Eksplorium *p*-ISSN 0854-1418

Volume 46 No. 1, May 2025: 1669–1688 *e*-ISSN 2503-426X

Fuzzy-Based Extreme Gradient Boosting for 5G Network Security

[1*]Kishore Malladi, [2]Appala Naidu Tentu, [3]Neelima Guntupalli, [4]Supriya Goel

[1*]Research Scholar, Department of Computer Science and Engineering, ANU College of Science, Acharya Nagarjuna University, Guntur, Andhra Pradesh

[2]Associate Professor, CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science (AIMSCS), University of Hyderabad Campus

[3]Assistant Professor, Department of Computer Science and Engineering, ANU College of Science, Acharya Nagarjuna University, Guntur, Andhra Pradesh

[4]Assistant Professor, CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science (AIMSCS)

[*]Corresponding Author Email Id: kishoremalladhi1@gmail.com

Article Received: 25 Feb 2025, Revised: 22 Apr 2025, Accepted: 01 May 2025

Abstract: In this work, an Intrusion Detection System (IDS) is developed for the SDN by including well-known machine learners and tree-based algorithms. The entire process is done as 1) Data preprocessing, 2) Feature extraction, 3) Dimensionality reduction, 4) Classification. The well-known NSL-KDD dataset is considered for this research. The Random Forest classifier aids in the feature extraction, and the Principal Component Analysis (PCA) is used for the dimensionality reduction. A Fuzzy-XGBooster classifier is proposed in this work, and it handles the classification part, and detects the normal and the anomaly class. The implementation part is done on the NSL-KDD dataset, and the performance is evaluated on several metrics. The proposed Fuzzy-XGBoost classifier achieved higher performance rate with the values of 0.999246 for accuracy, 0.998859 for precision, 0.998716 for recall, 0.998788 for F1 measure, 0.999485 for specificity, and 0.000515 for false alarm rate, respectively. Again, for the metrics Matthews Correlation Coefficient (MCC), Negative Predictive Value (NPV), False Positive Rate (FPR), False Negative Rate (FNR), Positive Predictive Value (PPV), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) the proposed Fuzzy-XGBoost classifier has achieved suitable values of 0.9981, 0.9993, 0.000602, 0.001317, 0.9988, 0.029031, and 0.000843 respectively.

Keywords: Intrusion detection, SDN, machine learner, Classification, NSL-KDD dataset.

1. INTRODUCTION

5G technology can handle millions of users and data for a given time, and hence it is more popular. Cyber security threats are one of the major reasons affecting the infrastructure of the 5G networks. Various security models and deployment has gained importance now, since many users access the 5G platform. One such important research is the defining the security model for the software defined networking (SDN) [1]. In the SDN, the user can access through the information in the network by gaining proper access. An unauthorised activity within the network platform can be referred to the network intrusion. The intrusion can be done either by an external attacker or the insider. The attacks in the network can be of many types like Denial-of-service (DoS) attacks, malware infection, brute-force attacks, distributed denial-of-service (DDoS) attacks, port scanning, botnet attacks, phishing, and so on [2] [3].These attacks can result in confidential information loss, reputation, monetary damages, and the hereby reduce the network stability [4].It is necessary to develop security framework for 5G platform for dealing with various range of attacks. Hence, Intrusion detection system (IDS) has gained lot of research interests [5]. IDS help in monitoring the traffic of the network, identifying the intruder, and warn the network admin regarding the threat, and the aiding in the prevention of the security breach [6] [7]. Hence, it is necessary to develop an IDS for the 5G network in order to prevent business loss. Developing IDS can help in many industries like finance, healthcare, and government and help in preventing the various intrusions [8]. Intrusion detection techniques fall into three major categories (i) A Misuse-based or Signature-Based Approach, ii) Anomaly based techniques, and iii) Hybrid technique derived from the signature and anomaly-based techniques. Signature based techniques help raising false alarms based on the intruder attack, while the anomaly-based approach identifies the inconsistencies raising through the attack [9]. Research related to the IDS faces various challenges, as the cyber world introduces various threats to the network environment. IDS system needs to tackle various threats effectively and accurately [10]. Vast number of users in the 5G platform can give arise to the abnormal patterns and attacks. In many cases, the intrusion may be miss detected as the false positives and false negatives, and hence affecting the network stability [11]. Miss detection of the intrusion attack as the false positive or false negative can have serious effects such as loss in system downtime, higher work load, security breach. Thus, it is necessary to develop a stable IDS system for handling higher workload of 5G network. Recently, various machine learning algorithms has been developed for tacking the issues related with the IDS. Machine learning algorithms have been evolving successfully in the recent decade. It is necessary to carefully diagnose and choose the right machine learner for the developing a stable IDS system and network protection [12]. In recent years, many machine learning based IDS has been developed for dealing with various public IDS dataset. Some machine learners perform well on the few datasets, but can suffer in datasets which require generalization [13]. Some research has used the CNN and the LSTM technique for developing the security framework as the hybrid model. As the attack pattern change over the time, the reliability of the IDS can affect over the time. Machine learners have generated lot of false alarm rate, and thus can increase false positive rate and thus generate lot of warning. Another challenge while including the machine learners is the use of high computational resources for the IDS [14]. In [15], deep machine learners have been used for development of IDS, and also have achieved good detection rate. Some of the major contributions of this work are explained as follows:

- The data from the standard NSL-KDD dataset, offering a deeper feature set and the various attacks in the network.
- The features are extracted using a Random Forest algorithm which can provide valuable patterns of values based on its tree structure.
- PCA is then used for dimensionality reduction to improve classifier performance.
- The Fuzzy-XGBoost-based IDS can then classify the data into normal and anomaly classes.

The research work is devised as follows: Section 2 discusses various existing works related to the use of machine learners for the SDN, and the research gaps are also discussed. Section 3 gives a brief inscription to the proposed Fuzzy-XGBoost-based IDS system, and the various steps in the IDS models. The implementation in the NSL-KDD dataset, and the results of the proposed Fuzzy-XGBoost-based IDS are detailed in section 4. Finally, the conclusion and the future scope are discussed in section 5.

2. LITERATURE REVIEW

Gupta, N., *et al.* [16] in 2022, proposed a cost sensitive based IDS scheme namely, CSE-IDS as an ensemble machine learning approach for dealing with the intrusions. The model was constructed in the three-layer format, using different deep learners. Layer 1 had a deep neural network (DNN) for traffic identification of the normal vs suspicious traffic in the network flow. The suspicious traffic identified from the Layer 1 was sent to the Layer 2, which contained the XGBoost algorithm for further classification. Layer 2 formed an output of three classes such as, normal class, majority attack class, and minority attack class. The minority attacks can have different attacks, and thus were subjected to further classification in Layer 3. Layer 3 used the random forest algorithm (RF) for identifying the minority attack subclass.

Z. A. E. Houda, *et al.* [17] in 2023 developed a novel framework, namely MiTFed, for the purpose of global intrusion detection in the networks. The system allowed multiple SDNs to form a collaborative network, and find the intrusion without affecting/ sharing the sensitive user details in the dataset. The construction of the MiTFed followed the following steps. 1) To develop a collaborative SDN without affecting the sensitive information within the data. This allowed for privacy preservation along with the intrusion detection. 2) They proposed Secure Multiparty Computation (SMPC) for building a secure model for the local updates in the system. 3) Finally, a blockchain model was designed with the Ethereum smart contract technique for maintain the system properties like flexibility, efficiency, decentralization, and trust. The proposed MiTFed was one of the prime works as a combined platform to prevent security threats in the both blockchain and the SDN-based networks.

M. A. Razib, *et al.* [18] in 2022 experimented with the use of a deep learner for IDS. The scheme used the deep learner for detecting threats arising from the cyber platform in IoT. The proposed scheme was constructed as the SDN framework driven through the deep learner. The deep learner for the scheme was a collaborative form of DNN and LSTM. Using the deep learner directly increased the IDS capability to effectively detect a wide range of threats arising in the networks.

L. M. Halman and M. J. F. Alenazi, [19] in 2023 developed a machine learning-based cyberattack detector (MCAD) for mitigating the cyber threats in SDN framework. The proposed MCAD was a three-layered framework, and it diverted abnormal traffic in the network from the normal network traffic. A Ryu controller was used in the MCAD for the learning switch application, and it learned the nature of the traffic. The MCAD allowed a wide spectrum of ML algorithms to be trained and attacks were introduced in different scenarios to test the effectiveness of the system.

R. Ben Said, *et al.* [20] in 2023 developed a hybrid scheme of the CNN and the LSTM for the intrusion detection. They considered the impact the data imbalance arising due to the data redundancy in the intrusion dataset. The imbalance dataset can have an adverse effect on the anomaly detection, and hence affected the performance of the deep learner. In this study, this disadvantage was overcomed by constructing hybrid model of Convolutional Neural Network (CNN) and bidirectional long short-term memory (BiLSTM) for detecting the attacks on the network. The proposed hybrid approach aided in both the binary and the multiclass classification.

Table 1 illustrates a comprehensive overview of recent IDS within SDN environments using various methods.

Table 1: Comparative analysis of further work

Author(s)&Year	Methods	Contribution	Drawbacks	Advantages of proposed method
Gupta et al. (2022) [16]	CSE-IDS (DNN + XGBoost + RF)	Multi-layer ensemble for unbalanced data	High complexity and slower real-time responsiveness.	Enhance efficiency and reduce false alarms
Houda et al. (2023) [17]	MiTFed (Federated + Blockchain)	Privacy-preserving collaborative IDS	Overlooks ML model tuning; significant resource cost	Combined accuracy with feature selection and minimized overhead
Razib et al. (2022) [18]	DNN + LSTM on SDN	Deep learner for IoT threat detection	No Feature selection, many false alarms	Low false alarms using PCA preprocessing
Halman & Alenazi (2023) [19]	MCAD (3-layer ML + Ryu controller)	Abnormal traffic isolation Through ML	Disregards Dimension reduction and model interpretability	Enhances clarity, feature handling, and processing time
Ben Said et al. (2023) [20]	CNN + BiLSTM	Balanced detection for multi-class data	Neglects feature optimization, poor runtime	Improves speed and minimize redundancy

2.1 Research Gaps

Majority of the existing literature works dealt with the use of the machine learners/ deep learners for the IDS in the SDN environment. Some work, faced overhead issues, privacy preservation, loss of sensitive information, data imbalance, and so on. Few other works dealt the large-scale Intrusion detection by framing hybrid model of the machine learners. One such major problem, is the usage of right feature extraction and feature selection steps, which has a direct impact on the training procedure of the classifier. Also, the majority of the attacks are left undetected due to the complexity of the dataset. Hence, this research gaps need to be addressed while developing a suitable IDS for the SDN environment.

3. PROPOSED FUZZY XG-BOOST METHODOLOGY AS IDS

This section explains the proposed Fuzzy-XG Boost-based IDS scheme for the SDN environment. The Fig.1 explains the various processes involved in the proposed Fuzzy-XG boost-based IDS framework.

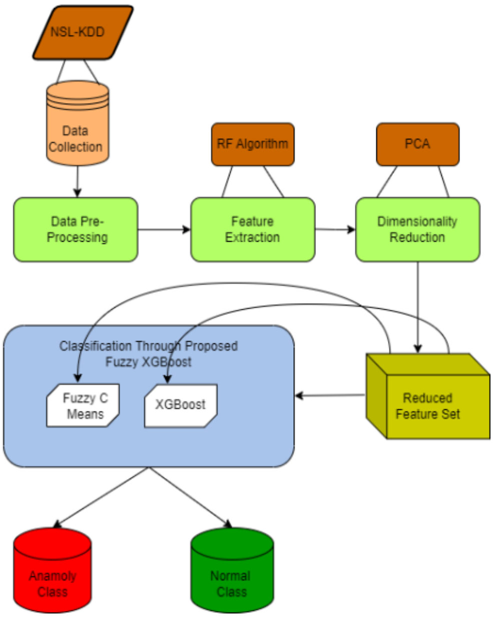

Figure.1 Proposed Fuzzy-XG Boost-based IDS framework in the SDN environment

Intrusion detection has four basic steps: 1) Data preprocessing, 2) Feature extraction, 3) Dimensionality reduction, and 4) Classification. Initially, the data from the NSL-KDD dataset are subjected to pre-processing, for removal of unwanted data. The pre-processed data is provided to the RF algorithm, and necessary features are extracted from it. The extracted features require dimensionality reduction through PCA, since the direct usage of the extracted features may reduce the performance of the classifier. Finally, the feature set is provided to the proposed fuzzy-XGBoost classifier training for identifying the normal and the anomaly class. The proposed Fuzzy-XGBoost classifier was designed by integrating the Fuzzy C means algorithm and the XGBoost classifier for the improved classification performance. The final output from the proposed Fuzzy-XGBoost classifier is the detection of the anomaly as the normal class and the anomaly class. The anomaly class contains the class information of various attacks, and the normal class classifies the normal traffic in the network.

3.1 Data Collection:

The data for this work is taken from the NSL-KDD dataset https://www.kaggle.com/datasets/hassan06/nslkdd, which is one of the standard datasets for the intrusion detection. The NSK-KDD dataset is collected with four sub datasets, namely KDD Test+, KDDTest-21, KDD train+, and KDD train+20 percent. The dataset contains the traffic record in the internet, and various intrusions. For each record in the dataset, there are 43 features for reference of the traffic, and the score for indicating the severity of the traffic. Also, four different set of attacks like Denial of service, user to root, probe, and remote to local are included. During training phase, to address a class imbalance, we used SMOTE (Synthetic Minority Oversampling Technique), which oversampled minority classes.

3.2 Data Pre-Processing:

Many data obtained from the dataset may contain irrelevant information and thus using the dataset directly for the feature extraction may prolong the training process. Also, the raw data usage can influence the anomaly detection rate. Hence, in this work, the dataset is subjected to the data cleaning. Here, the data cleaning is done through null value removal process. Most of the datasets have null value or the missing values due to the data entry errors or presence of in complete information. For example, if a column in the dataset has more than 50% of missing values, then it is fine to leave the column for the further evaluation as it may reduce the classification accuracy. Similarly, for the other case, the missing values can be filled using a simple imputer like mean, median, or the most frequent value. Let us consider the input dataset D, after the subjected to the data pre-processing, the dataset is represented as D_i. The cleaned dataset is then given to the feature extraction process. The input dataset D_i is fed to the RF algorithm and it selects the appropriate features for the classification. Here, the RF algorithm is used for the feature extraction purpose, since it is having the following advantages. Random forest algorithm is one of the well-known ensemble supervised machine learning techniques, which can be used for the various purposes of classification and regression problems. This work considered the use of the RF for extracting the features from the dataset. The random forest algorithm avoids the overfitting issue prevailing in the basic decision tree, by choosing a concept of randomization and diversity in the feature extraction process. The random forest work well particularly in for machine learners, since it can carefully extract the required features for the training. One of the major advantages of the RF algorithm is the discriminate feature extraction, thus it allows in finding the features without the loss of information.

In the RF algorithm, the decision tree is constructed as the different collection of feature extraction unit in the tree structure, and it is represented as $\{F(p, \emptyset_m)m = 1,2, \dots\}$. The term $\emptyset_m$ represents the set of independent identical random vectors. Each tree defined in the random forest casts a unit vote for defining a output class for input p . The main aim of the RF is to generate features from the ensemble of decision trees. It is necessary to achieve diversity among the ensemble of trees, and it is done as the randomization approach defined by the bagging/random subspace methods. For the single tree generation in the RF following steps are followed:

1) Consider the dataset D_i with N data samples. The training set of the RF is fed with N training set, each of the N records in the training set are subjected to the sampling in a random manner, but the sampling is replaced with an original data. This process of the random sampling and replacement is called as bootstrap sample process.
2) The sample record form the above process results in Q input training variables. For the Q training variables in the sample, a number q is selected in random from the Q such that $q \ll Q$.

3) The best split is obtained from the q attributes and it is used for the node splitting. The forest grows based on this process, and during the random forest growing, the value of q should be held constant.

By following the above steps, the trees of the random forest are grown, and it is done without data pruning. Repeating the above steps help in growing of the random forest trees. In the RF algorithm, the total number of trees to be generated in RF is pre-decided, by defining a total RF tree Q_{tree}. For controlling the depth of the random forest tree, a node controller parameter N_{size} is used. The parameter $N_{size}$is measure of total number of instances present in the leaf node, and the value is generally set as one. After building the forest, the forest undergoes training as mentioned above. For the extraction of the features, the training procedure runs through the entire grown trees, and extracts the required number of features. For each new instance, the tree provides an extracted feature output, and the response is recorded as a vote. The vote from each tree is subjected to the hash function, and the maximum number of votes from the tree is declared to be the final extracted output. The RF uses the majority voting scheme for finding the required feature extraction information. The bootstrap sample set drawn during the sampling process, results in loss of one third of the information, and it is referred to as OOB (Out-of-bag) data. Hence, for finding the accuracy of the feature extraction process, a generalization error term is formulated, and it is expressed as follows in eq. 1:

$$PE^{*} = W_{p,q}\big(G(P,Q)\big) < 0 \qquad (1)$$

In Eq. (1) $G(P,Q)$ specifies the margin function. The task of the margin function is to measure the extent of average number of votes (P,Q) in the suitable tree, which exceeds the other tree information. The terms P is specified as the predictor vector, and the term Q is used for the feature extraction. The margin function is mathematically expressed as follows:

$$G(P,Q) = V_m I(\emptyset_m(P) = Q) - \ max_{u \neq Q} V_m I(\emptyset_m(P) = u) \ (2)$$

The term $I(.)$ in Eq. (2) Refers to the indicator function. The margin function calculated is directly proportional to the confidence present through the RF feature extraction. Hence, the strength of the RF feature extraction process can be calculated to be expected value of the margin function is given in Eq (3).

$$RF_{strength} = E_{P,Q}G(P,Q) \qquad (3)$$

The generalization error of ensemble of random forest trees is bounded above by a function of mean correlation between base trees and their average strength. If ϑ is mean value of correlation, an upper bound for generalization error is given in Eq. (4)

$$PE^{*} \leq \vartheta(\frac{1-v^2}{v^2}) \qquad (4)$$

3.3 RF algorithm for feature extraction:

For extracting the suitable features from the input data matrix D_i, RF algorithm considers an instance domain P, and its finite discrete class information C . The class information for the given input p , and the instance N is represented as $D =$

$\{(p_j, q_j)p_j \in P, q_j \in Q\}_{j=1}^{N}$. The decision tree model is constructed as the collection of structured decision nodes, the class information from each leaf nodes are associated as a path vector $V(p) = \emptyset_m , m = 1\ tos.$ For the each input p the class information from the leaf node $\emptyset_m$ is formulated. For extracting the suitable features from the dataset, the class information is not necessary, rather the path vector $V(p)$is used for the feature extraction. Every node in the RF trees represent the binary feature. For the given instance p, the associated features on the path $V(p)$ from all the leaf nodes are corresponded to the value as 1. The other features of the leaf node are set to the value as zero. The procedure is repeated several times, to obtain the suitable features from the dataset. The final features are obtained through the concatenated as a binary vector from all the RF trees.

The feature extraction strategy must ensure to extract important features from the dataset, without the response variable. The entire process involved in the feature extraction through the RF is explained as follows:

Step 1: For the in-input dataset $D_i , i = 1\ to\ N$, a synthetic dataset ${D_i}^X$ is created with the random sampling of the training data, and the product of the marginal distributions T.

Step 2: The response from the dataset D_i is classified into the class 1, and the response of the synthetic datasets is recorded in the class 2.

Step 3: Both the input information D_i and synthetic information are subjected to the concatenation as matrix Z, and each response from the concatenated matrix Z is recorded as z.

Step 4: The random forest trees defined as the

$y = \{(p_j, q_j)p_j \in P, q_j \in Q\}_{j=1}^{N}$ is used for finding the response y.

Step 5: Now, an empty proximity matrix PM is created.

Step 6: For all the trees in the RF, repeat the steps of

- Find the node id $node\_id \leftarrow GetNodeID(l)$
- Find the feature id as $F\_id \leftarrow Hash(node\_id, F)$
- Increment ${p_{j,F\_id}}' = {p_{j,F\_id}}' + 1$
- Return $D' = \cup\ {p_j}'$

Step 7: The above steps are repeated for all the possible combinations of the data input.

Step 8: Scale the elements of the proximity matrix PM by dividing each elemens with the total number of trees. Then, find the dissimilarity matrix D' , through the multidimensional scaling, and it returns a total of F feature space represented as $G = \{G_r\} = D'$. The dimension of the feature space obtained from the algorithm depends on the total number of RF trees and the tree size. Defining the tree size allows finer control in the size of the feature space from the dataset. Again, it is necessary to apply the pruning procedure for defining the stopping criteria in the RF algorithm. Also, hash function can be used for mapping the features from the leaf nodes of the RF trees.

3.4 Dimensionality reduction: PCA algorithm

Dimensionality reduction is one of the key strategies in machine learning process, and it aids in selecting proper features for the training. Here, the dimensionality reduction training is done through PCA. PCA is one of the well-known unsupervised dimensionality reduction techniques. The technique performs the dimensionality reduction by carrying out orthogonal linear combinations on the original data, and also measures the highest variance. The orthogonal linear combinations are also said to be principal components (PC). The PC can be subjected to the total number of original variables in the dataset. In many cases, first set of PC contribute to the majority of the variance. The rest of the values are discarded as minimal information loss. Hence, it is necessary to apply the dimensionality reduction without loss of information from the real data. In this case, PCA based dimensionality reduction comes in handy. A Principal Component Analysis (PCA) was performed in our model, which is a potential to reduce the dimensionality of the feature space. We kept the top 25 principal components which explained 95% of the total variance explained for our dataset. This allows for the majority of the data variance and information to be kept/considered while removing redundant or unimportant component features. The major goal of the dimensionality reduction process is to represent the feature space in lower dimensional space, without the loss of information. Also, the dimensionality reductions need to consider the factor retaining originality of the extracted feature space. Dimensionality reduction process is formulated as follows:

Where, the value of $K \ll F$.

The various steps involved in PCA is explained as follows:

Step 1: The extracted features $G = \{G_r\}\ for\ r = 1\ to\ F$ are fed to the PCA, the feature space is defined with the vector of $1 \times F$.

Step 2: In the next step average value of the vector is calculated. The average of vector is computed as shown in Eq. (5)

$$\hat{G} = \frac{1}{F}\sum_{r=1}^{F} G_r \qquad (5)$$

Step 3: Now, the mean value is computed using the formula, $\emptyset_r = G_r - \hat{G}$.

Step 4: The computed mean value is used for building a covariance matrix using the matrix Y represented as, $Y = [\emptyset_r]\ \ r = 1to\ F$. Now, the covariance matrix is computed as shown in Eq. (6)

$$CM = \frac{1}{F}\sum_{h=1}^{F} \emptyset_h {\emptyset_h}^T = YY \qquad (6)$$

Step 5: The next step is to compute eigen values and eigen vectors from the covariance matrix CM. The eigen values are computed as, and the eigen vectors are represented as.

Step 6: Now, the feature vector is formulated based on the order of the eigen values and eigen vectors. Based on the eigen value, the eigen vector are arranged on the order of highest to lowest. This is one of the major significances of the principal component. Now, the eigen vector

p-ISSN 0854-1418
e-ISSN 2503-426X

sorted with the highest eigen value is selected to be the principal component. The feature vector is formulated with the highest eigen value.

Step 7: The principal components selected through the process and a feature vector is formulated. The feature vector is formed by multiplying the transpose of the vector with the original data.

3.5 Classification: Proposed Fuzzy-XGBoost classifier

After the dimensionality reduction, the reduced features are fed to the proposed Fuzzy-XGBoost classifier for detecting the anomaly class and the normal class. The proposed Fuzzy-XGBoost classifier is the integrated approach to the fuzzy c means classifier, and the XGBoost classifier. The fuzzy c means performs the classification through forming clusters for the appropriate class, and the XGBoost classifier uses the gradient booster for the classification. Both the classifiers have improved performance in the supervised classification scenario and dealing with large datasets, and thus can be employed in this research Fig.2

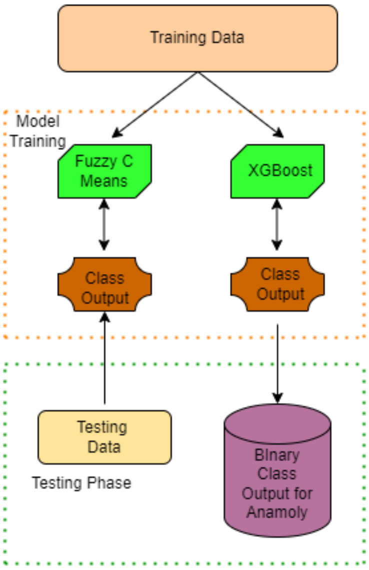

Figure 2: IDS framework with the proposed Fuzzy-XG Boost classifier.

3.5.1 Mathematical description of the Fuzzy-C means classifier

This section briefly explains the Fuzzy C means classifier. The features from the feature vector space H $H = \{H_h\}\ for\ h = 1\ to\ K$ is fed to the classifier. The fuzzy c means treats the features as objects and performs the classification of objects into various fuzzy clusters. For the fuzzy clustering, cluster centers or centroids are defined by the fuzzy logic. The fuzzy C means approach defines d fuzzy clusters to partition the K dimensional features. During the fuzzy clustering a fuzzy matrix L is defined with K rows and d columns. Each element of the fuzz matrix L is defined as $L_{ij}$and it is expressed as the degree of association or the membership function. The membership function L_{ij} is characterized as shown in Eq. (7)

$$L_{ij} \in (0,1)\ i = 1\ to\ K; j = 1\ to\ d \qquad (7)$$

$$\sum_{j=1}^{d} L_{ij} = 1 \; for \; i = 1 \; to \; K \qquad (8)$$

$$0 < \sum_{i=1}^{K} L_{ij} = 1 < K \;\; for \; j = 1 \; to \; d \qquad (9)$$

An objective function is defined for performing the clustering and it is declared as the minimization function as shown in Eq. (10)

$$O_y = \sum_{i=1}^{K} L_{ij} \sum_{j=1}^{d} {L_{ij}}^{y} q_{ij} \qquad (10)$$

The fuzzy C means approach defines d fuzzy clusters to partition the K dimensional features. During the fuzzy clustering a fuzzy matrix L is defined with K rows and d columns. Each element of the fuzz matrix L is defined as $L_{ij}$and it is expressed as the degree of association or the membership function. The membership function L_{ij} is characterized as shown in Eq. (11):

$$L_{ij} \in (0,1) \; i = 1 \; to \; K; j = 1 \; to \; d \qquad (11)$$

$$\sum_{j=1}^{d} L_{ij} = 1 \; for \; i = 1 \; to \; K \qquad (12)$$

$$0 < \sum_{i=1}^{K} L_{ij} = 1 < K \;\; for \; j = 1 \; to \; d \qquad (13)$$

An objective function is defined for performing the clustering and it is declared as the minimization function as shown in Eq. (14).

$$O_y = \sum_{i=1}^{K} L_{ij} \, , q_{ij} \qquad (14)$$

Where, q_{ij} is the Euclidean distance between the objects F_i and the cluster centre E_j, and it is calculated as $q_{ij} = \|F_i - E_j\|$. The term the weighing component for the fuzzy classification. Here, the term y is the scalar function used as the weighing component. It directly controls the fuzziness during the classification process. Now, the value of the centroid is calculated as shown in Eq. (15),

$$E_j = \frac{\sum_{i=1}^{K} {L_{ij}}^{y}, F_i}{\sum_{i=1}^{K} {L_{ij}}^{y}} \qquad (15)$$

Based on the centroid calculation, the clusters are formulated.

Now, the algorithm of the fuzzy c means is formulated as follows:

Step 1: Initialize the value y as $y > 1$for finding the membership values L_{ij} of the matrix.

Step 2. The cluster centres are found based on the Eq. (15).

Step 3: Then identify the Euclidean distance based on the computed values of the centroid and the membership function.

Step 4: Now, again update the membership function L_{ij} as mentioned in the Eq. (16).

$$L_{ij} = \frac{1}{\sum_{w=1}^{K} \frac{q_{ij}}{q_{iw}}^{\frac{2}{y-1}}} \qquad (16)$$

Step 5: Check for the convergence, if the value of the clustering is not converged then repeat the steps from 2.

The stopping criterion for the fuzzy algorithm is defined based on the centroid values, if the centroid value is small, then the objective cannot be further minimized. This can result in trap of the local minima. Hence, the value for the initialization of the fuzzy algorithm needs to be carefully selected.

3.5.2 Mathematical description of the XGBoost classifier

XGBoost is the well-known tree-based algorithms for the classification. This algorithm takes in account of the attributes in the dataset, i.e., the features extracted from the dataset, are provided to the XGBoost classifier, and it acts as the nodes of the tree. The algorithm checks the condition of the root node, and performs the node splitting accordingly. Each end node in the branch is considered as the leaf node, and it is not subjected to the node splitting. The entire concept of the XG boost is based on the gradient booster technique. The training is done as the supervised approach. The training data from the feature set has lot of information, and it is directly used for prediction of the class value. The main objective of the training process is finding the suitable parameters for the training, and the objective function definition is used for this purpose. The entire model performance is evaluated based on the objective function. The objective function for the training purpose comprises of two major parts. The initial part is the training loss function, and the other is the regularization parameter for setting the training process. Now, the objective function of the training model is formulated as shown in Eq. (17)

$$\mathrm{OB}(\theta) = \mathrm{L}(\theta) + \mathrm{R}(\theta) \quad (17)$$

Where, $\mathrm{L}(\theta)$ is the training loss function, and the term $\mathrm{R}(\theta)$ is the regularization loss function. The training loss function defines the actual way of training the XGBooster. The regularization term boosts the model to train the complex features and prevents in the overfitting. Thus, the regularization term aids in maintaining the training efficiency. The XGBoost adds the prediction results from each tree and finds the final class information. While constructing the trees for the XGBoost algorithm, it is necessary to consider the objective function related to the training process. Hence, the parameters used, and the features in the leaf node gain importance during the setup. For setting the parameters for the leaf node, the respective score in each leaf node is to be calculated. Also, it is complex to calculate the score value of all tree nodes in parallel. This is where, the objective function comes in handy. The XGBoost classifier adds up the score of the leaf nodes only satisfying the objective function. Hence, the objective of the XG-Boost classifier is taken as the second order degree of the Taylor's series. The objective function from the second order taylor series for the time step t is expanded as Eq. (18):

$$\mathrm{OB}(\theta)^{\mathrm{t}} = \sum_{\mathrm{i}=1}^{\mathrm{K}} [\mathrm{g_i r_t(p_i)} + \frac{1}{2}\mathrm{h_i r_t}^{2}(\mathrm{p_i})] + \mathrm{R}(\mathrm{r_t}) \quad (18)$$

Where, $\mathrm{g_i}$ and $\mathrm{h_i}$ are the inputs. Based on the above objective the XGBoost algorithm finds the predictive output. The loss functions are carefully monitored through the regularization term. The tree in the XGBoost can be formulated as Eq. (19):

$$\mathrm{r_t(p_i)} = \mathrm{s_{a(p)}} \quad (19)$$

The term s refer to the leaf score, and based on the function a the leaf score in each tree is calculated. Now, the complexity of the leaf node in the XGBoost is calculated as given in Eq. (20):

$$R(r_t) = \alpha X + \frac{1}{2}\,\beta \sum_{l=1}^{L} w_l^2 \qquad (20)$$

In the above equation, L defines the total number of leaf nodes. The XGBoost tree structure is formulated by calculating the regularization, leaf score and the objective function in each score level. After the gain calculation, the node is split as the left node and the right node. In the case of gain value short of the regularization, then the branch is dropped. This process is also referred as tree pruning. The XGBoost follows the steps, and accurately classifies the data.

4.DEVELOPING IDS USING INTEGRATED FUZZY-XG BOOST CLASSIFIER

In this work, the IDS framework is developed by the integrated approach of the Fuzzy and the XGBoost classifiers. The entire process of IDS development is explained in pseudocode at below,

```
INPUT: $D_i$ dataset containing feature and target (K&Y)
OUTPUT: Better prediction accuracy using trained hybrid approach
1. Preprocessing
        -Normalize or standardize features in K if necessary
2. Fuzzy C-Means Clustering (FCM)
        -Apply FCM to K to obtain:
        -Membership matrix $L_{ij}$ (size: N x $E_j$), where $L_{ij}$ indicates the degree of membership of i to cluster j
         - Cluster centers $E_j$
3. Feature Augmentation
        -Add the membership degrees (L) as new features to the original feature set:
        -K_augmented = concatenate (K, L)
4. Train an XGBoost Model
        - Train an XGBoost model, using K_augmented
5. Predictions
        - For new input k_new:
        a. Compute membership vector u_new using the trained FCM cluster centers
        b. Add u_new as membership to form x_new_augmented
        c. Use the trained XGBoost model to predict the target for x_new_augmented
        RETURN: Trained FCM and XGBoost model
```

Table 2. System performance is evaluated with standard classification metrics, while training time is reported to indicate computational demand.

Table 2: Parameter table of integrated Fuzzy-XG Boost classifier

Component	Parameter	Value/Setting
XGBoost	Learning Rate (eta)	0.1
	Max Depth	6
	Subsample	0.8
	Colsample by Tree	0.8
	Number of Estimators	100
	Objective	binary:logistic
	Booster	gbtree
Fuzzy C-Means (FCM)	Number of Clusters (c)	2 (Normal vs. Anomalous)
	Fuzziness Coefficient (m)	2
	Convergence Threshold	1.00E-05
	Maximum Iterations	100

5. RESULTS AND DISCUSSION

This section explains the implementation of the proposed Fuzzy-XGBoost classifier on the standard IDS dataset. The results are compared with the existing tree-based techniques like Decision tree and the XG Boost.

5.1 Evaluation metrics:

The evaluation of the entire techniques is done based on the standard metrics like accuracy, precision, recall, F1-measure, Specificity, and False Alarm rate, False positive rate (FPR), False negative rate (FNR), Negative prediction value (NPV), Positive prediction value (PPV), Mathews Correlation coefficient (MCC), Mean Absolute Error (MAE), and Root Mean Square error (RMSE). Each metric is mathematically expressed in terms of the true positive, true negative, false positive and false negative.

Accuracy: Accuracy is the measure of the total correct detection instances by the classifier to the total number of intrusion prediction Eq. (21).

$$Accuracy = \frac{TN+TP}{Tp+FP+TN+FN} \qquad (21)$$

Precision: Precision measures the average of the positive response by the classifier to the total true and false positives Eq. (22).

$$Precision = \frac{TP}{TP+FP} \qquad (22)$$

Recall: Recall is the measure of the total identification of the true instance to the actual true instance in the dataset Eq. (23).

$$Recall = \frac{TP}{TP+FN} \qquad (23)$$

F1 score: It is measure of the harmonic mean value between the precision and recall. It accesses the predictive ability of the classifier Eq. (24).

$$F\_Score = \frac{2(precision.Recall)}{Precision+ Recall} \qquad (24)$$

Specificity: Again, the specificity is the ability of the classifier to correctly predict the negative instances out of total negative instances in the dataset Eq. (25).

$$Specificity \ = \frac{TN}{TN+FP} \qquad (25)$$

false alarm rate: It is the ratio of the false positive detected by the classifier to the total self-samples detected Eq. (26).

$$FAR = \left(\frac{Fp}{(Fp+Tn)}\right) \qquad (26)$$

FNR: It is the ratio of the wrongly detected intrusions, to the total of correct and the wrong intrusion prediction Eq. (27).

$$FNR = \left(\frac{Fn}{(Fn+Tp)}\right) \qquad (27)$$

FPR: It is the total determination of the false positives in dataset, to the total response of the false positive and true negative Eq. (28).

$$FPR = \left(\frac{Fp}{(Fp+Tn)}\right) \qquad (28)$$

MCC: This metric classifies the efficiency of the binary classification model, and it is determined using the true positive, true negative, false positive and false negative respectively Eq. (29).

$$MCC = \left(\frac{((Tp \times Tn)-(Fp \times Fn))}{\sqrt{((Tp+Fp)(Tp+Fn)(Tn+Fp)(Tn+Fn))}}\right) \quad (29)$$

NPV: It is the measure of the likelihood of the classifier to determine the anomaly as the false information Eq. (30).

$$NPV = \left(\frac{Tn}{(Tn+Fn)}\right) \qquad (30)$$

PPV: It is the inverse of the NPV, which predicts the true positive, to the total positive response Eq. (31).

$$PPV = \left(\frac{Tp}{(Tp+Fp)}\right) \qquad (31)$$

RMSE: It is measured as the average difference in error response, from the predicted and the actual response Eq. (32).

$$RMSE = \sqrt{({O_{pred}}^2 - {O_{actual}}^2)} \qquad (32)$$

MAE: It is the magnitude in difference between the predicted and the actual outcome Eq. (33).

$$MAE = \left|(O_{pred} - O_{actual})\right| \qquad (33)$$

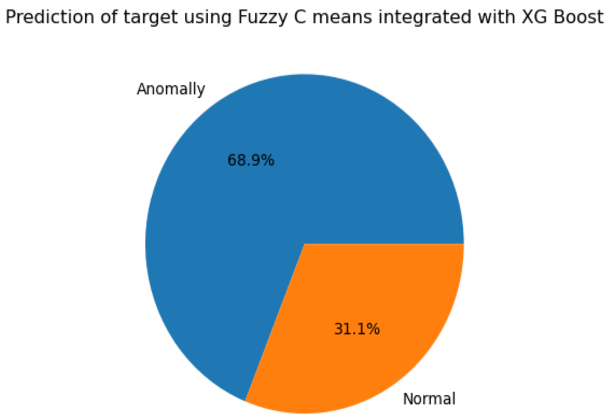

Figure 3: Detection plot of the proposed Fuzzy-XG Boost

Table.4**:** Comparative performance of the proposed Fuzzy-XG Boost vs Existing techniques	**Decision Tree**	**XG Boost**	**Proposed Fuzzy-XG Boost**
Accuracy	0.9625	0.9682	0.9992
Precision	0.9567	0.9599	0.9988
Recall	0.9542	0.9645	0.9987
F1- Measure	0.9555	0.9622	0.9987
Specificity	0.9685	0.9710	0.9994
FAR	0.031454	0.028997	0.000515
MCC	0.9458	0.9436	0.9981

NPV	0.9777	0.9773	0.9993
FPR	0.023354	0.024885	0.000602
FNR	0.030575	0.031102	0.001317
PPV	0.9678	0.9658	0.9988
RMSE	0.162455	0.165833	0.029031
MAE	0.026392	0.027501	0.000843

5.2 Comparative evaluation of proposed Fuzzy-XG Boost with the existing techniques

The Fig.4 gives the comparison of Decision Tree, XG-Boost and Fuzzy-XG Boost Classifier where the proposed technique outperforms the existing techniques and Fig.3 shows the prediction plot of the proposed Fuzzy-XG boost classifier on the NSL-KDD dataset. The proposed Fuzzy-XG Boost predicted the intrusion class as 68.9% and the remaining 31.1 % as the Normal class from the NSL-KDD dataset The performance evaluation study showed that the proposed Fuzzy-XG Boost classifier accurately estimates 99% of anomalies in the network, which is a dominant performance in the 5G environment.

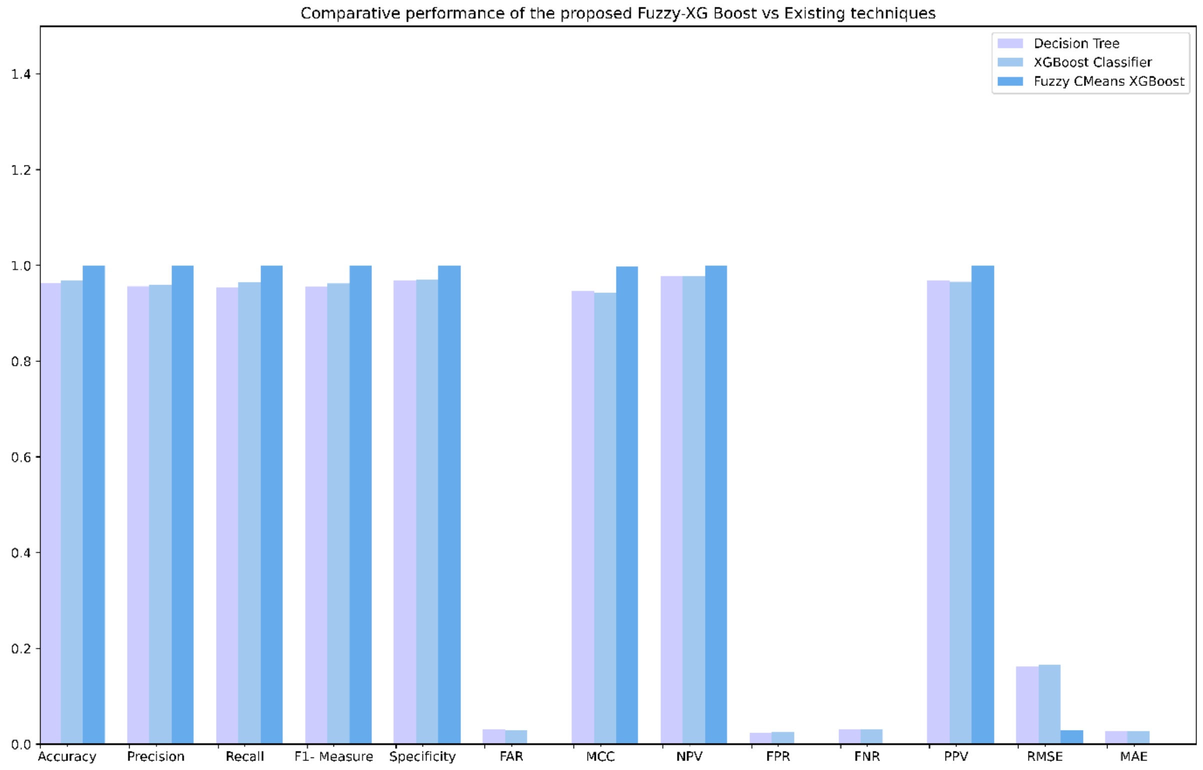

Figure 4: Comparative evaluation of proposed Fuzzy-XG Boost with the existing techniques

6. CONCLUSION AND FUTURE WORK

This work concentrated on developing an IDS framework for the SDN environment using the machine learners. For this purpose, the data was collected from the NSL-KDD dataset, and subjected to the pre-processing. RF classifier was employed in the feature extraction purpose, and the extracted features were given to the PCA for the dimensionality reduction. Final, set of features were given to the classification model trained with the proposed Fuzzy-XGBoost classifier to find the normal and anomaly class. The performance of the proposed Fuzzy-XGBoost classifier was evaluated on the metrics like accuracy, precision, recall, F1-measure, Specificity, False Alarm rate, MCC, NPV, FPR, FNR, PPV, RMSE, and MAE respectively and compared with the existing models like decision tree, and the XGBoost classifier. The proposed Fuzzy-XGBoost showed a higher performance rate, and thus had improved anomaly detection rate. the proposed Fuzzy-XGBoost classifier has an outstanding intrusion detection performance by attaining values as 0.999246, 0.998859, 0.998716, 0.998788, 0.999485, and 0.000515 for the metrics accuracy, precision, recall, F1-measure, Specificity, and False Alarm rate respectively. The proposed Fuzzy-XGBoost classifier has improved performance for the metrics MCC, NPV, FPR, FNR, PPV, RMSE, and MAE values of 0.9981, 0.9993, 0.000602, 0.001317, 0.9988, 0.029031, and 0.000843 respectively. As a future work directive, optimization algorithms can be incorporated in the fuzzy approach to prevent the trap of the local minima. Also, the performance can be evaluated with various other Intrusion detection datasets.

Declarations

Author Contributions

All the authors have contributed equally.

Conflicts Of Interest

The authors declare that they have no conflict of interest.

Notation list

Notation	Description
$G(P,Q)$	Margin Function
ϑ	Mean value of correlation
N	Number of data instance
p	Input
$I(.)$	Indicator Function
$V(p)$	Path vector
$\emptyset_m$	Leaf node
D_i	Input dataset
y	Scalar function
L(θ)	training loss function
R(θ)	regularization loss function

q_{ij}	Euclidean distance between the objects F_i and E_j
S	Leaf Node
d	Fuzzy Clusters
K	Dimensional features
L	Total no of leaf node
C	No of fuzzy clusters
E_j	cluster centre

REFERENCES

[1] Mazhar T, *et al.* Analysis of cyber security attacks and its solutions for the smartgrid using machine learning and blockchain methods. Future Internet Feb. 2023;15 (2):83.

[2] B. Gao, B. Bu, W. Zhang and X. Li, "An Intrusion Detection Method Based on Machine Learning and State Observer for Train-Ground Communication Systems," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 6608-6620, July 2022,

[3] Pallepati M. Network intrusion detection system using machine learning with data preprocessing and feature extraction. Int J Res Appl Sci Eng Technol Jun. 2022;10 (6):2360–5.

[4] Perera S, Jin X, Maurushat A, Opoku D-GJ. Factors affecting reputational damage to organisations due to cyberattacks. Informatics Mar. 2022;9(1):28.

[5] A. Zainudin, R. Akter, D. -S. Kim and J. -M. Lee, "Federated Learning Inspired Low-Complexity Intrusion Detection and Classification Technique for SDN-Based Industrial CPS," in IEEE Transactions on Network and Service Management, vol. 20, no. 3, pp. 2442-2459, Sept. 2023,

[6] L. Yang, Y. Song, S. Gao, A. Hu and B. Xiao, "Griffin: Real-Time Network Intrusion Detection System via Ensemble of Autoencoder in SDN," in IEEE Transactions on Network and Service Management, vol. 19, no. 3, pp. 2269-2281, Sept. 2022,

[7] Bandakkanavar R. Krazy Tech, TECHNICAL PAPERS Causes of CyberCrime and Preventive Measures Jan. 2023. Accessed: Jan. 03, 2023.

[8] Sarker IH. Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects. Ann. Data Sci., Sep. 2022.

[9] Jelen S. Intrusion detection systems: types, detection methods and challenges. securitytrails; Jul. 2020.

[10] Nalayini, C.M., Katiravan, J., Geetha, S. and JI, C.E., 2024. A novel dual optimized IDS to detect DDoS attack in SDN using hyper tuned RFE and deep grid network. Cyber Security and Applications, 2, p.100042.

[11] Duy, P.T., Khoa, N.H., Nguyen, A.G.T. and Pham, V.H., 2021. DIGFuPAS: Deceive IDS with GAN and function-preserving on adversarial samples in SDN-enabled networks. Computers & Security, 109, p.102367.

[12] Musleh D, Alotaibi M, Alhaidari F, Rahman A, Mohammad RM. Intrusion detection system using feature extraction with machine learning algorithms in IoT. J Sens Actuator Netw Mar. 2023;12(2):29.

[13] Vijayakumar DS, Ganapathy S. Machine learning approach to combat false alarms in wireless intrusion detection system. Comput Inf Sci Jul. 2018;11(3):67.

[14] J. E. Varghese and B. Muniyal, "An Efficient IDS Framework for DDoS Attacks in SDN Environment," in IEEE Access, vol. 9, pp. 69680-69699, 2021,

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

[16] Gupta, N., Jindal, V. and Bedi, P., 2022. CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. Computers & Security, 112, p.102499.

[17] Z. A. E. Houda, A. S. Hafid and L. Khoukhi, "MiTFed: A Privacy Preserving Collaborative Network Attack Mitigation Framework Based on Federated Learning Using SDN and Blockchain," in IEEE Transactions on Network Science and Engineering, vol. 10, no. 4, pp. 1985-2001, 1 July-Aug. 2023

[18] M. A. Razib, D. Javeed, M. T. Khan, R. Alkanhel and M. S. A. Muthanna, "Cyber Threats Detection in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework," in IEEE Access, vol. 10, pp. 53015-53026, 2022,

[19] L. M. Halman and M. J. F. Alenazi, "MCAD: A Machine Learning Based Cyberattacks Detector in Software-Defined Networking (SDN) for Healthcare Systems," in IEEE Access, vol. 11, pp. 37052-37067, 2023,

[20] R. Ben Said, Z. Sabir and I. Askerzade, "CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection," in IEEE Access, vol. 11, pp. 138732-138747, 2023.