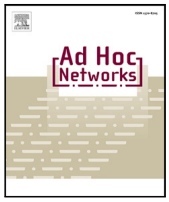
Ad Hoc Networks 155 (2024) 103407

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc



Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability

Syed Muhammad Salman Bukhari [a,1], Muhammad Hamza Zafar [b,1], Mohamad Abou Houran [c], Syed Kumayl Raza Moosavi [d], Majad Mansoor [e], Muhammad Muaaz [f], Filippo Sanfilippo [b,g,*]

[a] Department of Electrical Engineering, Capital University of Science and Technology, Islamabad, 44000, Pakistan
[b] Department of Engineering Sciences, University of Agder, Grimstad, 4879, Norway
[c] School of Electrical Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
[d] SEECS, National University of Sciences and Technology, Islamabad, 44000, Pakistan
[e] Department of Automation, University of Science and Technology of China, China, 28796, China
[f] Department of Information and Communication Technology, University of Agder, Grimstad, 4879, Norway
[g] Department of Software Engineering, Kaunas University of Technology, Kaunas, 51368, Lithuania

ARTICLE INFO

Keywords:
WSNs
Network intrusion detection
Federated learning
Denial of Service
SCNN-Bi-LSTM
Stacked CNN

ABSTRACT

As the digital landscape expands rapidly due to technological advancements, cybersecurity concerns have become more prevalent. Intrusion Detection Systems (IDSs), which are crucial for identifying unusual network traffic indicative of malicious activity, have become a necessity. These systems can be either hardware or software-based. However, traditional IDS models often fail to adequately protect data privacy and detect complex, unique breaches, particularly within Wireless Sensor Networks (WSNs). To address these limitations, this paper proposes a novel Stacked Convolutional Neural Network and Bidirectional Long Short Term Memory (SCNN-Bi-LSTM) model for intrusion detection in WSNs. This model leverages Federated Learning (FL) to enhance intrusion detection performance and safeguard privacy. The FL-based SCNN-Bi-LSTM model is unique in its approach, allowing multiple sensor nodes to collaboratively train a central global model without revealing private data, thereby alleviating privacy concerns. The deep learning methodology of the SCNN-Bi-LSTM model effectively identifies sophisticated and previously unknown cyber threats by meticulously examining both local and temporal linkages in network patterns. The model has been specifically designed to detect and categorize different types of Denial of Service (DoS) attacks using specialized WSN-DS and CIC-IDS-2017 datasets. Compared to traditional Artificial Deep Neural Network (ADNN) models, our proposed FL-SCNN-Bi-LSTM model demonstrated superior detection rates for complex and unknown attacks, significantly improving IDS performance. The model achieved a notable classification accuracy of approximately 99.9% precision and recall on both datasets, substantially reducing false positives and negatives. Our research underscores the potential of federated learning and deep learning in enhancing the security and privacy of WSNs. The proposed FL-SCNN-Bi-LSTM architecture not only facilitates the identification of complex cyber threats but also exemplifies how deep learning techniques can be employed to bolster intrusion detection systems while preserving user data privacy.

1. Introduction

Due to its numerous real-time applications in essential tactical monitoring, battlegrounds, building security monitoring, monitoring wildfires, and medical care, Wireless Sensor Networks (WSNs) have grown in importance as a study area [1]. A WSN is composed of a significant amount of independent nodes for sensors that are dispersed throughout various study areas to collect vital information and collaboratively transfer the gathered information wirelessly to a more potent node known as the sink node or Base Station (BS) [2,3]. The transportation of data over the network is facilitated by the appropriate WSN protocols. Ensuring the security of WSNs from various risks is of paramount importance. However, due to the limited resources of WSNs, such as battery life, memory, and computing power, achieving this objective presents a significant challenge [4]. Due to their limiting characteristics, conventional safety precautions like encryption might

* Corresponding author at: Department of Engineering Sciences, University of Agder, Grimstad, 4879, Norway.
E-mail address: filippo.sanfilippo@uia.no (F. Sanfilippo).
[1] The authors contributed equally to this work.

https://doi.org/10.1016/j.adhoc.2024.103407
Received 3 October 2023; Received in revised form 8 January 2024; Accepted 14 January 2024
Available online 20 January 2024

not always be sufficient for such networks. Because of their open and dispersed architecture and the constrained resources of their sensor nodes, WSNs are extremely susceptible to assaults. Additionally, as WSNs must often broadcast packets, sensor nodes can be arbitrarily placed across the environment, making it simple for an adversarial attacker to insert themselves into a WSN [5]. This open and distributed nature of WSNs not only makes them indispensable for various applications but also highly vulnerable to cyberattacks. An intruder can take control of a sensor node and use it to eavesdrop on conversations, transmit misleading messages, alter the authenticity of the data, and utilize network resources. The severity of these security challenges is further amplified by the escalating complexity and sophistication of modern malware. The 2017 Symantec Internet Security Threat Report [6] highlighted a dramatic increase in zero-day attacks, with over three billion instances reported in 2016 alone. Additionally, the Data Breach Statistics of 2017 [7] noted that approximately nine billion data records were compromised since 2013, indicating a significant shift in the target of cybercriminals from individual users to larger entities such as financial institutions. The Australian Cyber Security Centre's 2017 report [8] further emphasizes the growing need for more advanced IDS, in response to the varied sophistication of these attacks. A common and serious assault on WSNs is the Denial of Service (DoS) attack, which aims to cripple and interrupt the services they offer [9,10]. In light of these increasingly sophisticated threats, an Intrusion Detection System (IDS) becomes not just a tool but an essential component to detect both known and new attacks, alerting sensor nodes in real-time. The challenge, however, lies in the limited resources of WSNs, implementing effective IDSs to prevent or mitigate these risks is a formidable task [2,3].

When an intrusion takes place, IDS can identify suspicious or anomalous activity and sound an alert. Because sensor nodes are often created to be small, inexpensive, and lacking in hardware resources, implementing IDSs for WSNs is more challenging than for other systems. A dedicated dataset containing typical profiles and attacks in WSN is also lacking, making it impossible to identify an attacker's signature [2,3]. Given all of these obstacles, an IDS for WSNs must largely meet two criteria: it must be extremely accurate in recognizing an intruder, including unknown attacks, and it must also be computationally inexpensive to assure low influence on the WSNs' architecture [11]. In this research, a specialized WSN dataset is built to characterize four different forms of DoS assaults as well as the behavior under normal conditions. Because they may be used to identify intrusions and draw conclusions in certain network settings, Machine Learning (ML) techniques can be employed to build a strong IDS [12]. The majority of the time, putting into practice a successful custom IDS approach involves several difficulties. Based on the system's usability, capability, and accuracy, the underlying implementation concerns may be divided into many kinds of issues [13]. However, compared to earlier techniques depending on handmade signatures, IDS based on anomaly detection that use ML methodologies frequently have a greater False Positive Rate (FPR). As a result, analyzing data and identifying real-time incursions are difficult for ML anomaly-based systems. These systems' learning process requires high-dimensional training data to get around these constraints, which makes it more difficult and time-consuming than competing techniques [14].

The limitations of existing models in literature and the complexity of intrusion detection in WSNs a hybrid FL-SCNN-Bi-LSTM algorithm is investigated. The model's strength lies in its ability to learn from both local and temporal dependencies, enhancing its detection capabilities. Furthermore, the model employs federated learning, allowing multiple sensor nodes to collaboratively learn while preserving data privacy, a crucial aspect in environments where sensitive information protection is paramount. In building our model, we selected only strongly correlated data features from both datasets to enhance the overall relevance of the feature set. This refined feature selection process ensures the model focuses on the most significant data features, thereby improving detection performance. The proposed FL-SCNN-Bi-LSTM approach is used to train the model, effectively leveraging deep learning to analyze intrusion patterns. Under the privacy-preserving umbrella of Federated Learning, the model utilizes the spatial data collection capabilities of Convolutional Neural Networks and the temporal relationship capturing abilities of Bidirectional LSTM. This unique combination of methodologies enhances the model's intrusion detection capabilities while addressing issues related to data privacy and computational resource constraints. Fig. 1 shows the suggested model's fundamental design, giving a graphic picture of our innovative strategy's elements. The findings of this investigation demonstrate the potential of our approach to deliver a reliable, effective, and privacy-preserving solution for intrusion detection in WSNs.

1.1. Related work

Researchers have looked at intrusion detection systems using machine learning techniques in recent years, and they have provided remedies to the issues and restrictions of traditional IDS approaches [15]. Different techniques for determining the data sample type have been put forth in earlier works to categorize situations into normal and anomalous groups. Therefore, the literature on the most well-known IDS approaches is examined in this part. A hybrid network intrusion detection system utilizing the XGBoost algorithm was proposed by Dhaliwal et al. [16]. To benchmark their suggested strategy, they used the NSL-KDD dataset, and they compared their outcomes to those of other algorithms. The test results show that, in terms of classification accuracy, XGBoost outperforms Random Forest, Support Vector Machine (SVM), and Naive Bayes (NB). These findings demonstrate that the XGboost model with the highest performance is one where the parameter values are fixed. Amiri et al.'s [17] proposal for a network intrusion detection system that combines the PSO and XGBoost algorithms is known as PSO-XGBoost. In general, software-defined networks may incorporate intelligent algorithms to distinguish between usual and abnormal behavior in conventional networks [18]. For instance, [19] presents a wide range of clever techniques that may be applied to intrusion detection and the detection of distributed DDoS assaults in S.D.Ns. Actually, by using several approaches, such as neural network-based ones for classifying behaviors in SDNs like multilayered perception (MLP), SVM, evolutionary technique, fuzzy theory, Bayesian Networks, and Decision Tree (DT) is discussed in [19], with a breakdown of their advantages and disadvantages. In [20] introduces a technique for stopping low traffic assaults with large flows. A DDoS assault might come from any switch interface; hence the writers are looking for vulnerable interfaces. In such assaults, the attacker may be located in many subnets that are linked to various switches and do not deliver the flow directly to the controller. As a result, there is little data delivered to the various switches, making it impossible for the controller to notice the assault. The classification of interface flows is done first, and the judgment is determined using the Statistical Probability Ratio Test (SPRT). Jankowski and Amanowicz [21] present a technique for monitoring and spotting malicious activity utilizing SDN characteristics.

The authors have included a variety of modules, including one that extracts traffic flow data, one that recognizes and interprets traffic flows, and one that matches flows. In [22], The feature extraction process involves the combination of two modules, which then forward the result to a classifier. This classifier employs machine learning to identify the type of flow within the data segment. The module, situated on the Open Day Light controller, takes into account parameters such as Source IP address, Destination IP address, Source port, Destination port, and Protocol type to match the flow. For classification, we incorporate a Self-Organizing Map (SOM) and Learning Vector Quantization (LVQ). The effectiveness of these networks is then evaluated by comparing their False Positive (FP) and True Positive (TP) rates. This comparison is based on the categorization time and the percentage of errors,

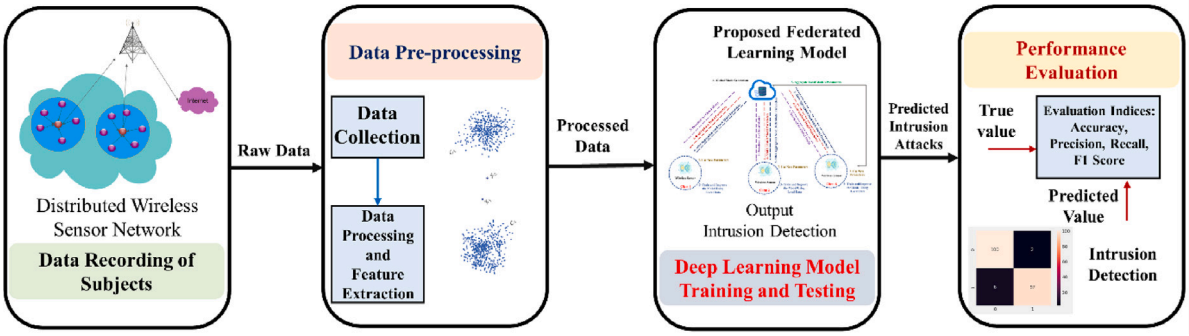


Fig. 1. Proposed structure for training and testing of federated learning assisted deep learning model for network intrusion prediction.

providing a comprehensive assessment of their performance. The study by Braga et al. in [23] presents a low overhead solution for detecting DDoS assaults based on the properties of the flow. This method uses a SOM neural network. It is utilized by the controller for NOX. The flows are divided into regular and assault classes by the neural network once it has been trained using the network's traffic characteristics. The NOX controller captures the characteristics of switch traffic flows while monitoring multiple switches during predetermined time intervals. Each incident is forwarded to a classifier module, where the neural network decides whether or not an attack has taken place. The effectiveness of neural networks in S-D-N security is examined in [24] since they are frequently utilized for intrusion detection in SDNs. To identify harmful assaults on a host, other algorithms like D.T., B.N., Naive Bayes (NB), and C4.5 decision tree are also utilized in [25]. In reality, [25] restricts the attacker's access by banning its subnet by utilizing limiting rules on the controller. A fuzzy logic-based decision-making strategy is provided in [26]. A statistics collection module, a processing module, and a decision module make up this design. On the controller, these modules are implemented as Java programs. The NSL-KDD dataset contains the following six properties, which may be utilized by a deep neural network to identify abnormalities in [27]. (Time, Protocol Type, SRC, DST, Bytes, Count, SRC) A comparison of conventional and DL algorithms for anomaly identification is described in [28,29], and [30]. These tests show how deep learning algorithms may gather network traffic information from numerous places and offer enhanced features. In [31], a hybrid deep network is used to construct an IDS that uses a recurrent neural network (RNN) and the Long Short-Term Memory (LSTM) architecture. The KDD Cup 1999 dataset was used to train this model, and it produced results for the categorization of assaults with a tolerable level of accuracy. An auto-encoder network is created and integrated as a module on the POX controller in [32] to identify DDoS assaults.

The scientists opted to train the network using actual network traffic, as opposed to other experiments that use predefined datasets. The outcomes demonstrate that this strategy outperforms a shallow neural network in terms of accuracy and detection rate. To produce SDN flow rules and anomaly detection in [33], the RNN is combined with the NSL-KDD dataset. Additionally, a unique dataset called CAIDA has been employed in [34] for DDoS detection. Similar strategies were used in [35–37]. In addition, the Generative Adversarial Network (GAN), another deep learning technique, is employed in [38] to distinguish between unauthorized and fraudulent WLAN transmissions. Although these studies show that machine learning approaches are excellent at detecting intrusions, they mostly focus on centralized, classical learning techniques, which raise serious privacy and communication overhead issues. Our research departs from these earlier investigations by using a Federated Learning (FL)-based methodology. With the proposed FL-based SCNN-Bi-LSTM model, the training process is distributed over several sensor nodes without compromising data privacy, and communication costs are minimized. By using deep learning approaches to successfully identify sophisticated and previously unknown cyber threats, the model also improves detection performance. Thus, our method not only gets over some of the drawbacks present in these prior efforts but also provides a more effective and private intrusion detection method.

The FL in the spectrum of intrusion detection systems (IDS) has not been investigated thoroughly. For instance, one research [1] suggested utilizing stacked unsupervised FL in a generalized cross-silo arrangement for a flow-based network intrusion detection system. This technique, which demonstrates effectiveness even in non-IID data silos, combines ensemble learning with a deep autoencoder and an energy flow classifier. Another study [2] proposed a cooperative method for exchanging cyber threat intelligence that makes use of FL to let several organizations jointly develop, train, and test a dependable ML-based network intrusion detection system. By enabling them to benefit from the experience of others while protecting the privacy of their data, this initiative considerably helps each organization. While novel, this research has not completely realized the promise of fusing FL with complex deep learning systems. By suggesting a Federated Learning-based Stacked Convolutional Neural Network with Bidirectional Long Short-Term Memory (FL-SCNN-Bi-LSTM), our study distinguishes itself in this regard. In addition to protecting data privacy and minimizing communication costs, this model also captures geographical and temporal interdependence in network patterns, making it possible to quickly identify sophisticated and previously unknown cyber threats. Additionally, when compared to conventional Artificial Deep Neural Network (ADNN) models and even the FL-based IDS suggested in [1,2], our proposed model has significantly increased accuracy for complex and unidentified attacks, demonstrating the effectiveness of our method for enhancing IDS performance. Table 1 shows the comparative study of what our proposed model offers in terms of evaluation metrics against recent papers proposed in the literature.

1.2. Contributions and paper organization

To improve the model's global weights, the model uses a hybrid deep neural network that combines the Stacked Convolutional Neural Network (SCNN-1D) algorithm and one of the improvised versions of recurrent neural network (Bi-LSTM). Federated learning is then used to prevent data breaches, which is a major concern in the modern world. The contributions of this work are listed below.

- Development of a Novel Hybrid Deep Learning Model: Introduce a groundbreaking FL-SCNN-Bi-LSTM model that ingeniously combines Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (Bi-LSTM), and LSTM networks. This model represents a significant leap in predictive analytics for cybersecurity, enhancing the capability to accurately identify diverse network intrusions.
- Innovative Approach to Data Privacy in IDS: Propose a federated learning framework for the FL-SCNN-Bi-LSTM model, enabling cooperative model training among nodes without the exchange

Table 1
Comparative Analysis of Different Techniques on Used Datasets.

Ref.	Year	Technique	Summary	Results	Privacy preserved	Centralized
[39]	2022	EECA-LSTM	The work combines empirical mode decomposition and principal component analysis to provide an improved empirical component analysis for feature selection. Utilizing LSTM, the chosen properties help categorize attack nodes.	Accuracy: 0.996 Recall: 0.996 Precision: 0.996 F1-Score: 0.995	✓	×
[40]	2022	LSTM-KPCA	In this study, an end-to-end model for network attack detection and classification based on recurrent deep learning models is proposed	Accuracy: 0.98 Precision: 0.91 Recall: 0.84	×	×
[41]	2023	Stacked unsupervised FL using deep encoder	In a cross-silo context, the study presents a versatile approach using stacked unsupervised federated learning for flow-based NIDS, incorporating a deep autoencoder, energy flow classifier, and ensemble learning for effective intrusion detection	Accuracy: 0.98 Recall: 0.88 Precision: 0.91 F1-Score: 0.90	✓	×
[42]	2024	FL-MA	This research significantly advances the current state of the art in IoT and WSN security by synergistically harnessing the potential of machine learning and the Firefly Algorithm.	Accuracy: 0.992 Recall: 0.981 Precision: 0.995 F1-Score: 0.962	×	✓
Our Model	2024	FL-SCNN-Bi-LSTM	Federated Learning assisted Hybrid Stacked CNN model with Bi-directional LSTM is proposed	Accuracy: 0.999 Recall: 0.999 Precision: 0.999 F1-Score: 0.999	✓	✓

of sensitive raw data. This methodology marks a transformative step in data privacy protection for Intrusion Detection Systems (IDS), particularly vital in scenarios requiring high levels of data confidentiality.

- Comprehensive Evaluation and Validation of the Model: Perform an extensive evaluation of the FL-SCNN-Bi-LSTM model using a variety of structured datasets relevant to intrusion detection systems. This rigorous testing, which assesses crucial performance indicators such as recall, accuracy, precision, and F1 score, validates the model's robustness and effectiveness in detecting a wide range of network intrusions.

Six subsections make up the remaining part of this document. In the first section, analytic and federated-based-DL approaches are used to assess existing methods for creating network intrusion detection models. Section 2 goes into further detail on the approaches' structure and the incorporation of a hybrid DL model with privacy protection based on federated learning. The properties of the input datasets utilized in this analysis are outlined in Section 3 in brief. The findings produced on datasets using the suggested approach are evaluated in Section 4. The experimental technique, as well as the CNN and Bi-LSTM structure training utilizing federated learning. In Section 5, datasets employing various methodologies and the suggested technique are compared.

2. Proposed method description

The overall design of the suggested system is thoroughly explained in this section. Data normalization has been performed on the supplied Intrusion datasets during preprocessing. The Min–Max Scaler is used to analyze the normalized data values in both datasets and calculate the coefficient.

With the benefits of Principal Component Analysis (PCA) and the empirical mode decomposition technique, this framework efficiently selects pertinent features in the CIC-IDS 2017 dataset, used for classifying different DDoS attacks. Initially containing 83 columns, PCA reduces this to 43 features. PCA achieves this by creating a set of principal components, which are eigenvector pairs, reducing the dimensionality necessary for classifying incoming data. In the formulation of the FL-SCNN-Bi-LSTM model for WSN intrusion detection, significant emphasis was placed on data pre-processing and feature selection. This process began with an in-depth examination of the dataset, identifying attributes indicative of DoS attack behaviors, such as irregular traffic volumes, abnormal packet sizes, and unusual transmission frequencies. PCA was strategically employed to distill the dataset to its most informative components, enhancing the focus on salient features for intrusion detection. This reduction in data complexity permitted quicker processing times and streamlined analysis, enhancing the model's computational efficiency and accuracy in threat identification. Post-PCA feature selection was pivotal in refining the model's detection and categorization of DoS attacks. By concentrating on the most informative data aspects, the model became sensitive to subtle variations and anomalies characteristic of sophisticated DoS attacks, improving its discernment between normal network behavior and potential intrusions. This heightened sensitivity was crucial in reducing false negatives and enhancing detection reliability. To sum up, the thorough method of feature selection and data pre-processing – particularly using PCA – was crucial in improving the FL-SCNN-Bi-LSTM model's capacity to identify DoS assaults in WSNs. This procedure enhanced the real-world cybersecurity context's accuracy, efficiency, and dependability while also making a substantial contribution to the development of intrusion detection techniques in WSNs [43]. After choosing pertinent attributes, the suggested system was put through the proposed Model (FL-SCNN-Bi-LSTM) classification of the attack nodes. Long-term sequence dependencies might be modeled by Bi-LSTM since it supports processing even minor data without classification errors. Additionally,

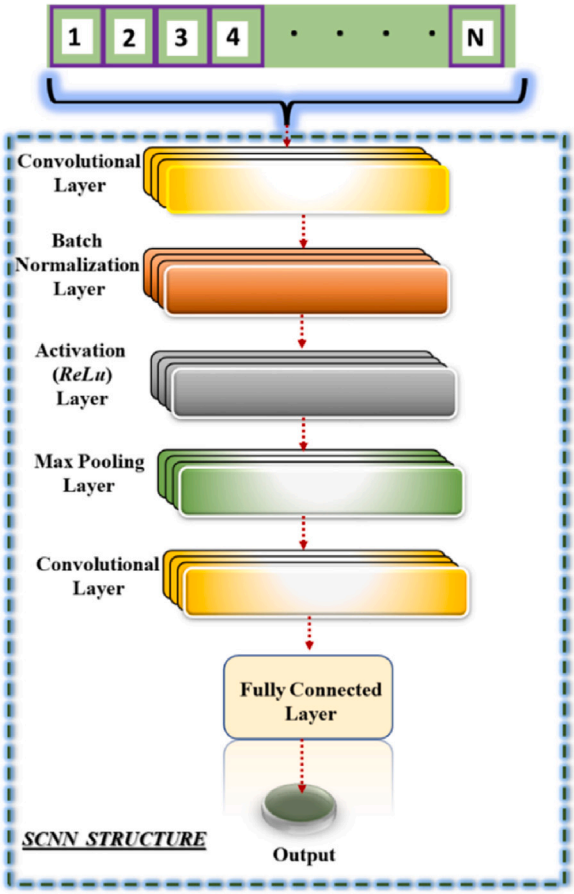


Fig. 2. Basic structure of stacked CNN.

Bi-LSTMs are capable of operating across a wide variety of variables, including input-gate bias, output-gate bias, and learning rate. As a result, the suggested system can identify attacks more effectively because of its efficient architecture.

2.1. Stacked-convolutional neural network (S-CNN)

A modified form of 2D CNNs called 1D-Convolutional Neural Networks (1D-CNNs) was developed recently [44]. The one-dimensional (1-D CNN), which is often an artificial neural feed-forward network made up of convolution layers and pooling layers, is one of the well-known deep learning techniques. The basic architecture of CNN is given in Fig. 2.

It is usual practice to process the input through layers, neurons, and activation processes using the Rectified Linear Unit (ReLU). Nonlinearity is typically added using this form of activation. Overfitting is typically avoided by using dropout layers and normalization (Scaling Data) approaches. Under equal conditions (same design, network, and hyperparameters), a 1D CNN is significantly less computationally challenging than a 2D CNN. The majority of 1D CNN applications, which often use tiny topologies, generally use networks with 50 neurons or less (with 1–2 hidden CNN layers), according to more recent studies. Due to its lower progressing requirements, the miniature form of 1D CNNs is especially well suited for low-cost and real-time applications [45], especially on portable or mobile computers. For applications like patient ECG, civil, time-series forecasting, high-power circuits, power engines or motors, etc. that had weak labeling and strong signal fluctuations, compact 1D CNNs fared better in the specialized study. The primary distinction between 1D-CNN and 2D-CNN is that the latter model uses 1D arrays as input vectors rather than the matrices that are usually used in 2D-CNNs. The basic mathematical equations for working CNN are given:

$$x_{o,fl}^{l} = f\left(\sum_{im} x_{i}^{l-1} * k_{io,fl}^{l} + y^{l}\right) \tag{1}$$

In Eq. (1), $x_{o,fl}^{l}$ represents the output of the lth layer, where l is the layer index. The function $f(.)$ is an activation function applied to the sum of the convolution of input x_{i}^{l-1} with the kernel $k_{io,fl}^{l}$, and the bias term y^{l}.

$$x_{o}^{l} = f\left[\max\left(\sum_{im} x_{i}^{l-1}\right) + y^{l}\right] \tag{2}$$

Eq. (2) describes the max pooling operation in the lth layer. Here, x_{o}^{l} is the output after applying the max pooling to the sum of the inputs from the previous layer $(l-1)$, followed by the addition of the bias term y^{l} and the application of the activation function $f(.)$.

$$x_{o}^{l} = f\left(x_{i}^{l-1} * z_{io}^{l} + y^{l}\right) \tag{3}$$

Eq. (3) represents a convolution operation in the lth layer. The output x_{o}^{l} is obtained by convolving the input from the $(l-1)$th layer, x_{i}^{l-1}, with the kernel z_{io}^{l}, adding the bias y^{l}, and applying the activation function $f(.)$.

In this framework, the number of filters F in each layer, and the parameters y and z can be learned. The use of 1D convolutions in CNNs allows for linear weighted summing of 1D arrays, leading to computational efficiency. These operations are carried out concurrently during both the forward and backpropagation processes, contributing to the model's overall performance in terms of both accuracy and computational cost.

2.2. Bidirectional long short-term memory (Bi-LSTM)

The return loop of the RNN model, which is used to analyze time series data, allows it to make good use of prior information. RNN, however, has storage and information limits. Because it is ineffective at learning long-term dependencies, the gradient disappears [45]. To address the shortcomings of the RNN, the LSTM was developed. The memory cells used to retain long-term historical data and the gate mechanism that controls it form the basis of the LSTM structure, input-gate i_t, forget-gate f_n, and output-gate o_n are the three types of gates found in a typical LSTM unit. In Fig. 3, these three gates are depicted.

In LSTM networks, each gate controls the state of the memory cells by conducting operations involving point-wise multiplication and sigmoid functions. The gates process the input data x_n at the current state and the output data h_{n-1} from the concealed state of the preceding layer. The 'forget gate' decides what information should be retained or discarded. It employs a sigmoid function to analyze both the current input x_n and the previously concealed state h_{n-1}. The output of the forget gate, denoted as f_n, varies between zero and one. A value close to zero implies that the data will be discarded, while a value near one indicates that the data will be retained. This mechanism is expressed in the following equation:

$$f_n = \sigma\left(W^{f} \cdot \left[h_{n-1}, x_n\right] + b_f\right) \tag{4}$$

Here, W^{f} represents the weight, and b_f the bias associated with the forget gate, while σ denotes the sigmoid activation function.

Next, the 'input gate' determines which new information needs to be added to the cell state. It also uses the sigmoid function, resulting in values ranging from zero (unimportant) to one (important). The input gate's operation is described as:

$$i_n = \sigma\left(W_i \cdot \left[h_{n-1}, x_n\right] + b_i\right) \tag{5}$$

The tanh function then processes the current input x_n and hidden state h_{n-1}, creating a vector of new candidate values, $\overline{C_n}$, which could be added to the state:

$$\overline{C_n} = \tanh\left(W_c \cdot \left[h_{n-1}, x_n\right] + b_c\right) \tag{6}$$

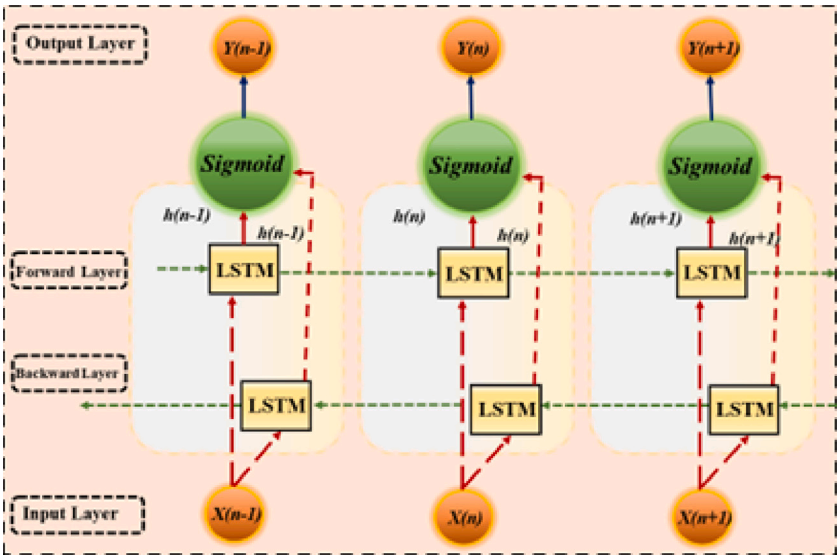


Fig. 3. Basic structure of Bi-LSTM.

The cell state C_n is updated by multiplying the old state C_{n-1} by f_n, and then adding the product of i_n and $\overline{C_n}$:

$$C_n = \left(f_n \odot C_{n-1}\right) + \left(i_n \odot \overline{C_n}\right) \tag{7}$$

In this equation, $\odot$ signifies element-wise multiplication and *tanh* is the hyperbolic tangent activation function.

Finally, the 'output gate' decides what the next hidden state h_n should be. It considers both the cell state and the output from the previous layer:

$$o_n = \sigma\left(W_o \cdot \left[h_{n-1}, x_n\right] + b_o\right) \tag{8}$$

$$h_n = o_n \odot \tanh\left(C_n\right) \tag{9}$$

In these final equations, W_o and b_o are the weights and bias for the output gate. The output gate uses the sigmoid function σ and the tanh function to calculate the final output h_n, which is the next hidden state.

In a nutshell, a standard LSTM network can only utilize the information it has already encountered in the sequence. In contrast, the Bi-LSTM (Bidirectional Long Short-Term Memory) architecture incorporates two LSTM layers—one processing the input sequence forwards (forward LSTM) and the other processing it backward (backward LSTM). The schematic diagram of Bi-LSTM is illustrated in Fig. 3. The forward LSTM layer captures information from the past to the current time step, while the backward LSTM layer gathers information from the future to the current time step. Subsequently, the outputs from both hidden layers at each time step are combined. This combination ensures that the hidden state h_n at any given time n in the Bi-LSTM network encompasses information from both the forward and backward directions. Symbolically, this is denoted by the sum of the output components from both these directions. Owing to its ability to assimilate information from both past and future contexts, the Bi-LSTM architecture is more effective than traditional LSTM and RNN (Recurrent Neural Network) models, particularly in tasks where the understanding of context in both directions is crucial.

2.3. Stacked convolutional neural network with bidirectional long short-term memory (SCNN-Bi-LSTM)

When dealing with large-scale, highly fluctuating data, the Stacked Convolutional Neural Network (SCNN) serves as an effective tool for extracting high-level characteristics from the input sequence. The SCNN excels in identifying complex patterns within the data, thanks to the hierarchical representation it generates. Each successive layer in the SCNN builds upon the features discovered by its predecessor, thereby enhancing the overall capability of the model to discern intricate intrusion patterns. Complementing the SCNN's prowess in spatial feature extraction is the Bidirectional Long Short-Term Memory (Bi-LSTM) component. Bi-LSTM captures temporal dependencies in both forward and backward directions through two distinct LSTM mechanisms: one processes the data sequence from start to end, and the other from end to start. This bidirectional processing grants each data point a comprehensive context, enabling the model to recognize the full spectrum of temporal patterns within the data. Such capability is particularly advantageous for detecting progressively evolving intrusions, like port scanning attacks. The synergy between the SCNN and Bi-LSTM components culminates in a powerful intrusion detection model. This model is adept at identifying both temporal and spatial patterns of intrusion, making it an ideal solution for Intrusion Detection Systems (IDS) in IoT networks. Given IoT networks' often constrained computational resources, the SCNN-Bi-LSTM model stands out for its thorough feature analysis and computational efficiency. The CNN-BiLSTM model's architecture is outlined in Fig. 4.

2.4. Hyperparameters of SCNN-BiLSTM

The performance of deep neural networks, like the stacked CNN and Bi-LSTM structure mentioned, is greatly influenced by their hyperparameters. The number of filters in each convolutional layer, their size, their stride, their padding, and the learning rate are all hyperparameters in this model that may be adjusted. For the model to perform at its peak, these hyperparameters must be tuned successfully.

In the formulation of the FL-SCNN-Bi-LSTM model for enhancing intrusion detection in Wireless Sensor Networks (WSNs), the selection and fine-tuning of hyperparameters were approached with a meticulous strategy, acknowledging their profound impact on the model's efficacy and efficiency. The hyperparameters, including the learning rate, number of layers and neurons, activation functions, dropout rate, and batch size, were carefully chosen based on empirical evidence and deep learning expertise. For instance, the learning rate was calibrated to a moderate level to balance rapid convergence with accuracy. The architecture of the model was configured with an optimized number of layers and neurons to effectively capture complex intrusion patterns without falling into the trap of overfitting.

The rationale behind these selections hinged on achieving a harmonious balance between the model's complexity and its ability to

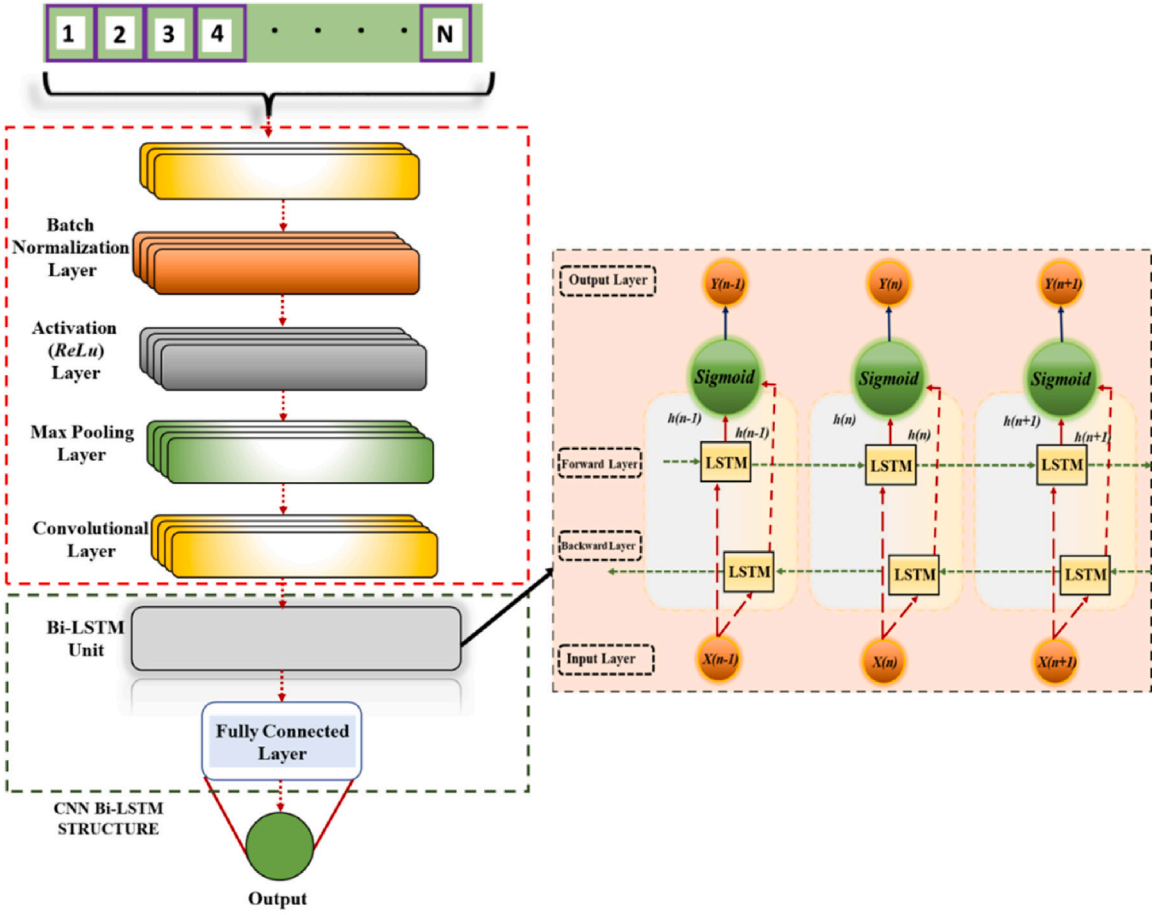


Fig. 4. CNN-Bi-LSTM basic architecture.

generalize across various scenarios. Activation functions like ReLU were chosen for their proven efficiency in deep learning contexts, while the dropout rate was fine-tuned to mitigate overfitting by intermittently deactivating neurons during the training phase.

These hyperparameters played a pivotal role in influencing the overall performance of the model. For example, the learning rate and batch size significantly affected the speed and stability of the training process, while structural parameters such as the number of layers and neurons directly impacted the model's learning capacity and generalizability. The versatility of these hyperparameters allowed the model to adeptly adapt to different operational environments within WSNs.

After extensive testing, we selected a filter size of 3, 64 filters per convolutional layer, "same" padding, 200 units for the Bi-LSTM's hidden layers, and a learning rate of 0.01. This configuration demonstrated the best balance between detecting intricate attack patterns and avoiding overfitting, ensuring the model's effective generalization to new data and optimizing detection performance on our validation set.

This strategic approach ensured that the model was not only equipped to handle standard detection challenges but was also capable of adapting to a range of scenarios within the evolving domain of network security, thereby solidifying its position as a significant contribution to the field of intrusion detection in WSNs. The architecture of the proposed model and a relative number of parameters used for both datasets are given in Table 2:

2.5. Federated learning

The availability of sufficient training data is a crucial need for Deep Neural Network (DNN) models to function superbly. By sending data from dispersed devices to a centralized server, this data is often used to build a global model. Exchanging data across many places and organizations, however, can be challenging, if not impossible, due to worries about data protection. Making efficient models with multi-party data while protecting data privacy is made more difficult. Federated Learning (FL) has been a possible remedy to these privacy problems in recent years. In 2016, FL was initially proposed by McMahan et al. [46]. In essence, FL employs a distributed learning approach to enable team training across many devices while reducing the danger of data leaking. Due to the development of edge computing, edge servers now have the processing capacity to perform extra computing activities, resulting in a setting that is intrinsically FL-friendly. The FL task eliminates the need to collect significant amounts of raw data because each participant trains their local model individually using local data. To a central server are just the model weights supplied. A global model is finally produced after several rounds, removing any possible privacy concerns. To decrease communication rounds, FedAvg, the most used FL optimization strategy, requires the client to perform multiple local epochs before talking with the central server. The submitted weights are combined using FedAvg. FedAvg has been further enhanced by several research [47–49]. Implementing the FedAvg-based synchronous technique is difficult, nevertheless, since edge nodes may switch between servers [50,51], and edge servers may give up on the training assignment at any time owing to network problems and other difficulties. Asynchronous aggregation systems, where the central server can update the global model before all clients have finished their local compute duties, have been the subject of certain investigations [52–55]. Additionally, the properties and volume of data may differ dramatically between devices due to the inherent statistical

Table 2
Parameters' information used for Proposed Model (SCNN-Bi-LSTM).

Model name	No. of parameters	No. of layers	No. of units/neurons	No. of filters	Activation
CNN-1D	3	3	Layer 1: 200 Layer 2: 100 Layer 3: 50	2	ReLU
Bi-LSTM	2	3	Layer 1: 200 Layer 2: 100 Layer 3: 50	N/A	ReLU
MaxPooling Layer	1	Each for CNN-1D layer	N/A	2	N/A

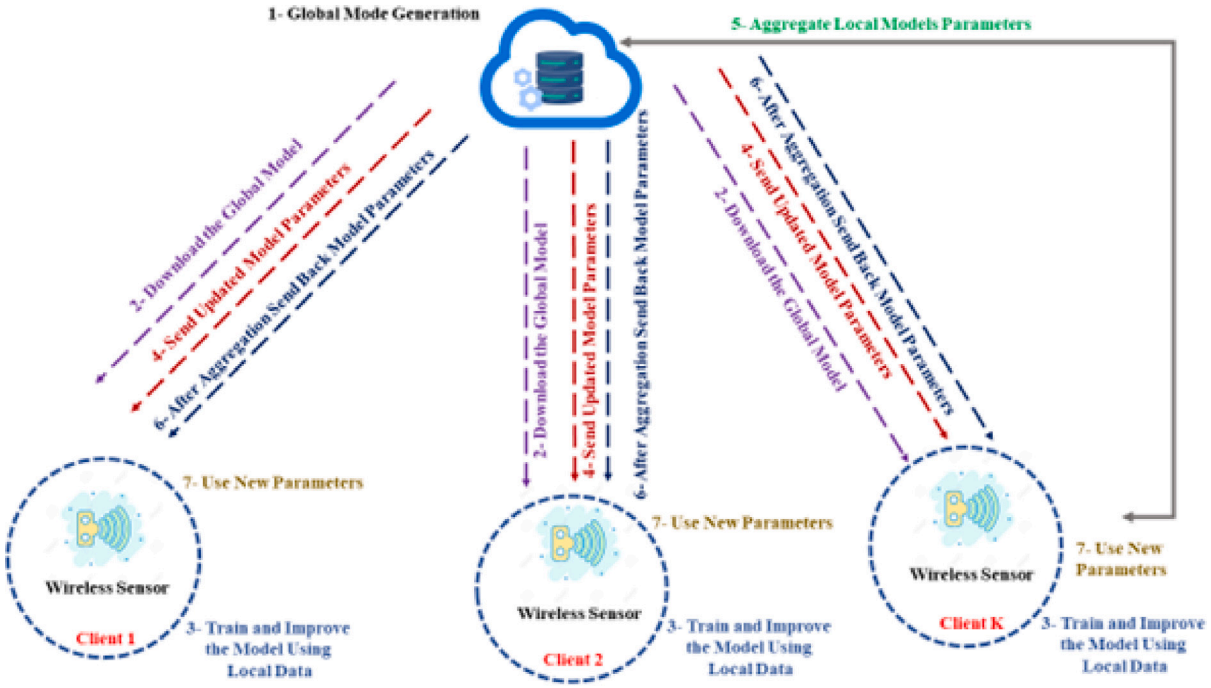


Fig. 5. Overview of the FL-based model for network intrusion detection.

heterogeneity of data. It is possible that not all participants' data will match the common global model as a consequence. In the field of FL research, personalization techniques, which adjust the global model to match particular data, have become more significant. The majority of customization techniques [56] modify the global model using client-specific local data. Transfer learning is a prime illustration [57]. Multi-task learning was used to FL by Smith et al. [58], where the model is created through a succession of linked tasks. Hanzely et al. suggested a gradient descent variation to integrate the local model with the global model [59]. Fig. 5 illustrates the basic idea of the FL-based IDS.

2.6. Federated learning-based stacked convolutional neural network with bidirectional long short-term memory (FL-SCNN-Bi-LSTM)

When trained on inadequate data, deep neural networks (DNNs), like our SCNN-Bi-LSTM model, run the danger of overfitting. Despite having achieved low error rates during training, this overfitting might cause the model to perform poorly on unobserved data. The traditional approach to this issue is to gather data from many sources and transfer it to a central server for model training. This strategy might, however, lead to significant communication overhead and privacy issues. By dispersing the training process over many devices, each of which holds a piece of the total data, federated learning (FL) addresses these problems. This strategy offers the following benefits:

- Reducing privacy hazards and potential legal difficulties by eliminating the need to upload local data to a main server.
- Since they can learn from data spread over several devices, models taught through FL often perform better than local models.
- The model is trained cooperatively by several devices, maximizing the effectiveness of computing resources.

The cloud server chooses random clients to train models locally through several transactions. Each chosen client receives the most recent model from the server and updates locally using local information. The users then input their model parameters, which represent the deviation between the starting parameters and the final values following training. The server averages their contributions before adding them to the global model.

We use an FL framework in the context of our study to improve the performance of network intrusion categorization utilizing information from numerous nodes. Our system's fundamental structure consists of one server and k clients $(C_1 ..., C_k)$ with limited computing capability as can be seen in Fig. 5. All clients and the server share the same global neural network architecture and learning objectives, and each client keeps a local model that can be trained using their unique input. The server chooses a subset of clients and provides them with the most current global model parameters at the beginning of each training round. The overview of the proposed hybrid model using the FL technique is given in Fig. 6.

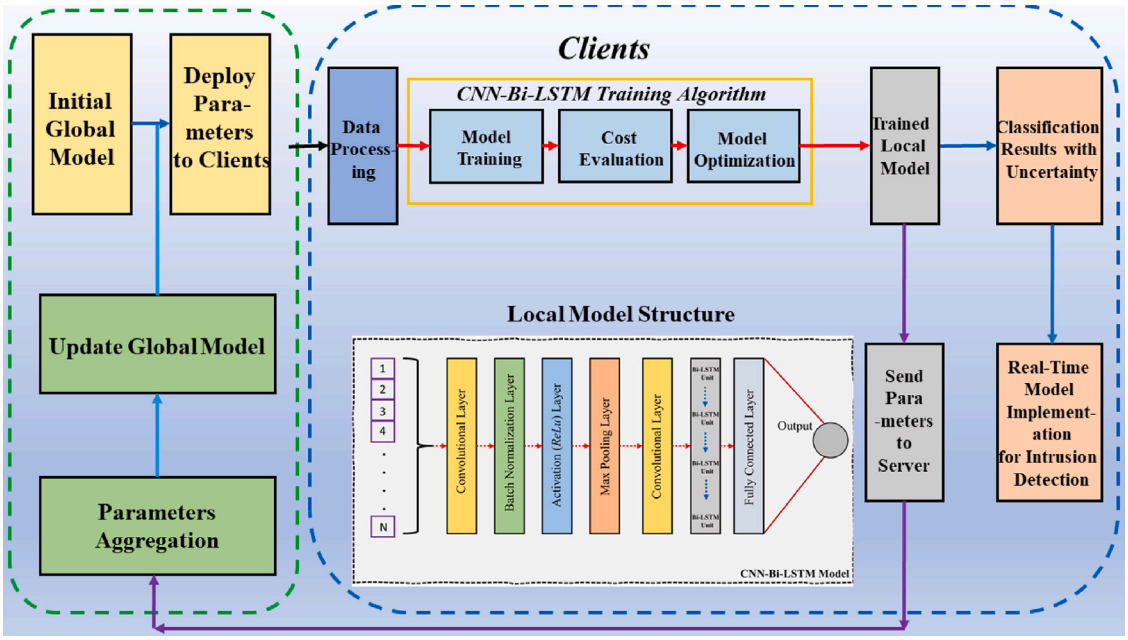


Fig. 6. Overall procedure of the federated SCNN-Bi-LSTM based network intrusion detection scheme.

- The global model parameters are downloaded from the server to each chosen client, who then locally trains their model with their data.
- Following training, each client computes and communicates the modifications to their model to the server.
- The changes are gathered by the server, averaged, and added to the overall model.
- The global model is iterated over numerous rounds, with each round possibly containing a different subset of clients, until it converges to the ideal state.

Wireless communication is used between the clients and the server. In the simplified system, we assume that the kinds and quantities of data collected for each client are different. Each client's computing power is not comparable and each client's communication delay with the server is variable and clients will cease to function during the training process. Also, it is considered that the global model and local model are updated at the server end and client end, respectively. All or a portion of the clients are chosen at the beginning of the training phase, and the most recent global model parameters are distributed to the clients. With gathered local data D_k, C_k does optimization over many iterations, such as adaptive moment estimation (Adam). Updated local model parameters include:

$$a_{t+1}^{k} \leftarrow a_t - \eta \nabla \mathcal{L}\left(a_t\right) \tag{10}$$

where $\eta \nabla \mathcal{L}\left(a_t\right)$ represents the batch gradient and η represents the learning rate. Then updates are transmitted to the server, where safe aggregation is carried out:

$$a_{t+1} \leftarrow w_t - \sum_{k=1}^{K} \frac{n_k}{n} a_{t+1}^{k} \tag{11}$$

where $n_k = \left|D_k\right|, n = \left|D_1 \cup \cdots \cup D_k\right|$. The technique is then done once more. The data that has been gathered from the various sensor nodes is pre-processed and distributed to each client manually. When the global model is integrated with the obtained model updates, a new global model is produced and made available to source clients. Also can be seen in Algorithm 1:

The suggested SCNN-Bi-LSTM model is used to show the general federated learning process using pseudocode. The server initializes the global model parameters to start the process. After startup, federated learning starts in a new cycle. The server chooses a subset of clients to take part in the training process inside each cycle. The most recent global model parameters are subsequently downloaded by these chosen clients from the server. By using these parameters and training the SCNN-Bi-LSTM model on their local data, each client then updates the parameters of their unique local models. Each client calculates the modifications made to their local model parameters, denoted by θ_k, when training is complete. Each client then transmits these calculated changes, which constitute model updates, to the server. The server calculates an average of all the updates after receiving them from all of the participating clients. This averaged update () contains the cumulative learning that was attained across all customers. The global model parameters are subsequently modified by the server using this averaged update, thereby incorporating the dispersed learning. With this update, a learning round concludes, and the server moves on to the following round of federated learning. With each round possibly including a new selection of clients, this procedure is repeated repeatedly. The model has successfully learned the underlying patterns in the dispersed data when the global model parameters converge to an optimal state, which occurs after repeated training. The final global model parameters are returned by the server when training is complete, and these parameters can be utilized to make predictions or conduct more research.

Algorithm 1 Federated Learning based SCNN-Bi-LSTM (FL-SCNN-Bi-LSTM)

```
Procedure FL-SCNN-Bi-LSTM (Server, Clients)
Initialize global model parameters θ
for each round t do
  Select a subset of clients C_t to participate in training
  for each client k in C_t in parallel do
    Retrieve global model parameters θ from server
    Update local model parameters by training on local data with SCNN-Bi-LSTM
    Compute and send model updates Δθ_t to the server
  end for
  Compute global model update Δθ = average(Δθ_t for each k in C_t)
  Update global model parameters θ = θ + Δθ
end for
return Global model parameters θ
End Procedure
```

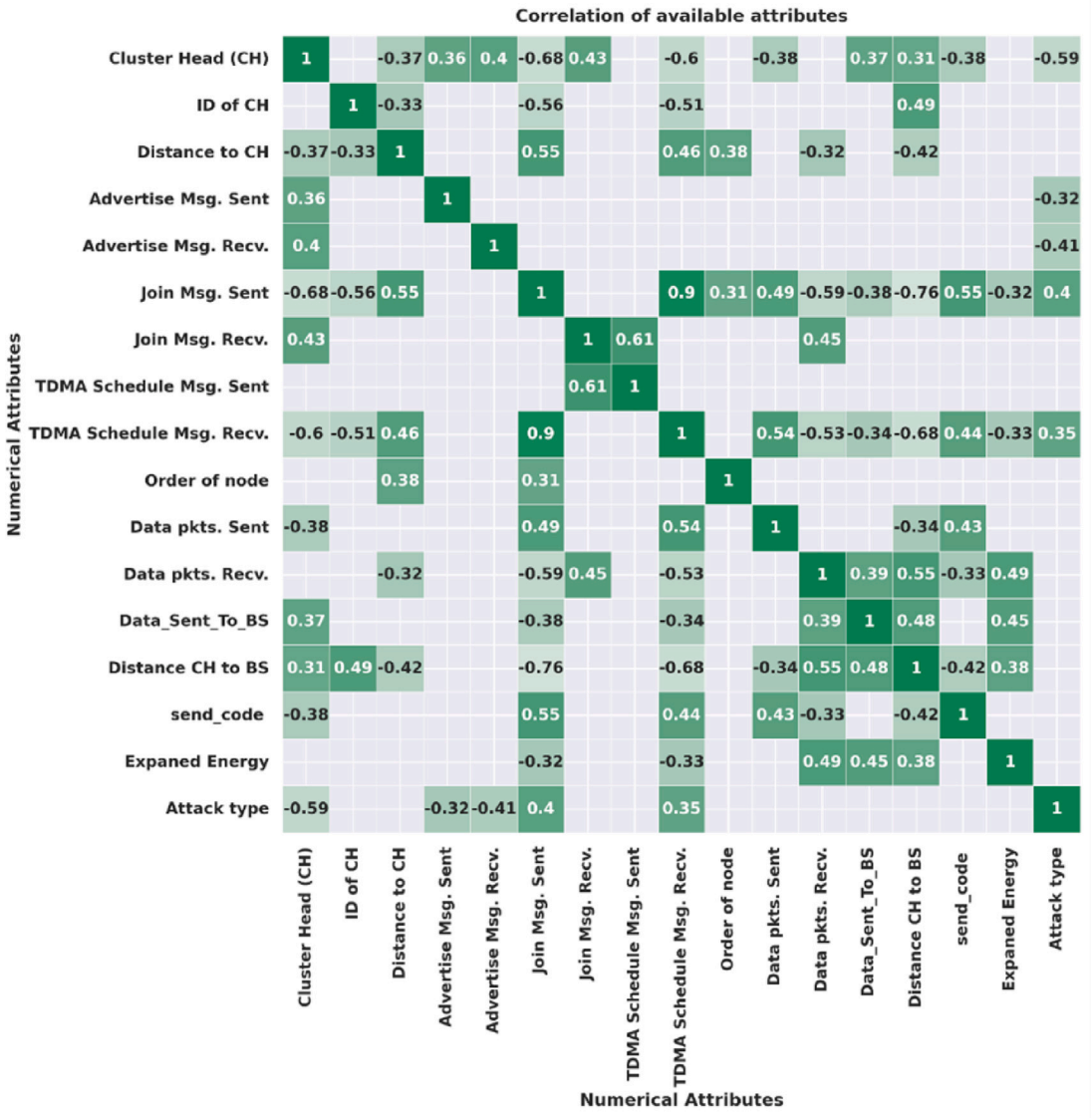


Fig. 7. Correlation matrix for Dataset 1.

Given the statistical heterogeneity of the data across many devices, we must take into account the possibility that the classic FL shared global model may not fit all datasets. The centralized technique of combining all the data into one model is likewise impacted by this unpredictability. We have used a personalizing technique to address this problem, dividing the model into local and global layers, and only utilizing the global layers for FL. Then, identical global layers and distinctive customization layers are made available to each FL member. With its capacity to effectively manage highly dimensional training data and recognize both well-known and novel cyberattacks, the proposed FL-SCNN-Bi-LSTM model offers a substantial advancement in intrusion detection approaches.

3. Dataset description and preprocessing

In this work, we used the SCNN-Bi-LSTM model to build a Federated Learning (FL)-based Deep Learning (DL) intrusion detection system utilizing two datasets: WSN-DS (Dataset 1) and CICISD-2017 (Dataset 2). The goal was to assess how different factors affected the effectiveness of intrusion detection systems. With the help of these datasets, we were able to undertake a thorough investigation of network intrusion characteristics and typical attack pathways for network intrusion detection. Our model's implementation of federated learning made it possible to conduct effective, privacy-protecting collaborative training across many devices. To undertake a thorough analysis of network intrusion features and attack vectors that are often applied in network intrusion detection, these two datasets were chosen. The datasets came from a range of suppliers and included several variables that would help our algorithm recognize network intrusions. Because it makes data analysis easier and improves the precision and speed of these algorithms, data pre-processing is a crucial step in the life cycle of Machine Learning (ML) or Deep Learning (DL) algorithms. However, we ran into certain issues with the obtained datasets, such as class imbalances and missing information. We used a variety of pre-processing techniques to solve these problems, making sure that our datasets were correctly cleaned and structured before being fed into the model.

3.1. WSN-DS (dataset 1)

In Dataset 1, there are a total of 19 characteristics, including the target variable (Attack Type). There are a total of 3,74,661 data samples, and none of the characteristic values were missing or null. Evaluation of the interdependence of data features and how every attribute affects the resultant feature are two examples of how feature association may

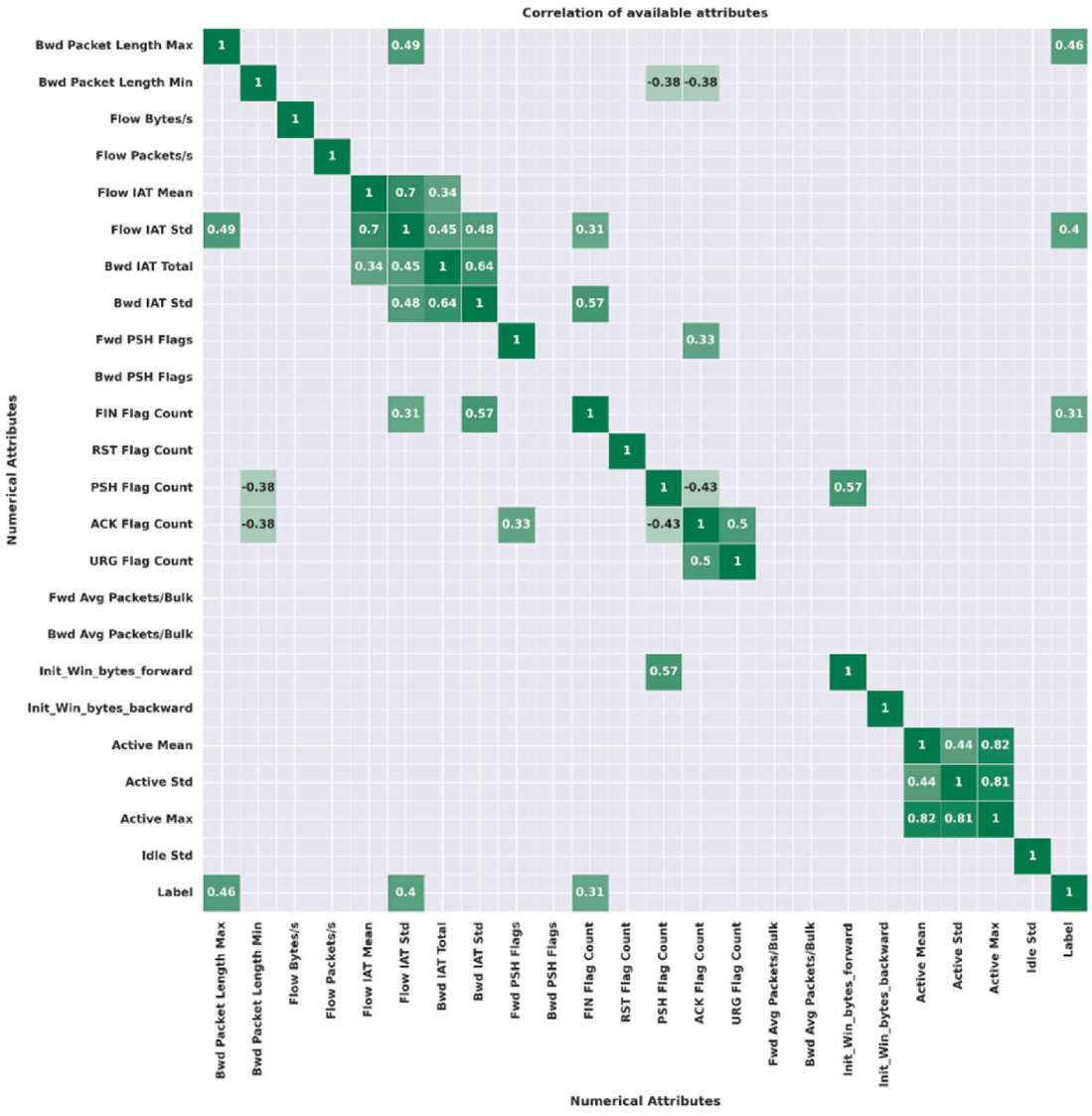


Fig. 8. Correlation matrix for Dataset 2.

be useful. Fig. 7 shows the feature correlation between several characteristics for Dataset 1. Except for "Join Message sent" and "TDMA schedule message received", which show a positive link with the target variable, all of the other characteristics that are accessible in Dataset 1 demonstrate a favorable negative connection with the target variable. A low-cost monitoring service is required to create the dataset and gather the necessary data from the transmitted and received packets within WSN. On the other side, we must make sure that the network-related data that is required to identify, categorize, and subsequently prevent various potential assaults is gathered. Each sensor will participate in the monitoring process in this study and should be able to monitor a set of its neighbors to spread the monitoring workload across the sensor nodes. Finding the right number of nodes for a sensor node to keep an eye on to keep track of all network sensors was a hurdle. The created dataset implements four different DoS attacks against the LEACH protocol: blackhole, grayhole, flooding, and TDMA assaults. In a black hole attack, the attacker influences the LEACH protocol by identifying itself as a cluster head (CH) at the start of the round. Any node that joined this CH during this round will thus forward the data packets to it for transmission to the BS. Assuming the role of the CH, the Blackhole attacker will keep dropping and not sending these data packets to the BS. In a grayhole attack, which is a sort of DoS attack, the attacker interferes with the LEACH protocol by posing as a CH for other nodes. To prevent some packets from reaching the BS, the forged CH drops some packets (selectively or arbitrarily) when it receives data packets from other nodes. A DoS attack known as a flooding assault involves the attacker having several effects on the LEACH protocol. This study simulates flooding assaults by sending a large number of powerful advertising C-H messages (ADV_CH). As a result, sensors will use more energy and take longer to decide which CH to join when they get a lot of ADV_CH notifications. To drain their energy, the perpetrator also tries to deceive others into choosing it as a C-H, particularly those nodes that are located far from it. Scheduling attacks take place when C-Hs create TDMA schedules for the data transmission time slots during the LEACH protocol's setup phase. The attacker that assumes the role of a C-H will provide each node with a certain period for data transmission. This is accomplished by switching the TDMA schedule's behavior from broadcast to unicast. This alteration will result in packet collisions, resulting in data loss. The correlation among data attributes available in Dataset 1 (WSN-DS) is shown in Fig. 7.

3.2. CIC-IDS 2017 (Dataset 2)

There are 24,96,897 data records in CIC-IDS (Dataset 2), and none of them have null values for any of the attributes. Since all characteristics in this dataset have previously undergone a numerical translation,

which is essential for deep learning algorithms, there are no category features to be found. The data instances for Dataset 2 are resampled to a minimum number of data values according to the machine and the target value attains maximum contribution from all attacks to be on the safe side. The link between the characteristics and various qualities is also shown in Fig. 8. Regarding the goal variable "Label", all attributes in Dataset 2 had positive values. Evaluations, however, reveal that many of them lack diversity and traffic volume, are outdated and are unworthy of trust. They also do not cover a lot of the known assaults. To address these issues with the earlier datasets, a Canadian institute developed the CIC-IDS-2017 dataset. It comprises several data formats for assessing anomaly detection techniques. This dataset includes more than 80 features related to the network traffic produced, including all currently used protocols such as TTP, SSH, FTP, TTPS, and Email. Additionally, it includes the most common threats identified by McAfee in 2016. In contrast to NSL-KDD, this dataset assesses network traffic based on variables such as time, both origin and destination addresses, origin and destination ports, and protocol information. The data is quite comparable to real-world data. As stated normalization is performed on feature value values. It provides further information on the characteristics of various assaults, and readers who are interested in specifics are directed to [60] for more data. Additionally, [61] reports the quantity and labels of assaults in this dataset. Fig. 8 shows the correlation matrix that displays data properties with correlations of more than 0.2. The correlation among different attributes available in Dataset 2 is given in Fig. 8.

Normal data is given the first class and the label encoder is used to gradually classify the other attacks, this is how attacks in the WSN-DS dataset are converted to numerical classes. The CIC-IDS-2017 datasets follow the same procedure: the normal data (Benign) is the first class, and the additional classes are numbered in order. In the initial stage, we assess the algorithm's performance and classification during the training phase. This evaluation uses a 20% subset from both datasets, which includes all types of attacks. Run the suggested model on the training set for a predetermined number of iterations. These iterations were chosen from a range of values that were tried during the experiments.

4. Evaluation of proposed methodology on datasets

These metrics, which are widely used in the field of machine learning, are selected to assess the proposed model in this paper:

- True positive: identifying an assault after it has occurred.
- False negative: detecting regular traffic when an actual assault has occurred.
- False positive, often referred to as a false alarm, occurs when an assault is detected when there has not been one.
- True negative: seeing typical traffic when regular traffic has already been present.
- Sensitivity Measure: i.e. anomalies discovered (recall).

To evaluate the expected accuracy of the provided approach, the specificity (Precision), sensitivity (Recall), F1-score (harmonic mean of Precision and Recall), and accuracy score of the focused class are computed. Accuracy, precision, recall, and F-1 measures are calculated in machine learning (ML) and deep learning using the "(TP) True Positive, (TN) True Negative, (FN) False Negative, and (FP) False Positive" rate. The projections are divided into optimistic and pessimistic predictions. To analyze the efficacy of ML/DL classification models, we employed four well-known performance assessment metrics: Accuracy, F1-score, Precision, and Recall. Machine learning professionals may characterize how well a classification model works by looking at a confusion matrix, which is a table. The confusion matrix, which is comprised of the outcomes of four tests, is used to determine a classifier's performance matrices. All above-mentioned metrics can be defined for a given dataset of size n as:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{12}$$

Accuracy: In network intrusion detection, Accuracy represents the proportion of total predictions (both intrusion and non-intrusion) that the FL-SCNN-Bi-LSTM model correctly identifies. High accuracy is crucial as it reflects the model's overall reliability in distinguishing between normal network behavior and potential security threats.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

Precision: Precision in network intrusion detection refers to the model's ability to correctly identify true intrusion attempts. It is the ratio of correctly predicted intrusion events to all events predicted as intrusions. This metric is significant as it reflects the model's effectiveness in minimizing false positives, ensuring normal network operations are not unjustly flagged as intrusions.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

Recall: Recall, or the true positive rate, measures the model's capacity to identify all actual intrusion attempts. A high recall indicates the model's effectiveness at detecting intrusions, minimizing the risk of overlooking genuine threats (false negatives), crucial for network integrity and security.

$$F1\ \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

F1-Score: The F1-Score is a harmonic mean of Precision and Recall, balancing these metrics. It is crucial in network intrusion detection as it encapsulates both precision in identifying true intrusions and the ability to recall all actual intrusion events. A high F1 score indicates a robust defense against network threats.

In Fig. 9, we present a consolidated view of the confusion matrices that delineate the predictive efficacy of our proposed model across two distinct datasets, both under the influence of Federated Learning (FL) and in its absence. For Dataset 1, part (a) of the figure reveals the confusion matrix results when FL is applied, showcasing the model's ability to discern between different classes with a high degree of accuracy. This is reflected in the number of true positives and negatives, as well as the limited instances of false positives and negatives. Conversely, part (c) of the same figure offers a comparative perspective by illustrating the model's performance without the aid of FL, allowing us to evaluate the impact of FL on the model's predictive capabilities. Similarly, for Dataset 2, part (b) of Fig. 9 provides insights into the model's classification prowess when enhanced with FL, indicating the model's robustness in handling new data while maintaining precision. Part (d), in contrast, showcases the model's performance devoid of FL's contribution, which serves as an essential benchmark to gauge the incremental value added by federated learning techniques. Collectively, these matrices serve not only as a testament to the model's overall performance but also underscore the nuanced improvements that federated learning imparts to the system's ability to predict and classify data accurately within the context of intrusion detection systems.

4.1. Evaluation results for WSN-DS (Dataset 1)

This section evaluated each ML/DL model's ability to predict the network intrusion detection outcomes utilizing all available datasets and characteristics. All prediction models were trained on the whole dataset, comprising 80% training and 20% testing subsets. The proposed model (FL-SCNN-Bi-LSTM) for Dataset 1 has the maximum accuracy (0.99%), F1-Score (0.99%), Precision (0.99%), and Recall (0.99%), according to the categorization results shown in Table 3. Other classifiers, including CNN-Bi-LSTM, KNN, SVM, RF (Random Forest), NN (Neural Network), and LightGBM, also performed well and provided high prediction accuracy. The suggested model worked effectively also ensuring data privacy with the provided dataset.

The confusion matrices for the proposed model applied to Dataset 1 are depicted in Fig. 9. Part (a) of the figure shows the results when the FL technique is utilized, while part (c) presents the outcomes without the use of FL. Each matrix provides a detailed account of the model's predictive accuracy by displaying the number of correct predictions and the instances of misclassification.

Fig. 9. Confusion matrices illustrating the performance of the proposed model: (a) with Federated Learning (FL) applied on Dataset 1, (b) with FL applied on Dataset 2, located in the top row from left to right, respectively. The bottom row shows (c) without FL applied on Dataset 1 and (d) without FL applied on Dataset 2, from left to right. Each matrix provides insights into the accuracy and misclassification rates for different configurations.

Table 3
Evaluation results generated on dataset 1.

Metrics	Proposed model with FL (FL-SCNN-Bi-LSTM)	Proposed model without FL (SCNN-Bi-LSTM)	KNN	RF	SVM	NN	LGBM classifier
Accuracy	0.997	0.995	0.984	0.961	0.85	0.89	0.982
F1-Score	0.996	0.994	0.96	0.967	0.83	0.894	0.928
Precision	0.998	0.995	0.964	0.969	0.852	0.89	0.912
Recall	0.996	0.994	0.958	0.961	0.851	0.9	0.95

Table 4
Evaluation results generated on dataset 2.

Metrics	Proposed model with FL (FL-SCNN-Bi-LSTM)	Proposed model without FL (SCNN-Bi-LSTM)	KNN	RF	SVM	NN	LGBM classifier
Accuracy	0.9993	0.9990	0.984	0.961	0.850	0.890	0.982
F1-Score	0.9993	0.9990	0.960	0.967	0.830	0.894	0.928
Precision	0.9993	0.9990	0.964	0.969	0.852	0.890	0.912
Recall	0.9992	0.9991	0.958	0.961	0.851	0.900	0.950

4.2. Evaluation results on CIC-IDS-2017 (Dataset 2)

The suggested model (FL-SCNN-Bi-LSTM) for Dataset 2 achieved the greatest accuracy of 0.999%, F1-Score with 0.999%, precision of 0.999%, and Recall with 0.999%, according to the evaluation criteria used to assess the proposed model's correctness as given in Table 4. On Dataset 2, other classifiers like KNN (K-nearest Neighbors), RF (Random Forest), NN (Neural Network), LightGBM, and SVM (Support Vector Machine) did very well. The suggested model worked quite well on the provided dataset.

The effectiveness of our model on Dataset 2 can also be assessed through the confusion matrices presented in Fig. 9. Part (b) of the figure illustrates the model's performance with the implementation of the FL technique, while part (d) shows the outcomes when the model operates without FL. The confusion matrices detail how accurately the model predicted the target variables, highlighting both correct predictions and misclassifications.

5. Comparative analysis

Comparing our suggested model to other FL-based models in the literature is essential for measuring our results given the surge in the popularity of FL for ensuring data privacy in a variety of applications. Our Federated Learning-based Stacked Convolutional Neural Network

Table 5
Comparative Analysis of different techniques with proposed model.

Dataset	Base proposed technique	Proposed model evaluated metrics (%) without FL	Proposed model evaluated metrics (%) with FL
WSN-DS (Dataset 1)	SCNN-Bi-LSTM	Accuracy = 0.995 F1-Score = 0.994 Precision = 0.995 Recall = 0.994	Accuracy = 0.997 F1-Score = 0.996 Precision = 0.998 Recall = 0.996
CIC-IDS2017 (Dataset 2)	SCNN-Bi-LSTM	Accuracy = 0.9990 F1-Score = 0.9990 Precision = 0.9990 Recall = 0.9991	Accuracy = 0.9993 F1-Score = 0.9993 Precision = 0.9993 Recall = 0.9992

with Bidirectional Long Short-Term Memory (FL-SCNN-Bi-LSTM) architecture shines out even among these sophisticated models. Unlike existing FL-based approaches, our method makes use of deep learning, particularly SCNN and Bi-LSTM, to offer improved detection accuracy. As a result, our technology can better detect network intrusion scenarios while simultaneously ensuring privacy by keeping data on local devices. The comparison between our suggested technique and the previous FL-based investigations is shown in Table 1. Even though FL has improved intrusion detection technologies, our method shows a modest advantage in terms of accuracy across the datasets studied. This improved performance demonstrates the potency of our suggested FL-SCNN-Bi-LSTM model, especially when combined with the strong privacy safeguards offered by our FL architecture. Therefore, our study points towards an approach that is promising for the development of intrusion detection systems that prioritize data privacy and detection performance while integrating FL and deep learning. Also in Table 5, overall relative results are shown together with comparison analysis, both with and without federated learning. We use base models and our recommended FL-based model to analyze the results of the evaluation on two separate datasets.

5.1. Comparative analysis for WSN-DS (Dataset 1)

Excluding Federated Learning (FL), the suggested model's performance on Dataset 1 was characterized by an accuracy of 0.995, a sensitivity of 0.994, a specificity of 0.995, and an F1-Score of 0.994. Interestingly, the implementation of Federated Learning (FL) improved all these parameters. Post-FL implementation, the model demonstrated an accuracy of 0.997, a sensitivity of 0.998, a specificity equal to 0.996, and an F1-Score of 0.996. These variations are discernible in the bar chart shown in Fig. 10(a) for Dataset 1 and in Fig. 10(b) for Dataset 2. The improved outcomes with FL usage imply that this learning strategy may be especially helpful in enhancing the functionality of the proposed model on Dataset 1.

5.2. Comparative analysis for CIC-IDS 2017 (Dataset 2)

On Dataset 2, our suggested model performed impressively. The model achieved an accuracy of 0.9993, a sensitivity of 0.9993, a specificity of 0.9992, and an F1-Score of 0.9993 specifically when using the Federated Learning (FL) approach. The accuracy, sensitivity, specificity, and F1-Score of the model without FL were all marginally lower than those of the model with FL, at 0.9990, 0.9990, 0.9991, and 0.9990, respectively. Despite being small, FL's performance improvement is considerable, demonstrating the benefits of using this cutting-edge machine-learning method. Due to FL's capacity to learn from decentralized data while preserving anonymity, it is feasible that its application helps create a model that is more reliable and efficient. This feature of FL may allow the model to detect a greater variety of intrusion patterns across many networks, enhancing the performance metrics in general. Both when using Federated Learning (FL) and when using the base model without FL, a similar trend in assessment measures can be seen in both datasets. The methodology suggested in this study, which uses several network intrusion datasets, produces an average score for the datasets of 0.99% across all assessment measures. The similarity of the findings achieved using various datasets emphasizes the consistency of the model's performance. Despite the small discrepancies in metrics between FL and non-FL approaches, FL is still the better option due to its added advantages. FL offers a formidable tool to prevent future data breaches – a significant worldwide concern – while enhancing performance and ensuring data privacy.

5.3. Strategies for model scalability and integration with emerging technologies

In the rapidly evolving landscape of network security and machine learning, the scalability of models and their integration with emerging technologies stand as pivotal factors determining their long-term success and applicability. This subsection delves into the various strategic approaches we have employed to ensure that our model not only meets the current demands of cybersecurity but is also primed for future advancements. By exploring both horizontal and vertical scaling methods, as well as the potential for integration with cutting-edge technologies such as IoT and cloud computing, we lay the groundwork for a model that is robust, adaptable, and forward-looking.

- **Vertical Scaling (Enhancing Computational Power):** We propose enhancing the computational power of individual nodes through hardware upgrades, such as faster processors or increased memory capacity. Additionally, software-level optimizations, including model quantization, are used to streamline the computational demands of the model. These optimizations aim to maintain high performance while minimizing the hardware requirements.
- **Distributed Computing Techniques:** For efficient handling of large-scale data, our model utilizes distributed computing frameworks like Apache Spark or Hadoop. These technologies enable the splitting and parallel processing of data across a cluster of machines. We employ data parallelism for distributing data segments across different nodes and model parallelism for dividing the model itself for simultaneous processing on multiple nodes.
- **IoT Integration:** Recognizing the expansive nature of IoT networks, our model is designed to be deployed on IoT devices directly. We focus on creating lightweight versions of the model that are optimized for the limited computational resources typical of IoT devices. Integration with IoT platforms allows for real-time data analysis, enhancing responsiveness and efficiency in dynamic environments.
- **Cloud Computing Integration:** We leverage cloud computing to offload intensive computational tasks, especially for model training and complex data analysis. By utilizing cloud-based services, the model can be scaled dynamically based on the workload, ensuring consistent performance. Additionally, our model is adapted for deployment as a cloud service, facilitating regular updates and maintenance through a centralized platform.

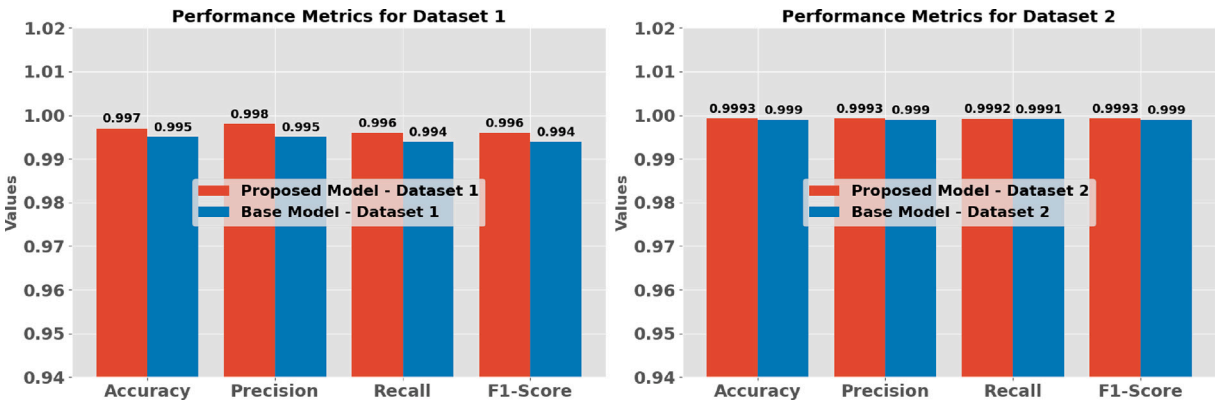


Fig. 10. (a) Comparative analysis for Dataset 1 (b) Comparative analysis for Dataset 2.

- **Use of Containers and Microservices:** To enhance the portability and flexibility of our model, we utilize containerization technologies like Docker. This allows for the deployment of the model across diverse environments with consistent performance. Furthermore, a microservices architecture is adopted, segmenting different functionalities of the model into independently scalable services. This approach ensures that each aspect of the model can be scaled according to its specific demands.
- **Adaptive Algorithms:** In response to the dynamic nature of network environments, our model incorporates adaptive algorithms that can adjust to changes in data patterns and network conditions. Online learning techniques are utilized to continuously update the model with new data, ensuring that the model remains effective against the latest security threats.
- **Horizontal Scaling (Adding Nodes):** To accommodate the growing volume of data in larger networks, our approach involves increasing the number of nodes within the federated learning framework. This strategy allows for the parallel processing of data across multiple devices or servers, enhancing the model's ability to handle extensive datasets without a bottleneck in processing capacity.

5.4. Limitations and challenges

While the FL-SCNN-Bi-LSTM model offers significant advancements in the field of intrusion detection within Wireless Sensor Networks (WSNs), it is imperative to thoroughly acknowledge and understand its potential limitations and the challenges that might arise. These considerations are crucial for a balanced and comprehensive view of the model's applicability and potential areas for future enhancement.

- **Scalability in Large Network Environments:** As the deployment scale of the model increases, particularly in extensive and complex network environments, scalability becomes a critical concern. The model must be capable of handling increased data loads and maintaining performance without incurring prohibitive computational costs. This aspect is particularly challenging in scenarios where network architectures and traffic patterns vary significantly from those in the model's initial testing and configuration.
- **Adaptability to Different Network Types:** The current testing of our model primarily focuses on specific network architectures, which raises questions about its adaptability to a broader range of network environments. Understanding and ensuring the model's effectiveness across various architectures is essential, especially considering the diverse nature of modern network systems.
- **Performance Under Fluctuating Network Conditions:** Network environments are often dynamic, with varying conditions such as traffic fluctuations and congestion. These factors can significantly impact the model's ability to detect intrusions accurately. Comprehensive testing under a variety of operational scenarios is necessary to evaluate the model's robustness and reliability in real-world conditions.
- **Dependency on Data Quality:** The model's performance is heavily reliant on the quality of data used for training and testing. In scenarios where the data is noisy, incomplete, or otherwise compromised, the model's accuracy and effectiveness could be adversely affected. This emphasizes the importance of implementing rigorous data quality management and preprocessing methods.
- **Ability to Detect Evolving Cyber Threats:** Given the rapidly changing landscape of cybersecurity threats, the model's capability to continuously adapt and identify new types of intrusions is of paramount importance. Ongoing research and development efforts are required to ensure the model remains effective against novel and sophisticated cyber threats.
- **Real-time Data Processing Limitations:** In high-traffic scenarios and environments with limited computational resources, such as edge devices in IoT networks, the model faces challenges in processing data in real time. This results in latency issues and necessitates further optimization to ensure timely and accurate intrusion detection.
- **Challenges with Training on Live Data Streams:** Live data streams, characterized by their unpredictability and variability, present significant challenges in maintaining consistent model performance. The model requires continuous updates and adaptations to these evolving data patterns, which can be resource-intensive and require sophisticated algorithms.
- **Strategies for Addressing These Challenges:** To mitigate these limitations, we propose the implementation of algorithmic optimizations and the utilization of cloud and edge computing resources to enhance computational efficiency. Additionally, adaptive learning techniques and real-time data preprocessing methods are suggested to address issues related to data variability and quality. For continuous model updates, online learning and incremental learning techniques are recommended to adapt to new data patterns without the need for complete retraining.

These detailed insights into the limitations and challenges of our model are intended to guide future research and development efforts, enhancing its real-time data processing capabilities and overall effectiveness in diverse and dynamic network environments.

Conclusion

In this article, we propose a classification technique for multiple Intrusion Detection System (IDS) datasets using a stacked CNN and Bi-LSTM hybrid deep learning model based on Federated Learning (FL).

We tested the FL-SCNN-Bi-LSTM algorithm on WSN-DS and CIC-IDS-2017 structured datasets to predict various network attacks. These datasets, sourced from online repositories and real-time sources, were used to analyze the threats and associated risk factors of network intrusion. We employed feature selection and data normalization as pre-processing techniques, and the classifier was then applied to the pre-processed datasets to create the FL-SCNN-Bi-LSTM. To evaluate the models' accuracy, we conducted assessments using efficiency calculations. We compared our proposed methods with popular classifiers including the SCNN-Bi-LSTM base model, SVM, LightGBM Classifier, KNN, and DNN. Our proposed approach consistently delivered superior classification accuracy, averaging 99.97% for Dataset 1 and 99.93% for Dataset 2. When compared to the base model and other classifiers, our approach showed a significant improvement in accuracy, indicating an increase of approximately 2%–3% in some cases. This improvement, although seemingly small, can have a substantial impact in the field of network intrusion detection, where even a slight increase in accuracy can lead to the detection of thousands of additional intrusion attempts. Looking ahead, we recognize that the field of network intrusion detection is rapidly evolving, and our methodology must adapt to stay effective. Future research could explore more advanced feature selection methods to further refine the model's ability to detect complex intrusion patterns. Optimizing the model for efficiency, particularly in real-time detection scenarios, is another vital area of focus. Addressing computational time constraints is essential for the model's performance in environments where quick response is critical. Enhancing data privacy measures during the federated learning process remains a priority, ensuring robust security of data. Additionally, adapting the model for scalability to larger and more complex networks and ensuring its adaptability to new types of network threats are essential for maintaining relevance in the ever-changing landscape of network security. Integrating this model with existing security systems could offer a more comprehensive defense strategy. Extensive real-world testing and deployment will be crucial in assessing the model's efficacy in diverse network environments. The integrity and security of contemporary network systems, as well as the maintenance of strong defenses against complex network attacks, depend on the ongoing improvement and adaption of techniques such as FL-SCNN-Bi-LSTM. In conclusion, even if the current model marks a significant advancement in network intrusion detection, the effectiveness of predictive classifiers will continue to rise with the ongoing development of novel feature selection strategies, optimization approaches, and data privacy technologies. Such continuous improvements will guarantee this method's continued importance in the intrusion detection space, providing a strong and dependable defense against network attacks.

CRediT authorship contribution statement

Syed Muhammad Salman Bukhari: Writing – review & editing, Writing – original draft, Methodology, Data curation, Conceptualization. **Muhammad Hamza Zafar:** Writing – review & editing, Writing – original draft, Methodology, Data curation, Conceptualization. **Mohamad Abou Houran:** Investigation, Formal analysis. **Syed Kumayl Raza Moosavi:** Software, Resources, Investigation. **Majad Mansoor:** Writing – review & editing, Writing – original draft, Visualization, Resources. **Muhammad Muaaz:** Writing – review & editing, Writing – original draft, Formal analysis. **Filippo Sanfilippo:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Investigation, Funding acquisition.

Declaration of competing interest

None. All authors claim that there is not any conflict of interest regarding the above submission. The work of this submission has not been published previously. It is not under consideration for publication elsewhere. Its publication is approved by all authors and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder.

Data availability

The links for the data are included in the paper.

Acknowledgment

This research is supported by Biomechatronics and Collaborative Robotics Group at the Top Research Center Mechatronics (TRCM), University of Agder (UiA), Norway.

References

[1] N. Marriwala, P. Rathee, An approach to increase the wireless sensor network lifetime, in: 2012 World Congress on Information and Communication Technologies, 2012, pp. 495–499, http://dx.doi.org/10.1109/WICT.2012.6409128.

[2] V.C. Gungor, B. Lu, G.P. Hancke, Opportunities and challenges of wireless sensor networks in smart grid, IEEE Trans. Ind. Electron. 57 (10) (2010) 3557–3564.

[3] M.A. Rassam, M. Maarof, A. Zainal, A survey of intrusion detection schemes in wireless sensor networks, Am. J. Appl. Sci. 9 (10) (2012) 1636.

[4] I. Butun, S.D. Morgera, R. Sankar, A survey of intrusion detection systems in wireless sensor networks, IEEE Commun. Surv. Tutor. 16 (1) (2014) 266–282.

[5] H. Modares, R. Salleh, A. Moravejosharieh, Overview of security issues in wireless sensor networks, in: 2011 Third International Conference on Computational Intelligence, Modelling & Simulation, 2011, pp. 308–311, http://dx.doi.org/10.1109/CIMSim.2011.62.

[6] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity 2 (1) (2019) 1–22.

[7] S. Tweneboah-Kodua, F. Atsu, W. Buchanan, Impact of cyberattacks on stock performance: a comparative study, Inf. Comput. Secur. 26 (5) (2018) 637–652.

[8] N. Sun, J. Zhang, P. Rimba, S. Gao, L.Y. Zhang, Y. Xiang, Data-driven cybersecurity incident prediction: A survey, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1744–1772.

[9] I. Almomani, B. Al-Kasasbeh, M. Al-Akhras, et al., WSN-DS: A dataset for intrusion detection systems in wireless sensor networks, J. Sens. 2016 (2016).

[10] N. Farooq, I. Zahoor, S. Mandal, T. Gulzar, Systematic analysis of DoS attacks in wireless sensor networks with wormhole injection, Int. J. Inf. Comput. Technol. 4 (2) (2014) 173–182.

[11] H. Sedjelmaci, S.M. Senouci, N. Ansari, A hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks, IEEE Trans. Syst. Man Cybern. A 48 (9) (2018) 1594–1606.

[12] X. Zuo, Z. Chen, L. Dong, J. Chang, B. Hou, Power information network intrusion detection based on data mining algorithm, J. Supercomput. 76 (7) (2020) 5521–5539.

[13] E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, B. Prieto, PCA filtering and probabilistic SOM for network intrusion detection, Neurocomputing 164 (2015) 71–81.

[14] S. Sheikhi, An efficient method for detection of fake accounts on the instagram platform, Revue d'Intell. Artif. 34 (4) (2020).

[15] S. Sheikhi, An effective fake news detection method using WOA-xgbTree algorithm and content-based features, Appl. Soft Comput. 109 (2021) 107559.

[16] H. Jiang, Z. He, G. Ye, H. Zhang, Network intrusion detection based on PSO-Xgboost model, IEEE Access 8 (2020) 58392–58401.

[17] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, N. Yazdani, Mutual information-based feature selection for intrusion detection systems, J. Netw. Comput. Appl. 34 (4) (2011) 1184–1199, Advanced Topics in Cloud Computing.

[18] G. Singh, N. Khare, A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques, Int. J. Comput. Appl. 44 (7) (2022) 659–669.

[19] J. Ashraf, S. Latif, Handling intrusion and DDoS attacks in software defined networks using machine learning techniques, in: 2014 National Software Engineering Conference, 2014, pp. 55–60, http://dx.doi.org/10.1109/NSEC.2014.6998241.

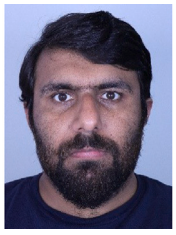
[20] P. Dong, X. Du, H. Zhang, T. Xu, A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows, in: 2016 IEEE International Conference on Communications, ICC, 2016, pp. 1–6, http://dx.doi.org/10.1109/ICC.2016.7510992.

[21] D. Jankowski, M. Amanowicz, On efficiency of selected machine learning algorithms for intrusion detection in software defined networks, Int. J. Electron. Telecommun. 62 (3) (2016).

[22] A. Azab, M. Khasawneh, S. Alrabaee, K.-K.R. Choo, M. Sarsour, Network traffic classification: Techniques, datasets, and challenges, Digit. Commun. Netw. (2022).

[23] R. Braga, E. Mota, A. Passito, Lightweight ddos flooding attack detection using NOX/OpenFlow, in: IEEE Local Computer Network Conference, 2010, pp. 408–415, http://dx.doi.org/10.1109/LCN.2010.5735752.
[24] A. Abubakar, B. Pranggono, Machine learning based intrusion detection system for software defined networks, in: 2017 Seventh International Conference on Emerging Security Technologies, EST, 2017, pp. 138–143, http://dx.doi.org/10.1109/EST.2017.8090413.
[25] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, B. Yang, Predicting network attack patterns in SDN using machine learning approach, in: 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN, 2016, pp. 167–172, http://dx.doi.org/10.1109/NFV-SDN.2016.7919493.
[26] S. Dotcenko, A. Vladyko, I. Letenko, A fuzzy logic-based information security management for software-defined networks, in: 16th International Conference on Advanced Communication Technology, 2014, pp. 167–171, http://dx.doi.org/10.1109/ICACT.2014.6778942.
[27] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, Deep learning approach for network intrusion detection in software defined networking, in: 2016 International Conference on Wireless Networks and Mobile Communications, WINCOM, 2016, pp. 258–263, http://dx.doi.org/10.1109/WINCOM.2016.7777224.
[28] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, R. Atkinson, Shallow and deep networks intrusion detection system: A taxonomy and survey, 2017, arXiv preprint arXiv:1701.02145.
[29] A. Javaid, Q. Niyaz, W. Sun, M. Alam, A deep learning approach for network intrusion detection system, in: Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies, Formerly BIONETICS, 2016, pp. 21–26.
[30] D. Kwon, H. Kim, J. Kim, S.C. Suh, I. Kim, K.J. Kim, A survey of deep learning-based network anomaly detection, Cluster Comput. 22 (2019) 949–961.
[31] J. Kim, J. Kim, H.L. Thi Thu, H. Kim, Long short term memory recurrent neural network classifier for intrusion detection, in: 2016 International Conference on Platform Technology and Service, PlatCon, 2016, pp. 1–5, http://dx.doi.org/10.1109/PlatCon.2016.7456805.
[32] Q. Niyaz, W. Sun, A.Y. Javaid, A deep learning based ddos detection system in software-defined networking (SDN), 2016, arXiv preprint arXiv:1611.07400.
[33] H. Li, F. Wei, H. Hu, Enabling dynamic network access control with anomaly-based IDS and SDN, in: Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, 2019, pp. 13–16.
[34] P. Manso, J. Moura, C. Serrão, SDN-based intrusion detection system for early detection and mitigation of DDoS attacks, Information 10 (3) (2019).
[35] A. Ahmim, L. Maglaras, M.A. Ferrag, M. Derdour, H. Janicke, A novel hierarchical intrusion detection system based on decision tree and rules-based models, in: 2019 15th International Conference on Distributed Computing in Sensor Systems, DCOSS, 2019, pp. 228–233, http://dx.doi.org/10.1109/DCOSS.2019.00059.
[36] M.A. Albahar, Recurrent neural network model based on a new regularization technique for real-time intrusion detection in SDN environments, Secur. Commun. Netw. 2019 (2019) 1–9.
[37] O. Faker, E. Dogdu, Intrusion detection using big data and deep learning techniques, in: Proceedings of the 2019 ACM Southeast Conference, 2019, pp. 86–93.
[38] M. Zhou, Y. Li, H. Yuan, J. Wang, Q. Pu, Indoor WLAN personnel intrusion detection using transfer learning-aided generative adversarial network with light-loaded database, Mob. Netw. Appl. 26 (2021) 1024–1042.
[39] L. Zhiqiang, G. Mohiuddin, Z. Jiangbin, M. Asim, W. Sifei, Intrusion detection in wireless sensor network using enhanced empirical based component analysis, Future Gener. Comput. Syst. 135 (2022) 181–193.
[40] V. Ravi, R. Chaganti, M. Alazab, Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system, Comput. Electr. Eng. 102 (2022) 108156.
[41] G. de Carvalho Bertoli, L.A.P. Junior, O. Saotome, A.L. dos Santos, Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach, Comput. Secur. 127 (2023) 103106.
[42] M. Karthikeyan, D. Manimegalai, K. RajaGopal, Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection, Sci. Rep. 14 (1) (2024) 231.
[43] C. Labrín, F. Urdinez, Principal component analysis, in: R for Political Data Science, Chapman and Hall/CRC, 2020, pp. 375–393.
[44] A.K. Ozcanli, M. Baysal, Islanding detection in microgrid using deep learning based on 1D CNN and CNN-LSTM networks, Sustain. Energy Grids Netw. 32 (2022) 100839.
[45] R. Ahmed, V. Sreeram, Y. Mishra, M. Arif, A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization, Renew. Sustain. Energy Rev. 124 (2020) 109792.
[46] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.
[47] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proc. Mach. Learn. Syst. 2 (2020) 429–450.
[48] W. Liu, L. Chen, Y. Chen, W. Zhang, Accelerating federated learning via momentum gradient descent, IEEE Trans. Parallel Distrib. Syst. 31 (8) (2020) 1754–1766.
[49] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, D. Liu, LoAdaBoost: Loss-based AdaBoost federated machine learning with reduced computational complexity on IID and non-IID intensive care data, PLoS One 15 (4) (2020) e0230706.
[50] L. Zhang, B. Cao, Y. Li, M. Peng, G. Feng, A multi-stage stochastic programming-based offloading policy for fog enabled IoT-ehealth, IEEE J. Sel. Areas Commun. 39 (2) (2020) 411–425.
[51] B. Cao, L. Zhang, Y. Li, D. Feng, W. Cao, Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework, IEEE Commun. Mag. 57 (3) (2019) 56–62.
[52] C. Xie, S. Koyejo, I. Gupta, Asynchronous federated optimization, 2019, arXiv preprint arXiv:1903.03934.
[53] Y. Chen, Y. Ning, M. Slawski, H. Rangwala, Asynchronous online federated learning for edge devices with non-iid data, in: 2020 IEEE International Conference on Big Data, Big Data, IEEE, 2020, pp. 15–24.
[54] W. Wu, L. He, W. Lin, R. Mao, C. Maple, S. Jarvis, SAFA: A semi-asynchronous protocol for fast federated learning with low overhead, IEEE Trans. Comput. 70 (5) (2020) 655–668.
[55] M. Cao, L. Zhang, B. Cao, Toward on-device federated learning: A direct acyclic graph-based blockchain approach, IEEE Trans. Neural Netw. Learn. Syst. (2021).
[56] K.C. Sim, P. Zadrazil, F. Beaufays, An investigation into on-device personalization of end-to-end automatic speech recognition models, 2019, arXiv preprint arXiv:1909.06678.
[57] K. Wang, R. Mathews, C. Kiddon, H. Eichner, F. Beaufays, D. Ramage, Federated evaluation of on-device personalization, 2019, arXiv preprint arXiv:1910.10252.
[58] V. Smith, C.-K. Chiang, M. Sanjabi, A.S. Talwalkar, Federated multi-task learning, Adv. Neural Inf. Process. Syst. 30 (2017).
[59] F. Hanzely, P. Richtárik, Federated learning of a mixture of global and local models, 2020, arXiv preprint arXiv:2002.05516.
[60] A. Shiravi, H. Shiravi, M. Tavallaee, A.A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Comput. Secur. 31 (3) (2012) 357–374.
[61] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108–116.

Syed Muhammad Salman Bukhari received his bachelors degree in electronics engineering in 2016 and his masters degree in electrical engineering in 2021 from the Capital University of Science and Technology, Islamabad, Pakistan. He is currently working on ML/DL Models in the different fields of research and AI. His research interests include machine learning, data science, deep learning, artificial intelligence, and optimization of models



Muhammad Hamza Zafar received his bachelors degree in Electronics engineering in 2015 and his masters degree in electrical engineering in 2022 from the Capital University of Science and Technology, Islamabad, Pakistan. He is pursuing a Ph.D. degree in the field of Robotics and AI. His research interests include the Robotics, Deep learning, swarm intelligence and data security.

Mohamad Abou Houran (Senior Member, IEEE) received the B.S. degree in electrical engineering from the Faculty of Mechanical and Electrical Engineering, Damascus University, Syria, in 2008, and the M.S. and Ph.D. degrees from the School of Electrical Engineering, Xi'an Jiaotong University (XJTU), China, in 2014 and 2020, respectively. He is currently an Assistant Professor with the School of Electrical Engineering, XJTU. His research interests include Wireless Power Transfer (WPT), Power Electronics, and Power Systems, and Interdisciplinary Research. He is a member of the IEEE Power Electronics Society (PELS). He is a reviewer for some IEEE and Elsevier journals, such as IEEE TRANSACTIONS ON POWER ELECTRONICS, IEEE ACCESS, and Applied Energy.

Syed Kumayl Raza Moosavi received his bachelors degree in Mechatronics engineering in 2016 and his masters degree in Artificial Intelligence in 2023 from the National University of Sciences and Technology, Islamabad, Pakistan. He will be pursuing a Ph.D. program in the field of Robotics and AI. His research interests include the Robotics, Artificial Intelligence and Swarm Optimization.

Majad Mansoor received his bachelors degree in Electronics engineering in 2015 and his masters degree in electrical engineering in 2021 from the University of Science and Technology of China. He is pursuing a Ph.D. degree in the field of Deep learning and Sustainable Technologies. His research interests include the deep learning, sustainable technologies, robotics and renewable energy.

Muhammad Muaaz received the M.Sc. degree in information and communication systems security from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2012, and the Ph.D. degree in computer science from Johannes Kepler University, Linz, Austria, in 2017. From 2018 to 2020, he held postdoctoral fellowships at the University of Agder, Grimstad, Norway, and Johannes Kepler University. He is currently working as a Researcher with the University of Agder. His research interests include the design and development of secure intelligent systems, information and communication systems security, data science, machine (deep) learning, mobile computing, and physiological and behavioral biometrics.

Filippo Sanfilippo holds a Ph.D. in Engineering Cybernetics from the Norwegian University of Science and Technology (NTNU), Norway, with a focus on alternative and flexible control approaches for robotic manipulators. His research interests include robotics, wearables, human–robot collaboration, collaborative robotics, artificial intelligence and control theory. He is currently appointed as a Professor at the Faculty of Engineering and Science, University of Agder (UiA), Grimstad, Norway. He is also an adjunct Professor at the Faculty of Informatics, Kaunas University of Technology, Kaunas, Lithuania. He carries a vast experience in participating to European research programs and various national projects from the Research Council of Norway (RCN). He is an IEEE Senior Member. He is the former Chair of the IEEE Norway Section. He is also the Chair of the IEEE Robotics and Automation, Control Systems and Intelligent Transportation Systems Joint Chapter. He is currently a member of the IEEE Region 8 Chapter Coordination Committee. He is a Member of the IEEE Public Visibility Committee, and of the IEEE R8 Awards \& Recognitions (A\&R) Committee. He is also the Treasurer of the Norsk Forening for Kunstig Intelligens (NAIS), the Norwegian Association for Artificial Intelligence. He has authored and co-authored several technical papers in various journals and conferences. He is a reviewer for several international conferences and journals.