IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 April 2024, accepted 28 April 2024, date of publication 2 May 2024, date of current version 23 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3395997

RESEARCH ARTICLE

Federated Learning for Privacy-Preserving Intrusion Detection in Software-Defined Networks

MUBASHAR RAZA[1,2], MUHAMMAD JASIM SAEED[1], MUHAMMAD BILAL RIAZ[3,4], AND MUHAMMAD AWAIS SATTAR[1]

[1]Department of Computer Science, Riphah International University, Lahore Campus, Lahore, Pakistan
[2]Department of Computer Science, Riphah International University, Sahiwal Campus, Sahiwal, Pakistan
[3]IT4Innovations, VSB—Technical University of Ostrava, Ostrava, Czech Republic
[4]Department of Computer Science and Mathematics, Lebanese American University, Byblos, Lebanon

Corresponding author: Muhammad Bilal Riaz (Muhammad.bilal.riaz@vsb.cz)

This article has been produced with the financial support of the European Union under the REFRESH – Research Excellence for Region Sustainability and High-tech Industries project number CZ .10.03.01/00/22_003/0000048 via the Operational Program Just Transition.

ABSTRACT Software-defined networking (SDN) is an innovative network technology. It changed the world of computer networking by providing solutions to many challenges. SDN provides programmability, easy and centralized network management, dynamic configuration, and improved security. Although SDN offers remarkable benefits but it provides centralized network management which is prone to attacks. So, intrusion detection systems (IDS) are essential to detect and prevent security attacks in SDN. Traditional IDS follow a centralized machine learning approach which causes vulnerabilities in IDS. Old-style IDS lack data privacy preservation, and solution for training data unavailability due to privacy. Federated learning (FL) is a distributed machine learning approach which provides a collaborative training approach without data sharing. In FL, training is performed on multiple nodes creating a global model without sharing the data. To address challenges and the limitations of traditional IDS, we proposed a FL based multi class classification IDS for SDN. FL delivers an efficient and scalable solution to address challenges of traditional IDS. The proposed model enhances security of SDN by not requiring the centralization of data. To test the impact and efficiency of proposed model, we used a latest and realistic cybersecurity dataset. We also compared the proposed model with state of art existing multi class classification studies. The results and their comparison with existing studies highlight the potential of proposed model to enhance network security while providing a privacy-preserving learning environment for intrusion detection.

INDEX TERMS Federated learning, intrusion detection, network security, software defined networks.

I. INTRODUCTION

SDN is a fascinating network technology. It solved many problems and challenges of traditional networks. It is a network management approach that provides many benefits. Although it is a major advancement in network management technology [1]. These days networks are growing unpredictably as a result of increasing number of devices on the Internet [2]. SDN rescued the computer networks for their growing demands and needs. SDN improved the security and performance of computer networks [3]. It added a factor of ease to managing networks. It offers a software-based centralized network management approach that makes computer networks dynamic, easy to configure, programmable, efficient, and reliable [4]. We can easily monitor networks and their traffic using SDN. Although SDN has many benefits, it also has some vulnerabilities [5]. Those vulnerabilities revolve around the centralized approach.

SDN consist of a layered architecture and it is divided into three planes. The first plane of SDN is application plane. The second plane of SDN is control plane, which is also known as control layer. The third plane of SDN is data plane

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

which is also known as infrastructure layer [3]. Each plane of SDN faces cyberattacks. The control plane is vital because it consists of a SDN controller, a central intelligence of SDN. SDN is a prominent technology that splits forwarding process of network packets from the routing process. The control plane of SDN is considered a brain, so in most cases, attackers try to hijack it [2]. If attackers somehow hijack the control plane, they can control the entire network.

Machine Learning proved helpful against cyberattacks in the last few decades [6]. These days machine learning is a prominent solution in cybersecurity. In this era, the number of devices has increased, and they are still increasing a huge number [6]. As the digital world grows, so does the number of cyberattacks. So, we need an automotive solution against cyberattacks. Machine learning provides this kind of solution. Machine learning can identify if an unauthorized device is trying to connect network. Machine learning can detect if a device has become a zombie. Machine learning can detect a new form of malware using previous signatures [7]. Machine learning can also be used to recommend security policies.

FL is a decentralized machine learning approach. It is a machine learning technique in which we train models on decentralized devices or servers. Training models on decentralized devices are performed without sharing the training data [8]. Each contributor has its local training data. FL is opposite of centralized machine learning, in which all the training data is gathered on a centralized server, and model training is performed. Instead of sharing the data, contributors share the weights of the model [9]. Weight sharing is performed in the iterations until the contributors get the best accuracy. FL permits multiple contributors to build a common model, which is also known as a global model. This global model contains the efficiency of all the contributors. FL solves many data-related issues, including data privacy, unavailability of training data due to privacy and unauthorized data access [10]. These days privacy and data security are prominent issues. Data privacy is preserved by FL [11].

In computer networking, we use IDS to deal with cyberattacks. An IDS monitors computer networks and takes predefined actions if it finds an anomaly in network traffic [12]. An IDS can be a dedicated hardware, software application, or combination [13]. The activity of IDS can vary as per the IDS category and the organization's requirements. Generally, an IDS reports unusual and malicious activity in the network to the administrators. Some IDSs are configured so that they try to prevent attacks; in this case, this kind of IDS is termed an intrusion prevention system (IPS).

We used a dataset [14] named as Edge-IIoTset based on cyber security to test the proposed model. Edge-IIoTset data set was specially created for IDS. According to authors, the data set can be used for FL. The authors designed a specific system to generate and capture network traffic. The designed system contains specific layers, including a SDN layer, network layer, IoT layer and edge computing layer. Edge-IIoTset consists of 61 input features and 2 target features. 61 input features consist of traffic data. First target variable is named as attack label. The attack label variable consists of two values, either 0 or 1. 0 indicates normal traffic, whereas 1 indicates intrusion. The second target variable is named as attack type. The attack type has fourteen classes of attack including MITM, Uploading, Ransomware, SQL_injection, Port_Scanning, DDoS_HTTP, DDoS_TCP, Password, Vulnerability_scanner, XSS, Backdoor, Fingerprinting, DDoS_UDP, and DDoS_ICMP.

A. RELATED WORK

Priyadarsini and Bera [3] called SDN the future of enterprise networks. The Authors said SDN is a dynamic approach to network configuration. They said SDN faces many server-based attacks due to its centralized control approach. DoS, spoofing, intrusion, and policy violation are some examples. They suggested that a dynamic security enforcement mechanism can help to deal with these attacks. Karmakar et al. [2] said that despite all the benefits, security of SDN remains an open issue. They discussed that a centralized architecture makes it vulnerable to attacks. Cited research paper proposed a threat model to deal with the attacks. In the control plane, a software error in the controller, a malicious application running on SDN controllers, topology poisoning, and threats from the other network devices can cause a serious security threat to SDN.

Apruzzese et al. [7] said that flooding attacks are one of the easiest attacks launched on networks. The attackers send low-rate traffic toward the control plane, which seems to be legitimate. In case attack becomes successful, then this causes a disconnection between control plane and data plane. They claimed that only IDS could prevent this attack. Novaes et al. [15] highlight different security attacks in SDN, which includes DDoS attacks and Portscan attacks. They proposed a mitigation system for these attacks. The proposed mitigation system is a combination of fuzzy and Long short-term memory (LSTM). Lee et al. [1] investigated how SDN architecture fails to prevent the arising security policy issues in the components. They proposed a fuzzy-based testing model to avoid the rising security policy issues among the SDN components.

1) INTRUSION DETECTION SYSTEMS FOR SDN

Haider et al. reported that DDoS attacks are the most common attacks these days. They are rapidly increasing in number on a daily basis [16]. They are a big problem for networks. SDN has a good immune system against security threats, but still, it has many vulnerabilities. DDoS attacks are a big security threat for SDN. The cited research article proposed a security solution against DDoS attacks. To mitigate the DDoS attack threat, the authors proposed an IDS that uses convolutional neural networks (CNNs) and an ensemble approach. They claimed that their proposed IDS is amazing and efficient against DDoS attacks.

Zhu et al. said there are many forms of Residual networks (Resnet) available these days [17]. Each form is an improvement to the previous version. Residual networks are used in a large number of IDS. Their performance and efficiency are quite impressive. The authors proposed another improved form of residual networks. They claimed that their proposed residual networks have a shorter detection time. They said that previous residual networks used in IDS take undesirable time to detect the attacks. The authors claimed that their proposed IDS has a low detection time.

Phan and Bauschert reported that cyber-attacks cause huge data losses [18]. In SDN, it is quite challenging to mitigate cyber-attacks. SDN improved cyber security as compared to traditional computer networks. SDN proved itself an innovative solution against challenges we faced in the last couple of decades. In SDN, communication protocols are required for communication between data plane and control plane. As control plane is a central intelligence of SDN so it requires mitigation solutions. The cited research paper proposed an IDS based on a double-deep Q-Network. They also proposed an intrusion response policy based on reinforcement learning. They focused on DoS attacks for model evaluation.

Sahoo et al. reported that SDN became a fascinating network architecture in this modern era [19]. It gives a chance to network administrators to take more control over computer networks in such an easy way. The controller of SDN is also known as the operating system of SDN. The centralized controller maintains and runs different applications and network services. Regardless of all these benefits, the controller of SDN is a prime target for attackers. DDoS attacks are growing rapidly in SDN because the control plane is vulnerable. The cited research article proposed an IDS grounded on a Support vector machine (SVM), kernel principal component analysis (KPCA), and genetic algorithms (GA). They tested their model against DDoS attacks.

ElSyed et al. said the SDN fulfills the requirements and demands of today's advanced data centers and enterprise networks [20]. The centralized approach of SDN provides many functions that are beneficial. SDN architecture provides a wide range of paybacks. Risk attacks are a big threat to SDN architecture. If attackers are able to control the control plane of SDN, then they can manage to control and route the traffic in their own way. Network intrusion detection systems (NIDS) can help to secure the SDN architectures. Deep learning is a good approach in NIDS. The cited research article proposed a CNNs-based IDS, which uses a regularizer method to avoid overfitting. Overfitting leads to poor performance of IDS.

Ravi et al. proposed an IDS [21]. This IDS is proposed specifically for SDN-based Internet of Things (IoT) networks. The demand for IoT networks is increasing as the number of IoT devices is increasing rapidly. The authors said an IDS is essential for security of SDN-based IoT networks. In their proposed model, the authors used (kernel-PCA) and gated recurrent units (GRU) to extract the features of data. Later on, they fused the features extracted by both techniques to train the neural networks. They used the final trained NN model for the classification of attacks in the network.

Yang et al. proposed an IDS for SDN [22]. They claimed that their proposed IDS could detect zero-day attacks. They used an unsupervised machine learning approach to build an IDS. They used an ensemble autoencoder to train the unsupervised machine learning model. They claimed that their proposed model has less complexity as compared to the previous NIDS. They also used a privacy preservation frame to protect the data privacy of users. In their proposed privacy-preserving framework, they added noise in the dataset to avoid data leakage and privacy breach of users.

Ahmed et al. said SDN had become part of many IT infrastructures since they were born [23]. Vehicular networks are also using SDN. An SDN-based vehicular network is also known as a software-defined vehicular network (SDVN). Ahmed et al. proposed an IDS and load balancing system for SDVN. For intrusion detection, they proposed an algorithm which is named by them active deep learning. They used a previously available dataset to test and validate their IDS. They claimed that their proposed system is more efficient as compared to the previous ones.

Prathibha et al. reported in their research work [24] that SDN was affected by DDoS, TCP SYN flood, and TCP ACK flood attacks. Another literature by Lanksky et al. [12] reported Brute force, DDoS, Worms, Malware, Web, and browser attacks. Another review paper [25] by Cui et al. found DDoS attacks in different research articles related to intrusion detection in SDN. Rasool et al. [26] reported that the centralized control approach causes serious vulnerabilities, which result in severe attacks. Authors focused on link flooding attacks, and they said it is one of the easiest attacks launched on networks. The attackers send low-rate traffic toward the control plane, which seems to be legitimate. In case this attack becomes successful, then this causes a disconnection between control plane and data plane. They claimed that only IDS could prevent this attack.

2) FEDERATED LEARNING

Data privacy preservation remained an open issue for decades. FL provides data privacy preservation [11]. FL provides a solution that solves the issue of centralized machine learning [27]. Li et al. proposed a smart healthcare system which uses FL for privacy preservation [28]. Their proposed system is a part of IoT. Their proposed system detects Alzheimer's disease. Their proposed system uses voice samples collected from user's IoT voice devices for disease detection. The Internet of medical things (IoMT) is very popular in the healthcare sector. Privacy of patient data is a big issue in IoMT. Han et al. proposed a teledermatology framework based on FL [29]. In their framework, mobile phones are used to collect images and transfer them.

Ines Feki et al. proposed a FL based approach to classify patients for COVID-19 using chest X-ray images [27]. Fan et al. performed their research on IoMT [30]. They proposed a FL based IoMT framework. Their proposed system provides healthcare services. Guo et al. worked on

TABLE 1. Limitations of previous studies.

Studies	Contribution of works	Limitations of works
Haider et al. [16]	To mitigate the attacks in SDN the main contribution of this research was CNNs and ensemble-based model against DDoS attacks.	This model can only detect DDoS attacks. Privacy of training data is vulnerable.
Sahoo et al. [19]	The main contribution of this research was SVM and GA based IDS against DDoS attacks in SDN.	The proposed model does not provide training data privacy and protection against unauthorized access to data. SVM could cause huge delays in training due to big data processing.
Zhu et al. [17]	This research proposed Resnet based IDS for SDN to classify network traffic in intrusion or normal.	Classification of traffic into intrusion or normal is not enough, we need to know the class of attack for proper mitigation.
ElSayed et al. [20]	CNNs based IDS was the main contribution of this research to classify traffic in SDN in intrusion or normal.	Training data privacy preservation is missing.
Phan et al. [18]	Double deep Q-Networks based IDS was proposed in this research to classify the traffic in SDN in intrusion or normal.	The model could classify only two classes.
Ravi et al. [21]	An IDS for SDN based IoT was proposed using GRU to detect network intrusion.	The data is shared among the IoT devices without taking care of privacy.
Yang et al. [22]	Ensemble Autoencoder was used to detect zero-day attacks in SDN.	Noise is added to training data to avoid data leakage, noise could harm data integrity.
Ahmed et al. [23]	A model named Active Deep Learning was proposed for Software defined vehicular network (SDVN) to detect intrusion.	Privacy of vehicle data is at risk. Data leakage expected.
Ahmad et al. [38]	A model was proposed based on FL for SDN based industrial CPS to detect attacks.	The model could detect six classes only.
Himanshi et al. [39]	A model was proposed to detect attacks on SDN from IoT based consumer devices.	They assigned the same label 1 to 9 classes and 0 to normal class of traffic.
Jie et al. [40]	A collaborative IDS for SDN based vehicular and ad hoc networks.	An IDS which can detect 5 classes of attack.
Zakaria et al. [41]	An FL based mitigation system was proposed for ORAN	A mitigation system which can detect 5 classes of attack in ORAN.
Zakaria et al. [42]	A blockchain and FL based IDS for vehicular edge computing.	An IDS which is evaluated using 10 classes of attack.
Zakaria et al. [43]	An IDS based on FL and explainable AI for IoT networks	Evaluation of model using old dataset.
Zakaria et al. [44]	A network mitigation system based on SDN and FL.	Accuracy achieved by the model is not promising.
Zakaria et al. [45]	A security framework based on FL for industrial IoT.	The proposed model is evaluated using 10 classes of attack.
Bukhari et al. [46]	An FL based IDS for Wireless sensor networks.	The proposed model is evaluated using 5 classes.

computer-aided diagnosis (CAD) [31]. They proposed a medical data processing method that uses FL to take care of medical data privacy. Elayan et al. worked on medical data privacy [32]. Their proposed model uses FL for healthcare data monitoring and analysis.

Driss et al. performed research work on intrusion detection vehicular sensor networks (VSN) using FL [33]. Their proposed system uses a combination of GRU and Random Forest based ensemble techniques. Li et al. proposed a FL based NIDS [34]. Huang et al. proposed an IDS using FL [35]. They proposed a system for intrusion detection in cyber-physical systems (CPS). Another research article by Zhao et al. proposed [36] an IDS using FL for IoT environments. Liang et al. proposed an IDS to detect cyber-attacks in advanced metering infrastructure (AMI) systems [37]. They used DNN as a training model with FL in their proposed model. According to them, AMI has a vital role in the smart grid (SG), so it is prone to cyber-attacks.

Zainudin et al. [38] proposed FL based IDS for SDN based Industrial CPS, they proposed a multi class model. Babbar and Rani [39] proposed an FL recommender IDS for SDN. Their model detects the attacks coming from IoT based consumer devices. Cui et al. [40] proposed a collaborative IDS for SDN based vehicular and ad hoc networks. They also proposed the multi class detection model.

Houda et al. [41] proposed a FL based jamming attack mitigation system for open radio access network (ORAN). They have tested both binary and multi class approach. Houda et al. [42] proposed a blockchain and FL based IDS for vehicular edge computing. Houda et al. [43] proposed an IDS based on FL and explainable AI for IoT networks. Houda et al. [44] proposed a network mitigation model based on FL, blockchain and SDN. Houda et al. [45] proposed a security frame work based on FL for industrial IoT applications. Bukhari et al. [46] proposed an FL based IDS for wireless sensor networks.

B. LIMITATIONS AND DRAWBACKS OF EXISTING STUDIES

Huge amount of data must be collected from different sites and must be sent to a distant central server for model training, which requires data transmissions causing delays. These data transmissions and delays have significant impact on model training. The whole process of traditional centralized model training is prone to attacks and data privacy violations. Data owners of network traffic are worried about the data privacy. Furthermore, this privacy concern of owners causes unavailability of training data. The traditional centralized model training failed to provide data privacy and security. Data transmissions to a central server from different sites can compromise the quality and integrity of data.

Major limitations of previous research works have been summarized in Table 1. The previous research works did not provide privacy of training data. The studies used centralized machine learning approach which has many drawbacks. Some of the previous research work could classify only DDoS attacks and some of them could classify traffic in intrusion and normal only. Previous studies overlooked the problem of big data processing, which could cause delays in training. Data leakage problem was overlooked as well. Some studies of them could classify the intrusion into important classes of attacks. To implement proper mitigation of attacks, identification of cyber-attack class is vital.

Let's compare the existing studies with this research work. The proposed model by Haider et al. [16] can only detect DDoS attacks. Privacy of training data in this model is vulnerable. Whereas the proposed model can detect 11 types of important attacks and provides privacy for training data. The study conducted by Sahoo et al. [19] does not provide training data privacy and protection against unauthorized access to data. SVM could cause huge delays in training due to big data processing. Whereas the proposed model overcomes the challenges of their study by using unique FL and approach. The study proposed by Zhu et al. [17] can only classify the traffic into normal or intrusion. Their study does not classify the attacks into specific type. Classification into specific type of attacks, helps to mitigate them and prevent them precisely. The proposed work classifies the attacks into specific types, which adds the uniqueness to this research work.

The research conducted by ElSayed et al. [20] does not provide privacy preservation and security to the training data where the proposed model provides the privacy. The model proposed by Phan and Bauschert [18] can classifies the attacks into two classes only whereas the proposed research can classifies important 11 attacks. Model proposed by Ravi et al. [21] shares the data among the IoT devices without taking care of privacy whereas overcomes this vulnerability. Yang et al. [22] added noise to training data to preserve the privacy but this approach hits the integrity of training data whereas the proposed model preserves the privacy of data without adding any noise. Model proposed by Ahmed et al. [23] shares the vehicle without taking care of any data security whereas the proposed model provided the data security.

Zainudin et al. [38] proposed an IDS which can detect 6 classes of attack whereas the proposed model is evaluated using 11 classes of attack and it achieved higher accuracy then their model. Babbar and Rani et al. [39] proposed and IDS which can detect 10 classes. They assigned the same label 1 to 9 class and 0 to normal class of traffic. So, their model is not able to tell us which type of attack detected. Furthermore, they evaluated their model on old datasets. Whereas the proposed model is evaluated using 11 classes and it can tell us the type of attack. Cui et al. [40] proposed an IDS which can detect 5 classes of attack whereas the proposed model is evaluated using 11 classes of attack and it achieved higher accuracy.

Houda et al. [41] proposed a FL based mitigation model for ORAN. They have evaluated their model using 5 classes of attack. Whereas we proposed an IDS for SDN which can detect 11 classes. Houda et al. [42] proposed an IDS which is evaluated using 10 classes whereas the proposed model is evaluated using 11 classes. Zkaria et al. used an old dataset to evaluate their proposed model whereas this research used a latest and state of art cyber security-based dataset. Houda et al. [44] proposed a network attacks mitigation framework. The proposed model achieved higher accuracy as compared to their model. Houda et al. [45] proposed a security framework for industrial IoT applications. They evaluated their model using 10 classes whereas the proposed model is evaluated using 11 classes and a latest cybersecurity dataset. Bukhari et al. [46] proposed an FL based IDS for wireless sensor networks. They have evaluated their model using 5 classes of attack whereas the proposed model is evaluated using 11 classes. The higher number of classes increases its scope and efficiency.

C. MOTIVATION

In section II of I and B of I, we discussed and observed that FL can be a practical approach to address the issues faced by traditional machine learning based IDSs. After getting

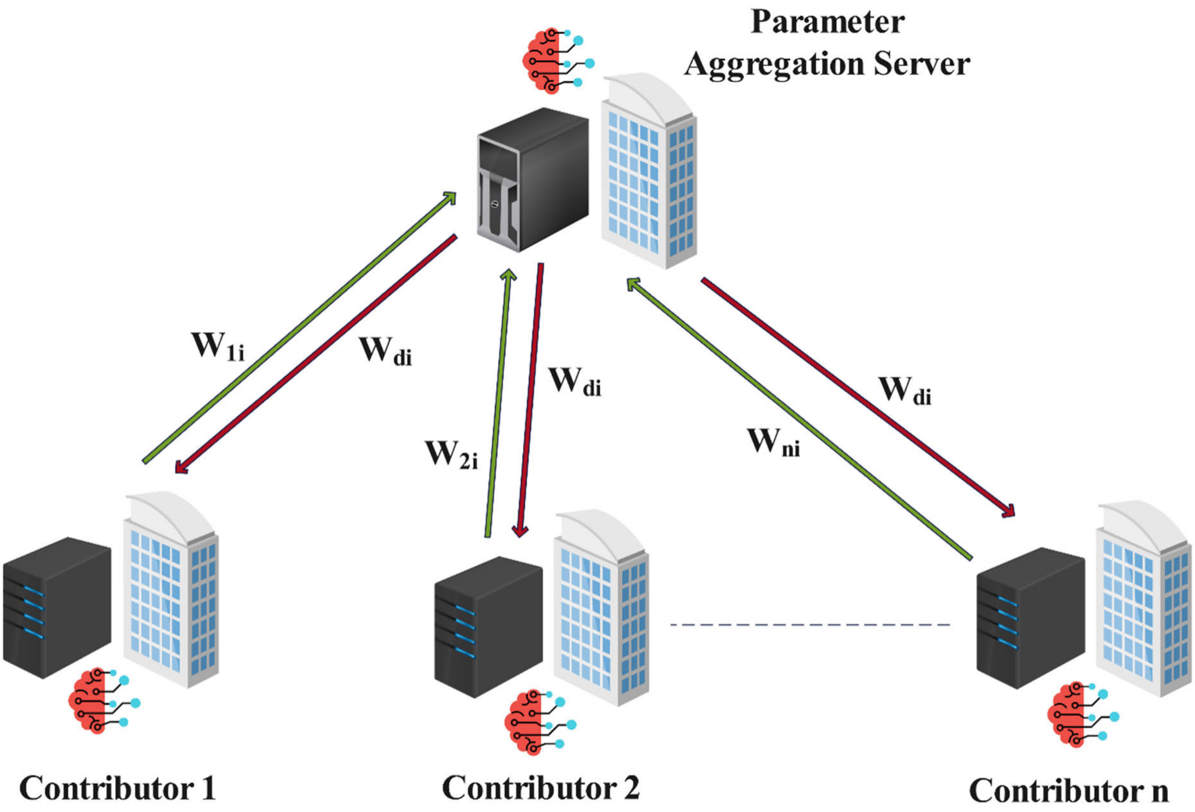

FIGURE 1. System architecture for privacy preserving intrusion detection in software defined networks.

motivation from those observations, this study proposes an FL based IDS, that can be implemented in SDN to protect it from cyber-attacks. There are many advantages of using FL approach, as compared to non-FL approaches [11], [27], [47]. FL provides edge devices-based training. In FL data is saved on local devices which ensures the security of data. FL is an upgradation of traditional machine learning approaches. Experimental results shows that the proposed IDS can be a practical and efficient approach against cyber-attacks.

D. PROBLEM FORMULATION

SDN is an innovative and comparatively new approach to networking. It faces many security challenges. IDS is a good solution to detect and prevent security attacks and breaches in SDN. Traditional IDSs have many flaws and draw backs. We already have discussed those drawbacks in detail in previous sections. To overcome their drawbacks, this study we propose FL based innovative IDS approach. Fig. 1 shows the system architecture and Fig. 2 shows the basic work flow of model. On client side there can be 1 to n contributors. Each contributor has been provided its own local training data. Each contributor trains their local model. After training the contributors sent their parameters to server. Server aggregates the parameters using Equation 13. Then server send the update to contributors. Now, contributors update their parameters. Here first global iteration completes. The process of training continues until the global model converges. The algorithm for client-side training is given in Algorithm 1 and algorithm for server side is given in Algorithm 2.

Algorithm 1 Contributor Side

```
1: Start
2: Local data splitting into training and validation
3: Initialize the weights and layers of training model
4: Initialize the epochs of training model
5: for number of epochs of training model
   i. Apply the feed forward step of ANN
      a. Calculate $Ƨ_i$ using equation 1
      b. Calculate $\hat{R}_k$ using equation 2
   ii. Calculate $\Psi$ using equation 5
   iii. Weights updating step
      a. Calculate $v_{jkl}^{c_i}$ using equation 9
      b. Calculate using $\Delta w_{ij}^{ci}$ equation 10
      c. Update the weights between hidden
         and output layer using equation 12
      d. Update the weights between hidden
         and input layer using equation 11
   iv. If number of local iterations do not end
      go to step 5 else go to step 6
6: Send optimized wights to aggregation server
7: Stop
```

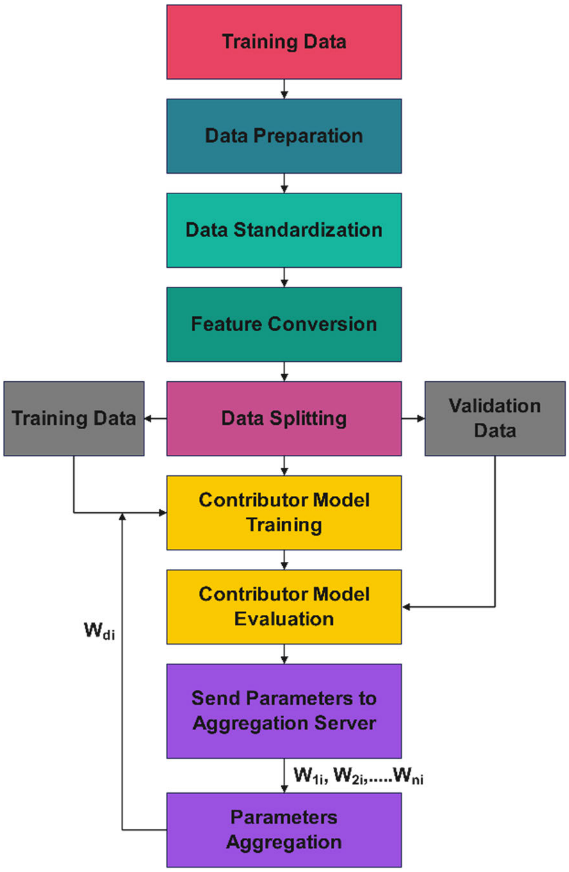

FIGURE 2. Basic work flow of model.

Algorithm 2 Server Side

```
1: Start
2: Initialize server with initial weights
3: for each global iteration
   i. each contributor
      a. Check weights dimension and equalize
      b. Aggregate weights using equation 13
    end
   end
4: Perform predictions
   i. For number of validation samples
      a. Calculate $Ƨ_i$ using equation 1
      b. Calculate $\hat{R}_k$ using equation 2
      c. Calculate $\psi$ using equation 5
6: Send parameter update to contributors
7: Stop
```

E. MAIN CONTRIBUTIONS

The main contributions of this research work are as follows:

- In this research work we implemented FL based IDS to perform multi class classification of security attacks in SDN. The proposed model ensures the data privacy preservation and avoids the data leakage problem by not requiring the transfer of training data on centralized server. The proposed model also eliminates the requirement of big data processing by not requiring the training data on central place.
- The proposed model is trained and validated using a latest cyber security dataset ‘‘Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning’’. The dataset was created using real time SDN traffic.
- As a baseline model of FL, we proposed an optimized ANN architecture which is simple, efficient and result oriented. The proposed ANN architecture has been evaluated by varying a range of hyperparameters.
- We used PCA to reduce the data dimensionality. Low dimensional data helps to reduce computation power and reduces latency.
- Most of the models proposed in the existing studies could classify binary or single class of attacks. This restriction made their scope and performance limited. The proposed model can classify attacks into 11 important classes. Such classification is beneficial for mitigating attacks. We also compared the proposed model with the latest and state of art existing multi class classification models to explain its effectiveness and novelty.
- We implemented a detailed experimental evaluation using various performance metrics and compared the results with latest existing studies, results show that the proposed model provides better scope and performance.

II. METHODOLOGY

This research work focuses to improve the security of SDN by improving IDS. In this research work, we used an advanced machine learning technique to implement an IDS for SDN. Centralized machine learning algorithms and techniques face many challenges. FL is an advanced machine learning technique which offers a solution to data privacy preservation in machine learning and improves the performance and security. The proposed model solves challenges of centralized machine learning and provides a highly efficient solution for intrusion detection in SDN.

There are few techniques and operations that we have performed to preprocess and clean the dataset. Duplicate records in a dataset are not good for any machine learning model. They badly affect the training and validation of the model. So, we must drop them to improve and maintain the model's accuracy and avoid misleading performance. First, we checked the null values in the dataset using isnull.sum() command and it showed 0 null values in each feature. Then we checked for na values using isna.sum() command, and the result was the same; there are no na values in the features. And at the end, we checked the duplicate values using duplicated-value_counts() commands, and it showed 235973 duplicate records. So, we dropped the duplicate records using the drop_duplicates(subset = None, keep = "first", inplace = True) command.

Machine learning algorithms do not accept and process categorical data. They require variables in numerical form. Label encoding is a process and method in which we convert categorical variables into numerical form. There are a few techniques and algorithms available in Python that convert the categorical variables into numerical form each one has its pros and cons. We used LabelEncoder from sklearn to convert categorical variables. It converts categorical data into 0 to n-1 integer values. LabelEncoder is a simple approach, which was suitable. We assigned labels to classes using LableEncoder.

Later on, we splitted the data into 70% and 30%. 70% data was used for training and the rest of the 30% was used for model validation. The 70% data was equally divided among the contributors for training purpose. The principal component analysis (PCA) is a feature extraction and dimension reduction technique. Principal component analysis reduces the data dimensionality, redundancy and preserves the maximum information [38], [39]. Principal component analysis converts the n features into k principal components by preserving the maximum information. In IDS, a huge amount of data is required for model training which requires high computation power and causes delays in the training process. Principal components are created using a mixture of early variables. Newly created variables or principal components are created in such a way that they have a minimum correlation. There are four classes in the attack-type target variable which has a negligible appearance in the dataset. We dropped those classes to improve the performance of the machine learning model. After applying the PCA and Data pre-processing technique final shape of dataset becomes (1957249, 23).

A. BASELINE MODEL ARCHITECTURE

Each contributor in FL uses a machine learning algorithm as a machine learning model and training algorithm. There are many machine learning algorithms available that can be used in FL as a baseline model. FL is based on play with parameters in the form of weights. So, in FL, parametric methods are preferred as a baseline model. Parametric algorithms provide a facility to learn, adjust and define the training parameters according to the dataset. In this case, we used ANN as a baseline model, which is a parametric method. we created a simulation environment using Python programming to test the proposed model. Before performing the training, we further divide the training data horizontally among contributors. The reason behind horizontal division is that in IDS, we need traffic data for training, and most traffic data have the same features, so it is the best fit for problem at the hand.

The architecture of the ANN baseline model contains 4 layers in total. First comes input layer, then we added two hidden layers, and at last, we added an output layer. The first three layers contain the relu activation function, and the output layer contains the softmax activation function. We used a sequential model from TensorFlow to implement the baseline model. The sequential model provides the facility to create different kinds of deep learning models containing layers. The input layer contains 64 neurons, first hidden layer contains 64 neurons, second hidden layer contains 32 neurons, and output layer contains 11 neurons.

Let's discuss and cover some details to explain the decision-making process for ANN architecture further. We started with 1 input layer, 1 output layer, and 1 hidden layer. It was the minimum requirement for ANN architecture. The output layer contains 11 neurons; we cannot change that number because it depends upon the number of output classes, and they are 11; in this case, we set the 11 neurons in the output layer. The number of neurons in the input layer is 64. As we discussed before, they should be more than input features; we have 23 input features for training, so first of all, we set the neurons in the input layer to 32. And we set 32 neurons in the hidden layer. At that time, we had 3 layers in total. We tested the model, and it gave an accuracy of around 80.

To further optimize the model, we increased the neurons in the input layer and hidden layer to 64; it boosted the accuracy to up to 95%. After that, we increased the neurons in the input layer and hidden layer to 128 and 128, but they were not practical; then we expanded it to 256 and 256, but the result was the same, it increased the execution and training time of the model it was like diminishing returns. So, we decided to keep it 64 and 64. To further optimize the model, we added another hidden layer with 32 neurons, which increased the model's accuracy by more than 98%. Then we increased neurons in the newly added layer to 64 and then 128, but it was ineffective. We added more hidden layers to the model, but they did not give positive results, so we decided to keep the architecture with 2 hidden layers.

B. HYPERPARAMETERS

Hyperparameters are parameters of deep learning models whose values are adjustable, and they are set before starting the model's training. Before setting the final values of hyperparameters, we set certain values and then train the model; this process is repeated until we get the best result. A very high learning rate causes the model to converge in less time, affecting accuracy. A very small learning rate causes the model to converge too late. So, the learning rate must not be too low or too high. In this case, we used a learning rate of 0.001. We tested the model by setting the learning rate from 0.1 to 0.001. And we got the optimal result on 0.001. We used the Adam optimizer. We also tried some other optimizers, but we found that Adam is the best suitable. We used two activation functions in the baseline model. Relu (Rectified Linear Unit) is used in input and hidden layers, whereas softmax is used in output layer. We used softmax in the output layer due to multi-class problem. Due to multiclass problem, we used categorical_crossentropy. The proposed model is different from traditional models due to the FL approach. So, we set 1 epoch for each contributor or local model. We have sufficient data, so we set the batch size to 64. It is the batch

TABLE 2. Simulation results.

Contributors	Global Iterations	Global Accuracy	Global Loss
3	10	98.21%	0.040
3	30	98.65%	0.033
3	50	98.64%	0.031
5	10	98.29%	0.040
5	30	98.43%	0.036
5	50	98.56%	0.033
10	10	98.16%	0.044
10	30	98.39%	0.038
10	50	98.43%	0.036
10	100	98.63%	0.033

size for each contributor. We tried 32, 64, 128, and 256 batch sizes, but only 64 was more effective.

The proposed system is tested by varying the number of global iterations and contributors. Simulation results show that changing the number of contributors does not affect the accuracy and performance of proposed model. More global iterations can give more precise results but we find that 50 to 100 global iterations are enough although they can be varied depending upon the dataset. More details about global iterations and their impact are discussed in section ''IV. Simulation Results'' and a summery is given in Table 2.

C. MODEL TRAINING

The following steps explain and highlight the overall training process of proposed system.

1) CONTRIBUTOR TRAINING

All the contributors or clients perform their training on provided local data. In the simulation environment, we created a function that trains a set number of contributors and stores their parameters.

2) AGGREGATION SERVER PARAMETERS UPDATE

Each contributor sends weights or parameters to the aggregation server after training.

3) PARAMETERS AGGREGATION

After receiving the parameters from contributors or clients, the aggregation server scales and averages the weights using mathematical operations. We used an algorithm named Federated Averaging (FedAvg) to perform aggregation.

4) PARAMETERS BROADCASTING

After performing the aggregation, the aggregation server broadcasts the averaged parameters to the clients.

5) CONTRIBUTORS UPDATE

The clients update their parameters and test the performance on receiving the parameters or weights. At this step, one common iteration completes.

6) REPEAT

On completing step 5, the system returns to step 1 and repeats until step 5. In this way, required number of common iterations can be performed until model converges.

D. COMMUNICATION ARCHITECTURE

In FL, the communication architecture is given by the FedAvg algorithm [11]. The transmission control protocol (TCP) can be used for sharing the parameters between contributors and the central server. Significant communication is required, and this process is repeated until the model converges, so this process can cause significant communication overhead if the number of clients participating in the FL process is too large. Due to this communication overhead, there can be latency introduced in the training process. The latency depends upon the communication overhead, and communication overhead relies on the efficiency of the communication network. So, a highly efficient communication network can reduce the communication overhead and latency. The purpose of FL is to preserve privacy, although the clients and central server can also be subject to cyberattacks. To minimize this concern, secure aggregation protocols and encryption methods can be used [48].

E. SYSTEM ARCHITECTURE

Earlier we have discussed the training steps of proposed systems. Let's discuss the proposed systems architecture in further detail using graphical representation. In Figure 1, the graphical model is shown. In Figure 1, W_{1i}, W_{2i}, up to W_{ni}, represent the weights sent by contributors to aggregation server whereas W_{di}, represents the aggregated weights sent by aggregation server to contributors. Controllers and aggregation servers are based on SDN controllers.

1) CLIENT SIDE

In this research, ANN architecture has been used as a baseline model for training. In the ANN architecture four layers has been used in total including one input layer, two hidden layers and one output layer. The first three layers contain the relu activation function, and the output layer contains the softmax activation function. Softmax has been used because the problem being addressed is multi-class problem. The softmax activation function is given in equation 1, where k = 1,23…n. The relu activation function is given in the equation 2, where j = 1,2,3…n. Equation 3 gives the mathematical form of input layer where j = 1,2,3…n. Equation 4 gives the mathematical form of output layer where k = 1,2,3…n. Equation 5 gives the mathematical form of error calculation, where c_k and $\hat{\mathrm{R}}_k$ represent the actual output and calculated output.

$$\text{Ƨ}_i = \frac{e^{(\tilde{\mathrm{N}}_i)}}{\sum_{j=1}^{K} e^{(\tilde{\mathrm{N}}_j)}} \tag{1}$$

$$\hat{\mathrm{R}}_k = \max(0, \text{ɧ}_k) \tag{2}$$

$$\tilde{\mathrm{N}}_j = n_1 + \sum_{i=1}^{r} (w_{ij} \times m_i) \tag{3}$$

$$\mathfrak{h}_k = n_2 + \sum\nolimits_{j=1}^{p} (v_{jkl} \times S_i) \tag{4}$$

$$\Psi = \frac{1}{2} \sum\nolimits_k (c_k - R_k)^2 \tag{5}$$

Equation 6 gives the mathematical form of rate of change in weights for the output layer.

$$\Delta v_{jkl}^{c_i} = - \in \frac{\partial \psi}{\partial v_{jkl}^{c_i}} \tag{6}$$

After performing mathematical operation on equation 6, equation 7 has been obtained.

$$\Delta v_{jkl}^{c_i} = - \in \frac{\partial \Psi}{\partial \hat{\mathrm{R}}_k} \times \frac{\partial \hat{\mathrm{R}}_k}{\partial \mathfrak{h}_k} \times \frac{\partial \mathfrak{h}_k}{\partial v_{jkl}^{c_i}} \tag{7}$$

After performing the substitution in equation 7, equation 8 has been obtained.

$$\Delta v_{vjkl}^{c_i} =\in (c_k - \hat{\mathrm{R}}_k) \times \hat{\mathrm{R}}_k(1 - \hat{\mathrm{R}}_k) \times \hat{\mathrm{R}}_j \tag{8}$$

And after substituting $\boldsymbol{\varphi}$ in equation 8, equation 9 has been obtained.

$$\Delta v_{jkl}^{c_i} =\in \varphi_k \hat{\mathrm{R}}_j \tag{9}$$

$$\Delta w_{ijl}^{ci} \propto -[\sum\nolimits_k \frac{\partial \Psi}{\partial \hat{\mathrm{R}}_k} \times \frac{\partial \hat{\mathrm{R}}_k}{\partial \mathfrak{h}_k} \times \frac{\partial \mathfrak{h}_k}{\partial \hat{\mathrm{R}}_j}] \times \frac{\partial \hat{\mathrm{R}}_j}{\partial \mathfrak{h}_j} \times \frac{\partial \mathfrak{h}_j}{\partial w_{ij}^{c_i}}$$

$$\Delta w_{ijl}^{ci} = - \in [\sum\nolimits_k \frac{\partial \Psi}{\partial \hat{\mathrm{R}}_k} \times \frac{\partial \hat{\mathrm{R}}_k}{\partial \mathfrak{h}_k} \times \frac{\partial \mathfrak{h}_k}{\partial \hat{\mathrm{R}}_j}] \times \frac{\partial \hat{\mathrm{R}}_j}{\partial \mathfrak{h}_j} \times \frac{\partial \mathfrak{h}_j}{\partial w_{ij}^{c_i}}$$

$$\Delta w_{ijl}^{c_i} =\in [\sum\nolimits_k (c_k - \hat{\mathrm{R}}_k) \times \hat{\mathrm{R}}_k(1 - \hat{\mathrm{R}}_k) \times v_{jkl}^{c_i}] \times \hat{\mathrm{R}}_k(1 - \hat{\mathrm{R}}_k)\beta_i$$

$$\Delta w_{ijl}^{c_i} =\in [\sum\nolimits_k (c_k - \hat{\mathrm{R}}_k) \times \hat{\mathrm{R}}_k(1 - \hat{\mathrm{R}}_k) \times v_{jkl}^{c_i}] \times Rj(1 - Rj)\beta_i$$

$$\Delta w_{ijl}^{c_i} =\in [\sum\nolimits_k \varphi_\kappa(v_{jkl}^{c_i})] \times \hat{\mathrm{R}}_j(1 - \hat{\mathrm{R}}_j)\beta_i$$

$$\Delta w_{ijl}^{c_i} =\in \varphi_j \beta_i \tag{10}$$

The mathematical form of final equation for input and hidden layer is given in equation 11.

$$w_{ijl}^{c_i} = w_{ijl}^{c_i} + \sigma_L(\Delta w_{jkl}^{c_i}) \tag{11}$$

The mathematical form of final equation for output and hidden layer is given in equation 12.

$$v_{ijl}^{c_i} = v_{ijl}^{c_i} + \sigma_L(\Delta v_{jkl}^{c_i}) \tag{12}$$

In equation 11 and equation 12 the $\boldsymbol{\sigma_L}$ represents the learning rate of proposed model.

F. AGGREGATION SERVER SIDE

Once the weights have been received on aggregation server, we need to preprocess the weights to avoid any conflict of dimension in the weights of contributors. There can be a conflict in the dimension of weights received from different contributors. First of all, we calculate the dimensions of weights using shape function. If the dimensions have conflict, then to avoid this conflict, we used zero padding technique. Suppose the highest dimension is (M x N) and the lowest dimension is (K x L) then to equalize dimensions, zeros would be added to all the low dimensional weights. To create the zero-padding matrix, we used zeros function from TensorFlow. Once the dimensions are scaled next step on aggregation server is to aggregate the weights.

To aggregate the weights from equation 11 and 12 on server side, equation 13 has been used, lets dissect this equation. $\boldsymbol{l_p}$ represents the datapoints used by a contributor, and l represents the size of overall training data, where $\frac{l_p}{l}$ is scaling factor and p = 1,2,3…P. The scaling factor is multiplied with weights of each contributor, where $\boldsymbol{\alpha}$ represents weights v and w. After scaling the weights of each contributor, we sum up the scaled weights.

$$\vartheta(\alpha_{ijl}^{c_i}) = \sum\nolimits_{p=1}^{P} \frac{l_p}{l} F(\alpha_{ijl}^{c_i}) \tag{13}$$

The basic work flow of the proposed model is shown in figure 2. This work flow diagram illustrates the working of each contributor and flow of data and parameters from initial stage to aggregation server. Each contributor in the system follows the same sequence and methodology.

III. SIMULATION RESULTS

To simulate the proposed model, we used Jupyter notebook, Python, Pandas, NumPy, Matplotlib, TensorFlow, Keras and scikit-learn (SKLearn). The system used for simulations has the specification of 16GB RAM, Intel Core i5-5300U CPU @ 2.30GHz, and 256GB solid-state drive (SSD).

Figure 3 shows global accuracy and global loss for 10 iterations and 3 contributors. It shows that global loss started from 0.08 and decreased gradually until it reached 0.05. Then the global loss drops further and touches 0.04. Later on, it keeps around 0.04. The global accuracy graph shows that initially, the model gave an accuracy of around 0.96, then it boosted to 0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it is near to 0.98.

Figure 4 shows global accuracy and global loss for 30 iterations and 3 contributors. It shows that global loss started from 0.08 and decreased gradually until it reached 0.05. Then the global loss drops further and touches 0.04. Later on, it keeps around 0.04. The global accuracy graph shows that initially, the model gave an accuracy of around 0.96, then it boosted to 0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it is around 0.98.

Figure 5 shows global accuracy and global loss for 50 iterations and 3 contributors. It shows that global loss started from 0.08 and decreased gradually until it reached 0.05. Then the global loss drops further and touches 0.03. But on average, it keeps around 0.04. The global accuracy graph shows that initially, the model gave an accuracy of around 0.96, then it boosted to 0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it is around 0.98.

Figure 6 shows global accuracy and global loss for 10 iterations and five contributors. It shows that global loss started from 0.08 and decreased gradually until it reached 0.055. Then the global loss drops further and touches 0.045. Later

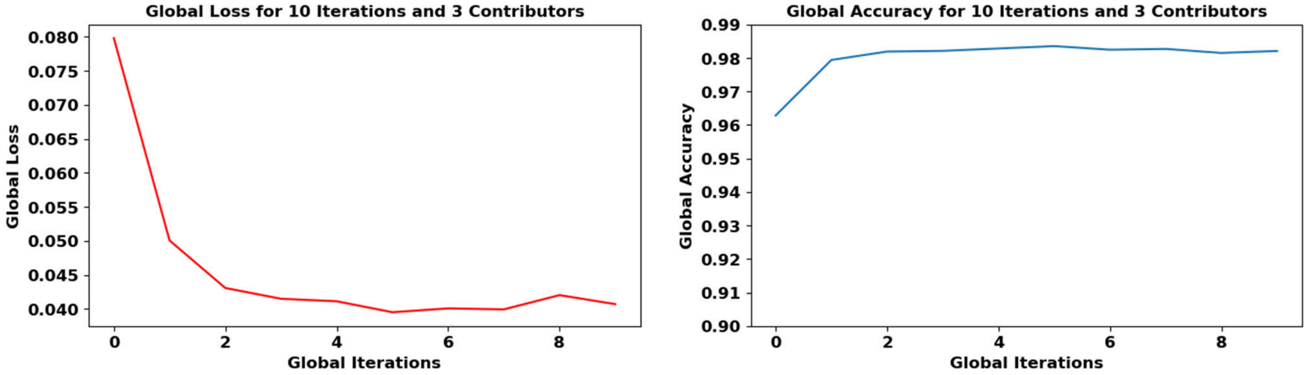

FIGURE 3. Global loss and accuracy, plot 1.

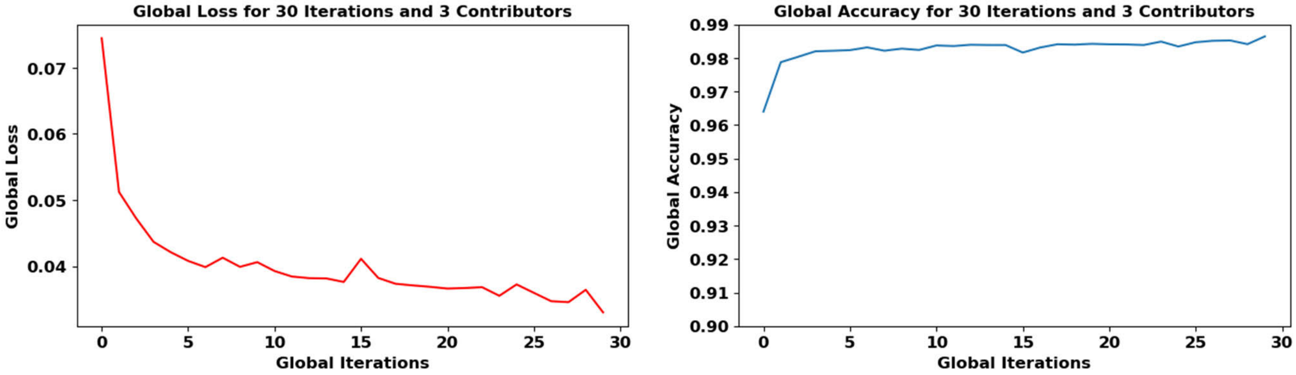

FIGURE 4. Global loss and accuracy, plot 2.

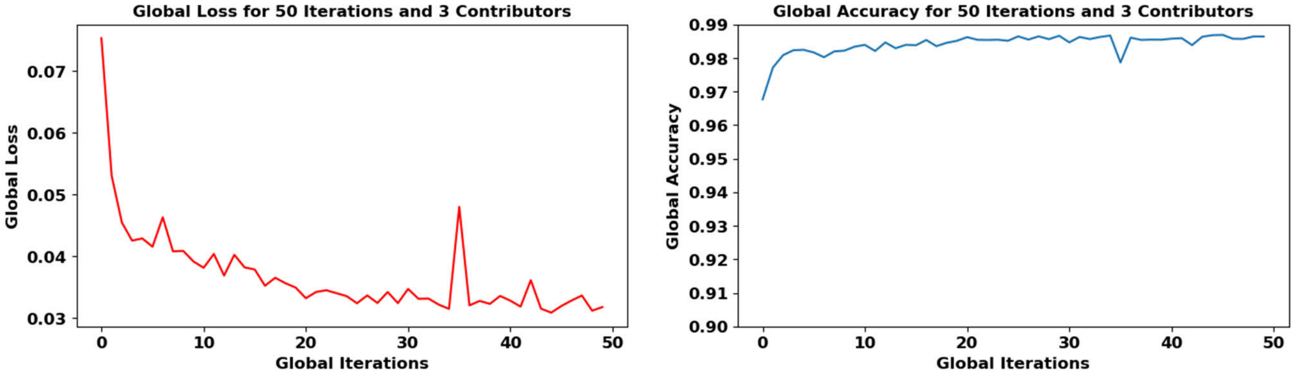

FIGURE 5. Global loss and accuracy, plot 3.

on, it keeps around 0.045. The global accuracy graph shows that initially, the model gave an accuracy of around 0.96, then it boosted to 0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it stuck to 0.98.

Figure 7 shows global accuracy and global loss for 30 iterations and 5 contributors. It shows that global loss started from 0.085 and decreased gradually until it reached 0.06. Then the global loss drops further and touches 0.045. Later on, it keeps around 0.04. The global accuracy graph shows that initially, the model gave an accuracy of around 0.96, then it boosted to 0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it stuck to 0.98.

Figure 8 shows global accuracy and global loss for 50 iterations and 5 contributors. It shows that global loss started from 0.083 and decreased gradually until it reached 0.04. Later on, it drops further to 0.035, but on average, it keeps around 0.04. The global accuracy graph shows that initially, the model gave an accuracy of around 0.96, then it boosted to

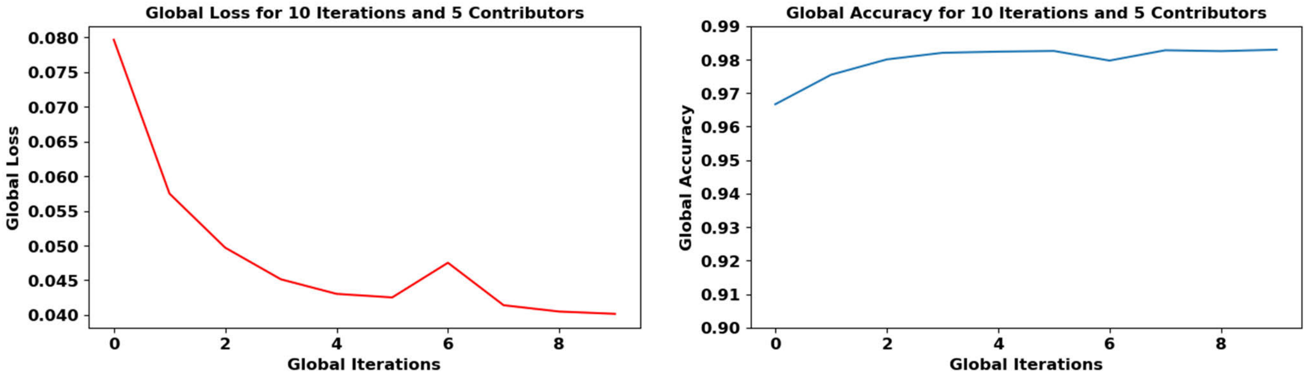

FIGURE 6. Global loss and accuracy, plot 4.

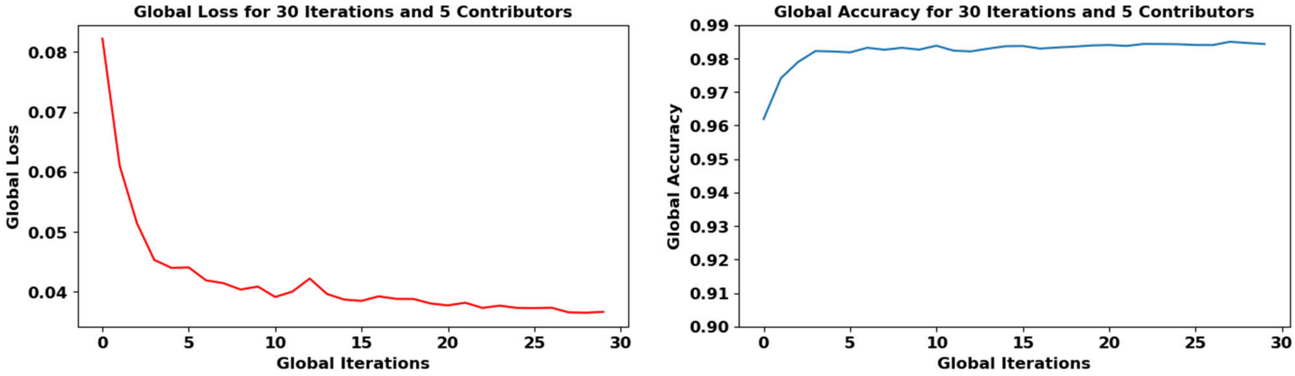

FIGURE 7. Global loss and accuracy, plot 5.

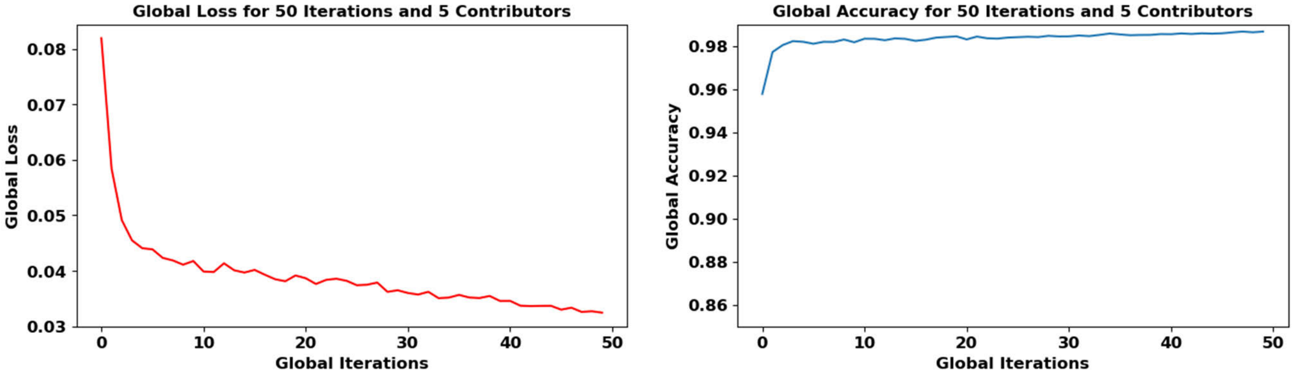

FIGURE 8. Global loss and accuracy, plot 6.

0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it stuck to 0.98.

Figure 9 shows global accuracy and global loss for 10 iterations and 10 contributors. It shows that global loss started from 0.09 and decreased gradually until it reached 0.05. Then the global loss drops further and touches 0.04. Later on, it keeps around 0.04. The global accuracy graph shows that initially, the model gave an accuracy of around 0.96, then it boosted to 0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it stuck to 0.98.

Figure 10 shows global accuracy and global loss for 30 iterations and 10 contributors. It shows that global loss started from 0.09 and decreased gradually until it reached 0.05. Then the global loss drops further and touches 0.04. Later on, it keeps around 0.04. The global accuracy graph shows that initially, the model gave an accuracy of around 0.96, then it

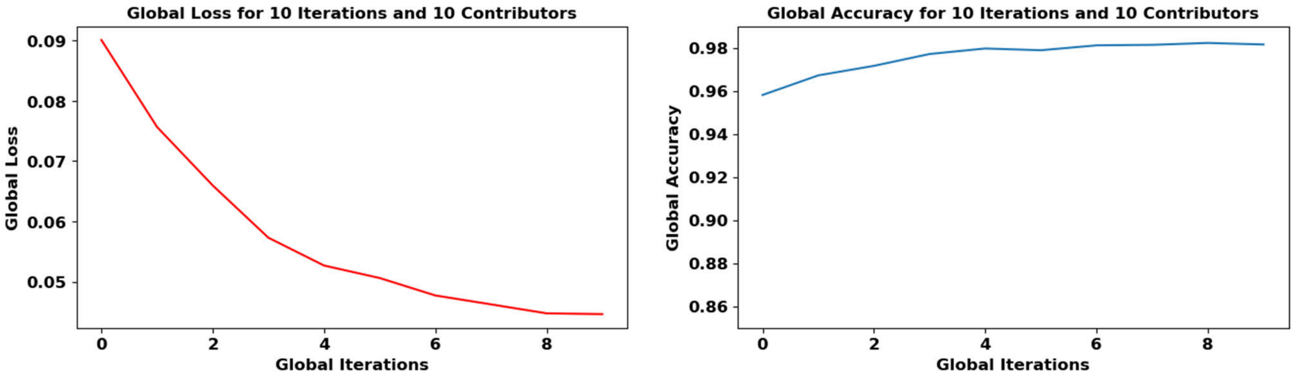

FIGURE 9. Global loss and accuracy, plot 7.

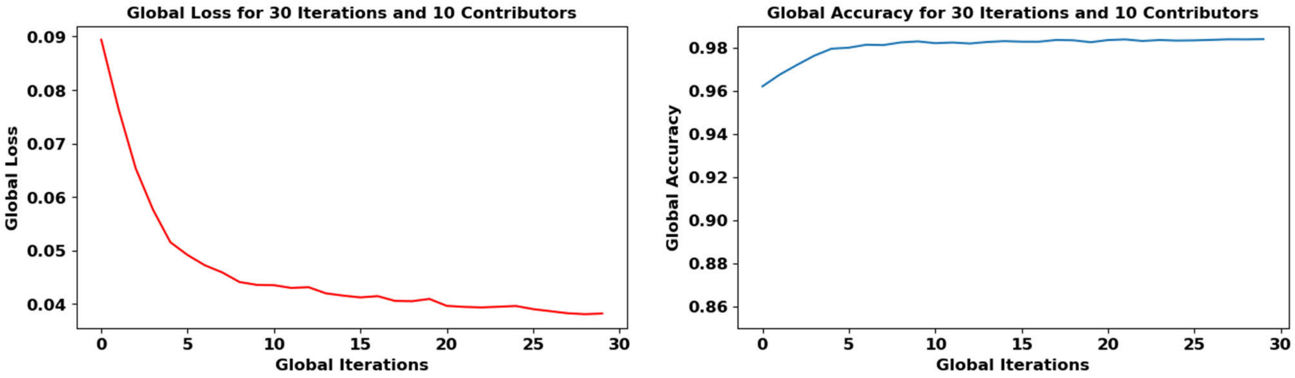

FIGURE 10. Global loss and accuracy, plot 8.

boosted to 0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it stuck to 0.98.

Figure 11 shows global accuracy and global loss for 50 iterations and 10 contributors. It shows that global loss started from 0.089 and decreased gradually until it reached 0.05. Then the global loss drops further and touches 0.04. Later on, it keeps around 0.04. The global accuracy graph shows that initially, the model gave an accuracy of around 0.96, then it boosted to 0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it maintained at 0.98.

Figure 12 shows global accuracy and global loss for 100 iterations and 10 contributors. It shows that global loss started from 0.094 and decreased gradually until it reached 0.05. Then the global loss drops further and touches 0.04. Later on, it keeps around 0.04. The global accuracy graph shows that initially, the model gave an accuracy of around 0.95, then it boosted to 0.98. Later on, it slightly increased above 0.98, but in most of the iterations, it maintained at 0.98.

Table 2 summarizes the achieved simulation results. It shows change in contributors to 3, 5, and 10. It also shows change in global iterations to 10, 30, 50, and 100. Simulation results show that lowest accuracy obtained is 98.16%, and the highest is 98.65%. The global loss keeps around 0.04. The lowest global loss obtained is 0.031, and the highest is 0.44. These simulation results show how effective the proposed model can be.

The evaluation of proposed model is investigated using several evaluation metrics including accuracy score, precision score, recall, and f1 score. The outcomes of these metrics help to estimate performance of model. These metrics give a score of around 0.98.6%, which meet the simulation results we discussed earlier. These results verifies that the performance and accuracy of the proposed model is excellent. Table 3 shows the score of evaluation metrics we used to evaluate the proposed model.

To prove the novelty and effectiveness of proposed model, we also performed the centralized IDS (CIDS) model training and evaluation using the Edge-IIoTset dataset. We trained the 5 contributors independently using their local datasets. The experimental results are given in Table 4. We found that there is a significant difference between the accuracy and performance of centralized IDS and FL based IDS. The accuracy and performance of centralized IDS is lower than FL based IDS. The highest accuracy achieved by an independent centralized contributor is 78.46% whereas the highest accuracy achieved by FL based IDS is 98.65%.

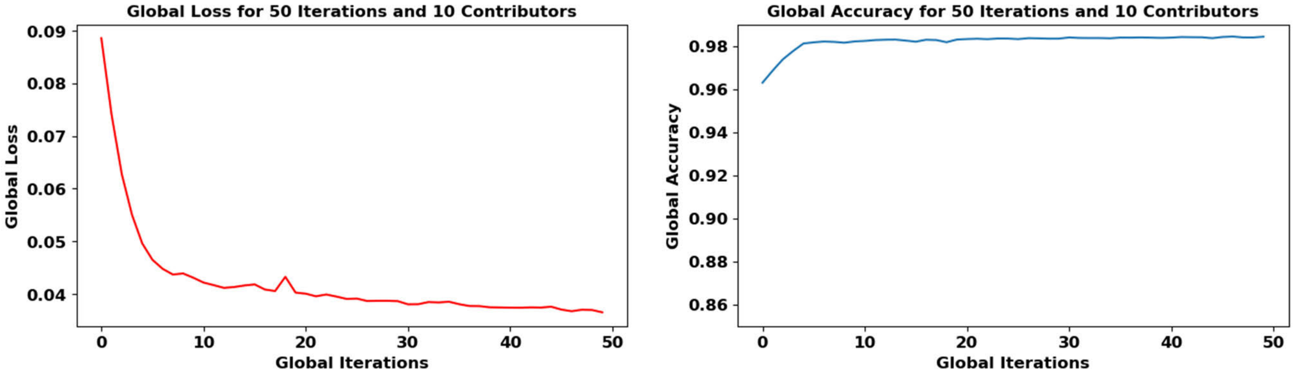

FIGURE 11. Global loss and accuracy, plot 9.

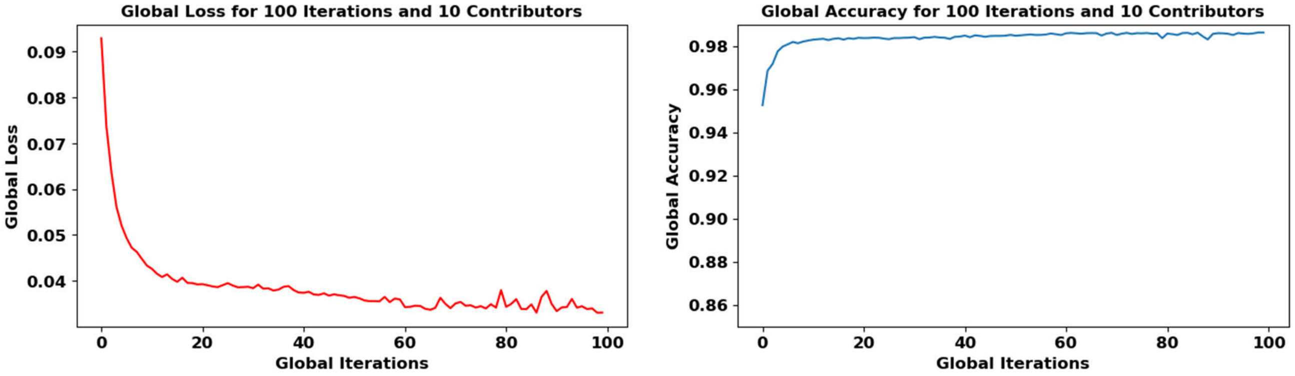

FIGURE 12. Global loss and accuracy, plot 10.

TABLE 3. Scores of evaluation metrics.

Sr. No.	Evaluation Metric	Score
1	Accuracy Score	0.9865
2	Precision Score (micro)	0.9863
3	Precision Score (weighted)	0.9864
4	Recall Score (micro)	0.9863
5	Recall Score (weighted)	0.9863
6	f1 Score (micro)	0.9863
7	f1 Score (weighted)	0.9864

TABLE 4. Results of centralized IDS.

Sr. No.	Centralized IDS	Accuracy Score
1	CIDS1	0.7845
2	CIDS2	0.7843
3	CIDS3	0.7846
4	CIDS4	0.7836
5	CIDS5	0.7744

IV. DISCUSSION

The proposed model can preserve data privacy of training data using FL. Further, the proposed model solves the problem of training data unavailability due to privacy concerns. We used ANN as a baseline model in proposed model. We tested the proposed model using the latest and most realistic cybersecurity dataset [14]. We evaluated the proposed model using seven evaluation metrics. These evaluation metrics helped to assess the proposed model in detail. The proposed model has been tested using different global iterations and contributors. According to the evaluation results of the proposed model, we obtained 98.65% accuracy.

In the literature review considered several research articles. Let's discuss their contributions. ElSayed et al. proposed a hybrid deep learning approach based on CNN to classify normal or attack classes [20] in SDN. They used precision, recall, and F1-score metrics to assess the model. Ahmed et al. proposed an intrusion detection algorithm named deep active learning [23]. They used F1-score metrics for evaluation. Ravi et al. proposed a deep learning approach for intrusion detection in SDN-IoT networks [21]. They used precision, recall, and F1- score metrics for evaluation. Phan and Bauschert proposed a reinforcement learning based solution for intrusion detection in SDN [18]. They proposed the solution to detect Dos attacks. Zhu et al. proposed a Resnet based solution for intrusion detection in SDN [17]. Yang et al. proposed autoencoder based intrusion detections system for

TABLE 5. Comparison results of proposed model with literature.

Studies	Machine Learning Techniques	Privacy Preservation	Accuracy	Detection Classes
Zhu et al. [17]	Resnet	No	93.52%	2
Phan et al. [18]	Q-Networks	No	85%	2
ElSayed et al. [20]	CNN	No	97%	2
Yang et al. [22]	Ensemble Autoencoder	No	97%	2
Ahmed et al. [23]	Active Deep Learning	No	96%	2
Xu et al. [51]	KNN	No	97.6%	2
Ahmad et al. [38]	FL	Yes	95.41%	6
Jie et al. [40]	FL	Yes	98%	5
Zakaria et al. [44]	FL	Yes	89%	5
Proposed Model	FL	Yes	98.65%	11

SDN [22]. They used unsupervised learning technique in their proposed model.

We discussed the findings and results of the proposed model and the results and techniques of earlier studies. Let's compare them with the proposed model. All of them used centralized machine learning techniques. We used FL that is a distributed learning technique. Further the proposed model obtained 98.65% accuracy whereas earlier models obtained accuracy around 85% to 97%. They used 3 to 4 evaluation metrics for evaluation. We used seven evaluation metrics for the evaluation of the proposed model. Their proposed models do not address the problem of data privacy preservation, and the problem of unavailability of training data due to privacy concerns. The proposed model addresses these problems. They used old datasets. None of them used the latest dataset we used. The cited research articles proposed IDS which can classify 1 or 2 classes only. The proposed model can classify 11 classes. This detailed comparison shows how the proposed model is better than earlier studies.

FL is a collaborative machine learning technique which uses weights or parameters for the training of global model [8], [47], [49], [50]. Every organization and individual being is concerned for the privacy of data. They do not willing to share their data specially when it comes to the data of network traffic. In FL different organizations can share the parameters of their model training instead of training data. IDS based on centralized machine learning techniques suffer from data privacy preservation and unavailability of training data due to privacy problem. FL provides a fascinating solution against these problems. So, in IDS for SDN, FL is more effective approach then centralized machine learning techniques.

Simulation results show that varying the number of contributors does not affect the accuracy and performance of proposed model. More global iterations can give more precise results but we find that 50 to 100 global iterations are enough although they can be varied depending upon the dataset. More global iterations require more computation power so the optimal number of global iterations play a crucial role in the training of proposed system.

A. IMPLEMENTATION CHALLENGES AND SOLUTIONS

SDN networks can involve a diverse range of devices with varying processing power, memory limitations, and communication bandwidths. This heterogeneity can impact the efficiency of the federated learning process. Carefully selecting the training devices with required computational power can optimize the learning process. FL based IDS involves exchanging model updates between devices and the central server. Excessive communication can strain network resources and increase latency. A highly efficient communication network can reduce the communication overhead and latency. Otherwise, we can Implement dynamic scheduling algorithms that consider factors like device load and communication bandwidth to optimize model update frequency.

B. PRACTICAL IMPLICATIONS AND APPLICATIONS

SDNs are widely used in telecommunications for network management, virtualization and configurations. The proposed IDS can be used to detect and prevent network intrusions in telecommunications networks. SDNs are used in cloud computing for network virtualization and resource allocation. The proposed IDS can be used to detect and prevent network intrusions in cloud computing environments. Networks implemented and used in IoT are prone to attacks. SDNs are used in IoT for network management. The proposed model can be used to detect and prevent network intrusions in IoT networks.

We have emphasized the importance of data privacy in SDNs and how FL can help preserve data privacy by enabling local data processing and training. The proposed IDS is designed to be scalable, making it suitable for large-scale SDNs. We have discussed how FL enables the system to learn from distributed data sources, reducing the need for centralized data storage and processing. FL reduces network latency by enabling local data processing and analysis, reducing the need for data transfer across the network and devices.

V. CONCLUSION

In this research article we have demonstrated the promising potential of FL in the context of intrusion detection within SDNs. We have validated the proposed model using a latest cyber security dataset and compared the proposed model with state of art existing studies. Achieving a remarkable accuracy rate of 98.65%, findings of this research highlight the effectiveness of collaborative machine learning approach in enhancing network security. However, achieving such high accuracy rates requires careful model selection, hyperparameter tuning and data processing. By distributing the training process across multiple network nodes while preserving data privacy, FL permits real-time threat detection and adaptation to evolving attacks. This not only enhances the overall security of SDNs but also minimizes the risk of data exposure and solves the problem of training data privacy concerns. The proposed model is a multi-class classification approach. As the landscape of cyber threats continues to evolve, this research work underscores the importance of leveraging cutting-edge technologies like FL. The proposed model is an innovative IDS to mitigate the flaws and challenges of traditional and centralized IDS.

VI. FUTURE DIRECTIONS

In the future, we will explore more datasets to include more important classes to the training data and to compare the performance of proposed model. This will increase the scope and effectiveness of proposed model and ensure the in-depth validation.

REFERENCES

[1] S. Lee, S. Woo, J. Kim, J. Nam, V. Yegneswaran, P. Porras, and S. Shin, "A framework for policy inconsistency detection in software-defined networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 3, pp. 1410–1423, Jun. 2022, doi: 10.1109/TNET.2022.3140824.

[2] K. K. Karmakar, V. Varadharajan, and U. Tupakula, "Mitigating attacks in software defined networks," *Cluster Comput.*, vol. 22, no. 4, pp. 1143–1157, Dec. 2019.

[3] M. Priyadarsini and P. Bera, "Software defined networking architecture, traffic management, security, and placement: A survey," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108047, doi: 10.1016/j.comnet.2021.108047.

[4] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, "Complementing IoT services through software defined networking and edge computing: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1761–1804, 3rd Quart., 2020.

[5] M. F. Hyder and M. A. Ismail, "Securing control and data planes from reconnaissance attacks using distributed shadow controllers, reactive and proactive approaches," *IEEE Access*, vol. 9, pp. 21881–21894, 2021.

[6] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: An overview from machine learning perspective," *J. Big Data*, vol. 7, no. 1, pp. 1–29, Dec. 2020.

[7] G. Apruzzese, P. Laskov, E. M. de Oca, W. Mallouli, L. B. Rapa, A. V. Grammatopoulos, and F. Di Franco, "The role of machine learning in cybersecurity," *Digit. Threats, Res. Pract.*, vol. 4, no. 1, pp. 1–38, 2022.

[8] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowl.-Based Syst.*, vol. 216, Jun. 2021, Art. no. 106775.

[9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[10] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[11] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3347–3366, Apr. 2023.

[12] J. Lansky, S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh, and A. M. Rahmani, "Deep learning-based intrusion detection systems: A systematic review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021.

[13] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, Dec. 2019.

[14] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.

[15] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proença, "Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment," *IEEE Access*, vol. 8, pp. 83765–83781, 2020.

[16] S. Haider, A. Akhunzada, I. Mustafa, T. B. Patel, A. Fernandez, K. R. Choo, and J. Iqbal, "A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 53972–53983, 2020.

[17] Z. Zhu, W. Zhai, H. Liu, J. Geng, M. Zhou, C. Ji, and G. Jia, "Juggler-ResNet: A flexible and high-speed ResNet optimization method for intrusion detection system in software-defined industrial networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4224–4233, Jun. 2022.

[18] T. V. Phan and T. Bauschert, "DeepAir: Deep reinforcement learning for adaptive intrusion response in software-defined networks," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 2207–2218, Sep. 2022, doi: 10.1109/TNSM.2022.3158468.

[19] K. S. Sahoo, B. K. Tripathy, K. Naik, S. Ramasubbareddy, B. Balusamy, M. Khari, and D. Burgos, "An evolutionary SVM model for DDOS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 132502–132513, 2020.

[20] M. S. ElSayed, N.-A. Le-Khac, M. A. Albahar, and A. Jurcut, "A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique," *J. Netw. Comput. Appl.*, vol. 191, Oct. 2021, Art. no. 103160.

[21] V. Ravi, R. Chaganti, and M. Alazab, "Deep learning feature fusion approach for an intrusion detection system in SDN-based IoT networks," *IEEE Internet Things Mag.*, vol. 5, no. 2, pp. 24–29, Jun. 2022.

[22] L. Yang, Y. Song, S. Gao, A. Hu, and B. Xiao, "Griffin: Real-time network intrusion detection system via ensemble of autoencoder in SDN," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 2269–2281, Sep. 2022.

[23] U. Ahmed, J. C. Lin, G. Srivastava, U. Yun, and A. K. Singh, "Deep active learning intrusion detection and load balancing in software-defined vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 953–961, Jan. 2023.

[24] S. Prathibha, J. Bino, Md. T. Ahammed, C. Das, S. R. Oion, S. Ghosh, and M. Afroj, "Detection methods for software defined networking intrusions (SDN)," in *Proc. Int. Conf. Adv. Comput., Commun. Appl. Informat. (ACCAI)*, Jan. 2022, pp. 1–6.

[25] Y. Cui, Q. Qian, C. Guo, G. Shen, Y. Tian, H. Xing, and L. Yan, "Towards DDoS detection mechanisms in software-defined networking," *J. Netw. Comput. Appl.*, vol. 190, Sep. 2021, Art. no. 103156.

[26] R. U. Rasool, U. Ashraf, K. Ahmed, H. Wang, W. Rafique, and Z. Anwar, "Cyberpulse: A machine learning based link flooding attack mitigation system for software defined networks," *IEEE Access*, vol. 7, pp. 34885–34899, 2019.

[27] I. Feki, S. Ammar, Y. Kessentini, and K. Muhammad, "Federated learning for COVID-19 screening from chest X-ray images," *Appl. Soft Comput.*, vol. 106, Jul. 2021, Art. no. 107330.

[28] J. Li, Y. Meng, L. Ma, S. Du, H. Zhu, Q. Pei, and X. Shen, "A federated learning based privacy-preserving smart healthcare system," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 2021–2031, Mar. 2022.

[29] B. Han, R. H. Jhaveri, H. Wang, D. Qiao, and J. Du, "Application of robust zero-watermarking scheme based on federated learning for securing the healthcare data," *IEEE J. Biomed. Health Informat.*, vol. 27, no. 2, pp. 804–813, Feb. 2023.

[30] J. Fan, X. Wang, Y. Guo, X. Hu, and B. Hu, "Federated learning driven secure Internet of Medical Things," *IEEE Wireless Commun.*, vol. 29, no. 2, pp. 68–75, Apr. 2022.

[31] K. Guo, T. Chen, S. Ren, N. Li, M. Hu, and J. Kang, ''Federated learning empowered real-time medical data processing method for smart healthcare,'' *IEEE/ACM Trans. Comput. Biol. Bioinf.*, early access, Jun. 2022.

[32] H. Elayan, M. Aloqaily, and M. Guizani, ''Sustainability of healthcare data analysis IoT-based systems using deep federated learning,'' *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7338–7346, May 2022.

[33] M. Driss, I. Almomani, Z. Huma, and J. Ahmad, ''A federated learning framework for cyberattack detection in vehicular sensor networks,'' *Complex Intell. Syst.*, vol. 8, no. 5, pp. 4221–4235, Oct. 2022.

[34] J. Li, X. Tong, J. Liu, and L. Cheng, ''An efficient federated learning system for network intrusion detection,'' *IEEE Syst. J.*, vol. 17, no. 2, pp. 2455–2464, Jan. 2023.

[35] X. Huang, J. Liu, Y. Lai, B. Mao, and H. Lyu, ''EEFED: Personalized federated learning of execution&evaluation dual network for CPS intrusion detection,'' *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 41–56, 2023.

[36] R. Zhao, Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, ''Semisupervised federated-learning-based intrusion detection method for Internet of Things,'' *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8645–8657, May 2023.

[37] H. Liang, D. Liu, X. Zeng, and C. Ye, ''An intrusion detection method for advanced metering infrastructure system based on federated learning,'' *J. Mod. Power Syst. Clean Energy*, vol. 11, no. 3, pp. 927–937, May 2023, doi: 10.35833/MPCE.2021.000279.

[38] 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,'' *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 3, pp. 2442–2459, Sep. 2023.

[39] H. Babbar and S. Rani, ''FRHIDS: Federated learning recommender hybrid intrusion detection system model in software-defined networking for consumer devices,'' *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2492–2499, Feb. 2024.

[40] J. Cui, H. Sun, H. Zhong, J. Zhang, L. Wei, I. Bolodurina, and D. He, ''Collaborative intrusion detection system for SDVN: A fairness federated deep learning approach,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 9, pp. 2512–2528, Sep. 2023.

[41] Z. A. E. Houda, H. Moudoud, and B. Brik, ''Federated deep reinforcement learning for efficient jamming attack mitigation in O-RAN,'' *IEEE Trans. Veh. Technol.*, early access, Jan. 2024.

[42] Z. A. E. Houda, H. Moudoud, B. Brik, and L. Khoukhi, ''Blockchain-enabled federated learning for enhanced collaborative intrusion detection in vehicular edge computing,'' *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 2024.

[43] Z. Abou El Houda, H. Moudoud, B. Brik, and L. Khoukhi, ''Securing federated learning through blockchain and explainable AI for robust intrusion detection in IoT networks,'' in *Proc. IEEE Conf. Comput. Commun. Workshops*, May 2023, pp. 1–6.

[44] 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,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 4, pp. 1985–2001, Jan. 2023.

[45] Z. A. E. Houda, B. Brik, A. Ksentini, L. Khoukhi, and M. Guizani, ''When federated learning meets game theory: A cooperative framework to secure IIoT applications on edge computing,'' *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7988–7997, Nov. 2022.

[46] S. M. S. Bukhari, M. H. Zafar, M. A. Houran, S. K. R. Moosavi, M. Mansoor, M. Muaaz, and F. Sanfilippo, ''Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-bi-LSTM for enhanced reliability,'' *Ad Hoc Netw.*, vol. 155, Mar. 2024, Art. no. 103407.

[47] A. Qayyum, K. Ahmad, M. A. Ahsan, A. Al-Fuqaha, and J. Qadir, ''Collaborative federated learning for healthcare: Multi-modal COVID-19 diagnosis at the edge,'' *IEEE Open J. Comput. Soc.*, vol. 3, pp. 172–184, 2022.

[48] T. Haibo, L. Maonan, and R. Shuangyin, ''ESE: Efficient security enhancement method for the secure aggregation protocol in federated learning,'' *Chin. J. Electron.*, vol. 32, no. 3, pp. 542–555, May 2023.

[49] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, ''When machine learning meets privacy: A survey and outlook,'' *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–36, Mar. 2022.

[50] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, ''Federated learning for healthcare informatics,'' *J. Healthcare Informat. Res.*, vol. 5, no. 1, pp. 1–19, Mar. 2021.

[51] Y. Xu, H. Sun, F. Xiang, and Z. Sun, ''Efficient DDoS detection based on K-FKNN in software defined networks,'' *IEEE Access*, vol. 7, pp. 160536–160545, 2019.

MUBASHAR RAZA received the B.Sc. degree in computer systems engineering from The Islamia University of Bahawalpur, Pakistan, in 2019, and the M.S. degree in computer science from Riphah International University, Lahore Campus, Pakistan, in 2023.

Since 2023, he has been a Lecturer in computing with Riphah International University, Sahiwal Campus, Pakistan. His research interests include machine learning, computer networks, network security, computer vision, natural language processing, robotics, and information security.

MUHAMMAD JASIM SAEED received the M.S. degree in computer science from Liverpool John Moores University, U.K., and the Ph.D. degree in computer communications and networks from Manchester Metropolitan University, U.K.

Currently, he holds the position of an Assistant Professor and the Head of the Department of Computing, Riphah International University, Lahore Campus, Pakistan.

MUHAMMAD BILAL RIAZ received the Ph.D. degree in applied science in Pakistan.

He is currently a Senior Researcher with the VSB—Technical University of Ostrava, Czech Republic. He has more than 200 publications in highly reputed journals. His research interests include computational mathematics, mathematics physics, fluid dynamics, differential equations, dynamical systems, chaos theory, fractional calculus, and its applications in sciences.

MUHAMMAD AWAIS SATTAR received the B.Sc. degree in electrical (computer) engineering from COMSATS University, Lahore, Pakistan, in 2014, the M.Sc. degree in electrical engineering from Rochester Institute of Technology, Dubai, United Arab Emirates, in 2017, and the Ph.D. degree in technical computer science and telecommunication from Lodz University of Technology, Łódź, Poland, in 2022.

He is currently an Assistant Professor of computing with Riphah International University, Lahore Campus, Pakistan. His research interests include machine learning, computer vision, and process tomography.