



ENSEMBLE-LEARNING FRAMEWORK AS A SECURITY MODEL FOR HARDENING THE SECURITY POSTURE OF SDN IN PERSPECTIVE OF CYBER LAW

KHALIQ AHMED¹, DILAWAR KHAN², MUHAMMAD KASHIF SHAIKH³, M. SADIQ ALI KHAN⁴, SAIMA TABASSUM⁵, AND MUHAMMAD ZAKIR SHAIKH⁶

^{1,*} Department of Computer Science, University of Karachi, Karachi, Pakistan and Department of Computer Engineering, Sir Syed University of Engineering and Technology, Karachi, Pakistan

²Department of Strategic and Nuclear Studies, Faculty of Contemporary Studies, National Defence University, Islamabad, Pakistan

³Department of Software Engineering, Sir Syed University of Engineering and Technology, Karachi, Pakistan.

⁴Department of Computer Science, University of Karachi

⁵Department of Business Administration, Sindh Madressatul Islam University, Karachi Pakistan.

⁶National Center for Robotics and Automation-CMS, Mehran University of Engineering and Technology, Jamshoro, Pakistan

Email Id: kkhazada@hotmail.com

Abstract--Information is considered the most important resource for any organization and to share this resource requires network usage and management. Network operators are attempting to adapt to the coordination of distinctive sorts of networks while meeting the difficulties of expanding traffic. The customary network tends to be unbending. After the forwarding strategy has been defined, the best method to alter it is to rearrange all the impacted devices. It is lengthy and focuses on adaptation and overcoming the challenges of portability and huge data. System administrators have greater versatility to configure their networks using Software Defined Networking (SDN) in the sight of cyber law. With SDN, network management moves from classifying usefulness, lawfulness as far as low-level device arrangements to building programming that encourages system administration and troubleshooting. However, with the rise in network technology, great concerns of security emerged so high that, nowadays, security is considered the major issue to address in any organization. For this reason, in this paper, we have proposed an ensemble learning framework for hardening the security posture of SDN in prospective of cyber law. The proposed framework is evaluated on two different datasets, and it has exhibited great results as using different techniques together improves the performance.

Keywords: Ensemble Learning, Security, Cyber Law, Software Defined Networking (SDN), Quality of Service (QoS), Controller-Application Layer

Table of Contents

Introduction

1. LITERATURE REVIEW

2. DATASET

3. METHODOLOGY

4. FIGURE OF MERITS

5. RESULT AND DISCUSSION

6. CONCLUSION

ACKNOWLEDGEMENT



INTRODUCTION


In reaction to the developing number of cyber-attacks on governments and commercial companies all through the world, arrange interruption discovery frameworks (NIDS) have been rapidly created in the scholarly world and industry. Cybercrimes yearly fetched is consistently expanding [1]. Pernicious insiders, the dissent of benefit attacks, and web-based assaults are the foremost harming cybercrimes. Organizations may lose the mental property in case malevolent malware penetrates the framework, possibly disturbing a country's basic national framework. To protect computer systems from unwanted access, businesses use firewalls, antivirus software, and intrusion detection systems (NIDS) [2-3].

The three essential construction planes in SDN design are the application plane, the control plane, and the information plane. There are two planes: the control plane and the information plane. The application plane organises administrative rules, controls the controller's arrangements, and monitors Quality of services (QoS). The Control plane is in charge of things like activity designing, activity administration, and organized administration. The foremost vital plane within SDN engineering is the information plane. The information plane is made up of components that work together to construct a basic arrange that advances organize an activity. North-bound interfacing connects the application and control planes, whereas south-bound interfacing connects the control and information planes [4-5].

An attacker can carry out assaults on the SDN controller, such as a Secure Shell (SSH) brute force attack, which can pose major security risks. Even if the network administrator recognizes a prospective assault and perpetrator, simultaneous attempts may be difficult to account for in real-time. As a result, particular security rules are required, which are often implemented on the SDN controller in the same way as firewall rules are done. However, setting these rules may be difficult, as the purpose is to prevent rogue nodes or attackers from gaining access while enabling legitimate users to do so without difficulty [6-7]. At the controller-application level, solicitation is already raised all-around identification and esteem components, inside a multi-occupant location, which might enable safety regarding tastes regarding special affiliations progressing to the actual framework. A safety type must be delivered to manage various types of favorable location basics regarding utilizes. The actual controllers tend to be an especially exciting home intended for strike from the SDN progress representing, available to unapproved arrive at and wrong use. In case the actual controller isn't anchored subsequently a good assailant will take the personality, take care of the actual area of just one, and accomplish cancerous workouts. A safety progression, by way of example, transport layer security (TLS) using regular evidence between the controllers and their particular knobs may minimize these kinds of dangers [8,9]. This particular safety identity can be discretionary in OpenFlow plus the standard regarding TLS isn't chosen. A complete safety distinct for your controller-switch screen must be depicted for you to safeguard the call and insurance files carried crosswise more than it. Correct when there are particular controllers from the construction, the possibility of unapproved usage of concentrates and modification regarding the arrangement and progress rerouting can happen. It could influence Denial of Service (DoS), which will get a weakening influence on the actual framework. SDN's high quality in open interfaces and regarded customs turn into a good help intended for aggressors [10,11]. Criminal individuals have distinct traits that can be utilized to distinguish them from legal users. Attackers frequently engage in behaviours like coordinated attacks and the sharing of password dictionaries.

Such patterns may be identified using a variety of approaches, including machine learning. Machine learning-based techniques have shown a lot of promise in terms of user categorization [12,13,].

Utilizing NIDS to recognize the assault handle early from the organize is one of the focus on zones for expeditiously settling cyber-attacks. Organize interruption discovery frameworks (NIDS) are



planned to distinguish hurtful movements such as infections, worms, and disseminated denial-of-service (DDoS) ambushes[14,15,16]. The speed, accuracy, and unwavering quality of inconsistency discovery are fundamental victory criteria for NIDS. To extend location exactness and diminish wrong caution rates [8] machine learning methods (ML) are utilized to construct NIDS. Profound learning (DL) methodologies have been utilized within the field of NIDS as a progressed stream of machine learning. The most recent development focuses on implementing NIDS with machine learning methodologies using a new network architecture, notably the software-defined network (SDN) [17,18,19,20,21].

Therefore mentioned research gap is still there and needs to be addressed and the recent advancements in machine learning and deep learning have exhibited great results for multiple research problems. The study's goal is to present a novel artificial intelligence-based security model for hardening SDN's security posture while imposing the least amount of overhead on performance. For this purpose, we have proposed an ensemble learning-based model to protect the controller application-level communication.

LITERATURE REVIEW

For SDN, suggested machine learning algorithms that may be used to tackle DDoS assaults[22]. The utilization of Bayesian networks, fuzzy logic, evolutionary algorithms, neural networks, and SVM in SDN anomaly detection was covered in the study. The paper goes through the benefits and drawbacks of different systems for anomaly detection in great depth and provide a comprehensive overview of how to utilize SDNs to safeguard networks and advocate using SDNs for security as a service[23,24,25]

In addition, give an overview of programmable networks with an emphasis on SDN. The article discusses the evolution of programmable networks, with a focus on the SDN framework. The study also covers the testing of SDN protocols and possible alternatives to the OpenFlow standard. Nazar et al. [26] Provide an overview of SDN from the perspective of OpenFlow, focusing on the fundamental principle, applications, and security features of OpenFlow. An examination of automatic SSH brute force assaults is presented by Chiba et al. [27]. Using information from the LongTail project, the study studied the behaviour of the attacker and the dynamics of the attacks, including the exchange of password dictionaries and planned attacks. The study's conclusions can be utilised to give the system administrator advice for SSH users. Sangodoyin [28] covers SVM, Expectation Maximization, Bayesian Networks, and K-Nearest Neighbors (KNN) as anomaly detection algorithms in SDN. The author describes several attack scenarios and how they are implemented in SDN applications. Qazi developed Atlas, a new architecture for L2/3/4-based regulation that takes the use of application awareness in SDN. Atlas interacts with the centralised management of the SDN's data reporting process by classifying traffic in SDN using a machine learning technology called C5.0 classifier and gathering ground truth data using a crowdsourcing strategy. Their suggested method has a 94 percent accuracy rate for identifying the top 40 Android apps and can identify mobile applications with great granularity [29].

Kim et al. provided a thorough introduction to SDN, focusing on the present issues with network configuration and management systems, and suggested numerous methods to improve network administration [30]. They concentrated on three important network management challenges: the ability to often change network settings and status, advanced language support for network configuration, a better user interface, and controlled network troubleshooting. Explain in FlowN, an SDN-based system that employs programmable controls on networking devices to let different tenants define their own routing and control policies [31]. Customized address space, topologies, and control signals may all be provided by FlowN to its tenants. They may also expand the mapping across real and virtual networks by using databases. Liu et al. go into great length about the

security aspects of SDN [32]. The article examines a variety of strategies and approaches for dealing with security issues, including learning algorithms. B4 is a novel system proposed by Jain et al. which connects Google's network equipment globally via a privatised Wide Area Network (WAN).

The Naive Bayes, Random Forest, SVM, and J48 algorithms were tested using a public dataset in [33]. When comparing the results, J48 has the greatest accuracy of 80 percent. The necessity of feature selection and dataset labeling was also addressed by the author. For benchmarking dataset, the author employed a Decision Table, Naive Bayes, Bayesian Network, and Decision Tree. The Bayesian Network showed an accuracy of 91.68 percent in predicting the assault based on historical network data. This project made use of the Longtail Project 19 data.

In contrast to previous research, we proposed ensemble learning techniques to detect vulnerable networks and restrict hostile individuals' access. The suggested method allows SDN controllers to create security rules that attempt to prevent future assaults by blocking a whole subnet, as opposed to only blocking certain IPs. As most attackers have been observed using a variety of IP addresses within the same subnet range, blocking the entire subnet makes sense.

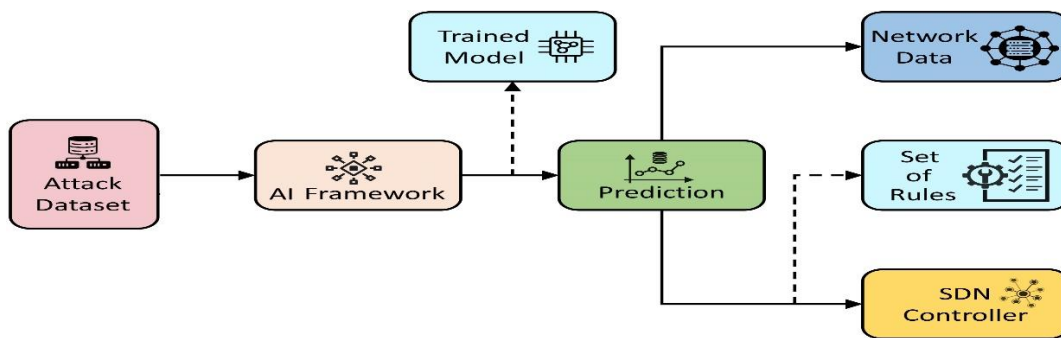


Figure 1: An AI-based framework for defining security rules for SDN controller

DATASET

Historical data is necessary to train the AI-based models to acquire effective classifiers for identifying possible susceptible hosts. The training aids the model's ability to learn and achieve better outcomes. Our objective is to leverage past attack patterns to predict which hosts will be targeted by an attacker. We forecast the probable host which can be targeted based on the attacker's IP. With the help of these predictions, the SDN controller's security rules may be developed, ensuring network security. We advise banning access to the entire subnet rather than just a specific IP in order to prevent additional assaults from the same attacker using a new IP in the same subnet.

To evaluation of the proposed model, we have used multiple datasets which are discussed as follows,

1.1. Scapy Dataset

The traffic flow generated by Scapy was used to create the dataset. In addition to ordinary traffic, the DDoS attack also employs a UDP flood attack. Finally, the hyper parameters of the models were evaluated using 70% of the dataset (This contained 6,250 for each traffic type and 18,750 data points from the dataflow table). The Scikit-learn Python toolkit's GridSearch function offers a local search for the ideal collection of hyper parameters. The standardization of the features In order to prevent classifier overfitting, the database was vectored. It's important to recall that the malicious data was obtained through three different forms of attacks (2083 data

samples are used in a controller assault, 2083 data samples are used in a dataflow table attack, and 2084 data samples are used in a bandwidth attack) [34].

2.2 NSL-KDD Dataset

This analysis uses the NSL-KDD dataset to prepare and evaluate the suggested show. The KDD-Cup 99 dataset may have been improved by the NSL-KDD dataset. The KDD-Cup 99 dataset was made for the Information Revelation and Information Mining competition, which is the world's biggest information mining competition [35]. The KDD-Cup 99 dataset encompasses an issue with copy records, which might impede the quality of the inputs and lead the learning calculation to favor the more visit record. To handle this challenge, the NSL-KDD is recommended and made transparently accessible to analysts. Despite the fact that NSL-KDD has some of the same issues as KDD-Cup 99, many scholars still use the data [36]. They can be used as a benchmark for contrasting various model designs.

Further in [37] the dataset is categorized and Pre-processed into various groups, one of which is the DOS category. We have extracted the DOS dataset to be used in this study.

RESEARCH METHODOLOGY

Figure 1 shows the general AI-based framework that is used to define the rules of security for SDN controllers.

Figure 2 shows the proposed framework to secure the network effectively. The first step towards artificial intelligence training is feature extraction. For network security, multiple important features are taken into account to train the network efficiently. For training the ensemble learning-based model we have used 20 important features that are discussed in table 1.

These 20 features are effectively used by the ensemble learning framework to classify any thread and help in updating the set of rules for the SDN controller. Ensemble learning is a novel machine-learning approach that solves the same issue by combining the efforts of several learners. Ensemble learning is frequently utilized in a variety of industries because it may considerably increase a learning system's classification ability. The framework uses various machine learning and a neural network-based classifier as a base classifier to get an individual prediction. The used classifiers in the ensemble learning framework are discussed as follows,

3.1 Support Vector Machine (SVM)

The SVM algorithm's objective is to recognize a hyperplane in an N-dimensional include space that recognizes between input information focuses. There are a few hyperplanes from which to select to part the two sorts of information focuses, which in our case are assaulting information and unique information. Our objective is to find a plane with the most prominent edge, or the most prominent separate among information focuses from distinctive classes. Expanding the edge separate gives a few support, making it simpler to classify ensuing input highlights. The objective of the SVM strategy is to maximize the separate between the information focuses and the hyperplane. Hinge loss is a loss function that aids in margin maximization. The hinge function is as follows,

$$C(x, y, f(x)) = 0 \text{ if } y \times f(x) \geq 1. \\ 1 - y \times f(x), \text{ otherwise.} \quad (1)$$



Table 1: Features utilized to classify traffic to set rules.

Number	Feature	Description
01	Duration	Total number of seconds with an established connection
02	Service	Destination network service (e.g., HTTP, ssh)
03	Byte Count	Count of bytes present in the flow
04	Type	Action type
05	Tp source	TCP Source Port
06	Count	Total number of connections
07	Tp dest	TCP Destination Port
08	cookie	controller-issued identifier
09	Port	Output port
10	Eth dest	Ethernet destination port
11	Priority	Flow entry priority level
12	Eth src	Ethernet source address
13	Ser	Service type
14	Prot	IP Protocol
15	Pack count	Total number of packets in a flow
16	IP dest	Destination IP
17	Len	The maximum length of the controller
18	IP src	Source IP
19	Vlan	Ethernet VLAN ID
20	Eth type	Ethernet frame type

3.2 Nearest Neighbor (K-NN)

The K-Nearest Neighbor strategy is based on the administered Learning approach and is among the foremost essential ML calculations. The K-NN show infers that the modern case/data and existing cases are comparable and places the modern case within the bunch that’s most congruous with the existing categories. The working principle of K-NN involves 5 steps which are as follows.

Selecting the number of K’s required as neighbors.



Calculating the Euclidean Distance (E.D) for every neighbor. The formulation of Euclidean distance is as follows,

$$E.D = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2} \tag{2}$$

Select the KNN categorized using ED.

Count the data points in each class among these k neighbors.

Allocating the updated data points to the class with the greatest number of neighbors.

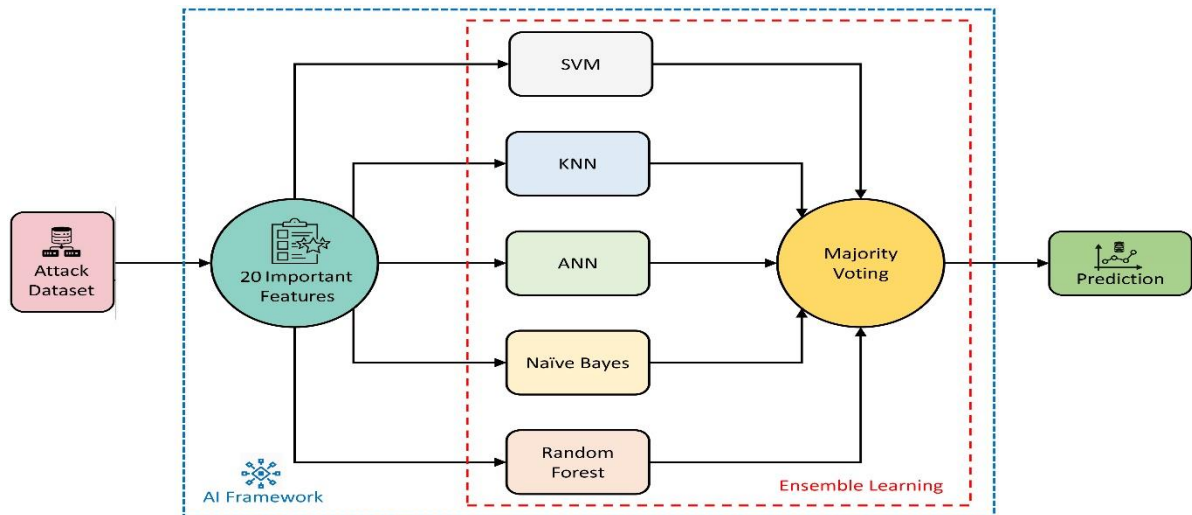


Figure 2: Proposed Ensemble learning framework

3.3 Artificial Neural Network (ANN)

Neural networks are a rough approximation of how the human brain learns. A Manufactured Neural Organize (ANN) is made up of Neurons, which are responsible for layer creation. Optimized parameters are utilized for these Neurons. The yield of each layer is given to the taking after layer.

Each layer has its nonlinear actuation work, which helps within the handle of learning and creating the ultimate yield of the ANN. In our case, we have utilized 3 completely associated layers where the primary two layers utilize the ReLU actuation work, and the final layer employments the sigmoid enactment function. These functions are formulated as,

$$\text{ReLU}(x) = \max(0, x) \tag{3}$$

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{4}$$

3.4 Naïve Bayes

A stochastic machine learning technique called a Naïve Bayes classifier is utilized to perform the classification task. The Bayes theorem forms an integral part of the classifier. The Bayes theorem allows us to calculate the likelihood function $P(a, x)$ from $P(a)$, $P(x)$, and $P(x | a)$ using $P(a)$, $P(x)$, and $P(x | a)$.

Consider the following equation:

$$P(a | y) = \frac{P(y | a) P(a) P(y)}{P(y)} \tag{5}$$

Above,

- $P(a | y)$ is the likelihood ratio of class (a, target) with the given prediction (y, features).
- $P(a)$ is the prior likelihood ratio of the class.
- $P(y | a)$ is the likelihood ratio of the predicted class.
- $P(y)$ is the prior likelihood ratio of the predictor.



3.5 Random Forest

A few choice trees make up an irregular timberland calculation. Sacking or bootstrap conglomeration is utilized to prepare the 'forest' shaped by the irregular woodland strategy. Sacking could be a meta-algorithm that increments the execution of machine learning strategies by gathering them. The irregular forest is an ensemble learning framework that uses multiple decisions to take the final decision. In our case, we have used its decision as one of the votes for our ensemble learning. A rainforest structure makes use of a variety of choice trees. There are three sorts of hubs in a choice tree: choice hubs, leaf hubs, and root hubs. The leaf hub of each tree expresses the extreme outcome that that particular option tree achieved. The ultimate product is chosen to employ a majority-voting method. In this circumstance, the extreme yield of the rainforest framework is the yield chosen by the larger part of choice trees. A straightforward arbitrary timberland classifier is delineated in figure 3.

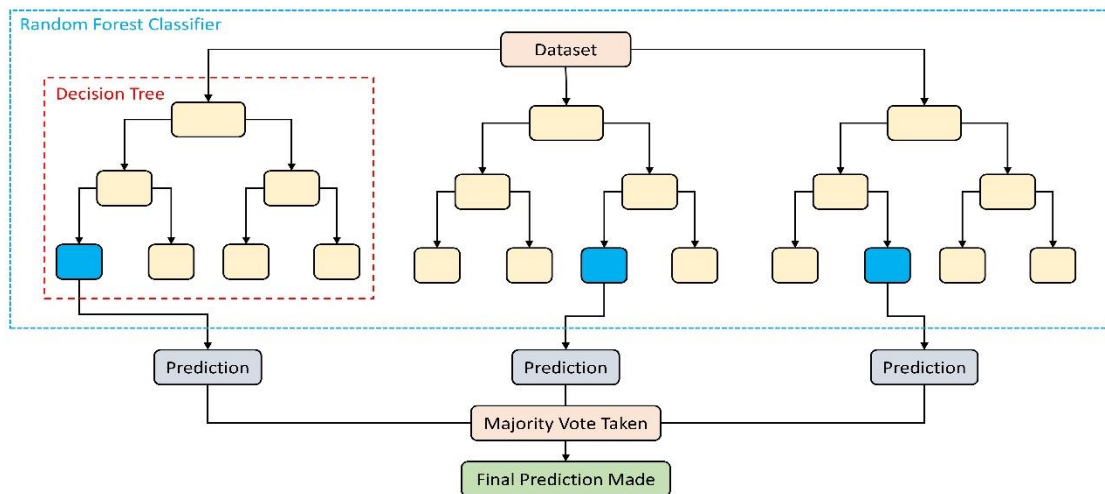


Figure 3: Random Forest framework

After getting a prediction from each classifier, a majority voting algorithm is applied to get the final prediction. In this, all the figures of merits, are mentioned below.

2. FIGURE OF MERITS

We utilized four habitually utilized measures to evaluate the modern methods and existing techniques'

execution, counting Affectability (moreover known as genuine positive rate), Specificity (moreover known as genuine negative rate), and Exactness (ACC). Taking after are the numerical expressions for these figures of merits,

$$\text{Sensitivity} = TP / TP + FN \tag{6}$$

$$\text{Specificity} = TN / TN + FP \tag{7}$$

$$\text{Accuracy} = TP + TN / TP + FP + TN + FN \tag{8}$$

Where acronyms are,

- TP: True Positive
- TN: True Negative
- FP: False Positive
- FN: False Negative.



Results and Discussions

We used two datasets to assess the effectiveness of the suggested framework, as mentioned in the dataset section. Table 2 displays the results gained by the proposed framework on the scapy dataset. The results achieved using the NSL-KDD dataset are also shown in Table 3. The ensemble learning architecture can achieve 0.985 sensitivity, 0.988 specificities, and 0.987 accuracies using the Scapy dataset. Although the NSL-KDD dataset's ensemble learning architecture can achieve 0.983 sensitivity, 0.991 specificities, and 0.987 accuracies.

The little differences in sensitivity and specificity among all classifiers demonstrate that the selected traits are not biased towards any one difference. The higher difference between these two metrics will replicate that the features are giving favor to a single case which shows that we cannot trust the performance. However, the selected feature vector hasn't shown any such signals. Figures 4 and 5 visually depict the outcomes obtained by various classifiers on the scapy and NSL-KDD datasets, respectively.

Conclusion

By separating the control plane from the data plane and changing the network architecture from proprietary to open and programmable, software defined networks (SDN) represent an innovation in networking. Due to the numerous advantages of current SDN architecture, many businesses are switching from conventional network design. As a modern technology, SDN faces a number of difficulties that could endanger cloud computing in the future. Security is one of the main concerns for the future of SDN technology. For this purpose, in this paper, we have proposed an architecture based on ensemble learning which hardens the security system of SDN in perspective of cyber law specifically at the controller-application level. The suggested approach can produce better results because ensemble learning tries to improve on the performance of individual classifiers.

Table 2: Results attained on Scapy dataset.

Classifier	Sensitivity	Specificity	Accuracy
SVM	0.937	0.931	0.932
KNN	0.881	0.907	0.903
ANN	0.940	0.946	0.9413
Naive Bayes	0.876	0.892	0.889
Random Forest	0.935	0.938	0.937
Ensemble	0.985	0.988	0.987

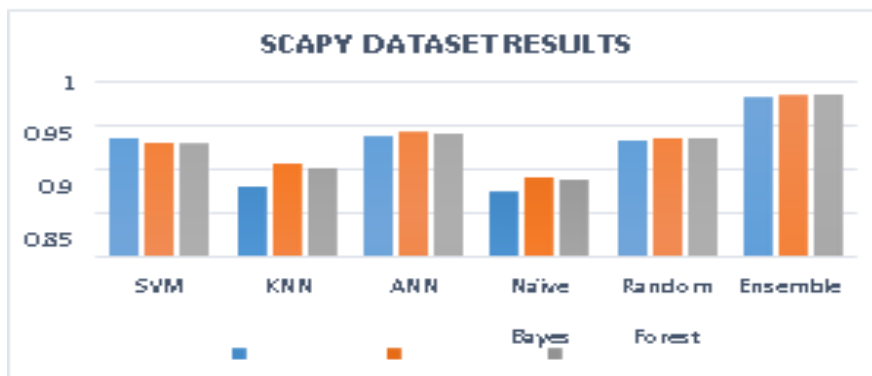


Figure 4: Visual representation of attained results on Scapy dataset



Table 3: Results attained on NSL-KDD dataset.

Classifier	Sensitivity	Specificity	Accuracy
SVM	0.932	0.946	0.9413
KNN	0.873	0.894	0.889
ANN	0.938	0.946	0.943
Naive Bayes	0.907	0.911	0.908
Random Forest	0.921	0.928	0.926
Ensemble Learning	0.983	0.991	0.987

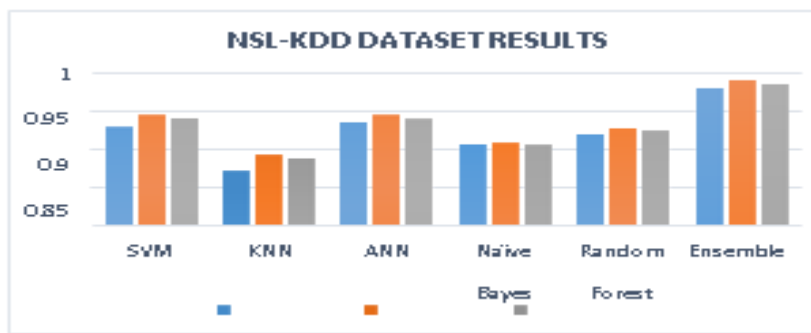


Figure 5: Visual representation of attained results on the NSL-KDD dataset

ACKNOWLEDGEMENT

Thanks, to the team members for their hard work and the University will arrange the resources.

REFERENCES


[1] Sharif, Md Haris Uddin, and Mehmood Ali Mohammed. "A literature review of financial losses statistics for cyber security and future trend." *World Journal of Advanced Research and Reviews* 15, no. 1 (2022): 138-156.

[2] Saini, Dinesh Kumar, and Jabar H. Yousif. "Vulnerability and Attack Detection Techniques: Intrusion Detection System." In *Cybersecurity*, pp. 17-26. CRC Press, 2021.

[3] Ozkan-Okay, Merve, Refik Samet, Ömer Aslan, and Deepti Gupta. "A comprehensive systematic literature review on intrusion detection systems." *IEEE Access* 9 (2021): 157727-157760.

[4] Shi, Yongpeng, Yurui Cao, Jiajia Liu, and Nei Kato. "A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks." *IEEE Network* 33, no. 1 (2019): 29-35.

[5] Karunakaran, Vasantharaj, and Angelina Geetha. "Applying Reinforcement Learning in SDN: A Packet Re-transmission Policy." In *Ambient Communications and Computer Systems: Proceedings of RACCCS 2021*, pp. 279-289. Singapore: Springer Nature Singapore, 2022.

- 
- [6] Swessi, Dorsaf, and Hanen Idoudi. "A survey on internet-of-things security: threats and emerging countermeasures." *Wireless Personal Communications* 124, no. 2 (2022): 1557-1592.
- [7] Tabrizchi, Hamed, and Marjan Kuchaki Rafsanjani. "A survey on security challenges in cloud computing: issues, threats, and solutions." *The journal of supercomputing* 76, no. 12 (2020): 9493-9532.
- [8] Aly, Wael Hosny Fouad, Hassan Kanj, Nour Mostafa, and Samer Alabed. "Feedback ARMA Models versus Bayesian Models towards Securing OpenFlow Controllers for SDNs." *Electronics* 11, no. 9 (2022): 1513.
- [9] Iqbal, Maham, Farwa Iqbal, Fatima Mohsin, Muhammad Rizwan, and Fahad Ahmad. "Security issues in software defined networking (SDN): risks, challenges and potential solutions." *International Journal of Advanced Computer Science and Applications* 10, no. 10 (2019).
- [10] Syed, Naeem Firdous, Zubair Baig, Ahmed Ibrahim, and Craig Valli. "Denial of service attack detection through machine learning for the IoT." *Journal of Information and Telecommunication* 4, no. 4 (2020): 482-503.
- [11] Prabakaran, Senthil, Ramalakshmi Ramar, Irshad Hussain, Balasubramanian Prabhu Kavin, Sultan S. Alshamrani, Ahmed Saeed AlGhamdi, and Abdullah Alshehri. "Predicting attack pattern via machine learning by exploiting stateful firewall as virtual network function in an SDN network." *Sensors* 22, no. 3 (2022): 709.
- [12] Wang, Xuerui, Zheng Yan, Rui Zhang, and Peng Zhang. "Attacks and defenses in user authentication systems: A survey." *Journal of Network and Computer Applications* 188 (2021): 103080.
- [13] Guo, Weilin, Liang Che, Mohammad Shahidehpour, and Xin Wan. "Machine-Learning based methods in short-term load forecasting." *The Electricity Journal* 34, no. 1 (2021): 106884.
- [14] Muhammad, M. U. U. A. H., and A. M. S. F. M. Saleem. "Intelligent Intrusion Detection System for Apache Web Server Empowered with Machine Learning Approaches." *International Journal of Computational and Innovative Sciences* 1, no. 1 (2022): 1-8.
- [15] Bedi, Punam, Neha Gupta, and Vinita Jindal. "I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems." *Applied Intelligence* 51 (2021): 1133-1151.
- [16] Naukarkar, Rupali Lalaji, and K. N. Hande. "Analysis of Implementing Network Intrusion Detection (NIDS) Algorithms Using Machine Learning." *International Journal of All Research Writings* 1 (2020): 2582-1008.
- [17] Yadav, Saneh Lata. "Deep learning approach for network intrusion detection systems." *In An Interdisciplinary Approach to Modern Network Security*, pp. 51-68. CRC Press, 2022.
- [18] Kanna, P. Rajesh, S. Gokulraj, K. Karthik, G. Vijaya, G. Sathish Kumar, and G. Rajeshkumar. "A REVIEW ANALYSIS OF ATTACK DETECTION USING VARIOUS METHODOLOGIES IN NETWORK SECURITY." *Journal of Pharmaceutical Negative Results* 13, no. 4 (2022): 1599-1614.
- [19] Alqahtani, Abdulrahman Saad. "FSO-LSTM IDS: Hybrid optimized and ensembled deep-learning network-based intrusion detection system for smart networks." *The Journal of Supercomputing* 78, no. 7 (2022): 9438-9455.
- [20] Sunagar, Pramod Chandrashekhar, and Anita Kanavalli. "Intrusion detection system using deep learning." *In Deep Learning Applications for Cyber-Physical Systems*, pp. 160-181. IGI Global, 2022.

- 
- [21] Basthikodi, Mustafa, Ananth Prabhu, and Anush Bekal. "Performance Analysis of Network Attack Detection Framework using Machine Learning." *Sparklinglight Transactions on Artificial Intelligence and Quantum Computing (STAIQC)* 1, no. 1 (2021): 11-22.
- [22] Ortet Lopes, Ivandro, Deqing Zou, Francis A. Ruambo, Saeed Akbar, and Bin Yuan. "Towards effective detection of recent DDoS attacks: A deep learning approach." *Security and Communication Networks 2021* (2021): 1-14.
- [23] Fernandes, Gilberto, Joel JPC Rodrigues, Luiz Fernando Carvalho, Jalal F. Al-Muhtadi, and Mario Lemes Proença. "A comprehensive survey on network anomaly detection." *Telecommunication Systems* 70 (2019): 447-489.
- [24] Dwivedi, Shubhra, Manu Vardhan, and Sarsij Tripathi. "Defense against distributed DoS attack detection by using intelligent evolutionary algorithm." *International Journal of Computers and Applications* 44, no. 3 (2022): 219-229.
- [25] Shukla, Nitin, Charu Gandhi, and Tanupriya Choudhury. "Leveraging Blockchain and SDN for Efficient and Secure IoT Network." In *Blockchain Applications in IoT Ecosystem*, pp. 151-166. Cham: Springer International Publishing, 2020.
- [26] Nazar, Muhammad Junaid, Saleem Iqbal, Saud Altaf, Kashif Naseer Qureshi, Khalid Hussain Usmani, and Sobia Wassan. "Software-Defined Networking (SDN) Security Concerns." In *Information Security Handbook*, pp. 19-38. CRC Press, 2022.
- [27] Chiba, Zouhair, Noredine Abghour, Khalid Moussaid, and Mohamed Rida. "Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms." *computers & security* 86 (2019): 291-317.
- [28] Sangodoyin, Abimbola O., Mobayode O. Akinsolu, Prashant Pillai, and Vic Grout. "Detection and classification of ddos flooding attacks on software-defined networks: a case study for the application of machine learning." *IEEE Access* 9 (2021): 122495-122508.
- [29] Thiruvengadam, Hemamalini, Ramya Gopalakrishnan, and Manoharan Rajendiran. "Dynamic Controller Deployment in SDN Networks Using ML Approach." In *Sustainable Communication Networks and Application: ICSCN 2019*, pp. 311-318. Springer International Publishing, 2020.
- [30] Barakabitze, Alcardo Alex, Arslan Ahmad, Rashid Mijumbi, and Andrew Hines. "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges." *Computer Networks* 167 (2020): 106984.
- [31] Meng, Weizhi, Wenjuan Li, and Jianying Zhou. "Enhancing the security of blockchain-based software defined networking through trust-based traffic fusion and filtration." *Information Fusion* 70 (2021): 60-
- [32] Liu, Yifan, Bo Zhao, Pengyuan Zhao, Peiru Fan, and Hui Liu. "A survey: Typical security issues of software-defined networking." *China Communications* 16, no. 7 (2019): 13-31.
- [33] Alehegn, Minyechil, Rahul Raghvendra Joshi, and Preeti Mulay. "Diabetes analysis and prediction using random forest, knn, naïve bayes and j48: An ensemble approach." *Int. J. Sci. Technol. Res* 8, no. 9 (2019): 1346-1354.
- [34] Muthamil Sudar, K., and P. Deepalakshmi. "An intelligent flow-based and signature-based IDS for SDNs using ensemble feature selection and a multi-layer machine learning-based classifier." *Journal of Intelligent & Fuzzy Systems* 40, no. 3 (2021): 4237-4256.
- [35] Hyndman, Rob J. "A brief history of forecasting competitions." *International Journal of Forecasting* 36, no. 1 (2020): 7-14.
- [36] JEYAKARTHIC, M., and A. THIRUMALAIRAJ. "ACCURATE CLASSIFICATION RESULT USING INSTANCE RULING CLASSIFIER ON KDD CUP DATASET." *Journal of Seybold Report ISSN NO 1533: 9211*.



- [37] *Riyad, A. M., MS Irfan Ahmed, and RL Raheema Khan. "An adaptive distributed intrusion detection system architecture using multi agents." International Journal of Electrical and Computer Engineering (IJECE) 9, no. 6 (2019): 4951-4960*