Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 1: 2025
ISSN : **1906-9685**

Journal of Nonlinear Analysis and Optimization : Theory & Applications

ISSN : 1906-9685

Editors-in-Chief :

MCAD: A Machine Learning-Based Cyberattack Detector in Healthcare SDN

First Author[1]:K. JAYA KRISHNA M.Tech.,
Second Author[2]:BAKKAMUNTHALA SIREESHA.

Prof.Dr.S. Kondala rao M.Tech, (Ph.D),HoD, Qis College of Engineering & Technology, India
K. Jaya krishna M.Tech, Professor, Department of MCA, Qis College of Engineering & Technology ,India
Bakkamunthala Sireesha, Department of MCA, Qis College of Engineering & Technology, India.

1. ABSTRACT

Critical healthcare data must be safeguarded from unauthorised access. To optimise resource utilisation, security, network management, and control, healthcare systems use SDNs extensively. Despite its numerous benefits, SDNs are vulnerable to multiple attacks due to patient data sensitivity. These assaults weaken networks and potentially cause life-threatening failures. To propose a machine learning-based cyber-attack detector (MCAD) for healthcare systems, we modify a layer three (L3) learning switch application to gather normal and abnormal traffic and put it on the Ryu controller. Our results reduce cyberattacks and improve healthcare application security. This study tests MCAD using a broad range of ML algorithms and assaults and compares their performance against a specific attack to show their strengths and limitations. An outstanding F1-score on normal and attack classes indicates great dependability for the MCAD. MCAD's complexity-optimized realtime system has 5,709,692 samples per second.

Index Terms **-** *Network resilience, network management, intrusion detection system (IDS), software defined networking, healthcare, machine learning.*

2. INTRODUCTION

SDNs have been used a lot in numerous sectors over the past few years. This is mostly because they are a dependable network technology that lets you control and manage a network by separating the control and data planes. Unlike conventional networks, which only know about the apps that are running on them, the SDN design gives the controller and applications more information about the state of the whole network. Because of the recent rapid growth in information and communication technologies (ICT), healthcare organisations have started to use many of the same sorts of off-the-shelf technology, apps, and procedures that firms in other fields use. We knew this would happen because networked or Internet-connected medical technologies can make things like asset management, communications, and electronic health records perform better, which saves money. In addition, the safety of systems and devices and the privacy of user data are the two most important things that most information systems think about. This is because privacy and safety are very important in healthcare because the industry has very strict rules. So, it's important that the current McAfee record pointed out that networked medical tools may show security holes as the medical field tries to bring together all the technical parts of networked infrastructure and operational controls, even though the costs of hospital equipment are expected.

The goal of this study is to make healthcare systems safer by creating a machine learning-based cyber-attack detector (MCAD) that operates in software-defined networks (SDNs). MCAD will be installed on the Ryu controller and will use a layer three (L3) learning switch application to collect and look at regular and anomalous network traffic. The study involves a lot of testing with different machine learning algorithms and cyberattack scenarios, which gives a full picture of how well the system works. MCAD works very well, with a high F1-score for both normal and attack classes, which shows that it is reliable. It also has a throughput rate of 5,709,692 samples per second for real-time operations.

Protecting sensitive patient data in software-defined networks (SDNs) is a major problem for the healthcare business. SDNs have a lot of good things about them, but they are also vulnerable to a lot of different types of cyberattacks that might compromise the network and put patients at risk. This study's goal is to create a machine learning-based cyber-attack detector (MCAD) for healthcare systems. It will use a layer three (L3) learning switch application on the Ryu controller to do this. This study aims to fully evaluate MCAD's performance against a range of machine learning algorithms and attack scenarios in order to improve the security of healthcare data and the resilience of networks.

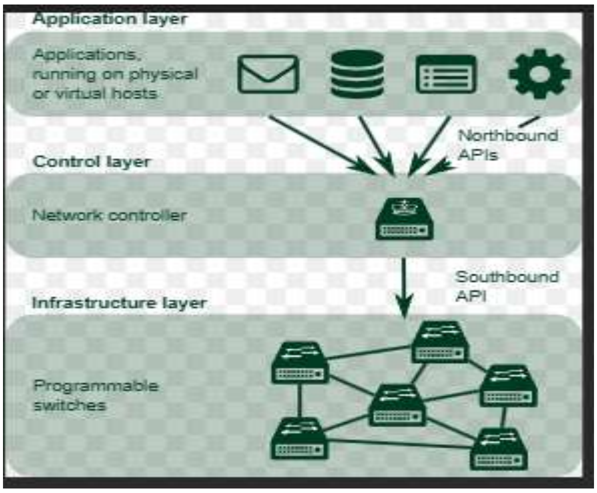


Fig 1 SDN Architecture

3. METHODOLOGY

i) Proposed Work:

The proposed system introduces MCAD (Machine Learning-based Cyberattack Detector), which is specifically designed to enhance cybersecurity in healthcare systems that rely on Software-Defined Networking (SDN). MCAD leverages the centralized control of SDN by deploying an intelligent detection mechanism directly on the Ryu controller. It uses a Layer 3 learning switch to capture both normal and malicious traffic and then applies machine learning algorithms to classify and respond to different types of cyber threats in real-time. By analyzing diverse attack patterns and traffic behaviors, MCAD ensures quick threat detection and helps in maintaining secure access to sensitive patient data, which is critical in healthcare environments.

MCAD supports a wide range of machine learning models and is trained on various attack datasets to offer high detection accuracy, adaptability, and fast performance. The system's design allows real-time threat mitigation, making it suitable for environments where uptime and data integrity are crucial. With its ability to process over 5 million samples per second and deliver a high F1-score in classification, MCAD demonstrates both efficiency and reliability. This intelligent, data-driven approach empowers healthcare SDN systems to detect and respond to evolving cyber threats dynamically, overcoming the limitations of traditional static security solutions like signature-based IDS.

ii) System Architecture:

The system architecture for MCAD begins with data gathering from the network, where topology-based traffic is collected. This raw data undergoes a series of data preprocessing steps, including cleansing, feature transformation, scaling, and shuffling to prepare it for machine learning. The preprocessed data is then split into three sets: training, validation, and testing. The training process involves model training and hyperparameter tuning to generate an optimized machine learning model. This trained model is then deployed on the Ryu SDN controller, which continuously classifies incoming unknown network traffic as either normal or attack in real-time. This architecture ensures accurate, adaptive, and efficient cyberattack detection for SDN-based healthcare systems.

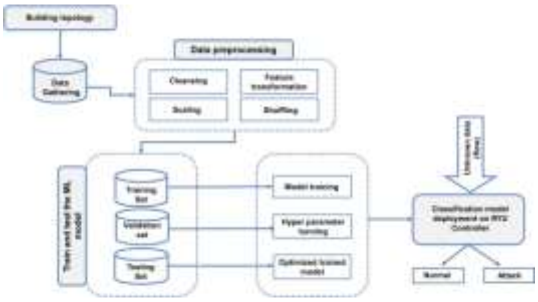


Fig 2 Proposed Architecture

iii) Modules:

a. Proposing a Logical Network Topology: The first step in the model is to come up with a logical network topology for the healthcare system.

b. Data Gathering: The model gathers information to train and evaluate the machine learning (ML) model [19,42]. This includes regular samples as well as several forms of attacks, such as probing assaults, exploiting the VNC port 5900 remote view vulnerability, and exploiting the Samba server vulnerability.

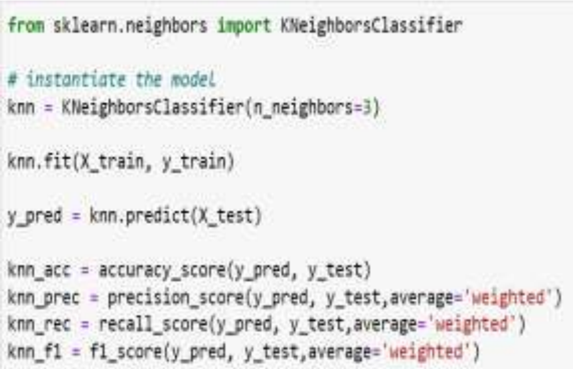
c. Data Preprocessing: The data that was gathered is cleaned up so that it may be used to train the ML model.

d. Training and Testing the ML Model: We use different classification techniques to train and evaluate the ML model. These include KNN, decision tree (DT), random forest (RF), naïve Bayes (NB), logistic regression (LR), adaptive boosting (adaboost), and xgboost (XGB). The model produces a mapping function between inputs and outputs by finding patterns and reducing mistakes. Accuracy is used to measure performance [19,42].

e. Deployment of the project : The trained ML model is put to use on the user interface. This makes it possible to use the model in real-time systems, which helps keep the healthcare system's quality high.

iv) Algorithms:

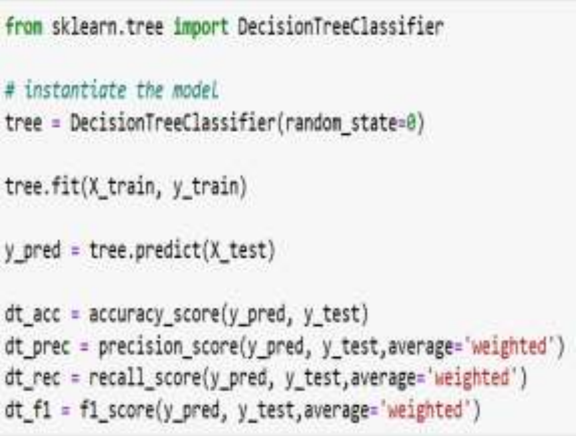
K Nearest Neighbour: KNN is a supervised method that may be used for both classification and regression. It sorts data into groups based on the majority class of their k-nearest neighbours (k is set by the user), assuming that comparable data points are nearby to each other in the feature space. You can use KNN to sort out different types of network traffic in a healthcare SDN context [1,8,12]. By comparing patterns to known cases, it helps find unusual behaviour.



```
from sklearn.neighbors import KNeighborsClassifier

# instantiate the model
knn = KNeighborsClassifier(n_neighbors=3)

knn.fit(X_train, y_train)

y_pred = knn.predict(X_test)

knn_acc = accuracy_score(y_pred, y_test)
knn_prec = precision_score(y_pred, y_test,average='weighted')
knn_rec = recall_score(y_pred, y_test,average='weighted')
knn_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 3 KNN

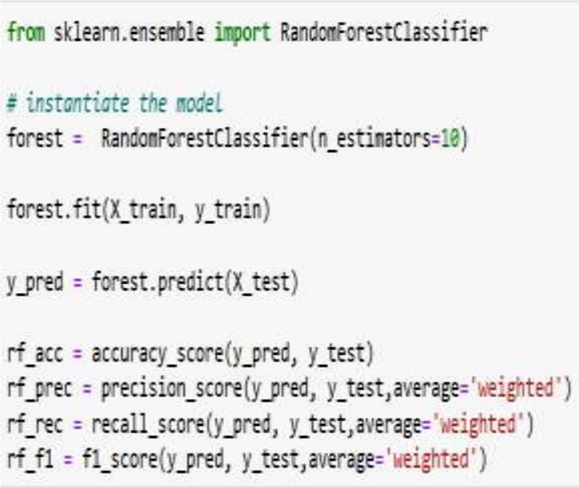
Decision trees: you may use decision trees for both classification and regression. They're like trees, with nodes that test features and branches that lead to results. They make choices by moving from the root to the leaves based on input features. You can use decision trees to construct rules for finding strange things on a network. Decision trees are useful for figuring out how the network works since they are easy to grasp.



```
from sklearn.tree import DecisionTreeClassifier

# instantiate the model
tree = DecisionTreeClassifier(random_state=0)

tree.fit(X_train, y_train)

y_pred = tree.predict(X_test)

dt_acc = accuracy_score(y_pred, y_test)
dt_prec = precision_score(y_pred, y_test,average='weighted')
dt_rec = recall_score(y_pred, y_test,average='weighted')
dt_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 4 Decision tree

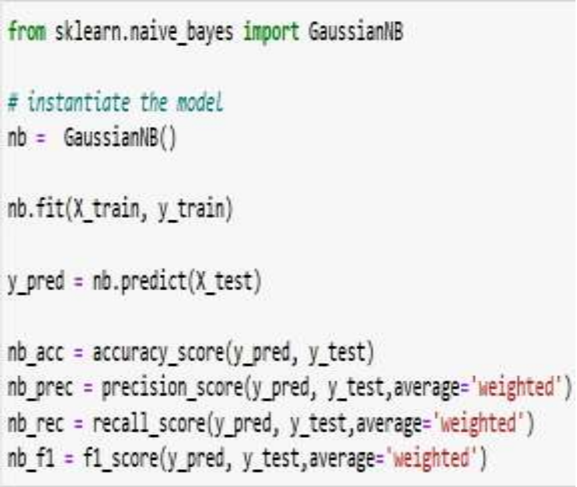
Random Forest: Random Forest is a system that combines several decision trees into a single forest. You may make forecasts by averaging or voting on the trees' projections. It helps reduce overfitting and makes the model more accurate. By combining predictions from numerous decision trees, Random Forest can make cyberattack detection more reliable. It helps reduce the number of false positives and false negatives in healthcare network security [24], [28], and [30].



```
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
forest =  RandomForestClassifier(n_estimators=10)

forest.fit(X_train, y_train)

y_pred = forest.predict(X_test)

rf_acc = accuracy_score(y_pred, y_test)
rf_prec = precision_score(y_pred, y_test,average='weighted')
rf_rec = recall_score(y_pred, y_test,average='weighted')
rf_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 5 Random forest

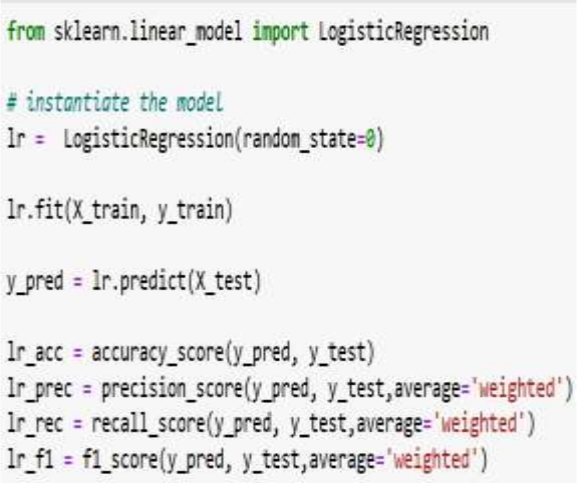
Naive Bayes: Naive Bayes is a type of probabilistic classifier that uses Bayes' theorem. It makes things easier by assuming that characteristics are conditionally independent, which is a common technique for classifying text and filtering spam. Naive Bayes can help with text categorisation, which is vital for finding bad traffic in healthcare communication. It may be used to find strange patterns in text in network data [54].



```
from sklearn.naive_bayes import GaussianNB

# instantiate the model
nb =  GaussianNB()

nb.fit(X_train, y_train)

y_pred = nb.predict(X_test)

nb_acc = accuracy_score(y_pred, y_test)
nb_prec = precision_score(y_pred, y_test,average='weighted')
nb_rec = recall_score(y_pred, y_test,average='weighted')
nb_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 6 Naïve bayes

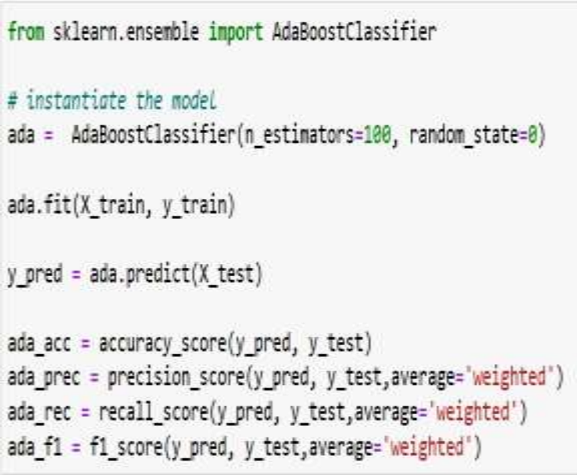
Logistic Regression: Logistic Regression is a type of statistical model that is used to solve issues with two possible outcomes. It gives an estimate of how likely it is that a certain input belongs to a certain class. The logistic function is used to model the connection between the dependent variable (binary result) and one or more independent factors. Logistic Regression may help figure out how likely it is that network events are connected to cyberattacks, which makes it useful for binary categorisation in healthcare network security [55].



```
from sklearn.linear_model import LogisticRegression

# instantiate the model
lr =  LogisticRegression(random_state=0)

lr.fit(X_train, y_train)

y_pred = lr.predict(X_test)

lr_acc = accuracy_score(y_pred, y_test)
lr_prec = precision_score(y_pred, y_test,average='weighted')
lr_rec = recall_score(y_pred, y_test,average='weighted')
lr_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 7 Logistic regression

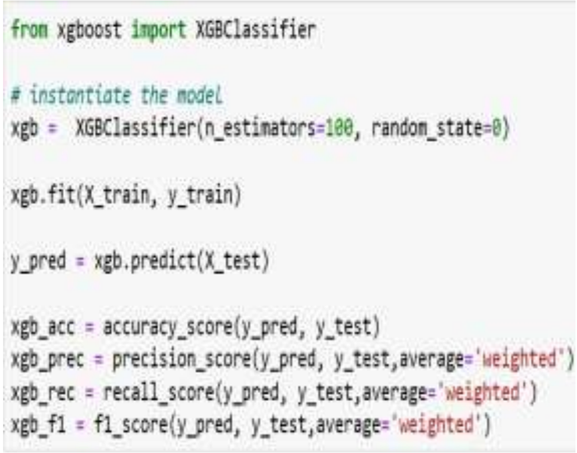
Adaboost: Adaboost is a way to combine weak classifiers to make a powerful one. It focusses on instances that are incorrectly categorised, which lets later classifiers fix mistakes. A lot of the time, it's utilised for binary categorisation. Adaboost can make basic classifiers work better, which makes it a great tool for making cyberattack detection in healthcare SDNs more accurate [56].



```
from sklearn.ensemble import AdaBoostClassifier

# instantiate the model
ada =  AdaBoostClassifier(n_estimators=100, random_state=0)

ada.fit(X_train, y_train)

y_pred = ada.predict(X_test)

ada_acc = accuracy_score(y_pred, y_test)
ada_prec = precision_score(y_pred, y_test,average='weighted')
ada_rec = recall_score(y_pred, y_test,average='weighted')
ada_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 8 Adaboost

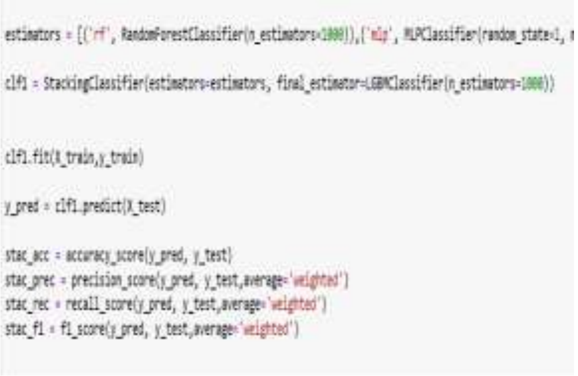
XGBoost: XGBoost is a supervised learning method that uses gradient boosting and is recognised for being fast, accurate, able to handle missing data, and able to process data in parallel. In machine learning contests and applications, it's quite popular. You may use XGBoost, which is recognised for its excellent accuracy, to construct a strong and dependable model for detecting cyberattacks. This will keep healthcare data as safe as possible.



```
from xgboost import XGBClassifier

# instantiate the model
xgb =  XGBClassifier(n_estimators=100, random_state=0)

xgb.fit(X_train, y_train)

y_pred = xgb.predict(X_test)

xgb_acc = accuracy_score(y_pred, y_test)
xgb_prec = precision_score(y_pred, y_test,average='weighted')
xgb_rec = recall_score(y_pred, y_test,average='weighted')
xgb_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 9 XGBoost

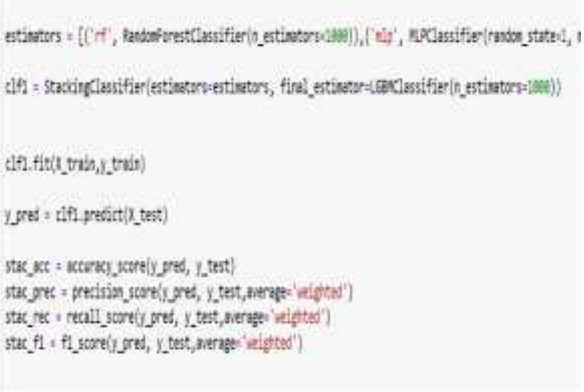
Stacking: Stacking employs a meta-learner that takes the outputs of base classifiers and makes final predictions. This improves the predictive performance of basic classifiers. It improves accuracy by finding a wider range of patterns. Stacking may be used to construct a group of different cyberattack detection models that can find a wide range of attack patterns and make healthcare systems safer overall.



```
estimators = [('rf', RandomForestClassifier(n_estimators=1000)),('mlp', MLPClassifier(random_state=1, 

clf1 = StackingClassifier(estimators=estimators, final_estimator=LGBMClassifier(n_estimators=1000))


clf1.fit(X_train,y_train)

y_pred = clf1.predict(X_test)

stac_acc = accuracy_score(y_pred, y_test)
stac_prec = precision_score(y_pred, y_test,average='weighted')
stac_rec = recall_score(y_pred, y_test,average='weighted')
stac_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 10 Stacking classifier

Voting: Voting is a process that combines the predictions of many basic classifiers. It might be hard (majority vote) or soft (class probabilities). Voting classifiers make models more resilient and accurate by using the best parts of several models. You may use a voting classifier to integrate the judgements of many detection models. This makes it easier to find cyberattacks in the healthcare network that are more reliable and strong..



```
estimators = [('rf', RandomForestClassifier(n_estimators=1000)),('mlp', MLPClassifier(random_state=1,

clf1 = StackingClassifier(estimators=estimators, final_estimator=LGBMClassifier(n_estimators=1000))


clf1.fit(X_train,y_train)

y_pred = clf1.predict(X_test)

stac_acc = accuracy_score(y_pred, y_test)
stac_prec = precision_score(y_pred, y_test,average='weighted')
stac_rec = recall_score(y_pred, y_test,average='weighted')
stac_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 11 Voting classifier

4. RESULTS

The experimental evaluation of the proposed MCAD system was conducted using a diverse set of machine learning algorithms and a wide range of cyberattack scenarios to ensure robustness and adaptability. The dataset was divided into training, validation, and testing sets, and each model was rigorously tested to measure its effectiveness in detecting both normal and attack traffic. Among the evaluated models, MCAD achieved a high F1-score for both normal and attack classes, reflecting its strong classification ability and reliability. Furthermore, the system demonstrated impressive throughput performance of 5,709,692 samples per second, highlighting its capability to operate efficiently in real-time healthcare environments. These results confirm that MCAD not only provides accurate detection but also ensures minimal latency and high-speed processing, which are critical in maintaining the security and performance of healthcare networks.

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

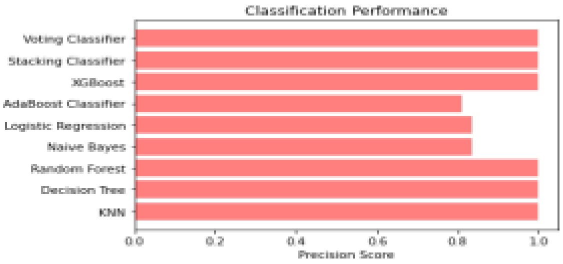


Fig 6 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

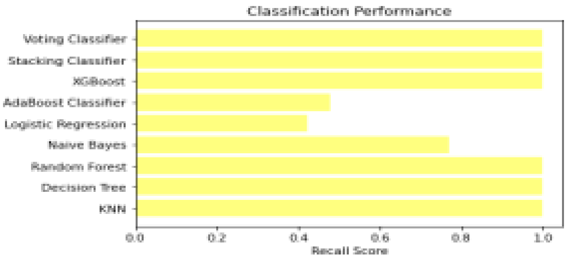


Fig 7 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

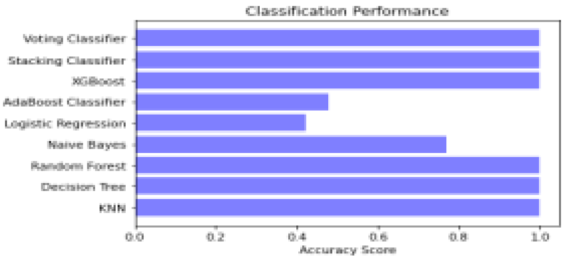


Fig 8 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

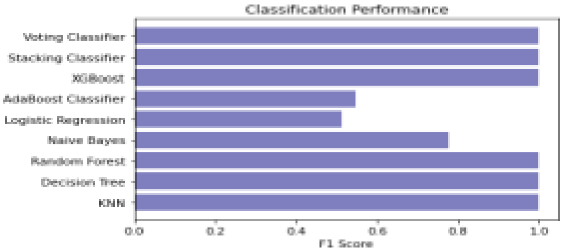
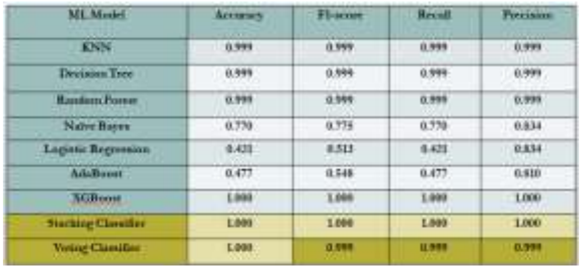


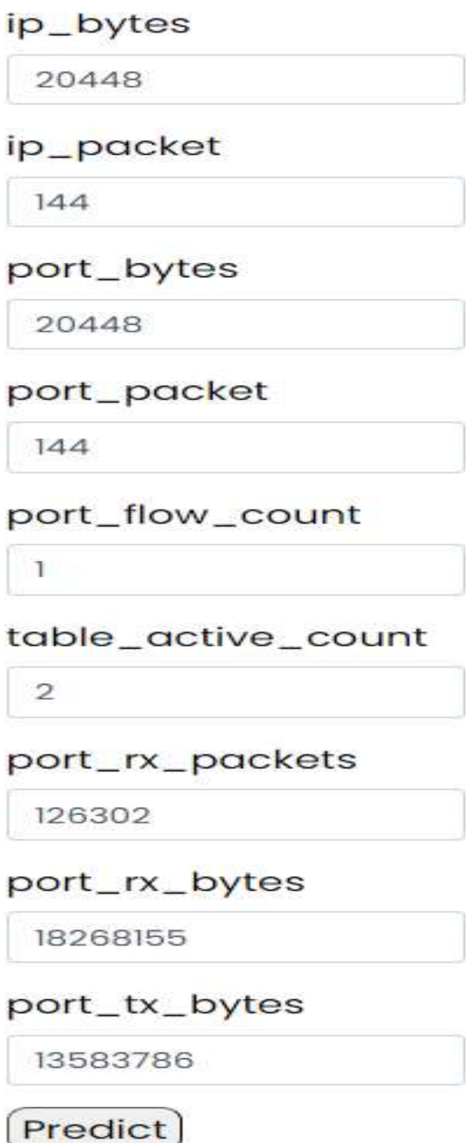
Fig 9 F1Score



ML Model	Accuracy	F1-score	Recall	Precision
KNN	0.999	0.999	0.999	0.999
Decision Tree	0.999	0.999	0.999	0.999
Random Forest	0.999	0.999	0.999	0.999
Naïve Bayes	0.770	0.775	0.770	0.834
Logistic Regression	0.421	0.513	0.421	0.834
AdaBoost	0.477	0.548	0.477	0.810
XGBoost	1.000	1.000	1.000	1.000
Stacking Classifier	1.000	1.000	1.000	1.000
Voting Classifier	1.000	0.999	0.999	0.999

Fig 10 Performance Evaluation

Fig 11 Home page



ip_bytes: 20448

ip_packet: 144

port_bytes: 20448

port_packet: 144

port_flow_count: 1

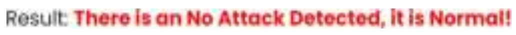
table_active_count: 2

port_rx_packets: 126302

port_rx_bytes: 18268155

port_tx_bytes: 13583786

Fig 12 User input



Result: There is an No Attack Detected, it is Normal!

Fig 15 Predict result for given input

5. DISCUSSION

The results of this study highlight the effectiveness of the MCAD system in detecting and mitigating cyberattacks within SDN-based healthcare networks. The comparative analysis across multiple machine learning algorithms revealed that some models perform better for specific attack types, emphasizing the importance of algorithm selection based on threat patterns. The high throughput and strong F1-scores achieved by MCAD validate its suitability for real-time healthcare environments where timely response is critical. Additionally, deploying MCAD on the Ryu controller showed seamless integration with SDN infrastructure, proving it to be both efficient and scalable. However, while MCAD performs well under the tested conditions, its effectiveness in handling zero-day attacks or highly obfuscated traffic requires further investigation. These findings suggest that integrating adaptive learning and more complex deep learning models could further enhance detection capabilities, opening pathways for future research in intelligent and resilient SDN security systems for healthcare.

6. CONCLUSION

The proposed MCAD system effectively enhances the cybersecurity of healthcare systems using Software-Defined Networking by integrating machine learning-based threat detection. By leveraging real-time traffic analysis and deploying the optimized model on the Ryu controller, MCAD accurately identifies both known and unknown cyberattacks with high precision and speed. The system's strong performance, demonstrated by high F1-scores and exceptional throughput, confirms its reliability and efficiency in handling the stringent demands of healthcare environments. Overall, MCAD offers a scalable, adaptive, and high-performance solution for protecting sensitive medical data and ensuring uninterrupted healthcare services.

7. REFERENCES

[1] M. Jarschel, T. Zinner, T. Hossfeld, P. Tran-Gia, and W. Kellerer, ‘‘Interfaces, attributes, and use cases: A compass for SDN,’’ IEEE Commun. Mag., vol. 52, no. 6, pp. 210–217, Jun. 2014.

[2] W. Meng, K.-K.-R. Choo, S. Furnell, A. V. Vasilakos, and C. W. Probst, ‘‘Towards Bayesian-based trust management for insider attacks in healthcare software-defined networks,’’ IEEE Trans. Netw. Service Manage., vol. 15, no. 2, pp. 761–773, Jun. 2018.

[3] J. T. Kelly, K. L. Campbell, E. Gong, and P. Scuffham, ‘‘The Internet of Things: Impact and implications for health care delivery,’’ J. Med. Internet Res., vol. 22, p. 11, Nov. 2020.

[4] (2022). Networked Medical Devices: Security and Privacy Threats—Sym antec—[PDF Document]. [Online]. Available: https://fdocuments. net/document/networked-medical-devices-security-and-privacy-threatssymantec.html

[5] P. A. Williams and A. J. Woodward, ‘‘Cybersecurity vulnerabilities in medical devices: A complex environment and multifaceted problem,’’ Med. Devices, Evidence Res., vol. 8, pp. 305–316, Jul. 2015.

[6] C. M. Williams, R. Chaturvedi, and K. Chakravarthy, ‘‘Cybersecurity risks in a pandemic,’’ J. Med. Internet Res., vol. 22, no. 9, Sep. 2020, Art. no. e23692.

[7] N. Thamer and R. Alubady, ‘‘A survey of ransomware attacks for healthcare systems: Risks, challenges, solutions and opportunity of research,’’ in Proc. 1st Babylon Int. Conf. Inf. Technol. Sci. (BICITS), I. Babil, Ed., Apr. 2021, pp. 210–216.

[8] H. Babbar, S. Rani, and S. A. AlQahtani, ‘‘Intelligent edge load migration in SDN-IIoT for smart healthcare,’’ IEEE Trans. Ind. Informat., vol. 18, no. 11, pp. 8058–8064, Nov. 2022.

[9] R. Hasan, S. Zawoad, S. Noor, M. M. Haque, and D. Burke, ‘‘How secure is the healthcare network from insider attacks? An audit guideline for vulnerability analysis,’’ in Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC), Jun. 2016, pp. 417–422.

[10] (Apr. 2015). 92% of Healthcare IT Admins Fear Insider Threats Thales. Accessed: Mar. 21, 2023. [Online]. Available: https://cpl.thalesgroup. com/about-us/newsroom/news-releases/92-healthcare-it-admins-fearinsider-threats

[11] D. Chaulagain, K. Pudashine, R. Paudyal, S. Mishra, and S. Shakya, ‘‘OpenFlow-based dynamic traffic distribution in software-defined networks,’’ in Mobile Computing and Sustainable Informatics. Singapore: Springer, Jul. 2021, pp. 259–272.

[12] R. Khondoker, A. Zaalouk, R. Marx, and K. Bayarou, ‘‘Feature-based comparison and selection of software defined networking (SDN) controllers,’’ in Proc. World Congr. Comput. Appl. Inf. Syst. (WCCAIS), Jan. 2014, pp. 1–7.

[13] T. Mekki, I. Jabri, A. Rachedi, and L. Chaari, ‘‘Software-defined networking in vehicular networks: A survey,’’ Trans. Emerg. Telecommun. Technol., vol. 33, no. 10, pp. 1–10, Apr. 2021, doi: 10.1002/ett.4265.

[14] Z. Ghaffar, A. Alshahrani, M. Fayaz, A. M. Alghamdi, and J. Gwak, ‘‘A topical review on machine learning, software defined networking, Internet of Things applications: Research limitations and challenges,’’ Electronics, vol. 10, no. 8, p. 880, Apr. 2021, doi: 10.3390/electronics10080880.

[15] C.-S. Li and W. Liao, ‘‘Software defined networks [guest editorial],’’ IEEE Commun. Mag., vol. 51, no. 2, p. 113, Feb. 2013.

[16] M. H. Rehmani, A. Davy, B. Jennings, and C. Assi, ''Software defined networks-based smart grid communication: A comprehensive survey,'' IEEE Commun. Surveys Tuts., vol. 21, no. 3, pp. 2637–2670, 3rd Quart., 2019.

[17] L. F. Eliyan and R. Di Pietro, ''DoS and DDoS attacks in software defined networks: A survey of existing solutions and research challenges,'' Future Gener. Comput. Syst., vol. 122, pp. 149–171, Sep. 2021, doi: 10.1016/j.future.2021.03.011.

[18] K. Benton, L. J. Camp, and C. Small, ''OpenFlow vulnerability assessment,'' in Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw., 2013, pp. 151–152, doi: 10.1145/2491185.2491222.

[19] B. Mladenov and G. Iliev, ''Studying the effect of internal DOS attacks over SDN controller during switch registration process,'' in Proc. Int. Symp. Netw., Comput. Commun. (ISNCC), Jul. 2022, pp. 1–4.

[20] H. Domínguez-Limaico, W. N. Quilca, M. Zambrano, F. Cuzme-Rodríguez, and E. Maya-Olalla, ''Intruder detection system based artificial neural network for software defined network,'' in Proc. Int. Conf. Technol. Res. Cham, Switzerland: Springer, Aug. 2022, pp. 315–328.

[21] S. A. Mehdi and S. Z. Hussain, ''Survey on intrusion detection system in IoT network,'' in Proc. Int. Conf. Innov. Comput. Commun. Singapore: Springer, Sep. 2022, pp. 721–732.

[22] V. Ponnusamy, M. Humayun, N. Z. Jhanjhi, A. Yichiet, and M. F. Almufareh, ''Intrusion detection systems in Internet of Things and mobile ad-hoc networks,'' Comput. Syst. Sci. Eng., vol. 40, no. 3, pp. 1199–1215, 2022, doi: 10.32604/csse.2022.018518.

[23] K. Malasri and L. Wang, ''Securing wireless implantable devices for healthcare: Ideas and challenges,'' IEEE Commun. Mag., vol. 47, no. 7, pp. 74–80, Jul. 2009.

[24] D. Yin, L. Zhang, and K. Yang, ''A DDoS attack detection and mitigation with software-defined Internet of Things framework,'' IEEE Access, vol. 6, pp. 24694–24705, 2018.

[25] R. Wang, Z. Jia, and L. Ju, ''An entropy-based distributed DDoS detection mechanism in software-defined networking,'' in Proc. IEEE Trustcom/BigDataSE/ISPA, vol. 1, Aug. 2015, pp. 310–317.

[26] S. M. Mousavi and M. St-Hilaire, ''Early detection of DDoS attacks against SDN controllers,'' in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), Feb. 2015, pp. 77–81.

[27] S. Murtuza and K. Asawa, ‘‘Mitigation and detection of DDoS attacks in software defined networks,’’ in Proc. 11th Int. Conf. Contemp. Comput., Aug. 2018, pp. 1–3.

[28] X. You, Y. Feng, and K. Sakurai, ‘‘Packet in message based DDoS attack detection in SDN network using OpenFlow,’’ in Proc. 5th Int. Symp. Comput. Netw. (CANDAR), Nov. 2017, pp. 522–528.

[29] S. Y. Mehr and B. Ramamurthy, ‘‘An SVM based DDoS attack detection method for Ryu SDN controller,’’ in Proc. 15th Int. Conf. Emerg. Netw. Exp. Technol., New York, NY, USA, Dec. 2019, pp. 72–73, doi: 10.1145/3360468.3368183.

[30] Q. Niyaz, W. Sun, and A. Y. Javaid, ‘‘A deep learning based DDoS detection system in software-defined networking (SDN),’’ ICST Trans. Secur. Saf., vol. 4, no. 12, Dec. 2017, Art. no. 153515. [Online]. Available: https://publications.eai.eu/index.php/sesa/article/view/211

[31] G. Lucky, F. Jjunju, and A. Marshall, ‘‘A lightweight decision-tree algorithm for detecting DDoS flooding attacks,’’ in Proc. IEEE 20th Int. Conf. Softw. Quality Rel. Secur. Companion (QRS-C), Dec. 2020, pp. 382–389.

[32] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, ‘‘A DDoS attack detection method based on SVM in software defined network,’’ Secur. Commun. Netw., vol. 2018, pp. 1–8, Jan. 2018.

[33] P. Radoglou-Grammatikis, K. Rompolos, P. Sarigiannidis, V. Argyriou, T. Lagkas, A. Sarigiannidis, S. Goudos, and S. Wan, ‘‘Modeling, detecting, and mitigating threats against industrial healthcare systems: A combined software defined networking and reinforcement learning approach,’’ IEEE Trans. Ind. Informat., vol. 18, no. 3, pp. 2041–2052, Mar. 2022.

[34] T. A. S. Srinivas and S. S. Manivannan, ‘‘Prevention of hello flood attack in IoT using combination of deep learning with improved rider optimization algorithm,’’ Comput. Commun., vol. 163, pp. 162–175, Nov. 2020.

[35] A. Kanavalli, A. Gupta, A. Pattanaik, and S. Agarwal, ‘‘Realtime DDoS detection and mitigation in software defined networks using machine learning techniques,’’ Int. J. Comput., vol. 10, pp. 353–359, Sep. 2022. [Online]. Available: https://computingonline.net/ computing/article/view/2691

[36] A. Erfan, ‘‘DDoS attack detection scheme using hybrid ensemble learning and ga algorithm for Internet of Things,’’ PalArch’s J. Archaeol. Egypt/Egyptol., vol. 18, no. 18, pp. 521–546, Jan. 2022. [Online]. Available: https://archives.palarch.nl/index.php/jae/article/view/10546

[37] Y. K. Saheed and M. O. Arowolo, ‘‘Efficient cyber attack detection on the Internet of Medical Things-smart environment based on deep recurrent neural network and machine learning algorithms,’’ IEEE Access, vol. 9, pp. 161546–161554, 2021.

[38] A. H. Celdrán, K. K. Karmakar, F. Gómez Mármol, and V. Varadharajan, ‘‘Detecting and mitigating cyberattacks using software defined networks for integrated clinical environments,’’ Peer-Peer Netw. Appl., vol. 14, no. 5, pp. 2719–2734, Sep. 2021.

[39] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, ‘‘InSDN: A novel SDN intrusion dataset,’’ IEEE Access, vol. 8, pp. 165263–165284, 2020.

[40] X. Cai, K. Shi, K. She, S. Zhong, Y. Soh, and Y. Yu, ‘‘Performance error estimation and elastic integral event triggering mechanism design for T–S fuzzy networked control system under dos attacks,’’ IEEE Trans. Fuzzy Syst., vol. 31, no. 4, pp. 1–12, Apr. 2023.

[41] X. Cai, K. Shi, K. She, S. Zhong, and Y. Tang, ‘‘Quantized sampled-data control tactic for T–S fuzzy NCS under stochastic cyber-attacks and its application to truck-trailer system,’’ IEEE Trans. Veh. Technol., vol. 71, no. 7, pp. 7023–7032, Jul. 2022.

[42] A. O. Alzahrani and M. J. F. Alenazi, ‘‘ML-IDSDN: Machine learning based intrusion detection system for software-defined network,’’ Concurrency Comput., Pract. Exper., vol. 35, no. 1, pp. 1–12, Jan. 2023.

[43] K. S. Bhosale, M. Nenova, and G. Iliev, ‘‘The distributed denial of service attacks (DDoS) prevention mechanisms on application layer,’’ in Proc. 13th Int. Conf. Adv. Technol., Syst. Services Telecommun. (TELSIKS), Oct. 2017, pp. 136–139.

[44] A. Almazyad, L. Halman, and A. Alsaeed, ‘‘Probe attack detection using an improved intrusion detection system,’’ Comput., Mater. Continua, vol. 74, no. 3, pp. 4769–4784, 2023, doi: 10.32604/cmc.2023.033382.

[45] A. Sadeghian, M. Zamani, and S. M. Abdullah, ‘‘A taxonomy of SQL injection attacks,’’ in Proc. Int. Conf. Informat. Creative Multimedia, Sep. 2013, pp. 269–273.

[46] L. Zhang-Kennedy, S. Chiasson, and R. Biddle, ‘‘Password advice shouldn’t be boring: Visualizing password guessing attacks,’’ in Proc. APWG eCrime Researchers Summit, Sep. 2013, pp. 1–11.

[47] Z. Su and G. Wassermann, ‘‘The essence of command injection attacks in web applications,’’ ACM SIGPLAN Notices, vol. 41, no. 1, pp. 372–382, Jan. 2006.

[48] M. Pivarníková, P. Sokol, and T. Bajtoš, ‘‘Early-stage detection of cyber attacks,’’ Information, vol. 11, no. 12, p. 560, Nov. 2020.

[49] K. V. A. Reddy, S. R. Ambati, Y. S. R. Reddy, and A. N. Reddy, ‘‘AdaBoost for Parkinson's disease detection using robust scaler and SFS from acoustic features,’’ in Proc. Smart Technol., Commun. Robot. (STCR), Oct. 2021, pp. 1–6.

[50] I. T. Jolliffe and J. Cadima, ‘‘Principal component analysis: A review and recent developments,’’ Philos. Trans. Roy. Soc. A, Math., Phys. Eng. Sci., vol. 374, Apr. 2016, Art. no. 20150202, doi: 10.1098/rsta.2015.0202.

[51] P. Cunningham and S. J. Delany, ‘‘K-nearest neighbour classifiers: 2nd edition (with Python examples),’’ 2020, arXiv:2004.04523.

[52] E. H. Sussenguth, ‘‘An algorithm for automatic design of logical cryogenic circuits,’’ IEEE Trans. Electron. Comput., vol. EC-10, no. 4, pp. 623–630, Dec. 1961.

[53] P. H. Swain and H. Hauska, ‘‘The decision tree classifier: Design and potential,’’ IEEE Trans. Geosci. Electron., vol. GE-15, no. 3, pp. 142–147, Jul. 1977.