat map is constructed to indicate a correlation between all features, where highly correlated variables can be combined or removed to improve model performance. The generated heat map given below in fig 2. Some of the features in the dataset had greater numerical values, whereas others had significantly lower values. Numerical quantities are therefore normalized. The following is an estimation of the normalization equation for numerical values:

$$i* = \frac{i-\mu}{\sigma} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Eq(1)$$

Where i = each unique value in the data. Table 1 shows features after pre-processing.

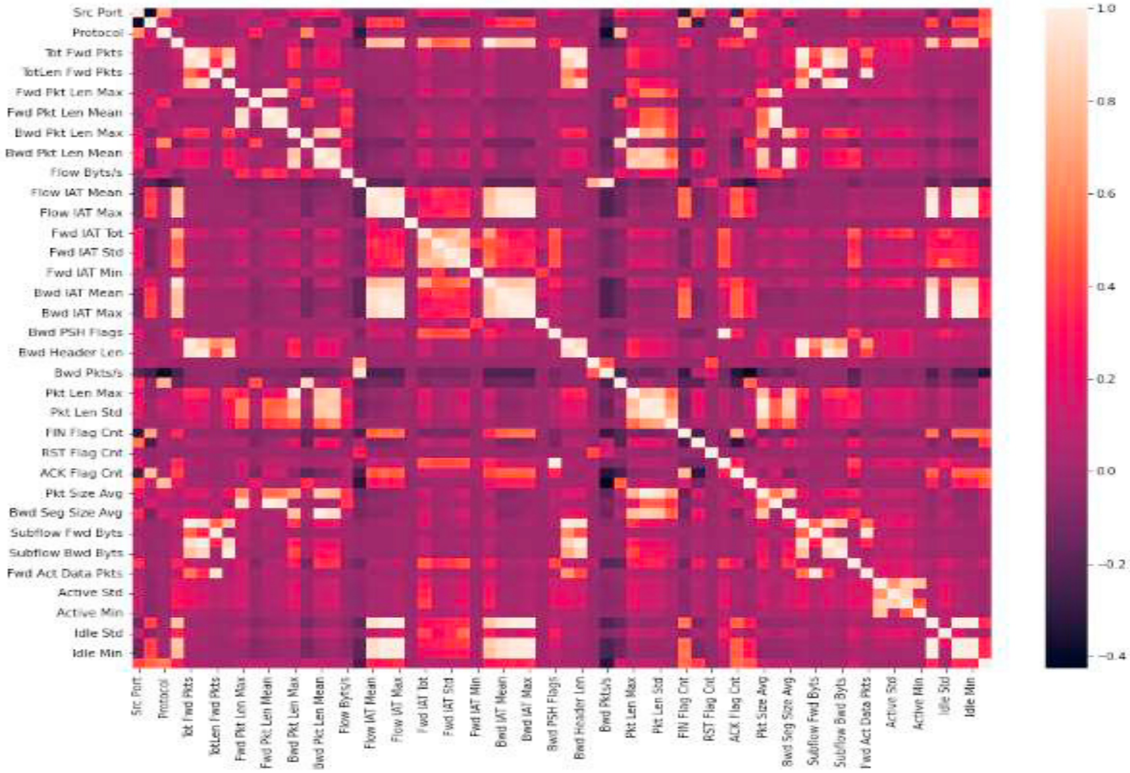
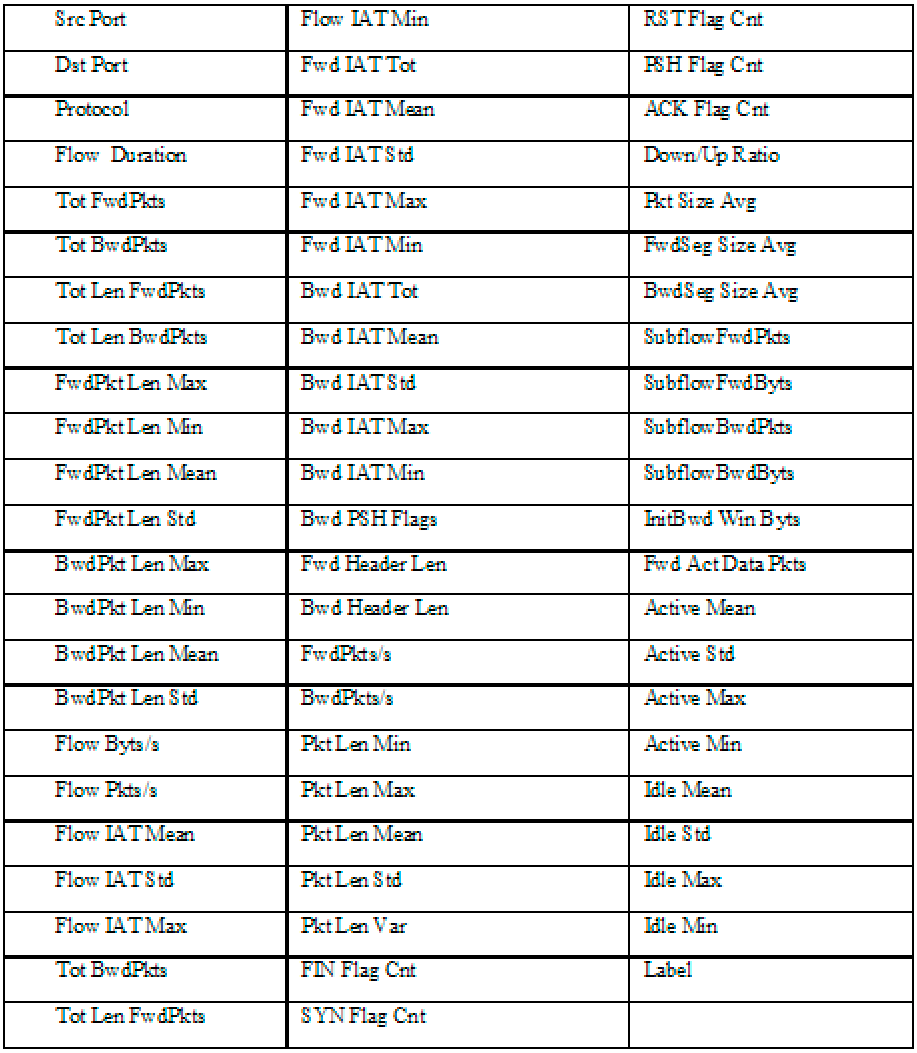


Figure 2. Heat map of features

Table 1. Attributes retained after pre processing



Src Port	Flow IAT Min	RST Flag Cnt
Dst Port	Fwd IAT Tot	PSH Flag Cnt
Protocol	Fwd IAT Mean	ACK Flag Cnt
Flow Duration	Fwd IAT Std	Down/Up Ratio
Tot FwdPkts	Fwd IAT Max	Pkt Size Avg
Tot BwdPkts	Fwd IAT Min	FwdSeg Size Avg
Tot Len FwdPkts	Bwd IAT Tot	BwdSeg Size Avg
Tot Len BwdPkts	Bwd IAT Mean	SubflowFwdPkts
FwdPkt Len Max	Bwd IAT Std	SubflowFwdByts
FwdPkt Len Min	Bwd IAT Max	SubflowBwdPkts
FwdPkt Len Mean	Bwd IAT Min	SubflowBwdByts
FwdPkt Len Std	Bwd PSH Flags	InitBwd Win Byts
BwdPkt Len Max	Fwd Header Len	Fwd Act Data Pkts
BwdPkt Len Min	Bwd Header Len	Active Mean
BwdPkt Len Mean	FwdPkts/s	Active Std
BwdPkt Len Std	BwdPkts/s	Active Max
Flow Byts/s	Pkt Len Min	Active Min
Flow Pkts/s	Pkt Len Max	Idle Mean
Flow IAT Mean	Pkt Len Mean	Idle Std
Flow IAT Std	Pkt Len Std	Idle Max
Flow IAT Max	Pkt Len Var	Idle Min
Tot BwdPkts	FIN Flag Cnt	Label
Tot Len FwdPkts	SYN Flag Cnt	

- Build Model: The following deep learning and machine learning models are used to create the anomaly sensor.

1. Decision tree(DT)
2. The Gaussian Naive Bayes (GNB) classifier
3. Support Vector Classifier(SVC)
4. Artificial Neural Networks(ANN)

1. Decision Tree (DT) similar to flowchart that shows decisions and their potential outcomes. DT is tree like architecture in which each inner node serves as a feature or attribute, each terminal node serves as a decision rule and each leaf node serves as the result. Decision trees are commonly used for classification and regression tasks.

2. The Gaussian Naive Bayes (GNB) classifier is simple probabilistic algorithm. It is   variation of Naive Bayes algorithm that presumes features follow a Gaussian distribution. GNB is particularly effective when dealing with numerical data. It is commonly used in various domains, including text classification, medical diagnosis, and pattern recognition.

3. SVC is a supervised ML model that analyses data and separates it into different classes using a hyper plane. It seeks to find ideal hyper plane that maximizes the separation of data points for different classes.

4. Artificial Neural Networks (ANN) is a ML model motivated by the organization and role of the human brain. It contains linked nodes, called neurons, arranged in layers. Through training process, ANN can learn complex patterns and Correlation in data. During training process, weights and biases are modified to reduce the error between anticipated and actual outputs. ANN is capable of solving various tasks, including classification, regression, and pattern recognition. Fig 3 and Fig 4 show the accuracy and loss in the training process.

Proposed ANN: Artificial neural networks are a computing systems made up of neurons. It composed of an output layer, few hidden layers and an input layer. The total neurons in the initial layer are dependent on the number of attributes; the number of output neurons is dependent on the number of output classes. Proposed ANN for IDS consists of 64 neurons in the input layers, one hidden layer with 64 neurons and an output layer with 6 neurons corresponding to 6 classes.

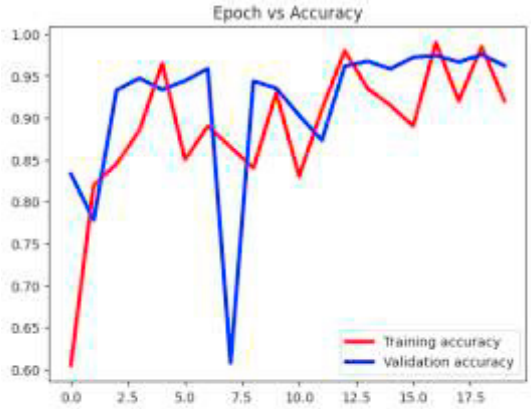


Figure 3. Accuracy obtained with model

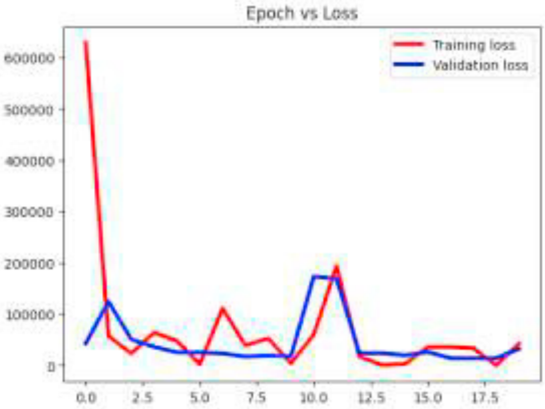


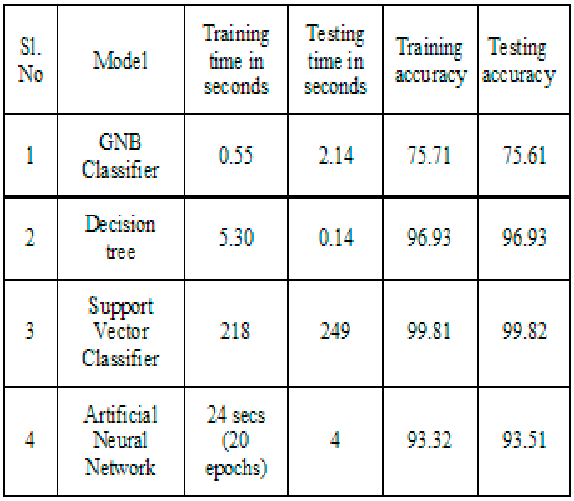
Figure 4. Loss incurred with model

Table 2 presents the outcomes achieved using different models. The Support Vector Classifier (SVC) achieved the highest accuracy for training and testing. However, it also exhibited the longest inference time among all models. Considering the deployment of the model in an SDN network, SVC is not an ideal choice. The Artificial Neural Network (ANN) achieved a testing accuracy of 94%, making it the next best model. The Decision Tree model performed even better with a testing accuracy of 97% and demonstrated moderate timing requirements. Additionally, the time needed for inference with the Decision Tree model was also significantly lower. Fig 5 shows model performances.

3.2 Deployment of IDS in SDN environment

SDN in Mininet: The IDM built in the previous step is deployed in SDN emulated in Mininet. In this study, we created a virtual network with 100 hosts, 15 switches, and a controller. IP addresses are assigned to clients from 10.0.0.1. to 10.0.0.100. Fig 6. Shows the topology of the network. All hosts are connected to the local switches, local switches are linked to the central switch, and controller is linked to all switches (local and central). We deploy the IDM in switches as shown in Fig 7.

Table 2. Performance comparison of IDM



Sl. No	Model	Training time in seconds	Testing time in seconds	Training accuracy	Testing accuracy
1	GNB Classifier	0.55	2.14	75.71	75.61
2	Decision tree	5.30	0.14	96.93	96.93
3	Support Vector Classifier	218	249	99.81	99.82
4	Artificial Neural Network	24 secs (20 epochs)	4	93.32	93.51

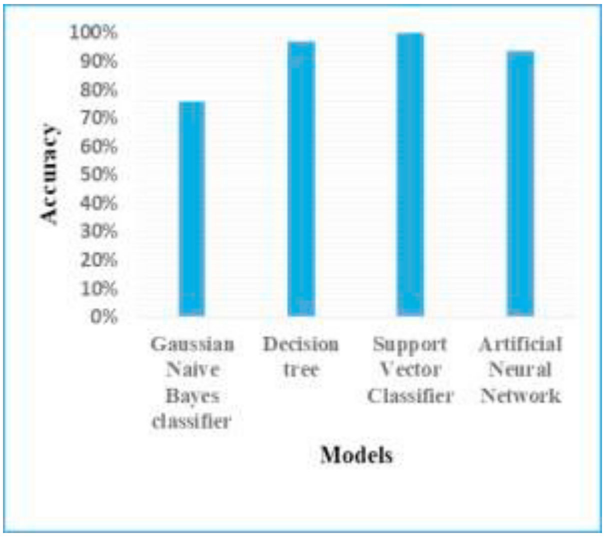


Figure 5. Model Performances

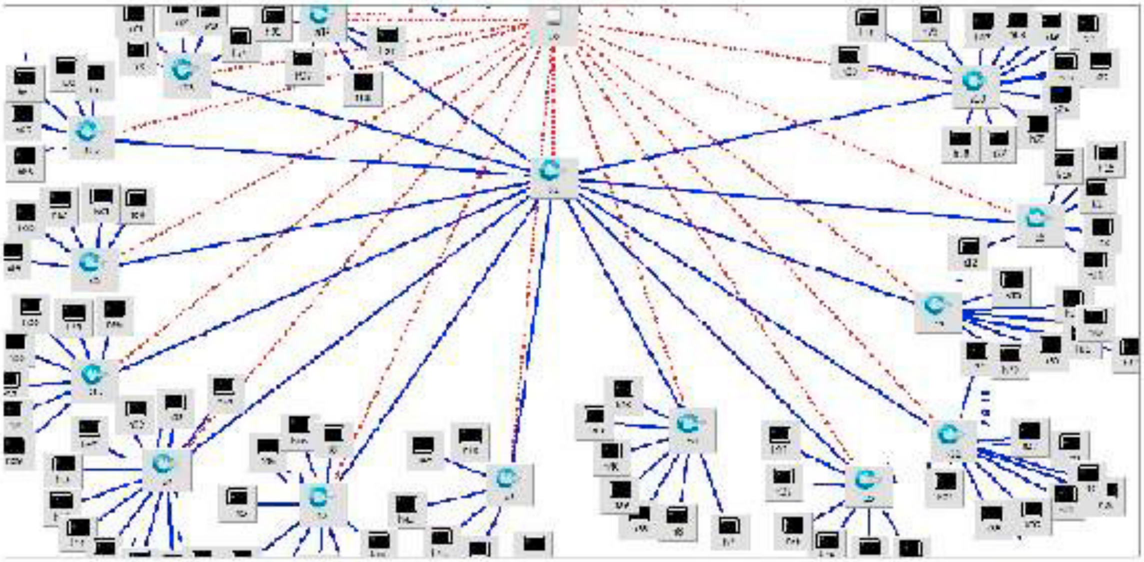


Figure 6. Topology for testing the IDS in SDN simulated using mininet

• Traffic generation: Then using scapy, we generated UDP packets with varying payload and data flow duration. Traffic has an interval of 0.025 secs. The Mininet network consists of a targeted node that serves as the end point for both regular and assault traffic (DOS and DDOS) generated by a client on the SDN network. The configuration and settings used to simulate assault traffic with multiple victims are the same as those used to simulate normal traffic. The only distinction is in the traffic interval and rate. The assault data flow interval is defined as follows: with a data flow rate at 8 packets/sec. In this multiple victim assault scenario, the traffic interval of 0.125 sec. We have collected packet information through switch, the collected data is live streamed to CICFlowMeter network analysis tool to generate the requisite features (Listed in Table 1).

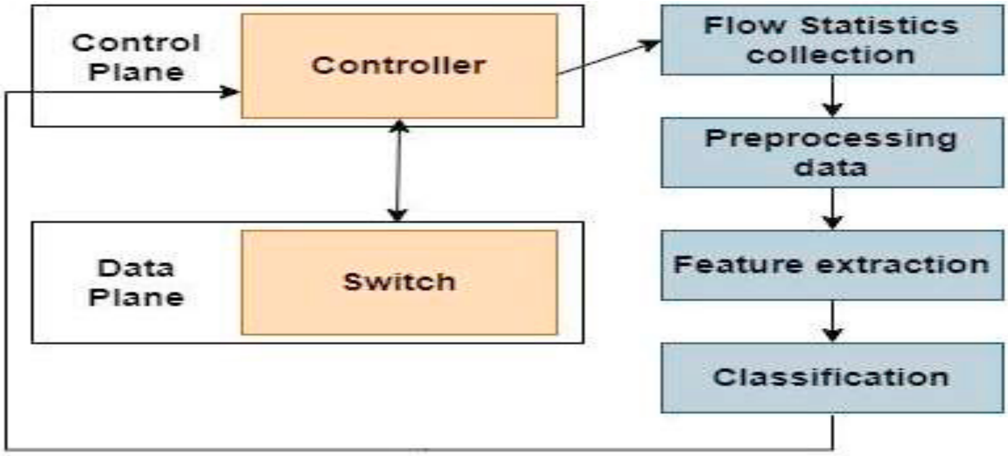


Figure 7. IDM deployed in SDN environment

3.3 *Intrusion Detection with IDS*

The generated features are live streamed to the saved ANN model to categorize given packet as regular or an attack. We assessed a multiple victim assault scenario with a 25 percent assault rate. Only Dos and DDos attacks are simulated using Mininet. The deployed IDM was able to detect all the attack traffic effectively.

3.4 Comparison with related works

The table 3 shows the results of proposed work with the related works in the literature. The results demonstrate the efficiency of our work.

Table 3. Proposed Methodology compared with related works.

Related works	Accuracy
SVM[17]	91.5%
Rnadom Forest[17]	93.3%
CANN[17]	92.2%
Random Forest[18]	87%
Adaboost [18]	85%
Proposed Decision tree	96.93%

3 Conclusion

The authors designed and developed IDM using four ML models. SVC outperformed all others with an accuracy of 99%. However ANN is found to be which is satisfactory in speed and efficiency. The ANN model deployed in SDN simulated using Mininet and normal traffic and DOS DDOS attack traffic generated. It was able to detect all the attack traffic successfully. However there is no provision to generate other attacks like SQL Injection, U2R Attack, Buffer Overflow Attack, Probe Attack, Brute Force Attack using Mininet.

The proposed work has undergone real-time testing using simulated network data in the Mininet environment. This testing enhances the credibility of the results, showcasing the practical viability of the proposed method in a simulated network context. In summary, the outcomes presented in Table 3 indicate that the proposed work excels in accuracy and efficiency and performs admirably in real-time network situations, underscoring its potential as a valuable addition to the field.

References

[1]. D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, vol. 103, pp. 14-76, 2015.

[2]. Abubakar, Atiku, and Bernardi Pranggono. "Machine learning based intrusion detection system for software defined networks." In *2017 seventh international conference on emerging security technologies (EST)*, pp. 138-143. IEEE, 2017.

[3]. Chuang, Hsiu-Min, Fanpyn Liu, and Chung-Hsien Tsai. "Early detection of abnormal attacks in software-defined networking using machine learning approaches." *Symmetry* 14, no. 6 (2022): 1178.

[4]. Farooq, Muhammad Shoaib, Shamyla Riaz, and Atif Alvi. "Security and Privacy Issues in Software-Defined Networking (SDN): A Systematic Literature Review." *Electronics* 12, no. 14 (2023): 3077.

[5]. statistical-based approach for detecting distributed denial of service against the controller of software defined network (SADDCS)

[6]. Tang, Tuan Anh, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, Mounir Ghogho, and Fadi El Moussa. "DeepIDS: Deep learning approach for intrusion detection in software defined networking." *Electronics* 9, no. 9 (2020): 1533.

[7]. Haider, Shahzeb, Adnan Akhunzada, Iqra Mustafa, Tanil Bharat Patel, Amanda Fernandez, Kim-Kwang Raymond Choo, and Javed Iqbal. "A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks." *Ieee Access* 8 (2020): 53972-53983.

[8]. Elsayed, Mahmoud Said, Nhien-An Le-Khac, and Anca D. Jurcut. "InSDN: A novel SDN intrusion dataset." *Ieee Access* 8 (2020): 165263-165284.

[9]. Ajaeiya, GeorgiA., Nareg Adalian, Imad H. Elhajj, Ayman Kayssi, and Ali Chehab. "Flowbased intrusion detection system for SDN." In 2019 IEEE Symposium on Computers and Communications (ISCC), pp. 787-793. IEEE, 2019.

[10]. Al-Adaileh, Mo Mohammad hammad A., Mohammed Anbar, Yung-Wey Chong, and Ahmed Al-Ani. "Proposed statistical-based approach for detecting distributed denial of service against the controller of software defined network (SADDCS)." In MATEC Web of Conferences, vol. 218, p. 02012. EDP Sciences, 2019.

[11]. .Anh, T. T., Lotfi, M., Des, M., Syed, A. R. Z., Mounir, G. and Fadi, E. (2020). DeepIDS: Deep Learning Approach for Intrusion Detection in Software Defined Networking

[12]. Lei, Lifeng, Liang Kou, Xianghao Zhan, Jilin Zhang, and Yongjian Ren. "An anomaly detection algorithm based on ensemble learning for 5G environment." *Sensors* 22, no. 19 (2022): 7436.

[13]. Towhid, Md Shamim, and Nashid Shahriar. "Early Detection of Intrusion in SDN." In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pp. 1-6. IEEE, 2023.

[14]. Alshammri, Ghalib H., Amani K. Samha, Ezz El-Din Hemdan, Mohammed Amoon, and Walid El-Shafai. "An efficient intrusion detection framework in software-defined networking for cybersecurity applications." CMC-Comput. Mater. Contin 72 (2022): 3529-3548.

[15]. Guo, Xian, and Wei Bai. "ML-SDNIDS: an attack detection mechanism for SDN based on machine learning." International Journal of Information and Computer Security 19, no. 1-2 (2022): 118-14119.

[16]. Abdallah, Mahmoud, Nhien An Le Khac, Hamed Jahromi, and Anca Delia Jurcut. "A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs." In Proceedings of the 16th International Conference on Availability, Reliability and Security, pp. 1-7. 2021.

[17]. Riyadh, M., Ali, B.J. and Alshibani, D.R., 2021. IDS-MIU: An Intrusion Detection System Based on Machine Learning Techniques for Mixed type, Incomplete, and Uncertain Data Set. International Journal of Intelligent Engineering & Systems, 14(3).

[18]. Zwane, Skhumbuzo Goodwill. "An Intrusion Detection System For Sdn-Based Tactical Networks: A Machine Learning Approach." PhD diss., University of Zululand, 2020.