Hindawi
Journal of Electrical and Computer Engineering
Volume 2024, Article ID 5134326, 16 pages
https://doi.org/10.1155/2024/5134326

Research Article

Balancing Data Privacy and 5G VNFs Security Monitoring: Federated Learning with CNN + BiLSTM + LSTM Model

Abdoul-Aziz Maiga,[1] **Edwin Ataro**,[2] **and Stanley Githinji**[3]

[1]*Pan African University, Institute for Basic Sciences Technology and Innovation (PAUSTI), Nairobi, Kenya*
[2]*Department of Electrical and Electronic Engineering, Technical University of Kenya, Nairobi, Kenya*
[3]*Department of Computing, United States International University-Africa (USIU-A), Nairobi, Kenya*

Correspondence should be addressed to Abdoul-Aziz Maiga; abdoul.maiga@students.jkuat.ac.ke

Received 28 October 2023; Revised 11 February 2024; Accepted 27 February 2024; Published 30 March 2024

Academic Editor: Serena Nicolazzo

The cloudification of telecommunication network functions with 5G is a novelty that offers higher performance than that of previous generations. However, these virtual network functions (VNFs) are exposed to internet threats when hosted in the cloud, resulting in new security challenges. Another fact is that many VNFs vendors with different security policies will be implied in 5G deployment, creating a heterogeneous 5G network. The authorities also require data privacy enhancement in 5G deployment and there is the fact that mobile operators need to inspect data for malicious traffic detection. In this situation, how can network traffic inspections be conducted effectively without infringing on data privacy? This study addresses this gap by proposing a novel state-of-the-art hybrid deep neural network that combines a convolutional neural network (CNN) stacked to bidirectional long short-term memory (BiLSTM) and unidirectional long short-term memory (LSTM) for the deep inspection of network flow for malicious traffic detection. The approach utilizes federated learning (FL) to facilitate multiple VNFs vendors to collaboratively train the proposed model without sharing VNFs' raw data, which can mitigate the risk of data privacy violation. The proposed framework incorporates transport layer security (TLS) encryption to prevent data tempering or man-in-the-middle attacks between VNFs. The framework was validated through simulation using open-access benchmark datasets (InSDN and CICIDS2017). They achieved 99.99% and 99.58% accuracy and 0.048% and 0.617% false-positive rates for the InSDN and CICIDS2017 datasets, respectively, for FL. This study demonstrates the potential of hybrid deep learning-based FL for heterogeneous 5G network VNFs security monitoring.

1. Introduction

Fifth-generation (5G) mobile network, also known as IMT-2020 [1], will have a significant impact on numerous fields and societies in the future [2]. It provides better user experience and flexibility in different scenarios with different Quality of Service (QoS) requirements. To achieve this, 5G integrates the concept of replacing hardware-based network functions with cloud (or software)-based network functions. Because network functions are no longer physical, they are called virtual network functions (VNFs). The concept of network function virtualization (NFV) [3] enables many new services such as massive Internet of Things (mIoT), virtual reality (VR), augmented reality (AR), telemedicine, autonomous vehicles (AVs), and many other services that require low-latency communication [4]. However, VNFs hosted in the cloud are susceptible to a range of security threats, while integrating numerous VNFs from various vendors with different security measures is necessary to meet 5G requirements [5]. This creates a complex and diverse network, resulting in new security challenges [6], as well as exacerbating existing security vulnerabilities.

Many critical attacks can target 5G VNFs from the Internet as follows [7]: an attacker who gains unauthorized access to a VNF, such as a load balancer, can reconfigure it to bypass an intrusion prevention system (IPS) and launch a Denial of Service (DoS) attack against a web service, thus making it inaccessible. An adversary can use an unexpected

VNF or VNF-in-the-middle technique to connect an unauthorized VNF to an active network that can eavesdrop on network traffic. Another type of attack is a flow classification anomaly, which consists of an attacker modifying the classification rules of VNFs, making end services inaccessible to users or authorizing malicious traffic. DDoS attacks, hyperjacking, exploitation of known vulnerabilities in open-source software, insecure interfaces, and malicious application programming interfaces (APIs) also target 5G VNFs [8]. Another type of attack is the side-channel attack [9].

Data privacy challenges and all potential risks in 5G networks [5, 10] have motivated authorities to emphasize the importance of data privacy, which must be strengthened. For mobile operators to develop machine learning (ML)-based security monitoring systems, they must consider the requirements of authorities for data privacy preservation. Another important factor to consider is the heterogeneity of 5G network VNFs provided by vendors with different technologies and security policies. Traditional state-of-the-art ML solutions are limited to sustainable application solutions for 5G network scenarios.

In this study, these critical challenges are addressed by proposing an all-in-one security framework tailored to 5G network VNFs security monitoring. It integrates federated learning (FL) with a customized state-of-the-art hybrid deep neural network model composed of a customized single-layer convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM) layer, and unidirectional long short-term memory (LSTM) layer (CNN + BiLSTM + LSTM). The FL allows different VNFs vendors to train the model without the need to share raw data and prevent data privacy violation. The model is designed to capture complex temporal dependencies in network traffic for the accurate detection of known and unknown malicious traffic. FL requires VNFs to train the model locally and share only the trained model hyperparameters with a server for aggregation. To ensure the safe transmission of the hyperparameters, the transport layer security (TLS) protocol is integrated. The overall framework has the advantage of providing an efficient malicious traffic detection model with a secure FL for data privacy preservation and trust collaboration among VNFs vendors.

The main contributions of this study to the field are as follows:

(1) A federated learning-based security monitoring system is proposed for 5G virtual network functions (VNFs) to improve privacy preservation and facilitate collaboration among VNF vendors to build strong security monitoring systems.

(2) A state-of-the-art customized hybrid deep neural network model (CNN + BiLSTM + LSTM) capable of capturing valuable network traffic features for the effective detection of known and unknown attacks is proposed.

(3) TLS 1.3 Encryption is implemented to guarantee the integrity of the data transmitted between the clients and the server during the federated learning process. This is very important in the 5G VNFs security application scenario, where the VNFs can be hosted in different clouds.

(4) The proposed model's performances are compared with the literature and presented.

(5) Finally, an architecture integrating the proposed system and the 3rd Generation Partnership Project (3GPP) 5G network architecture is suggested.

The remaining sections are organized as follows. Section 2 outlines the related work. Section 3 outlines the study's methodology. Section 4 describes the datasets and simulation process used to evaluate the proposed model. In Section 5, the results are discussed. Section 6 presents the implementation solution for the proposed system in the 3GPP 5G network architecture. Finally, the conclusion and future works are presented in Section 7.

2. Previous Works

Federated learning is a recent concept used by researchers for cyber security and is yet to be explored, particularly for its application in 5G virtual network function security. This section discusses previous studies on federated learning domains of applications, deep learning for security monitoring, and federated learning for security monitoring in 5G networks. In our context, "security monitoring" is limited to the detection of attacks through network traffic inspection.

2.1. Federated Learning Domains of Application. Jithish et al. [11] recently discussed the use of FL in many domains. The authors summarized the use of FL in smart homes, healthcare, electric vehicles, image processing, and smart grids. Subramanya and Riggio [12] used federated learning for VNF autoscaling in 5G networks. This concept has been specifically applied to multidomain 5G networks. Qu et al. [13] proposed in their paper a survivable service function chain (SFC) deployment method using federated learning. It has also been applied to multidomain networks. Sivalingam et al. [14] evaluated the application of deep learning with federated learning by reporting LSTM and gated recurrent unit (GRU) model utilization. Tam et al. [15] applied federated learning to massive IoT communication (mIOT) with deep reinforcement learning (DRL) to increase efficiency. To predict the resources required for VNFs during their migration in a network, Tang et al. [16] proposed FedBi-GRU, which is a combination of federated learning and bidirectional GRU. Gupta et al. [17] used federated learning to detect anomalies associated with various diseases in smart healthcare. In the field of telecommunications, Niknam et al. [18] discussed the use of federated learning for wireless communication, particularly in the 5G context. The authors discussed its applicability to edge computing and caching, spectrum management, and 5G core networks. The authors of [19–21] used federated learning to detect malware and anomalies in Internet of Things (IoT) networks and devices. As is evident, federated learning has been utilized in numerous areas; however, its potential for future telecommunication network security, such as 5G, is yet to be explored.

2.2. Deep Learning for Security Monitoring. This section examines the use of deep learning in security monitoring. A survey conducted by Xin et al. [22] discussed some of the key detailed studies on deep learning (DL) methods used for intrusion detection. The evaluated models were CNN, recurrent neural network (RNN), BiLSTM, and LSTM. The reported accuracies ranged from 79% to 99% based on the model. Abdulqadder et al. [23] proposed a security framework as a VNF that uses hybrid fuzzy logic with an artificial neural network (HF-ANN) for network flow packet classification as normal or malicious. Normal packets are allowed to access applications, whereas malicious packets are dropped from the VNF. The authors of [24] proposed a CNN-based hybrid deep learning model for intrusion detection in a software-defined network (SDN). The authors claimed that their model achieved accuracies of 99.28% for binary classification and 98.92% for multiclass classification. Another intrusion detection system for SDN was proposed by Assis et al. [25] using the GRU deep learning method. They reported the model to be promising through simulation using CICIDDoS-2019 and the CICIDS-2018 datasets. In the field of the Internet of Medical Things, Manimurugan et al. [26] proposed a deep belief network (DBN) algorithm model for intrusion detection. The reported accuracy of the model was 99.37%. Yao et al. [27] proposed a bidirectional generative adversarial network (BiGAN) to detect intrusions in IoT. The UNSW-NB15 and CIC-IDS2017 datasets were used to evaluate the proposed model. It achieved a 4% accuracy increase compared to the literature and a 4% false-alarm rate reduction while maintaining computational efficiency. In [28], the authors discussed the use of RNN, CNN, generative adversarial networks (GANs), and transformers for anomaly detection from log messages. Ferrag et al. [29] compared RNN, CNN, restricted Boltzmann machines (RBMs), DBNs, deep neural networks (DNNs), and deep autoencoders in terms of their performance in intrusion detection datasets. Their accuracies ranged from 97% to 98% based on customized hyperparameters. A hybrid deep learning model, DCNNBiLSTM, built through a combination of CNN and BiLSTM, was proposed by Hnamte and Hussain [30] for intrusion detection in the IoT. Their model achieved accuracies of 100% and 99.64% using the CICIDS2018 and Edge IIoT datasets, respectively. Ilango et al. [31] proposed a feedforward-convolutional neural network (FFCNN) for low-rate DDoS attacks in IoT networks. In [32], Yadav et al. proposed an autoencoder to detect attacks in IoT with a 5G network. Their model achieved an accuracy of 99.76%. The application of DL to intrusion detection has been demonstrated to be effective in various domains. This potential can be exploited to enhance intrusion detection in 5G VNFs. The decision to use deep learning combined with federated learning in this study was based on its good performance in malicious traffic detection, as can be observed in the literature.

2.3. Federated Learning for Security Monitoring in 5G Networks. According to our research, this is the first study to propose secured hybrid deep learning-based federated learning for the security monitoring of 5G VNFs [33, 34]. Most studies have focused on the security of IoT and its related applications or on the security of 5G network architecture layers.

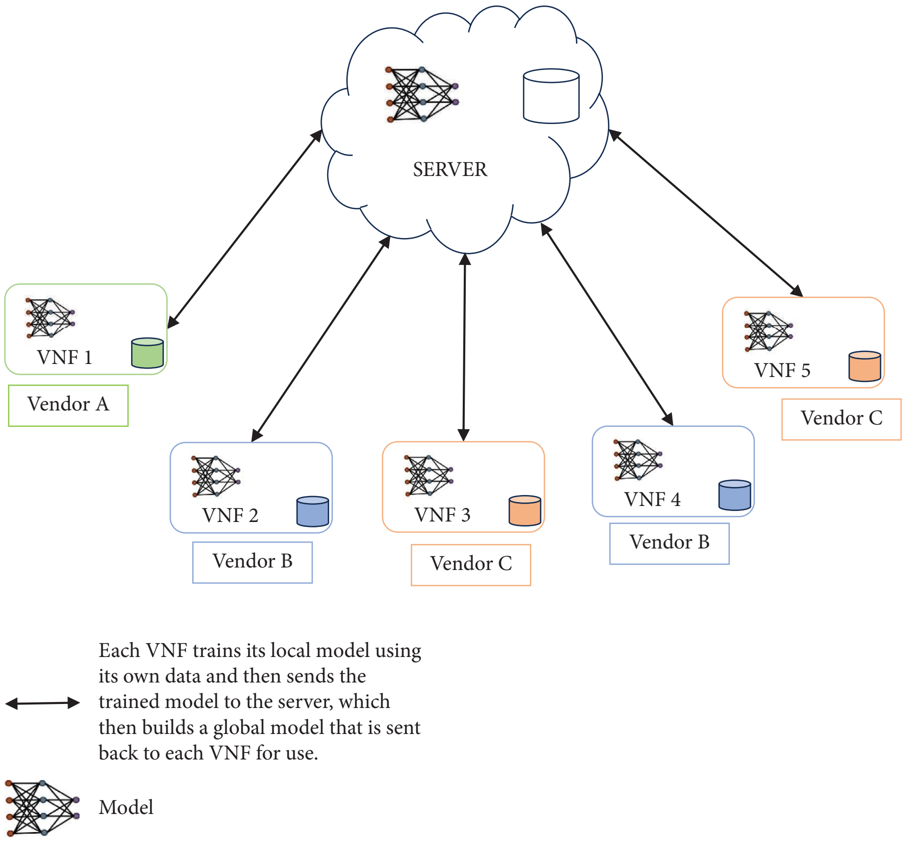
In this section, only the most related studies that used federated learning are discussed. Several researchers have investigated the use of federated learning for security monitoring in 5G networks. Bandara et al. [35] used blockchain with federated learning to detect attacks on 5G/6G networks. The authors considered a scenario with IoT device attacks in 5G/6G networks and reported their solution to be efficient. Boualouache and Engel [36] introduced multilayer perceptron (MLP)-based federated learning to detect passive mobile attackers in 5G vehicular edge computing. The simulation yielded a maximum accuracy of 95%. Fan et al. [37] proposed an IoT defender to protect 5G IoT against intrusion using federated transfer learning. The proposed model achieved an accuracy of 91.93%. Kholidy and Kamaludeen [38] used a Hashgraph-based federated learning approach (HFLA) to protect 5G networks from poisoning and membership inheritance attacks. The authors claimed that their model was superior to existing federated learning approaches. Jayasinghe et al. [39] adopted an ANN-based federated learning to secure 5G networks. The accuracy of the proposed model was 93.6%. For intrusion detection in 5G smart grids, Sun et al. [40] designed a neural network that utilized a transformer and hierarchical federated learning. The authors reported an accuracy of 99.48%. Belenguer et al. [41] proposed the GowFed to detect threats in industrial-level networks. This is a combination of federated learning and Gower dissimilarity matrices. A median accuracy of 95.5% was reported. Based on the literature reviewed thus far, the absence of federated learning applied to 5G VNFs security monitoring can be observed, making this study a new contribution to the literature.

3. Methodology

This section describes the techniques used to conduct experiments for evaluate and validate the proposed framework.

3.1. Federated Learning. Federated learning (FL) is used to overcome the weaknesses of traditional centralized machine learning methods regarding the preservation of data privacy and mitigation of computational costs during training. FL allows multiple devices (VNFs) to train a shared model without sharing raw local data. By employing federated learning, diverse VNFs belonging to different vendors can effectively contribute with their local data insights for training an improved global model while safeguarding the confidentiality and autonomy of their data sources. Figure 1 shows the FL architecture approach for heterogeneous 5G networks.

In this architecture, we consider a network with VNFs from three different vendors deployed in the same operator network: vendor A (VNF 1), vendor B (VNF 2 and VNF 4), and vendor C (VNF 3 and VNF 5). Only five VNFs were used as examples in this architecture. In real-world scenarios,



Each VNF trains its local model using its own data and then sends the trained model to the server, which then builds a global model that is sent back to each VNF for use.

Figure 1: Federated learning overview in the heterogeneous architecture.

there are many more network functions. The security policy of one vendor can differ from that of another and collaboration for sharing data cannot be guaranteed. Instead of sending raw data to the server to train the deep learning model, each VNF from each vendor trains the model using local data and sends only the training parameters to the server. The server aggregates all the VNFs models' parameters to build a single global model that will be distributed to each client for use. Finally, each VNF benefits from other VNFs models without the idea of raw data. Therefore, the risk of data privacy violations can be mitigated. The integrity of the data sent between entities is preserved by adding a TLS security layer (discussed later) for data encryption between the server and clients. Algorithm 1 describes the FL process secured by transport-layer security (TLS) protocol.

3.2. Proposed Model: CNN + BiLSTM + LSTM. The model proposed in this study is a customized hybrid deep neural network built from well-known neural network models from the literature. It is a combination of CNN, BiLSTM, and LSTM layers. Each layer parameters were carefully selected to obtain a high-performance model. To avoid repetition, the mathematical equations of the models are not described in this paper; readers will be redirected to previous studies that used the same models. A CNN is a type of neural network that uses filters to extract features from input data. Mostly used for image processing, it can also be used for processing time-series data. Readers can refer to previous articles [42–44] to gain a more comprehensive understanding of CNN and the mathematics behind them. BiLSTM is a type of RNN built from the following two LSTM layers: one that processes the sequence in the forward direction and the other in the backward direction. The architecture of BiLSTM is described in [30]. For LSTM, readers can refer to [45] for details.

The combination of a convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM), and long short-term memory (LSTM) is strategically chosen to synergistically capture spatial and temporal patterns in network data for deeper inspection of network traffic. The CNN component excels at spatial feature extraction, discerning local patterns indicative of malicious activity. Meanwhile, the BiLSTM and LSTM layers focus on learning sequential dependencies, leveraging bidirectionality to comprehend both past and future contexts and LSTM's ability to capture long-term dependencies. This comprehensive approach enables the model to robustly recognize complex hierarchical representations, combining the strengths of each architecture for enhanced accuracy in distinguishing normal traffic from malicious traffic. The CNN additionally contributes in reducing parameters count, mitigating overfitting concerns, particularly beneficial when dealing with limited labeled intrusion data.

```
Input: number of rounds $T$, number of local epochs $E$, and number of participating clients $N$
Output: final global model $W$ after $T$ rounds of federated learning
Initialize: Global model $W_0$;
for $t = 1$ to $T$ do
    for $i = 1$ to $N$ do
        Establish a secure TLS connection between the client and the server;
        Retrieve local data $X_i$ and associated labels $Y_i$
        Initialize: Local model parameters $W_i = W_t$;
        for $e = 1$ to $E$ do
            Update $W_i$ using $X_i$ and $Y_i$ through local training;
        end
        Securely send the updated local model $W_i$ to the server using the established TLS connection;
    end
    Aggregate and update the global model $W_{t+1}$ using the received local models;
    Send the updated global model $W_{t+1}$ to all participating clients using the TLS connection;
end
```

Algorithm 1: Federated averaging learning procedure with TLS encryption.

By combining these three algorithms, we were able to build an efficient customized hybrid model that performed better than most existing models in the literature. The overall architecture of the model is shown in Figure 2.

In the following, the characteristics of the model are described layer by layer.

(1) The first layer is a time-distributed one-dimensional convolution (Conv1D) layer. The number of filters was set to thirty two (32), and the kernel size parameter, which defines the size of the filters, was set to one (1). The input shape is specified as (1, 35, 1), indicating that the input data have a sequence length of 1, 35 features, and one channel. The activation function “softmax” is used in the layer.

(2) Time-distributed maxpooling and flattened layers were added to the previous layer. The first performs downsampling, and the second reshapes the input tensor into a one-dimensional vector, which is required before passing it to the next layer.

(3) The BiLSTM layer comes next, with thirty-two (32) units or memory cells, and uses softmax as the activation function, similar to the first layer.

(4) The next layer is the dropout layer. It takes 0.1 as a parameter, which indicates a dropout rate of 10%, helping to regularize the network and prevent overfitting.

(5) Another LSTM layer was added, but this time, it was a unidirectional LSTM layer. The number of units (memory cells) was set to 16, with softmax as the activation function.

(6) A dropout layer is added and takes 0.1 as a parameter.

(7) The output layer is a dense layer that is fully connected to a previously defined layer. It uses one neuron as the number of units and a hyperbolic tangent function (tan$h$) as the activation function. The output is a binary value: one (1) for malicious traffic and zero (0) for benign traffic.

3.3. TLS Version 1.3 for More Security. Although federated learning protects clients’ local data privacy, the training parameters exchanged between the clients and server can be vulnerable to numerous attacks and compromised. The TLS protocol, recommended by 3GPP [46] for communication security between 5G VNFs, can guarantee the integrity of the data between VNFs during the training process. The current TLS version (version 1.3) was used in the simulation to encrypt the proposed model parameters sent between the VNFs and the server. To achieve this, a self-signed certification generation method was used. The generation of certificates and keys was possible using the Openssl (version 1.1.1t) line command tool. The details of this process are as follows.

Step 1: an RSA private key of size 4096 bits is generated and stored in a file named “ca.key”.

Step 2: the key generated in step 1 is used to create a self-signed certificate and stored in a file named “ca.crt.” It is used as a root certificate authority (CA) for signing other certificates or for other purposes that require a trusted certificate. This is used by the client for authentication with the server.

Step 3: a server-side private key, with a size of 2048 bits, is generated and stored in a file named “server.key.” This server key is essential for establishing secure connections, decrypting data received from clients and creating digital signatures for authentication and data integrity.

Step 4: the previously generated server private key is used to create a certificate signing request (CSR), which is stored in the file name “server.csr.”

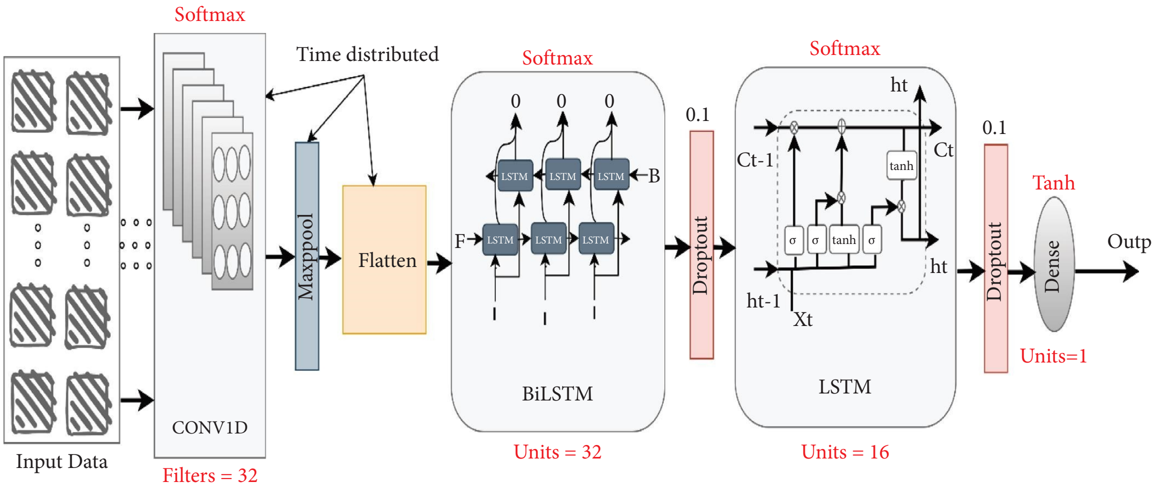


Figure 2: Proposed model (CNN + BiLSTM + LSTM) architecture.

Step 5: CSR “server.csr” is signed with the previously generated self-signed CA certificate “ca.crt” and its corresponding private key “ca.key”. The signed certificate is saved in the file “server.pem”.

After generating all necessary files in the simulation environment, the clients were configured to use the “ca.crt” file, and the server is configured to use the “ca.crt”, “server.pem”, and “server.key” files for authentication and encryption key exchange. All communications between the server and VNFs were encrypted during training. This prevents the data from being tempered by a Man-In-The-Middle (MITM) attack, thereby maintaining the integrity of the training hyperparameters.

4. Performance Evaluation

4.1. Datasets Overview and Preprocessing. The benchmark datasets used to evaluate the proposed model were InSDN [47] and CICIDS2017 [48]. These are among the most recent publicly available datasets used by researchers to evaluate the performance of intrusion detection systems.

4.1.1. Datasets Overview. The InSDN dataset, published in 2020, was developed specifically for a software-defined network (SDN) scenario, which makes it applicable to 5G VNFs in this study. SDN is a 5G-enabled technology block responsible for decoupling the control plan from the user plan for more efficiency in the network Quality of Service (QoS) flow [49]. The attacks in the dataset were classified into the following four vectors: attacks on the data plane, attacks on control-plane communication, attacks on the SDN controller, and attacks on the application plane. The total number of dataset instances was 343,889 for normal and attack traffic after labeling. Further details are presented in Table 1. The fully labeled dataset was preprocessed and used for model performance evaluation. The CICIDS2017 dataset is older than InSDN but closer to it in terms of features. It covers a comprehensive range of attack scenarios that have not been addressed in previous datasets. Used by many researchers to evaluate intrusion detection systems, this is the second choice for evaluating the proposed model. The initial dataset contains 2830743 samples with 79 features; however, it integrated many missing entries and redundant samples that needed to be cleared. The Benign samples (2273097) were very large compared to the attack samples. We retained all attack samples but randomly undersampled benign samples to avoid normal entry bias during the training process. Table 2 presents more details of the final dataset distribution before normalization and feature selection.

4.1.2. Normalization and Features Selection. Data normalization and feature selection were among the most important preprocessing tasks before proceeding with the experiments. The model supports only numerical values, whereas some features are object types that need to be standardized and normalized. In the following section, the normalization and feature selection processes are explained step by step.

Step 1: all features of type object are categorically encoded using the label encoder function of scikit-learn, except for the label feature, which is binary encoded by setting zero (0) to benign and one (1) to the others. Subsequently, all int64 values were converted to int32, and the other values were converted to float32.

Step 2: all redundant rows and rows containing missing or infinite values are removed.

Step 3: the correlation values for each feature were calculated, and those with a high correlation value greater than 85% were identified. Subsequently, all highly correlated features were excluded. This process aims to select only the relevant features for anomaly detection.

Step 4: *Z*-score normalization was performed for all feature values. It is a measure of the deviation of a particular data point from the mean relative to the variability (standard deviation) of the data. The formula for calculating the *Z*-score of a data point is given in equation (6).

Table 1: InSDN dataset distribution.

Dataset	Classes	Instances	Percentage (%)
InSDN	Normal	68,424	19.89711797
	Probe	98,129	28.53507963
	DDoS	73,529	21.3816086
	DoS	53,616	15.59107735
	DDoS*	48,413	14.07808915
	BFA	1,405	0.408562065
	Web attack	192	0.055831969
	BOTNET	164	0.047689807
	U2R	17	0.004943456

Table 2: CICIDS2017 dataset distribution.

Dataset	Classes	Instances	Percentage (%)
CICIDS2017	BENIGN	425,875	50.0000
	DoS Hulk	172,846	20.2905
	DDoS	128,016	15.0295
	PortScan	90,819	10.6631
	DoS goldeneye	10,286	1.2108
	FTP-Patator	5,933	0.6973
	DoS slowloris	5,385	0.6326
	DoS Slowhttptest	5,228	0.6149
	SSH-Patator	3,219	0.3787
	Bot	1,953	0.2296
	Web attack: brute force	1,47	0.1726
	Web attack: XSS	652	0.0766
	Infiltration	36	0.0042
	Web attack: SQL injection	21	0.0025
	Heartbleed	11	0.0013

$$Z = \frac{(X - \mu)}{\sigma}, \qquad (1)$$

where Z is the Z-score, X is the data point, μ is the mean of the data, and σ is the standard deviation of the data.

Step 5: after step 4, all the "Not a Number" (NaN) values are replaced with the mean value.

The abovementioned steps were implemented on both datasets, as required. Table 3 lists the datasets employed in the simulations after normalization and feature selection.

4.2. Simulation Process. The simulation was conducted using a Windows 11 system with an AMD Ryzen 7 4800H processor and a maximum of 4Ghz. It has 24GB of RAM and 512GB SSD of local storage, integrates a Radeon graphics card of 6GB, and NVIDIA GeForce RTX 2060 GPU. For comparison, the experiments were performed in two steps. The first step was to evaluate the performance of the CNN-BiLSTM-LSTM model using a centralized learning method, and the second step was to evaluate the model using federated learning. The Flower framework [50] combined with TensorFlow was used for the federated learning setup. Only TensorFlow is necessary for centralized learning. Python was used as the programming language. The Flower federated framework is used to connect the simulation VNFs and the server to be able to proceed with federated learning. TensorFlow was used to implement the CNN-BiLSTM-LSTM model on each VNF and load the local dataset onto the VNF. NVIDIA compute unified device architecture (CUDA) was used for the training with a GPU. CUDA serves as a parallel computing platform and programming model built by NVIDIA specifically for general-purpose computing on GPUs (graphics processing units). CUDA can be used to accelerate compute-intensive workloads on NVIDIA GPUs, extending its utility beyond graphics processing. CUDA support is integrated into widely used deep learning frameworks like TensorFlow and PyTorch, enhancing their capabilities for GPU acceleration.

The experiments were conducted using, respectively, three (3) clients, six (6) clients, nine (9) clients, and twelve (12) clients. The federated average (FedAVG) was used to aggregate the clients' local models to build a better global model. Before training the model, each dataset was divided into training (70%) and testing (30%) subsets. Each training set was then divided based on the number of clients to represent the local VNF data. Ten percent (10%) of the local data were used to validate the local model. The testing set was used to evaluate the performance of the global model on data that were not exposed during training and validation. For centralized learning, in contrast to federated learning, only one device was used to train the model, with 70% of the dataset used for training and 30% for testing. The optimal hyperparameters were determined and are listed in Tables 4 and 5. Customed hyperparameters were used for each dataset to test the model.

Table 3: Datasets information after preprocessing.

Dataset	Total samples	Selected features	Training samples	Testing samples
InSDN	343,889	40	240,723 (70%)	103,167 (30%)
CICIDS2017	851,750	35	596,225 (70%)	225,525 (30%)

Table 4: Optimal hyperparameters for FL and CL with the InSDN dataset.

Parameters	Federated learning	Centralized learning
Learning rate	0.0015	0.01
Number of rounds	8	—
Number of epochs	5	20
Batch size	128	128
Number of clients	3, 6, 9, 12	—
Loss function	Adam	Adam

Table 5: Optimal hyperparameters for FL and CL with the CICIDS2017 dataset.

Parameters	Federated learning	Centralized learning
Learning rate	0.01	0.01
Number of rounds	10	—
Number of epochs	40	200
Batch size	120	120
Number of clients	3, 6, 9, 12	—
Loss function	Adam	Adam

4.3. Evaluation Metrics. The proposed system model was evaluated using common metrics from the literature, such as accuracy (ACC), precision (PR), recall (R), F1-score, false positive rate (FPR), training time, and detection time. Accuracy defines the correct classification rate of the model, precision defines the positive predictive value, recall (also called sensitivity) represents the true positive rate (TPR), and the F1-score is defined as the average harmonic mean of the precision and the recall. The detection time is the time taken by the model to detect a network traffic as malicious or benign. The equations related to each metric are as follows:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}, \quad (2)$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}, \quad (3)$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}, \quad (4)$$

$$\text{F1-score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}, \quad (5)$$

$$\text{FPR} = \frac{\text{FP}}{(\text{FP} + \text{TN})}, \quad (6)$$

with TP = Ttue positives (positive instances predicted correctly). TN = true negatives (negative instances predicted correctly). FP = false positives (positive instances predicted incorrectly). FN = false negatives (negative instances predicted incorrectly).

5. Results and Discussion

All the simulation results presented in this section, which are displayed in the tables, were derived from the testing sets utilized to assess the final model's performance. The graphs are the training performance graphs that support the testing results presented. To conduct a comprehensive evaluation of training time, the model underwent training initially on a CPU and subsequently on a GPU. In the comparison with previous works concerning FL, only the best results obtained were used. The FL best results were obtained with 3 clients.

5.1. Main Findings. First of all, the proposed model is assessed using the CL method where all the data is used to train the model in a centralized server. The model training time was largely reduced using the GPU. The results are presented in Tables 6–10. The model outcomes are very good with 99.99% of accuracy for the InSDN testing set and 99.68% of accuracy for the CICIDS2017 testing set. An FPR of 0.0089% and 0.39% were observed for the same testing sets, respectively.

The assessment using federated learning (FL) was conducted with varying numbers of clients ranging from 3 to 12. The objective of this evaluation was to examine the model's performance as the number of clients increased while keeping the size of the dataset constant. The tables 7, 8, 9, and 10 show the results of the FL model in relation to the number of clients used. The model produced accurate results with accuracies ranging from 99.94% to 99.99% on the InSDN testing set and from 98.97% to 99.58% on the CICIDS2017 testing set. The best results were achieved with 3 clients in the

Table 6: Simulation results for centralize learning (CL).

Dataset\metric	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	FPR (%)	Training time (CPU) (s)	Training time (GPU) (s)	Detection time (ms)
InSDN	99.99	99.99	99.99	99.99	0.0089	244.74	45.86	0.091
CICIDS2017	99.68	99.61	99.76	99.68	0.39	4928.51	636.32	0.059

Table 7: Simulation results for federated learning (FL) with 3 clients.

Dataset\metric	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	FPR (%)	Training time (CPU) (s)	Training time (GPU) (s)	Detection time (ms)
InSDN	99.99	99.99	99.99	99.99	0.029	209.31	27.03	0.088
CICIDS2017	99.58	99.38	99.78	99.58	0.617	4709.67	342.32	0.073

Table 8: Simulation results for federated learning (FL) with 6 clients.

Dataset\metric	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	FPR (%)	Training time (CPU) (s)	Training time (GPU) (s)	Detection time (ms)
InSDN	99.98	99.98	99.99	99.99	0.048	208.81	24.20	0.066
CICIDS2017	98.98	98.31	98.69	98.99	1.72	4335.94	339.59	0.071

Table 9: Simulation results for federated learning (FL) with 9 clients.

Dataset\metric	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	FPR (%)	Training time (CPU) (s)	Training time (GPU) (s)	Detection time (ms)
InSDN	99.94	99.96	99.97	99.96	0.151	206.74	24.19	0.077
CICIDS2017	98.98	98.31	98.69	98.99	1.79	4333.82	340.03	0.071

Table 10: Simulation results for federated learning (FL) with 12 clients.

Dataset\metric	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	FPR (%)	Training time (CPU) (s)	Training time (GPU) (s)	Detection time (ms)
InSDN	99.94	99.96	99.97	99.96	0.156	206.41	24.03	0.065
CICIDS2017	98.97	98.31	98.68	98.98	1.89	4334.06	338.70	0.070

FL setup, and performance slightly decreased as the number of clients increased. This suggests that the FL model performs better when there is an ample amount of data for local models' training. The results also show very good computation improvement when the model is trained using a GPU rather than a CPU. GPUs are already known in the literature to be compute efficient in terms of time in DL models training. Another interesting metric is the proposed model detection time turning around 0.07 ms, which is a good sensitivity for real-world applications.

On both FL and CL, the proposed hybrid model yielded good performance. However, the model exhibited superior performance when employing CL compared to FL, as evidenced by its higher accuracy, precision, recall, F1-score, and lower false-positive rate (FPR). This can be attributed to the distinct training methodologies; FL involves training by multiple collaborative clients, resulting in the aggregation of local models to form a global average model, whereas CL employs centralized training in a single location. But in terms of training computation time, FL obtained a higher score because the training process is handled by many clients, reducing the global training time when compared to CL. What concerns data privacy? This is where FL takes a big advantage compared to CL which needs to collect data from different sources and send it to a centralized place for use. In the context of this study, where 5G VNFs can be provided by different vendors hosted in different clouds, FL can provide data privacy preservation between third-party VNFs, allowing them to train a high-performance intrusion detection model without sharing raw data to avoid data violation risks. If we consider the tradeoff between data privacy and high-performing malicious traffic detection, federated learning is preferable for real-world application scenarios.

In terms of latency (which is very important in 5G networks), FL is such that only the local trained model parameters are send through the network, reducing computational cost and mitigating bandwidth consumption compared to CL where large amount of raw data need to be sent to a centralized server. The proposed model stored in the system exhibited a size of 668.75 kilobytes, a notably compact scale when considering the substantial capacities of current computers and networks.

5.2. Proposed Model Comparison with Previous Works: Centralized Learning. In this subsection, the performance of the proposed model when trained using a centralized learning method is presented and compared to previous related works. The simulation results showed very good performance of the proposed model for both datasets (InSDN and CICIDS2017). The graphs in Figures 3 and 4 show the model training performance for each dataset. Table 11 summarizes the comparison results. Only previous studies that used datasets containing more than four types of attack classes were considered.

The proposed model performed better than previous models. It showed a very high accuracy for the InSDN dataset with a very low FPR, surpassing previous studies that used the same dataset. In addition, for the CICIDS2017 dataset, the model performed better than those in the literature that used the same dataset.

5.3. Proposed Model Comparison with Previous Works: Federated Learning. In contrast to centralized learning, federated learning involves using many virtual clients to train the proposed model. In the experiments conducted, TLS version 1.3 encryption was used to encrypt the data between the clients and the server during the training process. The proposed model performed well on the InSDN and CICIDS2017 datasets. Figures 5 and 6 support this conclusion. They show the training accuracy and loss graphs of the model for each dataset. Each round in FL is the results of several epochs of the local models training. It can be observed that the convergence of the model start stabilizing at 3 rounds on the InSDN training set and at 7 rounds on the CICIDS2017 training set. We kept the model training for few more rounds to allow it to achieve its best training capacity which yielded to the best accuracy on the testing sets described previously.

The evaluation using the InSDN testing dataset yielded better results, with 99.99% accuracy and 0.029% FPR. For the CICIDS2017 testing set, the model achieved 99.58% accuracy with an FPR of 0.617%. The GPU allowed a large reduction of the training time and the improvement of the detection time for both testing sets. To validate the proposed system, it was compared with previous studies that used federated learning for intrusion detection. Table 12 summarizes this comparison with the results of previous studies. We found that most previous studies did not consider the encryption of communication between clients and server during federated learning. In real-world scenarios, particularly for 5G virtual network functions, the server and VNFs can be hosted in different cloud networks. Thus, encryption between VNFs is required to avoid data tempering. The proposed system, which integrates an encryption layer, is more realistic for 5G network applications.

In both traditional and federated learning, the proposed model proved its efficiency by surpassing the models proposed in the literature for the same datasets. Despite the encryption of communication for FL, the training time of the model remained lower than that for CL. This can be explained by the fact that for FL, the computational cost is divided between the participating clients using their own local resources to train the model, whereas for CL, only one device is used to train the model. Using a GPU helped for better training time experience compared to when the model is trained using a CPU. We admit that the proposed hybrid deep neural network model based on a single-layer CNN, single-layer BiLSTM, and single-layer LSTM is efficient for adoption with FL under TLS encryption for 5G VNFs security monitoring. The framework can effectively detect malicious traffic, mitigate potential data privacy violations and data-tempering risks between VNFs from different vendors, and make it easy for vendors to collaboratively train a strong intrusion detection system without sharing raw data.

One limitation of the experiments was that the benchmark dataset sizes used were limited. The model underwent evaluation with 3, 6, 9, and 12 clients. The findings revealed that the performance of the federated learning (FL) model is adversely affected by an increase in the number of clients. This impact is attributed to the diminishing dataset size employed by the local models as the client count rises. As a result, the experiments conducted with 3 clients demonstrated superior performance. Conversely, the experiments involving 12 clients exhibited greater performance degradation. Despite these limitations, the proposed system proved to be effective for malicious traffic detection. The framework design is more suitable for 5G VNFs security-monitoring applications than those in the literature. The next section describes the implementation of the system in the 3GPP 5G architecture.

6. Proposed System Integration with 3GPP 5G Architecture

In this section, we present and describe the integration of the proposed solution into a 5G core network for security monitoring. The 5G network architecture, as defined by 3GPP, includes a significant component known as the network data analytic function (NWDAF). This network function plays a critical role in facilitating efficient data collection and analysis at the edge of a 5G network and is designed to work with artificial intelligent technologies [18]. By leveraging the existing infrastructure of NWDAF, the proposed solution can be seamlessly integrated without necessitating any modifications to the underlying 5G architecture. For a more comprehensive and detailed understanding of other 5G virtual network functions, readers can refer to the online accessible book, 3GPP TS 23.501 [56]. Figure 7 illustrates the implementation of the solution within the 5G core network. The NWDAF can play the role of an aggregation server allowing the core network VNFs from different vendors hosted in different clouds to collaboratively train the same model under the TLS security protocol. The advantage of this approach is that even different mobile operators can collaborate to train a common intrusion detection model with more diverse data without sharing sensitive raw data. Furthermore, FL is designed to be compatible with all types of network architectures, as long as

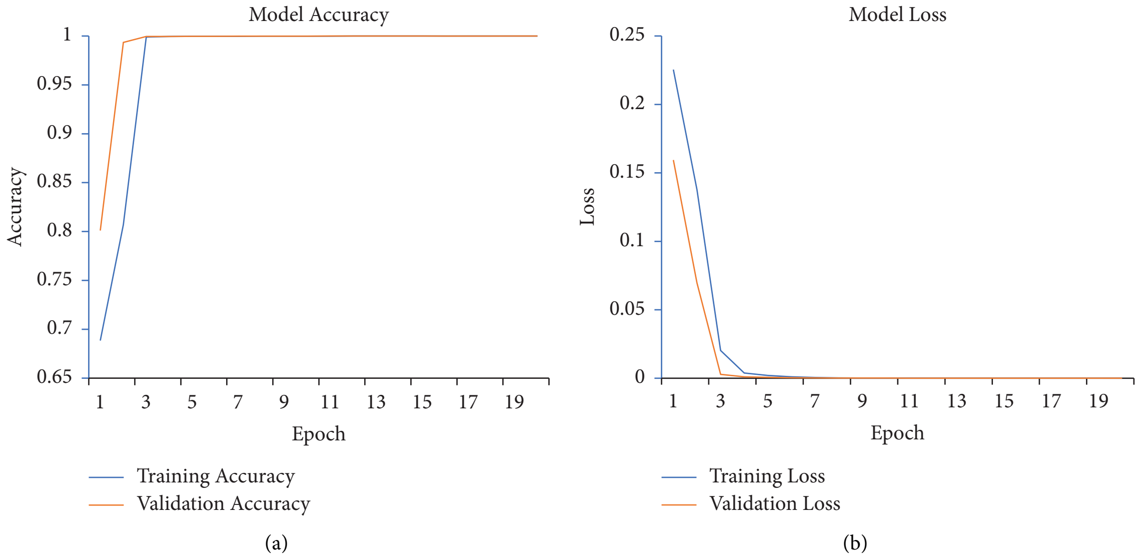


Figure 3: CL accuracy and loss graphs when the model is trained using the InSDN dataset. (a) Accuracy. (b) Loss.

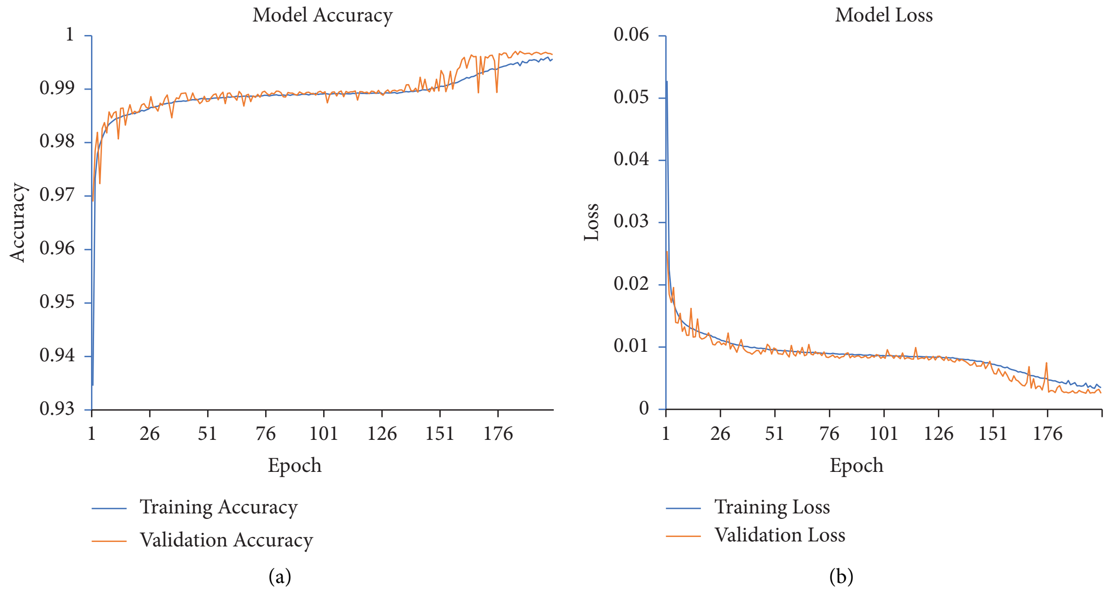


Figure 4: CL accuracy and loss graphs when the model was trained using the CICIDS2017 dataset. (a) Accuracy. (b) Loss.

Table 11: Proposed model compared with previous works (traditional learning).

Reference	Year	Technique used	Dataset utilized	Accuracy (%)	FPR (%)
Proposed model	2023	CNN-BiLSTM-LSTM	InSDN	99.99	0.0089
			CICIDS2017	99.68	0.39
[31]	2022	CNN	CICIDS2017	99.0	0.67
[26]	2020	DBN	CICIDS2017	97.73	—
[24]	2021	Hybrid CNN	InSDN	99.28	—
[51]	2021	DFFNN	NSL-KDD	99.0	1.0
			UNSW-NB15	98.9	1.1
[52]	2022	LSTM-BiLSTM-GRU-BiGRU	NSL-KDD	87.44	20.47
			UNSW-NB15	82.46	37.61
[53]	2022	MLP	UNSW-NB15	96.7	—

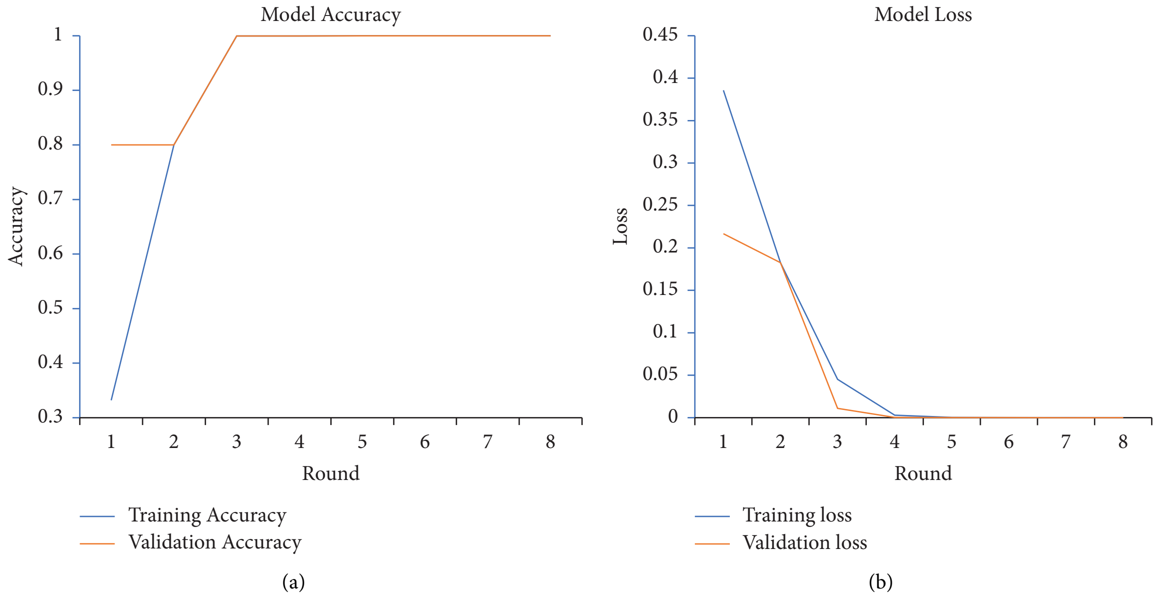


FIGURE 5: FL accuracy (a) and loss (b) graphs when the model is trained using the InSDN dataset.

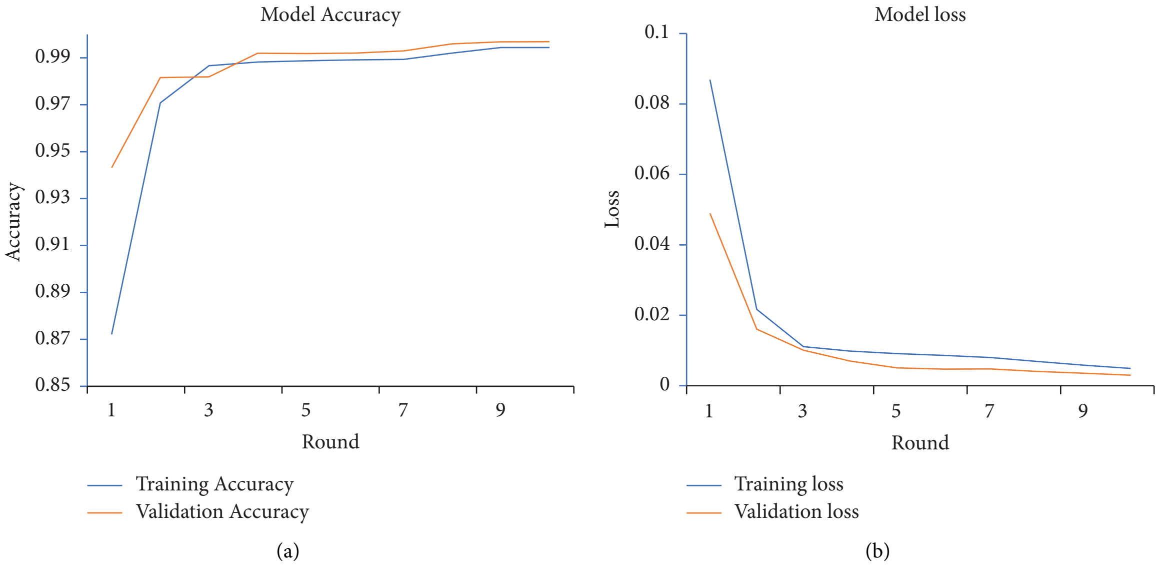


FIGURE 6: FL accuracy (a) and loss (b) graphs when the model was trained using the CICIDS2017 dataset.

TABLE 12: Comparison with previous studies that used FL for intrusion detection.

Reference	Year	Technique used	Dataset utilized	Accuracy (%)	Precision (%)	Encryption technique
[36]	2022	Multi-layer perceptron (MLP)	Private dataset	95	—	None
[37]	2020	Transfer learning	CICIDS2017	91.93	—	None
[39]	2022	ANN based-FL	UNSW-NB15	93.6	—	None
[40]	2022	Neural network that uses transformer	NSL-KDD	99.48	99.49	None
[41]	2023	Gower dissimilarity matrices	TON_IOT	95.5	96	None
[54]	2020	MLP	NSL-KDD	98.11	—	None
[55]	2021	CNN + GRU	Industrial CPS	99.20	98.86	None
Our model		CNN + BiLSTM + LSTM	InSDN CICIDS2017	99.98 99.58	99.98 99.38	TLS V 1.3

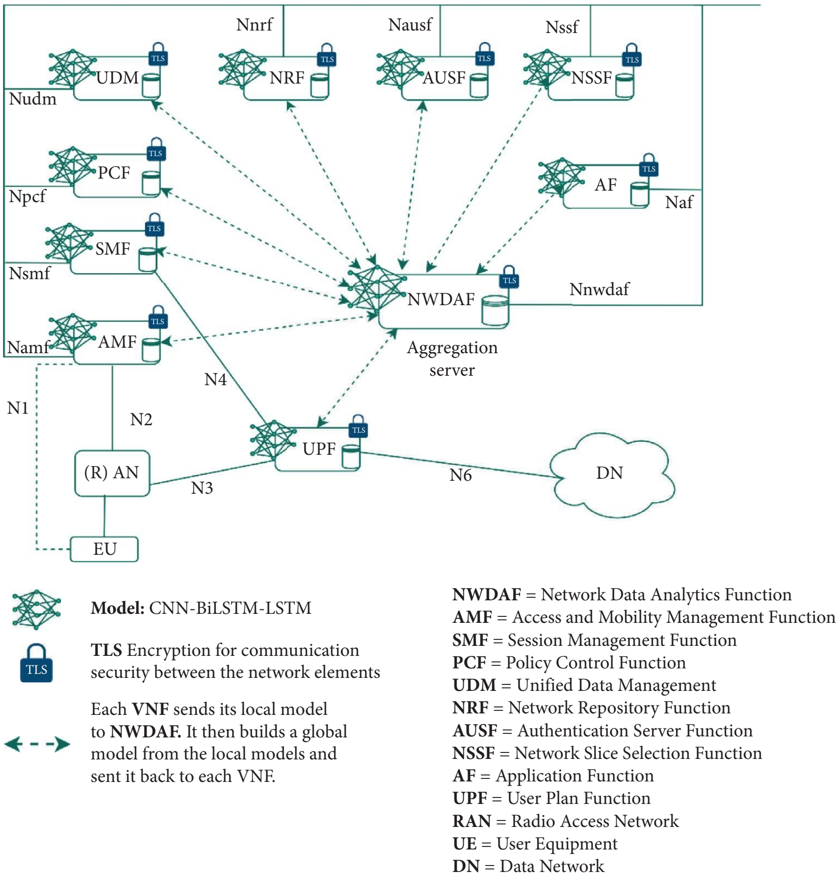

Nnrf
Nausf
Nssf
UDM
NRF
AUSF
NSSF
TLS
Nudm
PCF
Npcf
AF
Naf
SMF
Nsmf
NWDAF
Nnwdaf
AMF
Namf
Aggregation server
N4
N1
N2
UPF
(R) AN
N3
N6
DN
EU
Model: CNN-BiLSTM-LSTM
TLS Encryption for communication security between the network elements
Each VNF sends its local model to NWDAF. It then builds a global model from the local models and sent it back to each VNF.
NWDAF = Network Data Analytics Function
AMF = Access and Mobility Management Function
SMF = Session Management Function
PCF = Policy Control Function
UDM = Unified Data Management
NRF = Network Repository Function
AUSF = Authentication Server Function
NSSF = Network Slice Selection Function
AF = Application Function
UPF = User Plan Function
RAN = Radio Access Network
UE = User Equipment
DN = Data Network

Figure 7: Proposed solution integration with 5G core network.

the clients and server can interact with one another. In the context of 5G networks, all virtual network functions (VNFs) within the operator's network, regardless of architecture, are connected to the NWDAF for centralized monitoring, making the implementation of FL even more streamlined.

Implementing an FL network security monitoring system into a real world heterogeneous 5G network can present various challenges. Successful implementation requires collaboration among vendors providing VNFs within the system. Typically, each vendor employs its own team of engineers responsible for administering their VNFs within the telecommunications service provider network. The success of FL implementation hinges on fostering effective cooperation among these engineering teams who must work together to enable FL functionalities. However, achieving seamless collaboration proves challenging, as engineers often need to operate concurrently while ensuring the compatibility and interoperability of their respective VNFs. This collaboration is essential for overcoming barriers and ensuring the smooth integration of FL based security monitoring into the complex landscape of a 5G network. At this stage, quantifying the complexity of this procedure is challenging, given its dependence on factors such as the telecommunication operator network architecture, the enterprise culture, and the business policies established among vendors. In summary, the proposed model integration complexity will depend on how the network operator will handle the collaboration between the VNFs vendors located in different clouds.

7. Conclusion and Future Work

The virtualization and heterogeneity of 5G networks necessitate the reconsideration of existing security monitoring systems in terms of the data privacy required by authorities and deep network traffic inspection for malicious traffic detection. Determining the tradeoff between data privacy and efficient network traffic inspection for malicious traffic detection requires further research. Federated learning with a hybrid deep neural network model is proposed in this study to improve data privacy safeguarding between 5G VNFs while building strong and sustainable security monitoring systems. The proposed model performed well; its integration with the 3GPP 5G core network architecture is presented, and the implementation challenges are highlighted.

For further investigation, the following can be explored:

(i) The available datasets are limited in terms of sample size to accurately reflect future 5G core network VNFs scenarios. Therefore, it is necessary to address this gap by developing larger datasets for large scale simulation using hundreds of clients.

(ii) Researchers can explore the possibility of encrypted data inspection in a federated learning setup where different clients use different encryption technologies.

Data Availability

The datasets used to support the findings of this study are open access and have been cited in the article. However, the preprocessed versions can be obtained through request to the corresponding authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

The authors have contributed to this work as follows: Abdoul-Aziz Maiga contributed to conceptualization, methodology, software, and writing the original draft. Edwin Ataro contributed to validation, formal analysis, supervision, resources, reviewing and editing, and funding Acquisition. Stanley Githinji contributed to validation, formal analysis, conceptualization improvement, supervision, and reviewing and editing. All the authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research has been funded by the African Union (AU).

References

[1] H. Kim, "5G core network security issues and attack classification from network protocol perspective," *Journal of Internet Services and Information Security*, vol. 10, pp. 1–15, 2020.

[2] M. Gharbaoui, C. Contoli, G. Davoli et al., "An experimental study on latency-aware and self-adaptive service chaining orchestration in distributed NFV and SDN infrastructures," *Computer Networks*, vol. 208, Article ID 108880, 2022.

[3] M. Smine, D. Espes, N. Cuppens-Boulahia, and F. Cuppens, "Network functions virtualization access control as a service," in *Proceedings of the Data and Applications Security and Privacy XXXIV*, A. Singhal and J. Vaidya, Eds., pp. 100–117, Springer International Publishing, Berlin, Germany, 2020.

[4] T. Madi, H. A. Alameddine, M. Pourzandi, and A. Boukhtouta, "NFV security survey in 5G networks: a three-dimensional threat taxonomy," *Computer Networks*, vol. 197, Article ID 108288, 2021.

[5] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5G technologies: potential solutions, recent advancements, and future directions," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, pp. 196–248, 2020.

[6] G. Sahu and S. S. Pawar, "Security challenges in 5G network," in *Software Defined Networking for Ad Hoc Networks*, M. M. Ghonge, S. Pramanik, and A. D. Potgantwar, Eds., pp. 75–94, Springer International Publishing, Berlin, Germany, 2022.

[7] M. Zoure, T. Ahmed, and L. Réveillère, "Network services anomalies in NFV: survey, taxonomy, and verification methods," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1567–1584, 2022.

[8] A 5G, A.W.P, "Evolving-5G-Security-for-the-Cloud-2022-InDesign," 2023, https://www.5gamericas.org/wp-content/uploads/2022/09/Evolving-5G-Security-for-the-Cloud-2022-InDesign.pdf.

[9] J.-M. Chen, S. Chen, X. Wang, L. Lin, and L. Wang, "A virtual machine migration strategy based on the relevance of services against side-channel attacks," *Security and Communication Networks*, vol. 2021, Article ID 2729949, 17 pages, 2021.

[10] A. Dutta and E. Hammad, "5G security challenges and opportunities: a system approach," in *Proceedings of the 2020 IEEE 3rd 5G World Forum (5GWF)*, pp. 109–114, Bangalore, India, June 2020.

[11] J. Jithish, B. Alangot, N. Mahalingam, and K. S. Yeo, "Distributed anomaly detection in smart grids: a federated learning-based approach," *IEEE Access*, vol. 11, pp. 7157–7179, 2023.

[12] T. Subramanya and R. Riggio, "Centralized and federated learning for predictive VNF autoscaling in multi-domain 5G networks and beyond," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 63–78, 2021.

[13] H. Qu, K. Wang, and J. Zhao, "Survivable SFC deployment method based on federated learning in multi-domain network," *The Journal of Supercomputing*, vol. 79, no. 16, pp. 18198–18226, 2023.

[14] R. Verma and K. M. Sivalingam, "Federated learning approach for auto-scaling of virtual network function resource allocation in 5G-and-Beyond networks," in *Proceedings of the 2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, pp. 242–246, Paris, France, November 2022.

[15] P. Tam, R. Corrado, C. Eang, and S. Kim, "Applicability of deep reinforcement learning for efficient federated learning in massive IoT communications," *Applied Sciences*, vol. 13, no. 5, p. 3083, 2023.

[16] L. Tang, T. Wu, X. Zhou, and Q. A. Chen, "Virtual network function migration algorithm based on federated learning prediction of resource requirements," *Journal of Electronics and Information Technology*, vol. 44, pp. 3532–3540, 2022.

[17] D. Gupta, O. Kayode, S. Bhatt, M. Gupta, and A. S. Tosun, "Hierarchical federated learning based anomaly detection using digital twins for smart healthcare," in *Proceedings of the 2021 IEEE 7th International Conference on Collaboration and Internet Computing (CIC)*, pp. 16–25, Atlanta, GA, USA, July 2021.

[18] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[19] Y. Liu, N. Kumar, Z. Xiong, W. Y. B. Lim, J. Kang, and D. Niyato, "Communication-efficient federated learning for anomaly detection in industrial internet of Things," in *Proceedings of the GLOBECOM 2020- 2020 IEEE Global Communications Conference*, pp. 1–6, Taipei, Taiwan, August 2020.

[20] D. Man, F. Zeng, W. Yang, M. Yu, J. Lv, and Y. Wang, "Intelligent intrusion detection based on federated learning for edge-assisted internet of Things," *Security and Communication Networks*, vol. 2021, Article ID 9361348, 11 pages, 2021.

[21] X. Wang, Y. Wang, Z. Javaheri, L. Almutairi, N. Moghadamnejad, and O. S. Younes, "Federated deep learning for anomaly detection in the internet of Things," *Computers and Electrical Engineering*, vol. 108, Article ID 108651, 2023.

[22] Y. Xin, L. Kong, Z. Liu et al., "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[23] I. H. Abdulqadder, D. Zou, I. T. Aziz, B. Yuan, and W. Dai, "Deployment of robust security scheme in SDN based 5G network over NFV enabled cloud environment," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 2, pp. 866–877, 2021.

[24] 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," *Journal of Network and Computer Applications*, vol. 191, Article ID 103160, 2021.

[25] M. V. O. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença, "A GRU deep learning system against attacks in software defined networks," *Journal of Network and Computer Applications*, vol. 177, Article ID 102942, 2021.

[26] S. Manimurugan, S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, and R. Patan, "Effective attack detection in internet of medical Things smart environment using a deep Belief neural network," *IEEE Access*, vol. 8, pp. 77396–77404, 2020.

[27] W. Yao, H. Shi, and H. Zhao, "Scalable anomaly-based intrusion detection for secure internet of Things using generative adversarial networks in fog environment," *Journal of Network and Computer Applications*, vol. 214, Article ID 103622, 2023.

[28] V. Rey, P. M. Sánchez Sánchez, A. Huertas Celdrán, and G. Bovet, "Federated learning for malware detection in IoT devices," *Computer Networks*, vol. 204, Article ID 108693, 2022.

[29] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, Article ID 102419, 2020.

[30] V. Hnamte and J. Hussain, "DCNNBiLSTM: an efficient hybrid deep learning-based intrusion detection system," *Telematics and Informatics Reports*, vol. 10, Article ID 100053, 2023.

[31] H. S. Ilango, M. Ma, and R. Su, "A FeedForward–convolutional neural network to detect low-rate DoS in IoT," *Engineering Applications of Artificial Intelligence*, vol. 114, Article ID 105059, 2022.

[32] N. Yadav, S. Pande, A. Khamparia, and D. Gupta, "Intrusion detection system on IoT with 5G network using deep learning," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 9304689, 13 pages, 2022.

[33] R. O. Ogundokun, S. Misra, R. Maskeliunas, and R. Damasevicius, "A Review on federated learning and machine learning approaches: categorization, application areas, and blockchain technology," *Information*, vol. 13, no. 5, p. 263, 2022.

[34] M. Shaheen, M. S. Farooq, T. Umer, and B.-S. Kim, "Applications of federated learning; taxonomy, challenges, and research trends," *Electronics*, vol. 11, no. 4, p. 670, 2022.

[35] E. Bandara, X. Liang, S. Shetty, R. Mukkamala, A. Rahman, and N. W. Keong, "Skunk— a blockchain and zero trust security enabled federated learning platform for 5G/6G network slicing," in *Proceedings of the 2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 109–117, Napoli, Italy, June 2022.

[36] A. Boualouache and T. Engel, "Federated learning-based scheme for detecting passive mobile attackers in 5G vehicular edge computing," *Annales des Telecommunications*, vol. 77, no. 3-4, pp. 201–220, 2022.

[37] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "IoTDefender: a federated transfer learning intrusion detection framework for 5G IoT," in *Proceedings of the 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE)*, pp. 88–95, Guangzhou, China, June 2020.

[38] H. A. Kholidy and R. Kamaludeen, "An innovative hashgraph-based federated learning approach for Multi domain 5G network protection," in *Proceedings of the 2022 IEEE Future Networks World Forum (FNWF)*, pp. 139–146, Montreal, Canada, July 2022.

[39] S. Jayasinghe, Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "Federated learning based anomaly detection as an enabler for securing network and service management automation in beyond 5G networks," in *Proceedings of the 2022 Joint European Conference on Networks and Communications and 6G Summit (EuCNC/6G Summit)*, pp. 345–350, Grenoble, France, June 2022.

[40] X. Sun, Z. Tang, M. Du et al., "A hierarchical federated learning-based intrusion detection system for 5G smart grids," *Electronics*, vol. 11, no. 16, p. 2627, 2022.

[41] A. Belenguer, J. A. Pascual, and J. Navaridas, "GöwFed: a novel federated network intrusion detection system," *Journal of Network and Computer Applications*, vol. 217, Article ID 103653, 2023.

[42] J. Koushik, *Understanding convolutional neural networks*, arXiv preprint arXiv:1605.09081, 2016, https://doi.org/10.48550/arXiv.1605.09081.

[43] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, July 2017.

[44] J. Gu, Z. Wang, J. Kuen et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[45] N. Shenvi and H. Virani, "Forecasting of ionospheric total electron content data using multivariate deep LSTM model for different latitudes and solar activity," *Journal of Electrical and Computer Engineering*, vol. 2023, Article ID 2855762, 13 pages, 2023.

[46] A. H. Vasoukolaei, D. Sattar, and A. Matrawy, "TLS performance evaluation in the control plane of a 5G core network slice," in *Proceedings of the 2021 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 155–160, Thessaloniki, Greece, November 2021.

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

[48] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "A detailed analysis of the CICIDS2017 data set," in *Proceedings of the Information Systems Security and Privacy*, P. Mori,

S. Furnell, and O. Camp, Eds., pp. 172–188, Springer International Publishing, Berlin, Germany, 2019.

[49] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges," *Computer Networks*, vol. 167, Article ID 106984, 2020.

[50] D. J. Beutel, T. Topal, A. Mathur et al., "Flower: a friendly federated learning research framework," 2020, https://arxiv.org/abs/2007.14390.

[51] J. B. Awotunde, C. Chakraborty, and A. E. Adeniyi, "Intrusion detection in industrial internet of Things network-based on deep learning model with rule-based feature selection," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 7154587, 17 pages, 2021.

[52] P. B. Udas, M. E. Karim, and K. S. Roy, "SPIDER: a shallow pca based network intrusion detection system with enhanced recurrent neural networks," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 10246–10272, 2022.

[53] A. Ugendhar, B. Illuri, S. R. Vulapula et al., "A novel intelligent-based intrusion detection system approach using deep multilayer classification," *Mathematical Problems in Engineering*, vol. 2022, Article ID 8030510, 10 pages, 2022.

[54] N. A. Al-Athba Al-Marri, B. S. Ciftler, and M. M. Abdallah, "Federated mimic learning for privacy preserving intrusion detection," in *Proceedings of the 2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 1–6, Odessa, Ukraine, June 2020.

[55] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2021.

[56] T. Gpp, "23 501 system architecture for the 5G system (release 16)," *Technical Reports Series*, vol. 2018, 2018.