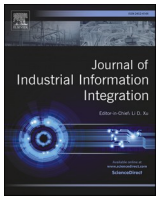
Journal of Industrial Information Integration 44 (2025) 100800

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.sciencedirect.com/journal/journal-of-industrial-information-integration



Design of a federated ensemble model for intrusion detection in distributed IIoT networks for enhancing cybersecurity

Ayushi Chahal *, Preeti Gulia, Nasib Singh Gill, Deepti Rani

Department of Computer Science & Applications, Maharshi Dayanand University, Rohtak 124001, India

ARTICLE INFO

Keywords:
Industrial Internet of Things (IIoT)
Intrusion detection
Federated learning
Ensemble learning

ABSTRACT

Automation has become possible by the reliance of Industry 4.0 on the Internet of Things (IoT) ecosystem. IIoT brings the next phase of digital transformation, which is defined by the convergence of the Industrial Internet of Things (IIoT) and Artificial Intelligence (AI). Industrial Internet of Things (IIoT) contributes in expansion of IoT network where large-scale data is generated continuously. Due to several security vulnerabilities in industrial information security management systems, the data can be breached by malicious attackers. Federated Learning is the best solution to address the challenge of heterogeneity and geographical locations in IIoT. This study proposes IIoT-IDFE (IIoT- Intrusion Detection Federated Ensemble) model for intrusion detection in heterogeneous IIoT environment. IIoT_IDFE model detects unwanted intrusions in two stages. In the first stage, local IIoT client devices use the Shared Local Ensemble (SLE) model to detect intrusion. In the second stage, instead of sharing actual data, the ensemble model is shared with a central federated server using the Broadcast Global Ensemble (BDE) model. By combining the advantages of ensemble and federated learning techniques, the proposed model guarantees a thorough approach to produce reliable aggregated predictions at the global scale. This allows IoT devices to maintain their privacy while improving the model's efficiency. Freely accessible industrial datasets i.e. "Edge-IIoTset" and "ToN-IoT" are used to implement the proposed intrusion detection method. Performance evaluation metrics, namely, accuracy, precision, recall, and f1-score are used to validate the performance and efficacy of the proposed IIoT-IDFE model. The performance evaluation with 99.99% to 100% accuracy confirms that the proposed model outperforms the state-of-art techniques.

1. Introduction

Industrial Internet of Things (IIoT) is a decisive component that may enable Industry 4.0 revolution across IoT industry. IIoT enables large number of IoT devices and applications within an industry to interconnect using the Internet. This automation can be achieved by integrating various technologies like big data analytics, Machine Learning (ML), Deep Learning (DL), predictive analysis, and manufacturing processes. IIoT intends to enable the control of the effective manufacturing/production of commodities and human monitoring in industrial environments. Through the creation of a smart network that facilitates data sharing, IIoT integrates many machines, sensors, actuators, and gadgets into cohesive environments [1].

IIoT helps to get data autonomously from a range of systems by bridging the gap between Information Technology (IT) systems and conventional industries. This digital shift of IIoT and AI is all powered by data. IIoT data is very dispersed among structures that are held or handled by several parties, and each of them can only access a portion of the entire data due to a variety of impediments [2]. Since, data is collected by various industrial equipments at different geographical locations, that tends to be heterogeneous. Due to different geographical locations, it is not possible to process the complete data collected by different IoT devices for any kind of data analysis as shown in Fig. 1. To access the complete data, one needs a centralized dataset which means integration of data from multiple decentralized data sources [3]. Most IIoT industries are not comfortable with sharing their data among industry partners due to privacy laws. Although, it can improve the scalability and reliability of the system, but privacy and security of data remain major concerns [4].

IIoT networks are susceptible to various types of attacks, primarily because IoT devices lack the computational and memory capacity required for more advanced safety measures. IIoT systems are

* Corresponding author.
E-mail addresses: ayushi.rs.dcsa@mdurohtak.ac.in (A. Chahal), preeti@mdurohtak.ac.in (P. Gulia), nasib.gill@mdurohtak.ac.in (N.S. Gill), deepti.rs.dcsa@mdurohtak.ac.in (D. Rani).

https://doi.org/10.1016/j.jii.2025.100800

Available online 13 February 2025

distributed and inherit all the issues associated with the requirement to ensure privacy, reliability, and accessibility [5]. Federated learning is a plausible and effective way around this problem. It is a machine learning technology developed by Google in 2016 that protects privacy [6]. When compared to standard centralized machine learning techniques, federated learning can dramatically overcome the privacy difficulties. Federated learning holds significant potential for connecting disparate data sources while protecting privacy. With the help of federated learning, multiple industries can train the data without disclosing relevant data and resources to a central data server [7].

Data collection and data flow from multiple geographically separated Industrial sites in a decentralized Industrial IoT environment involves the risk of attacks and data manipulation. This motivates to study in this specific area and work on an anomaly detection model for a decentralized IIoT environment. The main contributions of this study are:

- This study provides a detailed literature survey, elaborating the requirement of an anomaly detection model for distributed Industrial 4.0 architecture.
- A model is proposed for anomaly detection and prediction in an IIoT decentralized environment using an ensemble and federated approach.
- Results are discussed using an Industrial dataset named "Edge-IIoT-set" and "ToN-IoT" dataset.

The article is organized into five sections. Section 2 provides a detailed literature review of heterogeneous data collection in IIoT environment, different anomaly attacks in the IIoT network, multiple ML/DL techniques used to detect these attacks and the importance of federated learning for a decentralized organization. Section 3 elaborates all the background technologies that are used in the proposed model. Section 4 provides a detailed step-by-step explanation of the proposed model for anomaly detection in a decentralized IIoT environment. Section 5 discusses the results obtained after applying the proposed model over the two datasets i.e. Industrial dataset "Edge-IIoTset"and "ToN-IoT" dataset. This section explains the features of the dataset used and the performance matrices used to validate the model.

2. Related work

In this section, the detailed literature review is discussed which explains the heterogeneity of devices and data collected in an IIoT environment and how federated learning is used to handle this heterogeneity is explained in subSection 2.1. SubSection 2.2. discusses the existing literature on federated learning which can be used in an IIoT environment for anomaly detection.

2.1. Heterogeneity in IIoT

IoT environment is heterogeneous in nature, specifically when IoT sensors/devices are used for industrial applications. IIoT uses such devices that collect data in textual, audio, video or any other multimedia format. Also, not all IIoT sensors/devices need to collect the same kind of data simultaneously. There is always a possibility that any industrial section is collecting IoT data in some format with specific features, while at the same time any other industrial section is collecting IoT data in another format with other features and data collected by both sections needs to be processed together. Here comes the concept of heterogeneity.

F. Ilhan *et al.* [19] handle heterogeneous systems in IoT-enabled environments using an FL approach named, ScaleFL. For clients with constrained assets, ScaleFL addresses system heterogeneity by systematically downsizing the DNN model. Based on the resource constraint of each participating client, the model is automatically partitioned during FL along height (exits) and breadth (hidden dimensions).

E. Diao *et al.* [20] proposed a HeteroFL model for heterogeneous clients present in different geographical locations. HeteroFL provides a single global inference model for the training of various local models with different computational complexity and heterogeneous data. A.S. Dina *et al.* [17] used the focal loss method to balance out heterogeneous imbalanced data collected through IoT sensors.

D.N. Sachin *et al.* [21] developed a model named FedCure for Medical Industry in IoT-enabled environment. FedCure handles problems caused by heterogeneity in data and models by using several personalization strategies at the device level, making it easier to deploy personalized models that are suited to particular needs. The proposed framework uses edge computing techniques to provide effective processing and optimization in order to alleviate concerns associated with device heterogeneity. FedCure framework demonstrates outstanding accuracy and little communication overhead, particularly in tackling the difficulties brought on by heterogeneity.

B. Li *et al.* [22] addressed the problem of device heterogeneity using federated learning. Authors proposed a model named DeProFL which stands for Decentralized Prototype representation learning Federated Learning, based on time-varying communication topology and

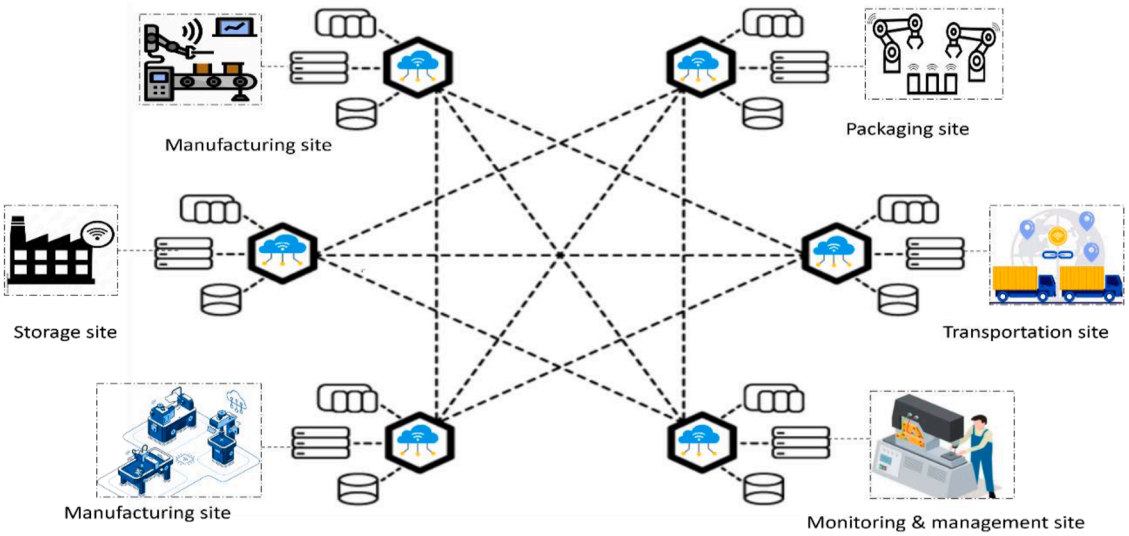


Fig. 1. Distributed IIoT architecture.

prototype representation learning. Prototype learning is used to build regular presentations of data points and models across devices to counteract the impacts of heterogeneity.

N. Abosata *et al.* [29] proposed a model named FT-CID (Federated Transfer learning-assisted Customized distributed IDS) for intrusion detection in heterogeneous IoT network. FT-CID works over a routing protocol for low-power and lossy networks (RPL). RPL is applied to different applications of IoT. Besides federated learning, it has also used transfer learning for server and edge global and local model parameters. FT-CID detects intrusions over IIoT network using RPL-IIoT dataset with 85.52% accuracy.

The literature review in this section shows that Industrial IoT faces the problem of data and device heterogeneity which can be overcome using a variant of the Federated Learning approach.

2.2. Federated learning for anomaly detection in IIoT

Federated learning makes it possible to share industrial data securely and distribute advanced analysis. It has become an important strategy in the Industrial Internet of Things (IIoT). The difficulties of data security and privacy in IIoT contexts can be addressed with federated learning [8].

M.M. Alani *et al.* [16] proposed an intrusion detection system for IIoT using the WUSTL-IIOT-2021 dataset. The used dataset has 48 features which have been reduced to 11 features with the help of the Recursive Feature Elimination (RFE) method of feature reduction. Shapley additive explanation (SHAP) technique is used to explain different features and the classifier is used to classify the attacks in the dataset. The proposed system gives an accuracy of 99.97%. Additionally, the suggested method demonstrated exceptional efficiency with a detection time of 0.1517 ms.

B. Shubyn *et al.* [9] suggested utilizing Federated Learning (FL) to enable a safe exchange of knowledge among intelligent manufacturing machines like Autonomous Guided Vehicles (AGVs). The authors used three simulated devices to gain experience. A two-layer FL architecture based on AI is used for local servers and the outcomes of the proposed methodology are used to validate the efficacy in real-world settings. LSTM model is used to train local model and Mean Squared Error (MSE), Mean absolute percentage error (MAPE), and Root Mean Squared Error (RMSE) are used as evaluation parameters.

N. Abosata *et al.* [10] proposed a model named FT-CID (Federated Transfer learning assisted Customized Distributed) for intrusion detection in a heterogeneous IoT environment. The proposed model can used for routing protocol for low-power and lossy networks (RPL). FT-CID model indirectly uses the local and global properties of various IoTs to achieve RPL privacy. Upon testing FT-CID model on a heterogeneous IoT network, the authors concluded that FT-CID detects RPL intrusions with an accuracy of 85.52%.

V. Kelli *et al.* [11] proposed a network flow-based Intrusion Detection System (IDS) using FL and active learning. The earlier technique is used to train models in private, and the subsequent one is a semi-supervised method used to adjust the global model to every node in the IDS. Within FL, authors have used a Deep Neural Network (DNN) with six dense layers for attack detection and classification. The proposed model increased the accuracy by 7.07%.

J. Xu *et al.* [12] provide a detailed analysis of using federated learning in the healthcare industry which is a large part of current IIoT. The authors highlight answers to the basic federated learning problems related to security, systems, and statistics, and pinpoint the possible applications in the medical field. M. Namratha *et al.* [13] focused on anomaly detection in patient data like temperature and heart rate. Federated learning played an important role in data security and privacy in this MIoT (Medical Internet of Things) scenario. Authors built a machine learning model locally on each device, and then employed the Secure Aggregation Principle on the centralized server to aggregate the training data. As a locally distributed model trains on it, the central machine-learning model gets better. This is updated with the total findings, which makes it more intelligent.

Y Liu *et al.* [14] developed a federated framework so that decentralized edge devices can work together to train a Deep Anomaly Detection (DAD) model in a way that can enhance the model's capacity for generalization. Authors also proposed a CNN-LSTM (Convolutional Neural Network-Long Short-Term Memory) model to detect any abnormalities in IIoT network.

R. Lazzarini *et al.* [15] presented the use of the Federated Learning model for Intrusion Detection in an Industrial IoT environment. FedAvg is used as an aggregation technique in the global model and a shallow artificial neural network (ANN) serves as the shared local model. Authors have performed binary and multiclass classification in a flower framework using two datasets i.e. ToN_IoT and CICIDS2017. Accuracy, precision, recall and F1-score are used as evaluation parameters resulting in 0.9815, 0.9829, 0.9815, and 0.9816 values respectively.

N.A. Jalali *et al.* [18] analyzed data addressing the general operations, configurations, and relationships between IoT devices in addition to centralized data and Federated learning systems. The authors examine general security flaws that malicious parties are trying to exploit, as well as security issues with the FL system. In this study, defensive strategies to improve safety and security in the FL system is also identified to accomplish the security trinity of safety, reliability, and accessibility.

M.M. Rashid *et al.* [24] suggest using Federated Learning (FL) to identify unauthorized intrusions using Edge-IIoTset in order to assess the effectiveness of the suggested approach. Authors have used CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network) in federated system. The performance test shows the dependability and efficacy of the suggested intrusion detection model by utilizing the FL approach to achieve an accuracy (92.49%) that is comparable to the accuracy provided by the traditional centralized ML models (93.92%).

X. Wang *et al.* [41] designed a method using federated learning technique for anomaly detection in IIoT environment named FLAD (Federated deep reinforcement Learning empowered Anomaly Detection). Authors used deep reinforcement learning. FLAD is composed of three different models i.e. Global Anomaly Detection Center (GADC), the Local Anomaly Detection Center (LADC), the Regional Anomaly Detection Center (RADC). FLAD attains high throughput, low latency, and high accuracy in anomaly detection for the protection of privacy in a range of IIoT applications.

Although the above literature in this section provides a lot of ways to classify and detect anomalies over the IoT network, but the methods discussed above used a centralized ML/DL approach. The centralized approach encountered several obstacles, including those related to the effectiveness of the system, real-time analysis and prompt reaction, consumption of electricity, high network bandwidth requirements, and, above all, confidentiality and safety, which influence the system's entire functionality. Federated Learning came out to be the way to handle the problem of centralized ML/DL i.e. privacy and security.

3. Background

This section explains the details of basic models used for ensemble learning this study. Also, it explains federated learning concepts in detail.

3.1. Ensemble learning

Ensemble learning refers to a process of combining different machine learning algorithms to obtain more accurate and robust predictive results as compared to an individual model. Multiple predictions obtained from different machine learning algorithms are combined together to generate better predictive performance. Multiple ML models are combined together to solve a complex problem that can be resolved more efficiently as compared to traditional ML models. Every individual

machine learning model may have its own strengths and weaknesses which may lead to improved generalized performance as shown in Fig. 2. However, ensemble learning is more expensive approach as compared to traditional models, but the performance achieved by ensemble learning overcomes this drawback. An unbiased model is constructed using an ensemble approach that aims to train or learn. The model that learns from the newly constructed model is called hypothesis. Ensemble models are constructed using combination of multiple base models which are fitted to learn from a set of data and aggregated to produce a final decision. Ensemble algorithms have been categorized into mainly three classes which are: bagging, boosting, and stacking. These ensemble classes contain several commonly used algorithms such as Random Forest, XGboost, Light Gradient Boost, CatBoost etc [25].

3.1.1. XGBoost

It is the most commonly used by data scientists due to some features like tree pruning, parallel processing, and regularization. It generates one of the most robust, efficient, and accurate models. An enhanced version of Gradient Boosting (GB) called Extreme Gradient Boosting (XGB) was developed to enhance performance by lowering the false alarm rate and increasing accuracy. In order to classify data with greater precision as it enters a network, XGB aids in the building of a smarter classification model. One of the most intriguing ensemble-boosting techniques with favorable results is XGB. When the machine is overloaded with data, XGB can handle over-fitting issues well. Because there are so many data entries in this instance, the classifier has to be able to adjust more quickly. Optimizing the optimum degree of hyper-parameters regularization gives XGB more power. By adjusting the hyperparameters, this decision-tree-based gradient-boosting machine learning approach may improve the accessibility, precision, as well as effectiveness of ensemble-based intrusion detection systems. It can manage the bias-variance trade-offs consistently. The primary disadvantage of this model is its high memory utilization and sluggish operating performance. With boosting approaches, overfitting may be minimized by carefully adjusting the hyper-parameters [32].

3.1.2. LGBM

A histogram-based decision tree approach called Light Gradient Boosting shortens the simulation times and uses less memory on a computer while increasing model effectiveness. Compared to other boosting ensemble decision tree algorithms, LGBM is significantly more optimized. It is a far better learning algorithm that is quicker, more dispersed, stronger, and more advanced. Large data flow can be effectively handled by LGBM. LGBM, like many other boosting algorithms, computes the superlative split and learns decision trees using a pre-sorted method and a Histogram-based approach. Two novel methods used by LGBM are: Exclusive Feature Bundling (EFB) and Gradient-based One-Side Sampling (GOSS). In order to separate the data samples and get a split value, GOSS downsamples each instance determined by the gradient sizes [33].

3.1.3. CatBoost

Catboost is popular among data scientists due to its high predictive accuracy, fast training, and handling of overfitting. It reduces the need for extensive pre-processing. CatBoost, which stands for "categorical boosting," employs target-based statistics, one_hot_max_size (OHMS), and permutation approaches to focus on categorical columns. At each fresh split of the existing tree, CatBoost applies the greedy strategy to solve the exponential increase of the combination of the features. The first improvement of CatBoost over Gradient Boosting is how it handles categorical variables with large cardinality. CatBoost employs one-hot encoding for categorical data with low cardinality [34].

3.1.4. Random Forest

Random forest is used to make predictions and classify the data using the bagging technique of ensemble learning. Bagging is a technique that is used to minimize the variance of the prediction function. Random forest is an enhanced version of the bagging technique that combines multiple unrelated decision trees and averages their outputs to give the final outcome. Random forest comprises multiple decision trees that provide stable predictions [31]. It is used to avoid the over-fitting problem of the Decision Tree. Random forest is created by generalizing multiple decision trees and then their mode is considered as the output of the system. It is considered the predominant method for prediction tasks as compared to other ensemble techniques.

3.1.5. Ensemble voting method

A collection of machine learning models is trained on the same dataset for majority voting, and each model employs a different algorithm or method. Every model generates a forecast for every test case; the model with the highest percentage of votes wins as the final output prediction. The ensemble approach might not be able to produce a reliable forecast in the situation if none of the guesses receives more than half of the votes [41]. Fig. 3 represents a voting method.

3.2. Federated learning

A distributed machine learning approach known as "federated learning" allows many client nodes to work together with a central server node to develop a model without exchanging training data [26]. Federated learning may be employed to deal with data privacy problems by training machine learning models at edge nodes (IoT devices) locally before relocating to the cloud network. Federated learning expedites information sharing among various IoT devices configured in an industry or an organization via collaborative trained ML models [39].

3.2.1. Benefits of federated learning in different industry applications

Reduced bandwidth and computational cost: In Federated learning,

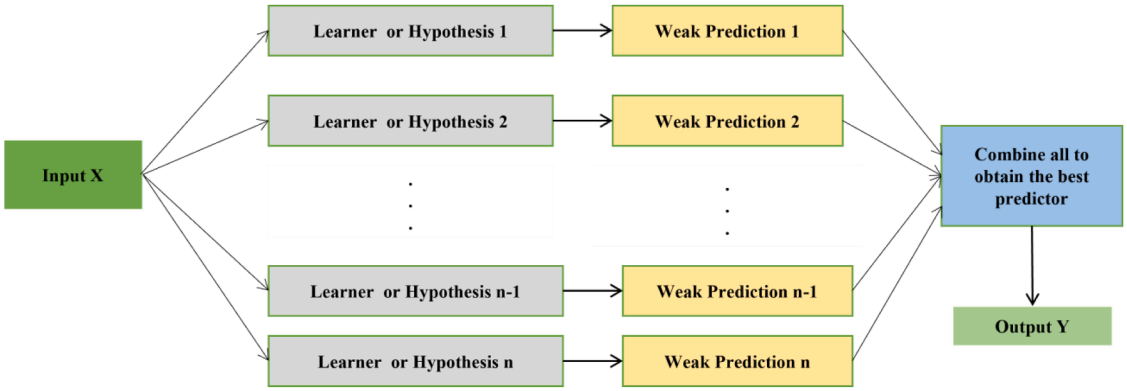


Fig. 2. Ensemble Model working.

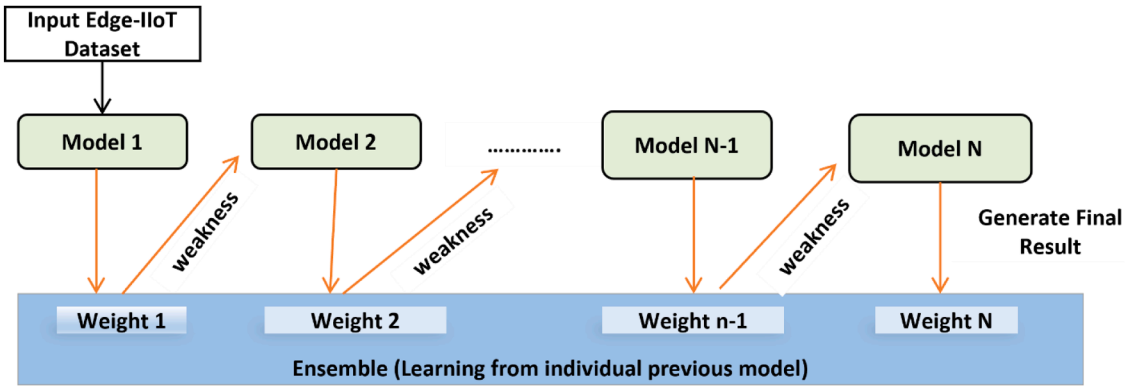


Fig. 3. Ensemble Voting Model.

workload distribution and training on parallel machines reduce the computational cost needed to train a super-quality global model [40]. The training is performed by multiple edge devices connected across the network which accelerates the inference and limits the network bandwidth. Federated learning reduces the exposure of sensitive data by ensuring that it remains on the local device or server. Instead of raw data, the model training process relies on aggregated updates. In FL, only weights are transferred rather than the data in order to reduce bandwidth. FL contributes to enhance data privacy and resist adversarial attacks in following ways:

- Enhancing data privacy: The model training process relies on aggregated updates instead of raw data. This distributed approach provides several privacy advantages such as data locality, secure aggregation, and regulatory compliance, differential privacy integration etc.
- Resisting adversarial attacks: Federated learning is also effective in maximizing the robustness of models against adversarial attacks, which are attempts to deceive ML systems. Adversarial examples are maliciously crafted inputs designed to fool machine learning models. FL's decentralized nature can mitigate their impact, as the model is trained on diverse and distributed data, making it less prone to overfitting specific adversarial patterns. FL uses robust aggregation techniques to handle outliers or poisoned updates to minimize the influence of adversarial contributions from compromised devices. Methods like anomaly detection can identify devices submitting anomalous model updates andflagging potential adversarial activity.

FL never shares the raw data of one client node with another client node or the server node. This sets FL apart from conventional distributed optimization and necessitates handling diverse data. There are two main options for FL [27]:

- cross-device: where FL is applied between edge devices, and
- cross-silo: where FL is applied between major institutions.

Federated learning is generalized in three simple steps:

Step 1: Sharing Global Model

In first step, general parameters are passed by the central server to all the local clients in IIoT surroundings.

Step 2: Local Model Training

In this step, data collected by local clients are trained and modeled by local Machine Learning or Deep Learning models at IoT sensors/devices or edge devices.

Step 3: Updating the Global model

To maintain the security and privacy of the data on every local client, only local models are shared with the central server rather than sharing data itself. Using the information sent by the local model, the central server updates the Global model by aggregation using existing FedAvg, and FedSGD algorithms.

Step 4: Step 2 and Step 3 keep on going in a loop until a certain accuracy is achieved.

The federated learning procedure is also explained for N clients in Fig. 4.

Federated learning can be categorized into the following [28]:

- **Horizontal Federated Learning:**

This technique is also called sample-based Federated Learning. In Horizontal FL, every device's local data has the same characteristics across different instances. All clients have the same feature with regard to a domain, driven facts, and other FL outcomes.

- **Vertical Federated Learning:**

This technique is also called feature-based federated learning. In this technique, every client employs a third-party organization to exchange just common data through encryption logic between them. It is utilized for unrelated domains to exchange common data via a global learning machine model.

- **Federated Transfer Learning:**

When two data sets differ in terms of both sample and feature, this kind of FL is used. A pair of entities are proposed, each in a different geographic region. The client groups associated with the two entities have a minor intersection, or since their businesses are different, there is only a small amount of feature space overlap between them. Since FTL protocols are comparable to Vertical FL, this scenario necessitates their application in order to offer the solutions for the full sample and feature space inside a federating framework.

4. IIoT-IDFE methodology and framework

In this section, the combined strategy of using Ensemble Learning (EL) and Federated Learning (FL) for anomaly detection in the IIoT environment is explained. The "Edge-IIoTset" dataset is chosen for this study. The dataset is explained in next section. This model focused on the horizontal FL concept. Four different clients are considered to work in the IIoT environment. These clients can detect anomalies/attacks/intrusions over the network using ensemble models. To increase the accuracy of a complete IIoT environment, it would be great if all the

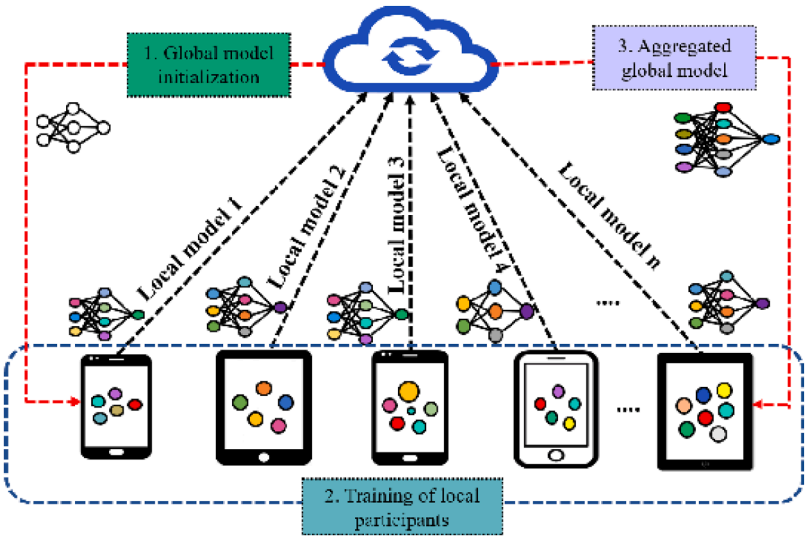


Fig. 4. FL procedure considering N participants [23].

clients could work in cooperation. However, it would not be possible for clients to share their actual dataset due to privacy and safety concerns. Hence, Federated Learning plays an important part here. The proposed model is described in two subsections here, the first elaborates the working of the Ensemble Learning model as local model and the second elaborates the working of Federated Learning as a global model.

In Fig. 5, the proposed model's federated approach is shown. As it shows, multiple decentralized IIoT clients are connected with a single federated server. IIoT federated server sends parameters to IIoT clients so that the clients can execute their tasks. All clients have their own Machine learning models which are further used to train local data collected by the IIoT devices at the terminal. The proposed model uses the Ensemble Learning model for local model training part.

4.1. Shared local ensemble (SLE) model as clients' local model

The proposed model used the Ensemble Learning model locally to ensure privacy named as Shared Local Ensemble (SLE) model. SLE model is used at every client of IIoT environment. SLE uses different ensemble models for anomaly detection in Industrial IoT such as Light Gradient Boosting, XGBoost, CatBoost, and Random Forest. These are well-known ensemble models, and every model is trained on the client dataset separately. Fig. 6 shows the flow chart of all the steps taken on the local ensemble model for every client.

Initially, IIoT client dataset is taken and split between training and testing with 4:1 ratio. The training dataset is encoded using a "label encoder". Since the dataset used is an Industrial IoT dataset, it is found to be imbalanced when evaluated. This imbalance is removed by using SMOTE (Synthetic Minority Oversampling Technique) + ENN (Edited Nearest Neighbors). SMOTE is used to handle oversampled data and ENN manages under-sampled data. Algorithm 1 shows the complete step-wise process that takes place at IIoT client edge devices locally.

Shared Local Ensemble model works into two steps:

Step 1: Train and validate Ensemble model

Developing a method that blends federated learning with a variety of machine learning techniques, including Light Gradient Boosting, XGBoost, CatBoost, and Random Forest, entails coordinating a cooperative model training procedure over decentralized data sources while

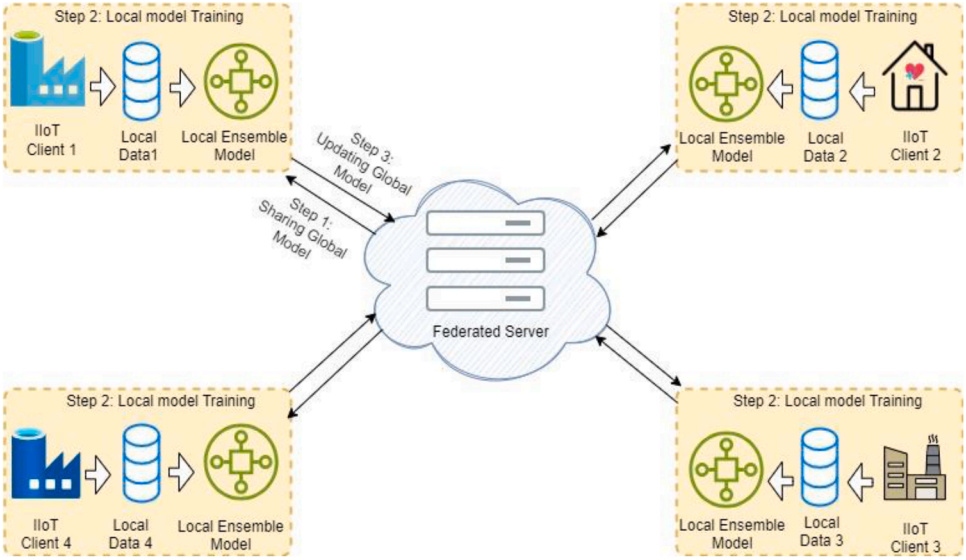


Fig. 5. Federated Learning Model in IIoT.

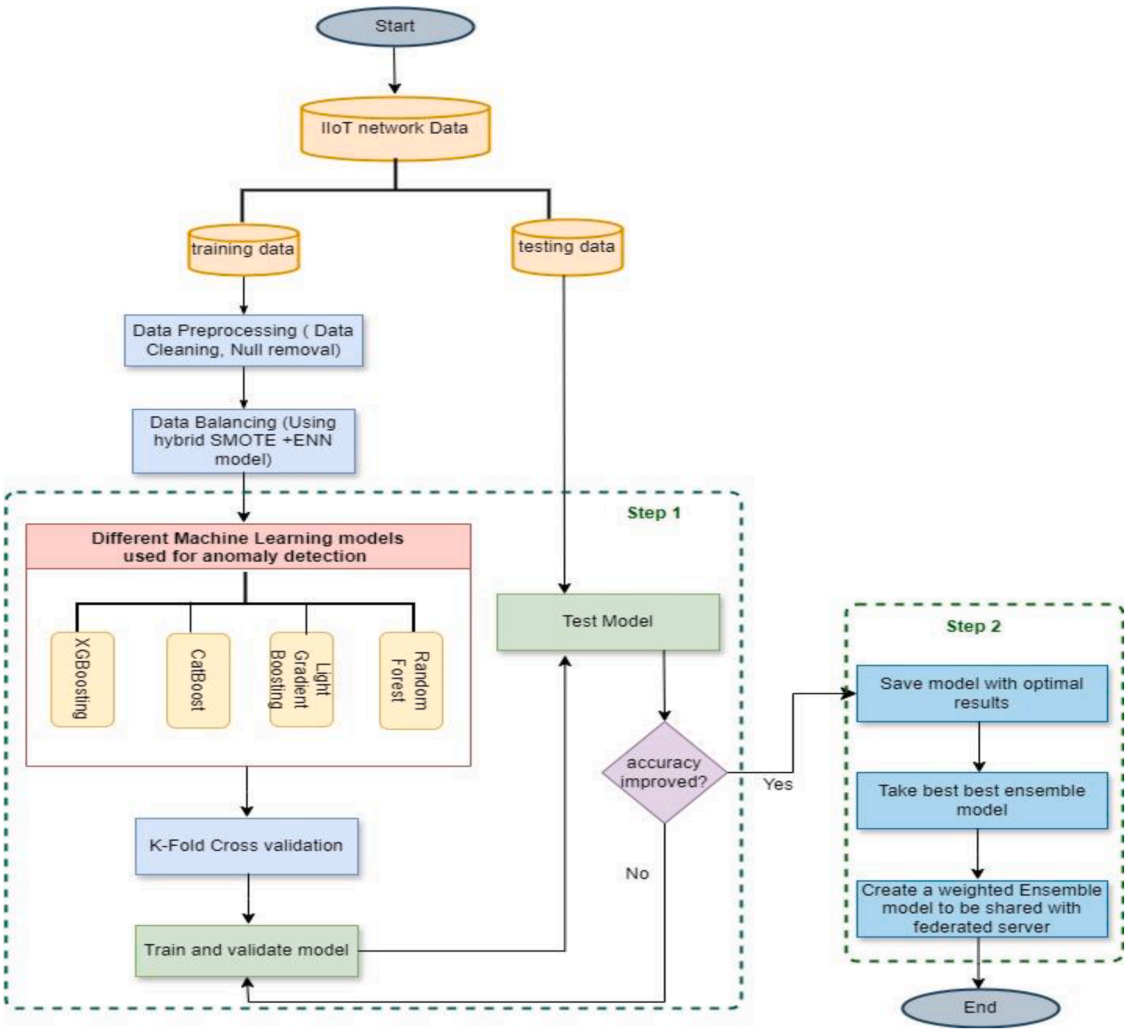


Fig. 6. Flowchart for Shared Local Ensemble (SLE) Model.

utilizing the individual strengths of each ensemble member to improve predictive performance. Federated learning protects privacy and security by training models on decentralized data locally rather than centrally, whereas ensemble techniques like as Random Forest, XGBoost, LightGBM, and CatBoost are excellent at identifying intricate trends and enhancing model resilience through aggregation.

The "train_test_split" function is used for evaluating the performance of a machine learning model. Scikit-learn is used to split underlined dataset into training sets and test sets. This is vital for evaluating the performance of a machine learning model. By default, 20% of data is test set and 80% of data is considered as training sets. Train set is a set of data that is used to fit the model. On this subset of dataset, the model is trained and the data is learned by the model. The "EdgeIIoT" dataset consists of total 157,800 records with 34 features (columns). Here, 33 features are given as input and one contains output classes. Categorical columns have been converted into numerical values using label encoding technique (using LabelEncoder() function) and they can be fitted by ML models. It is an important function to preprocess a machine learning project. Balanced data is then given to all the ensemble models i.e. Light Gradient Boosting, XGBoost, CatBoost, and Random Forest. All of these use the boosting method to create an ensemble tree except Random Forest which uses the bagging method. The "K-fold cross-validation" is applied to the ensemble models to get better accuracy. The resulting accuracy is validated using a testing dataset. Numbers of experiments have been carried out to analyze model robustness, diversity, and addressing data heterogeneity. More than 50 experiments were conducted. 5–10 experiments have been conducted for each client and for each round. Conducting multiple experiments allows for a more thorough exploration of hyperparameters across the federated clients and the aggregation strategies used in the ensemble.

Step 2: Create a weighted local ensemble model

Model with the best accuracy from step 1, is chosen to create a weighted ensemble model, that is shared with the federated server to update the global model. Using local datasets, several federated clients must be coordinated in order to build a reliable ensemble model. Using the ensemble approaches, each client separately trains a base learner with characteristics unique to its local data distribution. To effectively integrate their prediction powers, the generated models are then aggregated into a global ensemble model by a voting process or weighted averaging. Every IIoT client uses voting method of ensembling (explained in Fig. 7), to ensemble the SLE from previous federated round and the updated BGE model.

This can be explained mathematically as follows:

Let, $X = \{X_1, X_2, \ldots, X_k\}$ represents individually partitioned dataset for IC = $\{IC_1, IC_2, IC_3 \ldots\ldots IC_k\}$different IIoT clients.

$E = \{E1, E2, .., Ep\}$ be the local model models i.e. Random Forest, LightGBM, XGBoost, and CatBoost that are used to classify and detect attacks on the local dataset for different individual IIoT clients. Each IIoT client i trains a base learner E_i using an ensemble method (LightGBM, CatBoost, XGBoost, or Random Forest i.e. E1, E2, .., Ep) on

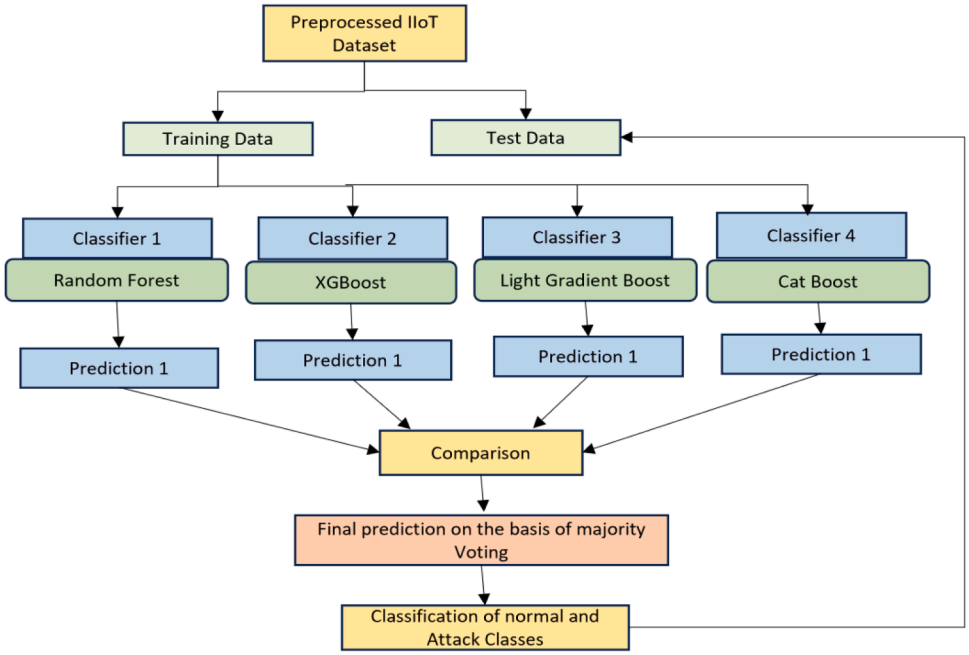


Fig. 7. Voting in SLE for round 1.

its respective local dataset X_i. The main goals of cooperatively training these K IIoT clients, while keeping data private and improving model's performance.

Local models are aggregated at each client using a voting mechanism as given in Eq. (1).

$$\underset{y}{\operatorname{argmax}} \sum_{i=1}^{N} w_i I[E_i(x) = y] \tag{1}$$

Where each model E_i votes for class labled as y based on its

Algorithm 1
IIoT Client Training Algorithm.

Input:
a set of Ensemble models $E = \{E_1, E_2, .., E_p\}$
local data set $X = (X_{train}, X_{test})$
BGE: Broadcast Global Ensemble Model
Output:
SLE: Shared Local Ensemble Model.
Data Cleaning
Remove irrelevant, redundant, and less useful instances
Fill the missing value with 0 or a relevant value
Data Balancing
Apply (SMOTE+ENN) method to balance out the data
Data Preprocessing
Cleaning Data by eliminating unnecessary, and duplicated instances.
Use 0 or a pertinent value to fill in the missing value.
Data Balancing
Apply (SMOTE+ENN) method to balance out the data.
Data Transformation
If (categorical or null values)
Then
Label encoder () /*to transform all features in string or categorical form into numerical*/
Else
One-hot encoder () /*to transform all features in string or categorical form into numerical */
Selecting usable features
function IC_r1(E)
$A = 0$
for each model $i = 1$ to p do
$Ei' \leftarrow$ Training (X_i, X_{train})
$Acc_i \leftarrow$ Evaluating (Ei', X_{test})
$Acc \leftarrow Acc \cup Acc_i$
$BE \leftarrow$ Choosing_bestModel(Acc)
return BE

function IC_r(BE, BGE_{r-1})
$G' \leftarrow$ Training (BGE_{r-1}, X_{test})
$BE \leftarrow BE \cup G'$
$SLE_r \leftarrow$ voted_ensemble_Learning(BE)
return SLE_r

prediction, I[.] represents an indicator function which evaluates to 1 if the model E_i predicts y for input x, and 0 otherwise.

Local model SLE consists of M no. of weak learners which are characterized by θ_m. Every IIoT client {IC_1, IC_2, IC_3…….IC_k} focuses on minimizing their respective objective function (Eq. (2)) as per the ensemble model selected by voting method.

$$\sigma_k^m = \frac{1}{n_k}\sum_{i=1}^{n_k} l\left(M_m\left(x_i^k;\theta_m\right),\ y_i\right) \tag{2}$$

Where, l is the loss function i.e. cross-entropy loss for classification.

4.2. Broadcasted global ensemble (BGE) model

As shown in Fig. 5, every IIoT client has its own private dataset which is processed using the SLE Model. There is a central federated server which is used to collect information in the form of models from the IIoT clients to help increase the efficiency of detection attacks over the IIoT network. At the federated server, the proposed Broadcasted Global Ensemble (BGE) Model is maintained using the SLE Model of every client. The proposed Federated Ensemble model "IIoT-IDFE" complete framework is shown in Fig. 8. Algorithm 2 explains the working steps for global model at federated server.

BGE model consists of local SLE models. The objective of BGE is to minimize the weighted sum of local objective function of each SLE

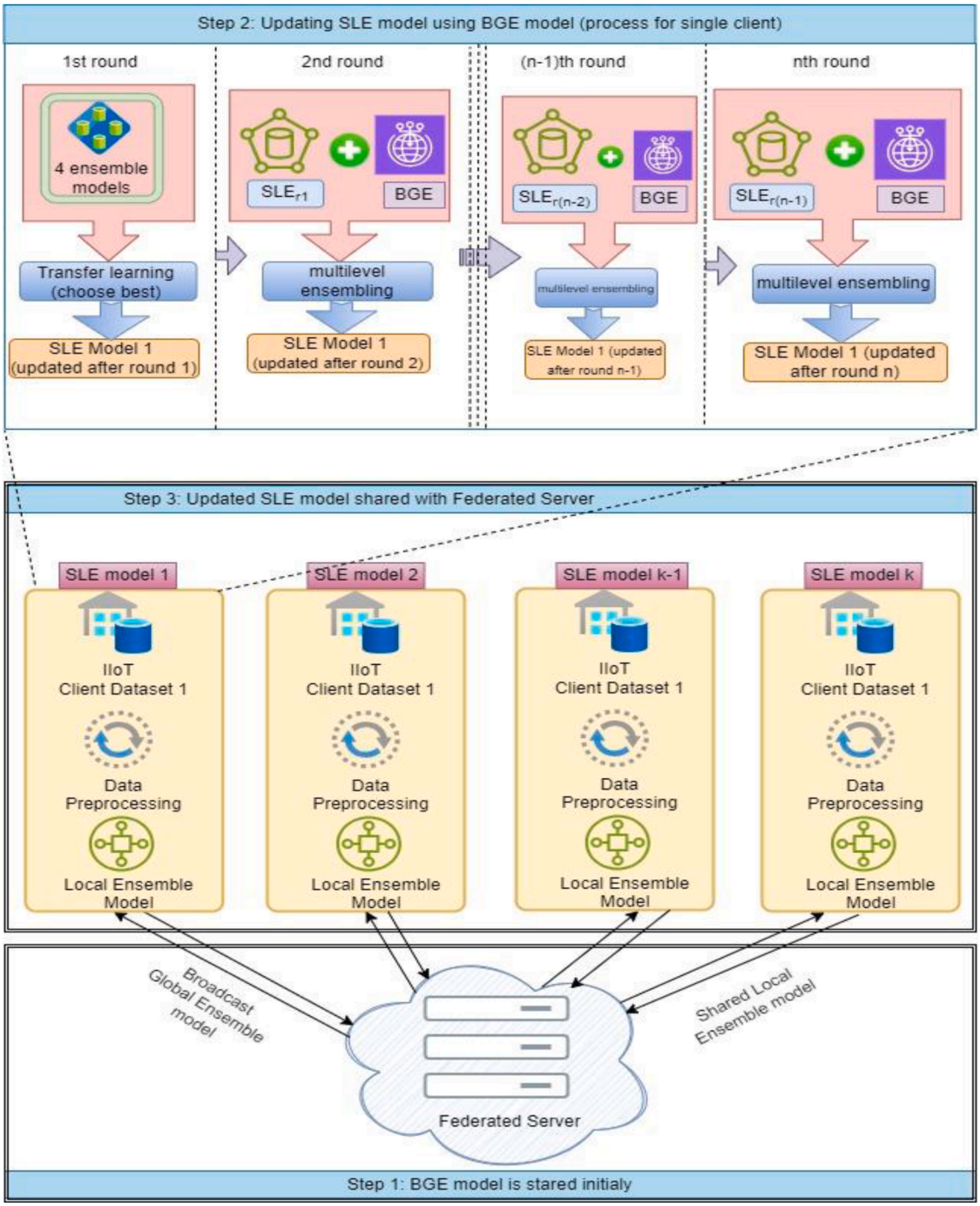


Fig. 8. IIoT-IDFE Model Framework.

Algorithm 2
Federated Server Aggregation Algorithm.

Input:
K: the number of IIoT Clients.
$SLE_{(r-1)}$: Shared Local Model of the previous round
Output:
BGE: Broadcast Global Ensemble Model.
function Aggregating(k)
r ← 1
BE ← φ
while true **do**
for each client $k = 1$ to K **do**
if $r = 1$ **then**
BE ← IC_r1(E)
else
$SLE_r^k \leftarrow$ **IC_r(BE, BGE_{r-1})**
$SLE_r^k \leftarrow \sum_{k=1}^{K} SLE_{r-1}^k$
if
$Acc_r^k = Acc_{r-1}^k$
then
Break;
$BGE_r^k \leftarrow$ Fedaggregation $(\sum_{k=1}^{K} SLE_{r-1}^k \cup BGE_{r-1}^k)$
$r = r + 1$
return BGE.

model. Objective function for global model BGE is given in Eq. (3).

$$\sigma^m(\theta_m) = \sum_{k=1}^{K} \frac{n_k}{n} \left(\sigma_k^m(\theta_m)\right) \tag{3}$$

At initial point central server decides global parameters θ_0^m to all IIoT clients (i.e. IC_1, IC_2, IC_3…….IC_k). Further for each round $R = (r_1, r_2, \ldots, r_n)$ central server provides global parameter θ_t^m. Every client IC_k updates its parameters by performing ensembling of BGE and respective SLE selected model. Updated parameters can be calculated using Eq. (4).

$$\theta_n^{k,m,(r)} = \theta_n^{k,m,(r-1)} - \eta \nabla \alpha_k^m \left(\theta_n^{k,m,(r-1)}\right) \tag{4}$$

Where, η is a learning rate initialized as 0.01.

After updation at local IIoT clients, updated parameters $\theta_n^{k,m}$ are again shared with the central server. Server aggregates the updated parameters of all the SLE to generate new parameters for BGE model as given in Eq. (5).

$$\theta_r^m = \sum_{k=1}^{K} \frac{n_k}{n} \theta_{(r-1)}^{k,m} \tag{5}$$

This process keeps on going until results converges.

The goal is to maximize the accuracy of the aggregated global model (BGE) while ensuring data privacy by keeping local datasets to respective clients. The performance of the combined federated learning system

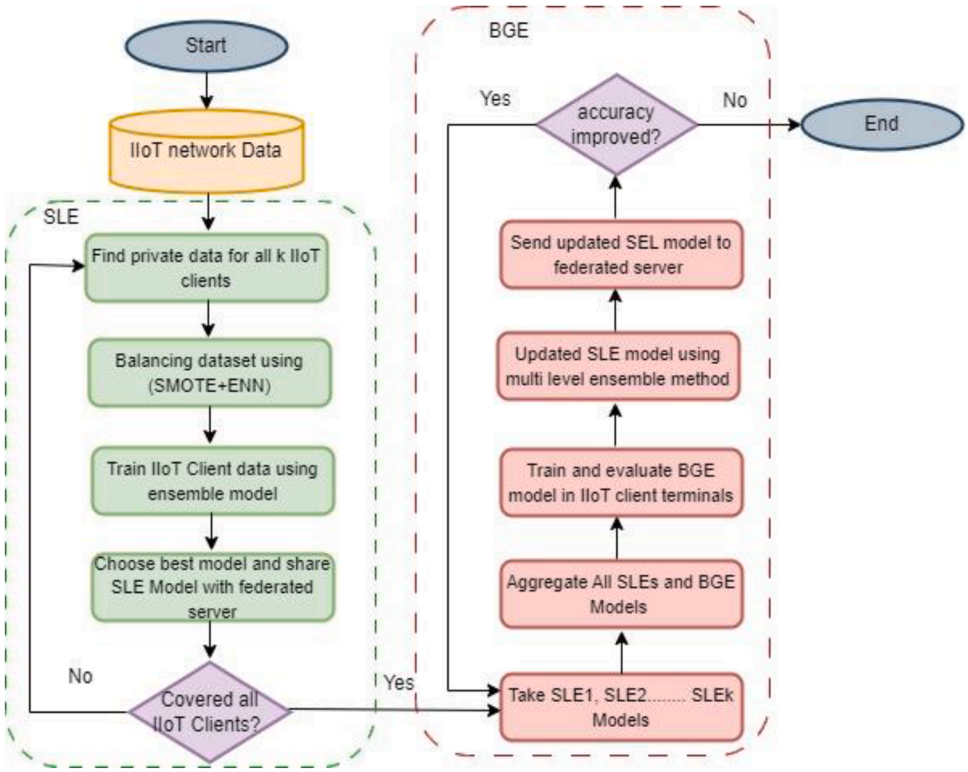


Fig. 9. Workflow of IIoT-IDFE Model.

is evaluated based on metrics such as accuracy, precision, recall, and F1-score. By combining the advantages of ensemble and federated learning techniques, the proposed model guarantees a thorough approach to produce reliable aggregated predictions at the global scale.

Locally, SLE Model is executed which uses four different ensemble methods i.e. Light Gradient Boosting, XGBoost, CatBoost, and Random Forest (explained in Section 3.1.) for intrusion detection in an IoT-enabled environment. In the first round, initially a Global Model is shared with IIoT clients so that they can create an SLE model for round 1. From Round 1, the best ensemble model is selected which is shared with the central federated server which is further used to create a BGE Model for the next round. The workflow of BGE Model is explained in Fig. 9. Steps for BGE model are explained as follows:

Step 1: BGE Model is trained and tested in round (r-1).
Step 2: BGE Model from round (r-1) is shared with IIoT client in round r. (where r represents no. of rounds for complete execution in the IIoT-IDFE model).
Step 3: In round (r-1), for every IIoT Client (IC) node (where, IC1, IC2, IC3…….ICk are k no. of IIoT clients) a multi-level ensemble technique is used at federated server to update the BGE Model for round r. This is done by averaging the federated results of the IIoT client models (SLE_1, SLE_2, …… SLE_k) for round (r-1) and BGE Model for round (r-1). Eq. (6) shows the federated averaging done at the federated server to update the global model.

$$W_p = \sum_{k=1}^{K} \frac{s_k}{s} w_p^k \tag{6}$$

Where, p is the size of all IIoT client's datasets in total, s_k is the size of the single IIoT client dataset. W_p represents the updated weight of global model after round (r-1).

Federated averaging approaches guarantee that the BGE model respects data ownership and privacy requirements while learning from the aggregate knowledge of all clients.

Step 4: The updated SLE model is used to detect anomalies at IIoT client personal dataset.
Step 5: Now, updated (SLE1, SLE2, …… SLEk) models in (r-1) are used for aggregation to update BGE model for round r.
Step 6: Step 2,3,4,5 kept on repeating until further improvement is stopped in federated rounds.

Through the use of ensemble techniques (LightGBM, CatBoost, XGBoost, Random Forest), the proposed method IIoT-IDFE seeks to improve model generalization and prediction accuracy across decentralized and possibly heterogeneous data sources by utilizing their various advantages within a federated learning framework.

Combined federated-ensemble learning model unlocks powerful capabilities for distributed, privacy preserving, and robust machine learning model. An adaptive federated-ensemble architecture helps in designing a dynamic architecture that adapts the ensemble size, composition and aggregation methods based on resource constraints, client diversity and task requirements. Implementing hierarchical ensembles where local models contribute to regional ensembles, and regional ensembles aggregate into a global ensemble. Combining local learning with global insights bridges privacy and performance. It offers explainable, robust, and secure decision-making for critical applications.

Device heterogeneity refers to the variability in computational power, memory, and communication capabilities across devices. FL manages devices and system heterogeneity through several techniques:

- **Asynchronous Federated Learning:** Instead of waiting for updates from all devices, the central server can proceed with updates from a subset of devices, incorporating asynchronous model aggregation. This reduces latency and enables slower devices to participate without stalling the entire process.
- **Device Selection:** The server can dynamically select a subset of devices for each training round based on availability, performance, or other criteria (e.g., minimizing energy consumption or ensuring fairness).
- **Compression and Efficient Communication:** Techniques like model pruning, quantization, or gradient sparsification are employed to reduce the size of model updates, making them suitable for devices with limited bandwidth or processing power.
- **Adaptive Workload:** Devices can be assigned workloads based on their capabilities, such as fewer local epochs or smaller batch sizes for less powerful devices.

Data heterogeneity arises because data across devices is usually non-independent and identically distributed, unbalanced, and sparse. FL manages data heterogeneity through:

- **Federated Averaging (FedAvg):** FedAvg combines model updates from devices through weighted averaging, where weights are proportional to the number of data samples on each device. This helps balance the contributions of devices with varying dataset sizes.
- **Personalization Layers:** To handle diverse data distributions, FL can include personalization layers or fine-tuning mechanisms. While the base model is trained collaboratively, devices adapt the model locally to better suit their specific data.
- **Clustered Federated Learning:** Devices with similar data distributions are grouped into clusters, and separate models are trained for each cluster. This approach reduces the impact of non-IID data by focusing on sub-populations with shared characteristics.
- **Regularization Techniques:** Techniques like **FedProx** (Federated Proximal) add a regularization term to the local loss function, penalizing drastic deviations from the global model. This helps stabilize training under non-IID conditions.
- **Data Augmentation and Synthesis:** Devices or the central server can generate synthetic data or augment existing data to reduce the disparity between datasets and enhance the model's generalization.

4.2.1. Parameter optimization for heterogeneous data

FL optimizes model parameters considering both system and data heterogeneity by leveraging advanced techniques:

- **Adaptive Learning Rates:** The learning rate can be adjusted dynamically for individual devices based on their data characteristics or the magnitude of gradient updates.
- **Weighted Aggregation:** Model updates are aggregated using a weighted scheme based on the amount and quality of data or the reliability of devices. This ensures that updates from devices with sparse or low-quality data have a limited impact.
- **Gradient Clipping:** To prevent large updates from devices with high variance data, gradient clipping is applied to normalize contributions and stabilize convergence.
- **Gradient Divergence Minimization:** Algorithms such as FedNova (Federated Normalized Averaging) normalize gradient contributions to mitigate the effects of different data distributions or varying numbers of local epochs across devices.
- **Scalable and Robust Aggregation:** Advanced aggregation methods like **median-based aggregation, Krum**, or **trimmed mean** help

deal with outliers, noisy updates, and adversarial behavior, ensuring robust parameter updates.

Federated Learning employs a suite of strategies to address the diverse challenges of heterogeneity. These include adaptive aggregation, asynchronous updates, regularization techniques, and efficient communication protocols. By accounting for differences in device capabilities and data distributions, FL ensures scalable, robust, and fair model training in heterogeneous environments.

5. Results and discussion

This section delves into the outcomes of the present study and engage in a comprehensive analysis of the obtained experimental results, comparative analysis, exploring their significance, and potential implications.

5.1. Dataset used

In the present paper, a freely accessible industrial dataset called "Edge-IIoTset" and "ToN-IoT" has been used to implement the proposed intrusion detection method [30]. "Edge-IIoTset" contributes to the research related to cybersecurity in IIoT. M. A. Ferrag et al. [30] created a realistic testbed which aims to emulate the real-world system of IIoT to carry out cyber-attacks as close as possible to the real world. Authors created a testbed consisting of 7 layers i.e. cloud computing layer, NFV layer, Blockchain layer, fog layer, SDN layer, edge layer, and IoT/IIoT perception layer. Each layer uses different hardware and operating system. IoT/IIoT perception layer used multiple IoT Sensors and Actuators to create Edge-IIoTset Dataset. Industrial sensors, equipment, and edge computing nodes installed in manufacturing plants and other industrial facilities are the main sources of Edge-IIoT datasets. Different devices used in this dataset are given as:

- Industrial sensors and devices: Edge-IIoT dataset is frequently sourced from a variety of industrially implanted sensors and devices. These sensors could be flow meters, vibration sensors, sensors for pressure, sensors that measure temperature, etc.
- Equipment Data: Information gathered from industrial workplaces and manufacturing plants' machinery, equipment, and production lines. Data from SCADA (Supervisory Control and Data Acquisition) systems, PLCs (Programmable Logic Controllers), and other control systems are included in EdgeIIoT dataset.
- Edge computing devices: Information created and gathered at industrial network edge computer nodes. Data from edge servers, gateways, and other edge devices that perform data preprocessing prior to transmission to centralized systems are included in Edge-IIoT dataset.
- IoT Devices: Information gathered from a range of IoT devices used in industrial environments, including smart cameras, barcode scanners, and RFID tags for computerization and surveillance.
- Simulation and Synthetic Data: To augment real-world data, Edge-IIoT contains simulated or synthetic data, which enables researchers to test algorithms in various contexts.

Because of the constant stream of sensor data, the 'Edge-IIoTset' dataset ranges in size from modest to large-scale, with quantities ranging from gigabytes to terabytes. They primarily include time-series data, which is crucial for industrial automation activities including monitoring and predictive maintenance. It contains 1,57,800 rows (observations or records) including 1,107,448 normal samples and 87,016 malicious or attack samples which have been collected based on 41 features. The dataset 'Edge-IIoTset' of size 2.7 GB has been collected and developed using a testbed emulator at Washington University at St. Louis. Scale of the dataset can be described as

- Volume: Depending on how many sensors, devices, and data points are gathered over time, Edge-IIoT datasets can be categorized into moderately sized or large-scale.
- Velocity: Due to the dynamic nature of industrial processes, data typically occurs in real-time or very close to real-time from sensors and devices. High data intake velocities need the use of effective data processing and evaluation methods.
- Variety: A variety of data kinds are displayed in dataset, including time-series data, numerical sensor readings, categorical data from equipment status, and even unstructured data from operator notes or maintenance logs.
- Veracity: - Because noise, anomalies, missing values, and sensor errors are frequent in industrial settings, it is imperative to ensure data quality and dependability in 'Edge-IIoTset' dataset.

In order to manage anomalies, guarantee data quality, and enable applications like anomaly detection and predictive maintenance, characteristics like data velocity, diversity, and veracity provide problems that call for strong data processing and analysis approaches. Given the sensitivity of industrial data, security and privacy issues are critical, requiring strict procedures to preserve the confidentiality and integrity of data in edge computing contexts. 'Edge-IIoTset' dataset is an essential tool for IoT-driven intrusion detection research.

The dataset contains a set of attributes: Four types of popular cyber-attacks have been used in this scenario, namely, 'Backdoor', 'Command Injection', 'DoS', and 'Reconnaissance'. The dataset is freely accessible and can be downloaded from the link: "https://www.kaggle.com/code/waleedgul/predict-attack-and-attack-type". Table 1 shows dataset descriptions and Fig. 10 depicts the statistical distribution of multiple classes in dataset using pie chart. Different classes of attacks present in the dataset are shown in Fig. 11.

5.2. Evaluation parameters

The aim of present study is to correctly predict and detect type of traffic present in the IIoT network traffic using machine learning classifiers. Here, ensemble-based machine learning techniques have been used for multi-class classifiers have been used to assess confusion matrix with true positive (TP), false negative (FN), true negative (TN), and false negative (FN) prediction values. The proposed intrusion detection model has been evaluated on "Edge-IIoTset" and "ToN-IoT" datasets using various evaluation metrics, namely, accuracy, precision, recall, and F1-score.

Accuracy: Accuracy refers to the percentage of traffic samples correctly classified as normal and anomalous. Accuracy can be obtained using the mathematical formula given in Eq. (7).

$$\text{Accuracy} = \frac{\text{true positive } (TP) + \text{ true negative}}{\text{true positive } + \text{ false negative } + \text{ true negative } + \text{ false negative}} \tag{7}$$

Sensitivity or Recall: Sensitivity which is also called recall refers to the percentage of network traffic samples out of anomalous samples

Table 1
Dataset Description.

Dataset Name	Edge-IIoTset
No. of observations (records/ rows)	157,800
No. of features (Samples)	63
Traffic (Normal and Attack)	'Backdoor', 'DDoS_HTTP', 'DDoS_ICMP', 'DDoS_TCP', 'DDoS_UDP', 'Fingerprinting', 'MITM', 'Normal', 'Password', 'Port_Scanning', 'Ransomware', 'SQL_injection', 'Uploading', 'Vulnerability_scanner', 'XSS'
Distribution of Normal and Attack Samples	Normal= 84.6% and Attack= 15.4%

classified correctly as anomalous. Sensitivity or recall can be calculated as given in Eq. (8).

$$Sensitivity\ or\ Recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{8}$$

Specificity: Specificity is the proportion of samples accurately categorized as normal out of the entire volume of normal traffic. Eq. (9) provides the mathematical representation used to calculate sensitivity.

$$Specificity = \frac{true\ negative}{true\ negative + false\ positive} \tag{9}$$

Precision: The percentage of abnormal samples among all samples that are accurately categorized as abnormal is known as precision. Eq. (10) represents the mathematical way of calculating precision.

$$Precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{10}$$

F1-Score: F1-score is harmonic mean of precision and recall which can be calculated using computational formula given in Eq. (11).

$$F1 - Score = 2 * \frac{\text{Precision} * \text{Recall}}{Precision + Recall} \tag{11}$$

5.3. Experimental analysis

An experimental analysis is conducted in this section to validate the proposed approach. Firstly, the selected dataset and performance metrics have been illustrated. Then the results obtained by the experiment have been analyzed and compared.

Experimental Setup: Creating a federated learning setup involves several steps, including simulating the distributed environment, training local models on client devices, and aggregating the updates on a central server. The experimental implementation of proposed work has been implemented on Python 3 programming language. In order to implement the proposed scheme, some necessary python libraries have been employed. Numpy and Panda provide powerful tools. Numpy offers highly optimized and efficient multidimensional array operations for computations. These are stored in contiguous memory blocks. Panda is a versatile library that offers efficient data manipulation and analysis for efficient operations in data structures. Additionally, Keras and

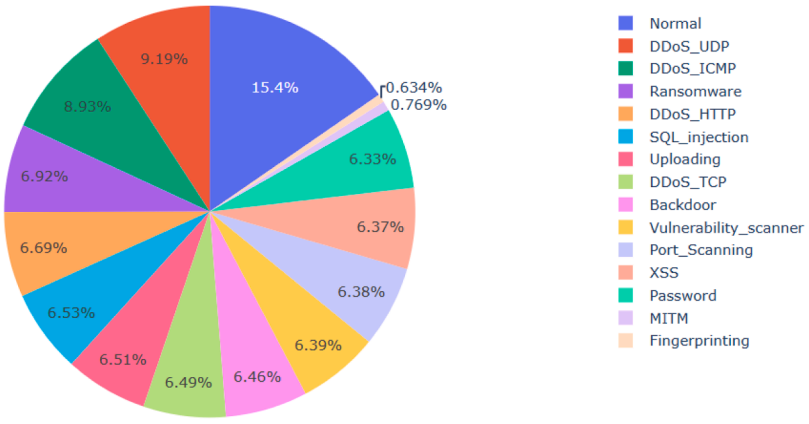


Fig. 10. Statistical distribution of Classes of Edge-IIoTset.

```
Index(['frame.time', 'ip.src_host', 'ip.dst_host', 'arp.dst.proto_ipv4',
       'arp.opcode', 'arp.hw.size', 'arp.src.proto_ipv4', 'icmp.checksum',
       'icmp.seq_le', 'icmp.transmit_timestamp', 'icmp.unused',
       'http.file_data', 'http.content_length', 'http.request.uri.query',
       'http.request.method', 'http.referer', 'http.request.full_uri',
       'http.request.version', 'http.response', 'http.tls_port', 'tcp.ack',
       'tcp.ack_raw', 'tcp.checksum', 'tcp.connection.fin',
       'tcp.connection.rst', 'tcp.connection.syn', 'tcp.connection.synack',
       'tcp.dstport', 'tcp.flags', 'tcp.flags.ack', 'tcp.len', 'tcp.options',
       'tcp.payload', 'tcp.seq', 'tcp.srcport', 'udp.port', 'udp.stream',
       'udp.time_delta', 'dns.qry.name', 'dns.qry.name.len', 'dns.qry.qu',
       'dns.qry.type', 'dns.retransmission', 'dns.retransmit_request',
       'dns.retransmit_request_in', 'mqtt.conack.flags',
       'mqtt.conflag.cleansess', 'mqtt.conflags', 'mqtt.hdrflags', 'mqtt.len',
       'mqtt.msg_decoded_as', 'mqtt.msg', 'mqtt.msgtype', 'mqtt.proto_len',
       'mqtt.protoname', 'mqtt.topic', 'mqtt.topic_len', 'mqtt.ver',
       'mbtcp.len', 'mbtcp.trans_id', 'mbtcp.unit_id', 'Attack_label',
       'Attack_type'],
      dtype='object')
```

Fig. 11. Set of Features of Edge-IIoTset.

Tensorflow have been used for machine learning operations. Scikit learn has been used for providing unified and consistent interface as well as rich functionality to machine learning.

Python 3 was the programming language we used for our experiments on Jupyter notebook. NumPy and other popular libraries are used, together with multi-dimensional arrays and matrices, to build our method. We also made use of the robust data-structure modification as well as analysis tools provided by Pandas. Moreover, Keras and TensorFlow are utilized for ML and DL. Furthermore, Scikit-learn offers a wide range of supervised and unsupervised machine learning algorithm implementations. Moreover, we oversampled minority classes using SMOTE and under-sampled majority classes using ENN to raise the overall model efficiency.

Dataset Preparation, Preprocessing and feature Selection: Preprocessing of raw data needs to be carried out before moving toward experimental processes and it is an essential step to improve the performance of the operation applied on the dataset and to reduce the complexity. The operations applied during preprocessing are as follows:

***Data Cleaning*:** Data cleaning refers to the process of removing the less prominent rows with missing values, irrelevant values, or noisy data. The irrelevant data might be either removed or replaced with suitably fitted values. In the programming part, the data has been grouped as well as duplicates, noisy and missing values were removed like "NAN" and infinite values and replaced with relevant and fit data values. Some additional features were removed that are not much helpful in prediction, using "data.drop()" function. Features removed are "frame.time", "ip.src_host", "ip.dst_host", "arp.src.proto_ipv4","arp.dst.proto_ipv4", "http.file_data","http.request.full_uri","icmp.transmit_timestamp","http.request.uri.query","tcp.options","tcp.payload","tcp.srcport", "tcp.dstport", "udp.port", "mqtt.msg", "icmp.unused", "http.tls_port", 'dns.qry.type', 'dns.retransmit_request_in', "mqtt.msg_decoded_as", "mbtcp.trans_id", "mbtcp.unit_id", "http.request.method", "http.referer", "http.request.version", "dns.qry.name.len", "mqtt.conack.flags", "mqtt.protoname", "mqtt.topic".

***Feature Engineering*:** Feature engineering refers to the process of transforming raw data into suitable features. Feature engineering is also called feature extraction. It aims to build more accurate and efficient model. The most relevant and promising features need to be selected, extracted, and transformed from the available data using feature selection method. Less significant features can be removed from the dataset and essential practical operations are performed only using most significant features.in order to reduce the complexity. Other benefits of feature selection are: speeding learning algorithm, reduce dimensionality of feature space, improved predictive accuracy in classification, and improved comprehensibility of results. Some operations of feature engineering includes feature selection, feature scaling, etc. Dimensionality reduction is a useful approach for data compression in order to reduce the space complexity as well as the computation time in case of large numbers of features. High-dimensional dataset with large number of features can increase space and time complexity. A new set of features can be obtained by removing unnecessary features. Remove columns (features) by specifying label names or column names using dataframe. drop() function.

Train-set and Test-set: The data need to be split or divided into features (X) and labels (y). Data frames are divided into X_train, X_test, y_train, and y_test. Different combinations of train and test data have been used in various rounds of experiments. Entire data have been split using **train_test_split(X, y, test_size=0.2, random_state=0)** function. The test_size feature scaling takes float or int value that should be between 0.0 and 1.0. It represents the proportion of the dataset to be included in test split. Random_state controls the shuffling before split. A portion of global dataset has been divided into training and test samples and assigned to different client nodes in different proportions.

Randomly selected data distribution of various traffic classes has been shown in Table 2. After data-preprocessing and feature engineering, it has been divided into training and test sets are genearted as shown in Table 3. To strongly evaluate the proposed model, it has also been implemented an another IIoT dataset i.e. 'ToN_IoT'. Table 4 specified the details of dataset "ToN_IoT" with details like features (Input and output) and instances.

Development of Proposed IIoT-IDFE model: After preparation and preprocessing of raw data present in the dataset chosen for practical implementation, it is trained and tested using one or more classifiers. In the present study, the efficiency and benefits of the federated learning approach have been showcased. The global IIoT dataset has been divided into training samples and testing samples in different percentage ratios. The considered training dataset has been divided into five subsets each containing 20% of traffic data samples.

Result Analysis: The main goal of the proposed work is to present a mechanism to correctly detect threats present in the traffic of IIoT network. This model shows how federated learning and ensemble-based machine learning approaches together create a promising method for sharing data with privacy. Threat detection can be used for various applications including digital forensics, information security, ethical hacking etc. where privacy is a crucial element to be maintained. In this study, the authors suggest an approach for threat detection by evaluating the federated + ensemble learning model by evaluating it on two distinct industrial IoT datasets using different metrics. The decentralized Federated learning approach has segregated global data into different combinations of training and test data samples, each representing the data of different nodes. In the present scheme, the global raw data has been balanced using different scaling and normalizing techniques to obtain an efficient intrusion detection mechanism.

Creating a federated and ensemble-based combined model involves aggregating predictions or model parameters from multiple models trained on different nodes or clients in a federated learning setup. LightGBM is a highly efficient gradient boosting framework in terms of time efficiency and accuracy that is already designed to handle large datasets and can be adapted for federated learning.

Classifiers: The proposed IIoT-IDFE Model is evaluated in terms of accuracy, precision, recall, and f1-score. In evaluating the federated ensemble combined model over multiple rounds, several metrics such as accuracy, precision, and recall are crucial for understanding its performance and robustness across a distributed environment. The model was trained and evaluated using a federated setup simulated across virtual workers (clients) with distinct datasets, ensuring data privacy and security.

The results of the classifiers have been optimized by tuning the hyperparameters of these classifier algorithms. Table 5 provides the results of all the ensemble classifiers used in first round of Federated Learning. Here, the entire dataset has been distributed among five IIoT clients and each client used 20% samples with the same features. The results have been obtained using four classifiers, namely, random forest, XgBoost, Light Gradient Boost, and CatBoost. In case of IIoT client 1, the

Table 2
Data Distribution for Edge_IIoTset Dataset in Training and Testing.

Attack Classification	Total	Train	Test
Normal	19,380	15,504	3876
DDoS_UDP	11,593	9274	2319
DDoS_ICMP	11,268	9014	2254
Ransomware	8753	7002	1751
DDoS_HTTP	8434	6747	1687
DDoS_TCP	8262	6609	1653
SQL_injection	8254	6603	1651
Uploading	8228	6582	1646
Port_Scanning	8111	6489	1622
Backdoor	8102	6481	1621
Vulnerability_scanner	8059	6447	1612
XSS	8028	6422	1606
Password	7977	6381	1596
MITM	981	785	196
Fingerprinting	810	648	162

Table 3
Description of Split dataset (Train and Test) on six different client nodes.

No. of Instances		2000		4000		5000	
Set	Class	R1 (70:30)	R2 (80:20)	R3 (70:30)	R4 (80:20)	R5 (67:33)	R6 (80:20)
Train	Attack + Normal	1400	1600	2800	3200	3350	4000
Test	Attack + Normal	600	400	1200	800	1650	1000

Table 4
Details of "ToN_IoT" Dataset.

Datasets	Features	No. of Instances	Input Features	Output Feature = 'type' (Classes)
TON_IoT (IoT_Fridge)	6	587,077	'date', 'time', 'fridge_temperature', 'temp_condition', 'label'	'normal', 'backdoor', 'ddos', 'injection', 'password', 'ransomware', 'xss'
TON_IoT (IoT_Garage_Door)	6	591,447	'date', 'time', 'door_state', 'sphone_signal', 'label'	'normal', 'backdoor', 'ddos', 'password', 'injection', 'scanning', 'ransomware', 'xss'
TON_IoT (IoT_GPS_Tracker)	6	595,687	'date', 'time', 'latitude', 'longitude', 'label'	'normal', 'backdoor', 'ddos', 'injection', 'password', 'ransomware', 'scanning', 'xss'
TON_IoT (IoT_Motion_Light)	6	452,263	'date', 'time', 'motion_status', 'light_status', 'label'	'backdoor', 'ddos', 'injection', 'normal', 'password', 'ransomware', 'scanning', 'xss'
TON_IoT (IoT_Thermostat)	6	442,229	'date', 'time', 'current_temperature', 'thermostat_status', 'label'	'backdoor', 'injection', 'normal', 'password', 'ransomware', 'scanning', 'xss'
TON_IoT (IoT_Weather)	7	650,243	'date', 'time', 'temperature', 'pressure', 'humidity', 'label'	'normal', 'backdoor', 'ddos', 'injection', 'password', 'ransomware', 'scanning', 'xss'

Table 5
Classification of Edge_IoT Dataset for round R1.

IIoT Clients	Ensemble Model	Precision	Recall	F1-score	Accuracy
IIoT Client 1	RF	0.93	0.92	0.93	0.93
	XGB	0.97	0.96	0.95	**0.97**
	LGB	0.97	0.95	0.96	0.96
	CB	1.00	1.00	1.00	**1.00**
IIoT Client 2	RF	0.95	0.94	0.94	0.96
	XGB	0.96	0.97	0.96	0.96
	LGB	0.98	0.98	0.97	**0.98**
	CB	0.97	0.97	0.96	**0.97**
IIoT client 3	RF	0.93	0.94	0.93	0.92
	XGB	0.98	0.99	0.98	**0.99**
	LGB	0.95	0.96	0.96	0.95
	CB	0.98	0.99	0.98	**0.98**
IIoT client 4	RF	0.92	0.93	0.93	0.93
	XGB	0.97	0.97	0.98	**0.98**
	LGB	0.94	0.93	0.93	0.93
	CB	0.99	0.98	0.98	**0.99**
IIoT client 5	RF	0.95	0.93	0.92	0.94
	XGB	0.98	0.97	0.98	**0.97**
	LGB	0.96	0.97	0.95	0.95
	CB	0.99	0.98	1.00	**0.99**

CatBoost classifier returns the best accuracy, precision, recall, and f1-score with 100% result. In the second node client2, the best accuracy and recall has been obtained using the XGBoost classifier. At the machine of client 3, again the best accuracy results have been obtained using XGBoost and CatBoost algorithms i.e. 99% accuracy. At IIoT client 4 end, 98% accuracy has been obtained using XGB classifier and 99% has been obtained using CatBoost classifiers. In the same way, these two algorithms XGB and CatBoost outperform at client 5 node.

Fig. 12 shows the evaluation matrices for every client (when $k = 5$, in current scenario). As shown by Fig. 12, there is not much difference in the performance of different ensemble learning classifiers, but XGBoost and CatBoost classifiers outperform as compared to other classifiers in most of the rounds.

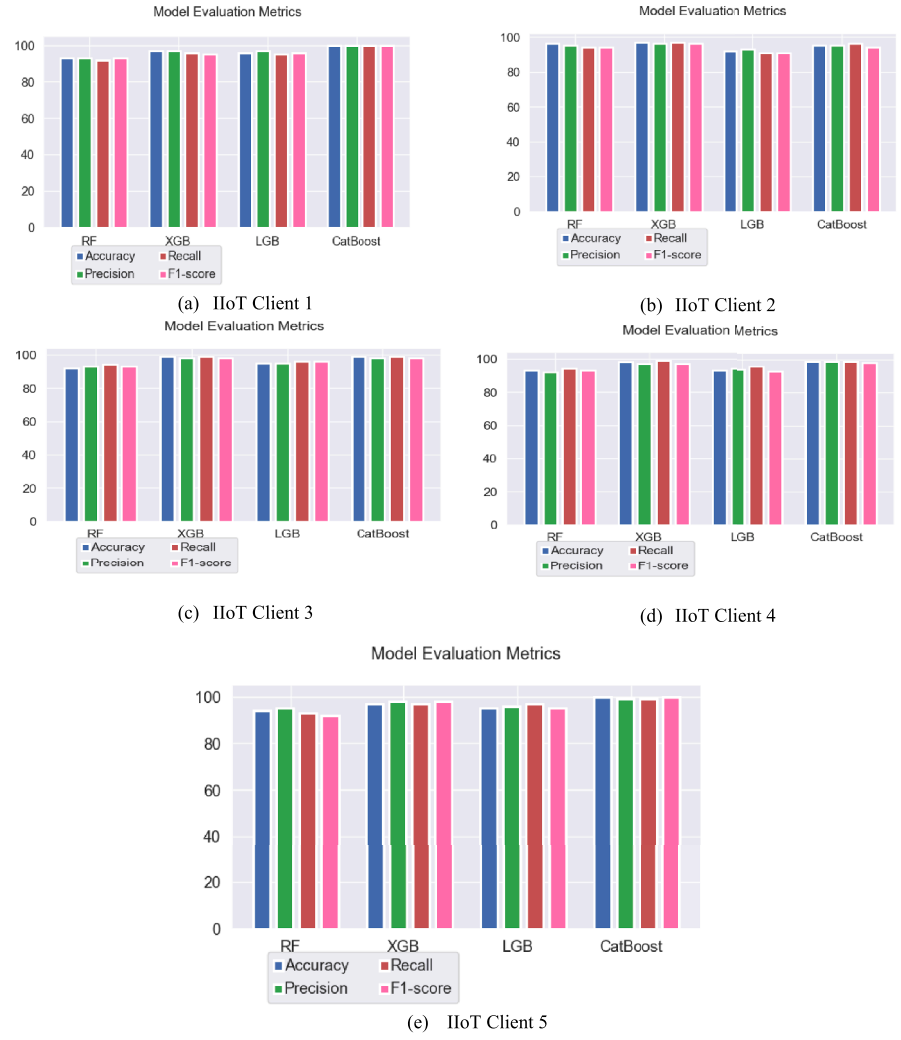
The main goal of this study is to select the best classifier algorithm. On the basis of results obtained using various ensemble algorithms applied at each local node, the best learning algorithms can be identified. Best learning models obtained at each node are shared with the federated server. These ensemble models are aggregated at the federated server node in order to construct a new global learning model. The global learning model with updated parameters is sent back to each local client node to train the private data of each client node.

Table 5 presents how global federated weights are calculated using the proposed model. The best accuracy results are taken from R1 and then in form of weight, the values are forwarded to R2. These results are aggregated in order to obtain global federated results which are further forwarded to next round R3. Again, the best results in terms of accuracy are calculated and forwarded to next level. In this way, the same procedure is executed until we get the stable and best global accuracy results. Means the accuracy results of round n should be matched with round n-1. As shown in Table 6, total 15 rounds have been executed. The accuracy values of various clients are calculated upto reaching to the best results. The final performance of the global federated server model in terms of accuracy reaches to 99% which is better that earlier round. However, in some rounds including round 1, CatBoost outperforms and returns accuracy 100%. But that is not stable performance.

In another experiment, we simulate the practical with "ToN-IoT" dataset. This practical implementation has been carried out to strongly validate the performance of the proposed model. Although "ToN-IoT" is a set of several datasets containing records extracted from fridge, garage door, GPS tracker, motion light, thermostat and weather monitoring device in IoT-enabled smart environment. Like the performance evaluated on "Edge-IIoT" dataset, the model's performance will be evaluated on "ToN-IoT" using several parameters. The classification results for this dataset have been presented in Table 7.

The goal of the proposed Federated Ensemble Model is to choose the best two models that have been developed by applying local ensemble learning approach for every node. The results obtained from local approaches are evaluated in three rounds to evaluate the goodness of the proposed model. This model works on each dataset contained in "ToN-IoT". The results computed in different rounds have been presented in Table 7. This table presents the results obtained and analyzed in round 1. The results in terms of F1-score, accuracy, precision and recall have been calculated and analyzed. The results have been evaluated in three rounds for each client node where best results obtained from one round are passed to next consecutive round.

Table 8 shows the results for Ensemble Federated Learning Algorithm with n rounds in ToN_IoT Dataset. After analyzing the accuracy results for ToN_IoT_Fridge, in round R2 it is found that using XGB 99.2% data is accurately classified and using LGB it is 99.2%, while XGB



(a) IIoT Client 1 (b) IIoT Client 2

(c) IIoT Client 3 (d) IIoT Client 4

(e) IIoT Client 5

Fig. 12. Model evaluation metrics for IIoT clients.

Table 6
Results of Ensemble Federated Learning Algorithm with n rounds.

Round	R2			R14		R15	
Client	Model	Accuracy	.....	Model	Accuracy	Model	Accuracy
C1	Catboost	1.00	.....	CatBoost	1.00	CatBoost	1.00
C2	LGB	0.98	.....	CatBoost	0.98	CatBoost	0.98
C3	XGB	0.99	.....	LGB	0.97	LGB	0.97
C4	CatBoost	0.99	.....	CatBoost	1.00	CatBoost	1.00
C5	XGB	0.97	.....	XGB	0.99	XGB	0.99
Server	Global Federated R1	0.98	.....	Global Federated R13 or R_{n-2}	0.99	Global Federated R_{n-1}	0.99

classifies 98.9% data truly. In the case of client C2, using XGB, 96.9% data is truly classified. Here, CatBoost outperforms, which classifies 99.9% data. In the case of client node C3, XGB and CatBoost gives 99.9 and 99.8% accuracy respectively. The objective of federated ensemble round to select the best two models on the basis of local ensemble results. These results are combined to obtain a new combined model which plays role in result sharing with federated server. These results are presented in round 2 which are trained using federated ensemble. The same procedure has been deployed on "ToN_IoT (Garage_Door) dataset". The same process can be applied on rest of the datasets in "ToN_IoT".

5.4. Comparative analysis

The performance and efficiency of the proposed model have been compared with existing models proposed for intrusion detection. Rashid et al. [35] evaluated the proposed federated learning model on Edge-IIoTset to evaluate its efficiency, reliability and effectiveness. Although the accuracy achieved is a little lesser than a centralized learning-based model, but the authors mainly focused on privacy and security. Farahani et al. [2] also used the federated learning technique for designing the model of privacy preservation in the business domain. Attota et al. [38] also used federated learning with the assistance of ML to predict anomalies in MQTT-IoT dataset and obtained an accuracy 94.1%. Dina et al. used FNN and CNN algorithms in designing the proposed intrusion detection model and obtained a performance near to 93%. M.M. Rashid et al. [24] also proposed a model for intrusion detection using federated learning with weighted federated aggregation of the CNN/RNN models' classification and obtained an accuracy score

Table 7
Classification of Ton_IoT Dataset for round R1.

	IIoT Clients	Ensemble Model	Precision	Recall	F1-score	Accuracy
ToN_IoT (IoT_Fridge)	IIoT Client 1	RF	0.856	0.835	0.846	0.884
		XGB	0.976	0.965	0.979	**0.998**
		LGB	0.923	0.956	0.955	0.942
		CB	0.984	0.995	0.987	**0.998**
	IIoT Client 2	RF	0.866	0.865	0.836	0.894
		XGB	0.956	0.935	0.959	**0.969**
		LGB	0.923	0.956	0.955	0.942
		CB	0.954	0.945	0.956	**0.971**
	IIoT Client 3	RF	0.799	0.796	0.792	0.792
		XGB	0.898	0.899	0.898	**0.899**
		LGB	0.887	0.896	0.897	0.895
		CB	0.899	0.889	0.898	**0.898**
ToN_IoT (IoT_Garage_Door)	IIoT Client 1	RF	0.913	0.912	0.923	0.923
		XGB	0.977	0.976	0.973	**0.977**
		LGB	0.967	0.965	0.964	0.967
		CB	0.969	0.988	0.987	**0.989**
	IIoT Client 2	RF	0.925	0.914	0.924	0.926
		XGB	0.946	0.947	0.946	0.946
		LGB	0.948	0.968	0.967	**0.968**
		CB	0.957	0.957	0.956	**0.957**
	IIoT Client 3	RF	0.893	0.894	0.893	0.892
		XGB	0.938	0.949	0.938	**0.959**
		LGB	0.945	0.946	0.956	0.955
		CB	0.948	0.959	0.958	**0.958**
ToN_IoT (IoT_GPS_Tracker)	IIoT Client 1	RF	0.893	0.892	0.893	0.893
		XGB	0.974	0.966	0.975	**0.977**
		LGB	0.971	0.954	0.962	0.961
		CB	0.982	0.988	0.978	**0.989**
	IIoT Client 2	RF	0.965	0.954	0.958	0.963
		XGB	0.986	0.978	0.966	0.976
		LGB	0.989	0.988	0.976	**0.989**
		CB	0.988	0.987	0.986	**0.987**
	IIoT Client 3	RF	0.935	0.941	0.937	0.946
		XGB	0.988	0.993	0.984	**0.991**
		LGB	0.975	0.976	0.968	0.985
		CB	0.982	0.993	0.987	**0.984**
ToN_IoT (IoT_Motion_Light)	IIoT Client 1	RF	0.913	0.932	0.931	0.933
		XGB	0.967	0.956	0.958	**0.974**
		LGB	0.957	0.955	0.964	0.976
		CB	0.962	0.943	0.937	**0.968**
	IIoT Client 2	RF	0.895	0.924	0.946	0.966
		XGB	0.986	0.977	0.965	0.976
		LGB	0.998	0.989	0.971	**0.985**
		CB	0.987	0.988	0.988	**0.989**
	IIoT Client 3	RF	0.939	0.946	0.957	0.968
		XGB	0.983	0.974	0.978	**0.981**
		LGB	0.958	0.986	0.969	0.953
		CB	0.981	0.986	0.958	**0.983**
ToN_IoT (IoT_Thermostat)	IIoT Client 1	RF	0.937	0.929	0.963	0.976
		XGB	0.983	0.985	0.985	**0.986**
		LGB	0.972	0.955	0.962	0.969
		CB	0.946	0.963	0.959	**0.997**
	IIoT Client 2	RF	0.953	0.947	0.954	0.968
		XGB	0.961	0.978	0.946	0.969
		LGB	0.981	0.968	0.973	**0.986**
		CB	0.973	0.967	0.966	**0.999**
	IIoT Client 3	RF	0.939	0.947	0.936	0.929
		XGB	0.983	0.993	0.968	**0.979**
		LGB	0.954	0.961	0.968	0.953
		CB	0.985	0.991	0.987	**0.989**
ToN_IoT (IoT_Weather)	IIoT Client 1	RF	0.938	0.929	0.917	0.978
		XGB	0.977	0.976	0.968	**0.978**
		LGB	0.977	0.978	0.979	0.979
		CB	0.978	0.978	0.963	**0.995**
	IIoT Client 2	RF	0.915	0.924	0.934	0.926
		XGB	0.962	0.957	0.961	0.963
		LGB	0.989	0.998	0.979	**0.998**
		CB	0.989	0.989	0.989	**0.998**
	IIoT Client 3	RF	0.935	0.921	0.935	0.928
		XGB	0.968	0.979	0.978	**0.999**
		LGB	0.965	0.967	0.987	0.965
		CB	0.968	0.979	0.989	**0.999**

Table 8
Results of Ensemble Federated Learning Algorithm with n rounds in ToN_IoT Dataset.

Dataset	Round	R2		R3		R4	
	Client	Model	Accuracy	Model	Accuracy	Model	Accuracy
ToN_IoT (IoT_Fridge)	C1	XGB	**0.988**	XGB	**0.998**	XGB	**0.998**
		LGB	0.982	FEL_R1	**0.992**	FEL_R1	0.985
		FEL_R1	**0.992**	FEL_R2	0.976	FEL_R3	**0.997**
	C2	XGB	**0.989**	XGB	**0.989**	XGB	0.989
		CB	**0.991**	CB	**0.991**	LGB	0.971
		FEL_R1	0.958	FEL_R2	0.897	FEL_R3	**0.991**
	C3	XGB	**0.989**	XGB	**0.997**	XGB	**0.998**
		LGB	0.958	FEL_R1	**0.982**	FEL_R1	**0.998**
		FEL_R1	**0.992**	FEL_R2	0.992	FEL_R3	0.995
ToN_IoT (IoT_Garage_Door)	C1	XGB	**0.988**	XGB	**0.988**	XGB	**0.998**
		LGB	0.958	FEL_R1	**0.992**	FEL_R1	0.962
		FEL_R1	**0.995**	FEL_R2	0.982	FEL_R3	**0.995**
	C2	XGB	**0.992**	XGB	**0.997**	XGB	0.969
		CB	**0.991**	CB	**0.991**	CB	**0.991**
		FEL_R1	0.958	FEL_R2	0.897	FEL_R3	**0.995**
	C3	XGB	**0.999**	XGB	**0.999**	XGB	**0.999**
		CB	0.958	FEL_R1	**1.000**	FEL_R1	**1.000**
		FEL_R1	**0.999**	FEL_R2	0.952	FEL_R3	0.955

Table 9
Comparative analysis of proposed model with state-of-art.

Model/ Reference	Year	Dataset	Application	Technique	Accuracy (%)
Farahani et al. [2]	2021	IIoT	Privacy-preserve	Federated learning	95.61%
Alani et al. [16]	2023	WUSTL-IIoT-2021	Intrusion Detection	ML	99.7%
Attota et al. [38]	2021	IoT-MQTT	Anomaly Detection in IoT	ML + Federated learning	94.1%
Rashid et al. [35]	2022	Edge-IIoTset	Intrusion Detection in IIoT	Federated Learning	92.49%
Dina et al. [36]	2023	WUSTL-IIoT-2021	Intrusion Detection	FNN/CNN	93.26%
Rashid et al. [24]	2022	Edge-IIoTset	Intrusion Detection in IIoT	CNN/RNN + Federated	91.87%
Priyadarshini et al. [37]	2024	NSL-KDD	Intrusion detection in IoT	ML+ Federated	98.99%
IIoT-IDFE	2024	Edge-IIoT Dataset	Intrusion Detection in IIoT	Ensemble + Federated	**99.98%**
IIoT-IDFE	2024	ToN_IIoT Dataset	Intrusion Detection in IIoT	Ensemble + Federated	**99.99% – 100.0%**

91.87%. Priyadarshini et al. [37] proposed a federated learning-based model for intrusion detection in IoT systems. The accuracy achieved by Alani et al. [16] was 99.7%.

In this study, the IIoT-IDFE Model is based on federated learning which has been assisted by ensemble learning to polish its performance. The ensemble learning scheme improves the performance of the federated learning model using a voting scheme. The proposed model has been evaluated on the 'Edge-IIoTset' dataset and 'ToN-IoT' datasets using various metrics and is compared with other existing models in Table 9.

Combined Federated ensemble model uniquely addresses existing gaps in traditional machine learning, federated learning and ensemble learning models by combining their strengths. Some specific gaps and challenges with solutions have been mentioned in Table 10:

The above features make federated ensemble models a powerful solution for modern machine learning applications, particularly in healthcare, finance, and other privacy-sensitive domains. Overall, the federated ensemble combined model showcases promising performance in a distributed learning environment, leveraging ensemble techniques to aggregate diverse model outputs while preserving data privacy. This approach not only enhances accuracy but also ensures reliable precision and recall across multiple rounds, validating its efficacy for collaborative and secure machine learning applications [43]. Federated ensemble-based threat detection models on edge devices face challenges due to the limited computational resources (e.g., memory, processing power, energy, and bandwidth) inherent in such devices [44]. The complexity of these models impacts computational efficiency, but there are various optimization strategies to address these issues. Following optimization strategies can be deployed to optimize them to fit resource-constrained devices:

- Reducing Model Complexity: Model complexity and computational requirements can be reduced by eliminating redundant weights in individual models. Pruning can be applied specifically to larger models. Reduce parameter precision for storage and computational efficiency.
- Efficient Ensemble Design: Train a smaller set of models that maintain diversity while offering similar accuracy to the full ensemble. Optimize the feature set used by the ensemble models, reducing computational and memory requirements [45].
- Optimization of Federated Learning: Only update a subset of ensemble models in each round to reduce computation and communication costs. Randomly exclude certain models or layers from participation in specific training rounds. Use intermediary hierarchical aggregation points (e.g., edge servers) to combine updates locally before sending them to a central server [46].
- Communication Efficiency: Transmit only significant gradients or model updates to reduce data size and compress updates before transmission.
- Adaptive Computation: Perform updates incrementally over time rather than processing the full dataset at once.

By reducing model complexity, optimizing communication, and leveraging collaborative frameworks, federated ensemble-based threat detection models can be adapted for resource-constrained edge devices. The goal is to balance accuracy with time efficiency, ensuring robust threat detection without overwhelming device resources. These optimizations make such models feasible and effective for real-world edge computing scenarios operated on WSN [47].

Table 10
Existing gaps and challenges with Federated-Ensemble model solution.

Features	Gaps	Federated Ensemble Model Solution
Privacy Preservation in Data Sharing [42]	Traditional ensemble methods often require centralized access to raw data, leading to privacy concerns.	Data remains decentralized on client machine. Combines locally trained models without sharing sensitive data, ensuring compliance with data protection regulations
Handling Non-IID and Heterogeneous Data	Federated learning struggles with non-iid (non-independent and identically distributed) data across clients, leading to biased or underperforming global models.	Treats locally trained models as diverse base models in the ensemble. By aggregating predictions from these varied models, it naturally accommodates data heterogeneity and performs well across diverse client datasets.
Mitigating Single Model Limitations	Centralized or federated models often face limitations in representational power and may fail to generalize across complex datasets.	Combines multiple models into an ensemble, enhancing representational capacity and generalization ability. Leverages model diversity to overcome individual model weaknesses.
Improved Fault Tolerance [43]	In federated learning, client dropout, communication failures, or partial updates can degrade the global model's performance.	As each client contributes independently to the ensemble, the model can still perform well even if some clients drop out. Redundancy in ensemble predictions ensures robustness against client failures.
Scalability and Flexibility [44]	Traditional ensemble methods often struggle to scale to large datasets or distributed systems.	Scales naturally to a large number of distributed clients without requiring centralized data storage. Easily integrates new client models into the ensemble without retraining the entire system.
Reduction of Communication Overhead	Federated learning models often require frequent updates of global model parameters, increasing communication costs.	Requires only model outputs or weights for aggregation rather than full parameter synchronization. Reduces communication overhead compared to traditional federated learning setups.
Enhanced Robustness to Adversarial Clients	Only Federated learning can be vulnerable to adversarial attacks or poisoned data from malicious clients.	Aggregating predictions from diverse models dilutes the impact of malicious contributions.Weighted voting or model reliability assessment can further minimize adversarial effects.
Enhanced Model Interpretability	Lack of interpretability in machine learning models can hinder trust and compliance in critical IIoT security systems.	Develop explainable federated ensemble models that provide insights into why a specific security event was flagged.

6. Conclusion

Centralized machine learning and deep learning approaches face various challanges due to the necessity of sharing data with all IIoT stakeholders. Federated Learning is a feasible solution to this problem of data privacy. Centralized models pose risks to data privacy as the number of IoT-connected devices increase rapidly. The challenges of implementing prevalent centralized ML methods stem from the requirement of consolidating massive volumes of data in a single location. Consequently, federated learning emerges as a secure alternative for ensuring both privacy and security.

This study introduces IIoT-IDFE (IIoT-Intrusion Detection Federated Ensemble) Model, leveraging two ensemble models: the Shared Local Ensemble (SLE) Model and the Broadcast Global Ensemble (BGE) Model. In each federated learning training round, every IIoT client receives an updated model from the central federated server, amalgamating SLE and BGE models. This updated model is then utilized to train the local IIoT client dataset for enabling IoT devices to enhance the model collectively while maintaining their individual privacy. The technique, relying on local IoT device data for federated training, ensures confidentiality and privacy.

The model's evaluation on the publicly available "Edge-IIoTset" dataset and "ToN-IoT" dataset, containing IIoT network traffic, demonstrates improved performance compared to various state-of-the-art intrusion detection models. Developed through a federated learning and ensemble learning approach, the proposed model prioritizes data accuracy, privacy, and robustness. A limitation of the IIoT-IDFE model is its dependency on a dedicated dataset obtained from an open-source reference. Difficulty in hyperparameter tuning can also be a limitation of combined federated ensemble learning. Federated ensembles often involve tuning hyperparameters at both the local model and ensemble levels that might increase the complexity. Coordinating hyperparameter tuning across decentralized clients can be inefficient and time-consuming. These problems can be solved by choosing the right ensemble aggregation method (e.g. averaging, weighted voting, or meta-models) . While federated ensemble models hold great promise, careful design and experimentation which are essential to address these limitations and ensure robust, scalable, and privacy-preserving performance. Future endeavors aim to enhance the model's applicability by utilizing real-time datasets collected through network testbeds.

All the work is original and contributed by all the authors.

Funding

No funding required.

CRediT authorship contribution statement

Ayushi Chahal: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Preeti Gulia:** Writing – review & editing, Visualization, Validation, Supervision, Project administration, Investigation, Conceptualization. **Nasib Singh Gill:** Writing – review & editing, Validation, Supervision, Project administration. **Deepti Rani:** Writing – original draft, Visualization, Software, Resources, Investigation, Formal analysis, Data curation.

Declaration of competing interest

There is no conflict of interest.

Data availability

Data will be made available on request.

References

[1] D. Sharma, A. kumar, N. Tyagi, S.S. Chavan, S.M.P. Gangadharan, Towards intelligent industrial systems: a comprehensive survey of sensor fusion techniques in IIoT, Measure.: Sens. 32 (2024), https://doi.org/10.1016/j.measen.2023.100944.
[2] B. Farahani, A.K. Monsefi, Smart and collaborative industrial IoT: a federated learning and data space approach, Digital Commun. Netw. 9 (2023) 436–447, https://doi.org/10.1016/j.dcan.2023.01.022.
[3] R. Kanagavelu, Z. Li, J. Samsudin, S. Hussain, F. Yang, Y. Yang, R.S.M. Goh, M. Cheah, Federated learning for advanced manufacturing based on industrial IoT data analytics, in: C. Toro, W. Wang, H. Akhtar (Eds.), Implementing Industry 4.0:

The Model Factory as the Key Enabler For the Future of Manufacturing, Intelligent Systems Reference Library, Springer International Publishing, Cham, 2021, pp. 143–176, https://doi.org/10.1007/978-3-030-67270-6_6.
[4] Ma, C., Yu, H., Li, Z., Yang, Z., 2022. Federated learning framework based on data value evaluation in industrial IoT. Security and communication networks 2022, e7424094. https://doi.org/10.1155/2022/7424094.
[5] F.A.M. do Nascimento, F. Hessel, A decentralized federated learning architecture for intrusion detection in IoT systems, in: L. Barolli, F. Hussain, T. Enokido (Eds.), Advanced Information Networking and Applications, Lecture Notes in Networks and Systems, Springer International Publishing, Cham, 2022, pp. 256–268, https://doi.org/10.1007/978-3-030-99587-4_22.
[6] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A. y, 2023. Communication-efficient learning of deep networks from decentralized data. https://doi.org/10.48550/arXiv.1602.05629.
[7] G.D. Govindwar, S.S. Dhande, An approach of federated learning in artificial intelligence for healthcare analysis, in: S. Kumar, S. Hiranwal, S.D. Purohit, M. Prasad (Eds.), Proceedings of International Conference on Communication and Computational Technologies, Algorithms for Intelligent Systems, Springer Nature, Singapore, 2023, pp. 97–107, https://doi.org/10.1007/978-981-99-3485-0_8.
[8] S.I. Manzoor, S. Jain, Y. Singh, H. Singh, Federated learning based privacy ensured sensor communication in IoT networks: a taxonomy, threats and attacks, IEEE Access 11 (2023) 42248–42275, https://doi.org/10.1109/ACCESS.2023.3269880.
[9] B. Shubyn, D. Mrozek, T. Maksymyuk, V. Sunderam, D. Kostrzewa, P. Grzesik, P. Benecki, Federated learning for anomaly detection in industrial IoT-enabled production environment supported by autonomous guided vehicles, in: D. Groen, C. de Mulatier, M. Paszynski, V.V. Krzhizhanovskaya, J.J. Dongarra, P.M.A. Sloot (Eds.), Computational Science – ICCS 2022, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2022, pp. 409–421, https://doi.org/10.1007/978-3-031-08760-8_35.
[10] N. Abosata, S. Al-Rubaye, G. Inalhan, Customised intrusion detection for an industrial iot heterogeneous network based on machine learning algorithms called FTL-CID, Sensors 23 (2023) 321, https://doi.org/10.3390/s23010321.
[11] V. Kelli, V. Argyriou, T. Lagkas, G. Fragulis, E. Grigoriou, P. Sarigiannidis, IDS for industrial applications: a federated learning approach with active personalization, Sensors 21 (2021) 6743, https://doi.org/10.3390/s21206743.
[12] J. Xu, B.S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, J. Healthc. Inform. Res. 5 (2021) 1–19, https://doi.org/10.1007/s41666-020-00082-4.
[13] M. Namratha, M.K. Anusree, Niha, S. Pooja, M.R. Arpana, Anomaly detection in medical IoT devices using federated learning, in: T. Senjyu, C. So–In, A. Joshi (Eds.), Smart Trends in Computing and Communications, Lecture Notes in Networks and Systems, Springer Nature, Singapore, 2023, pp. 259–270, https://doi.org/10.1007/978-981-99-0769-4_25.
[14] Y. Liu, N. Kumar, Z. Xiong, W.Y.B. Lim, J. Kang, D. Niyato, Communication-efficient federated learning for anomaly detection in industrial Internet of Things, in: GLOBECOM 2020 - 2020 IEEE Global Communications Conference. Presented at the GLOBECOM 2020 - 2020 IEEE Global Communications Conference, 2020, pp. 1–6, https://doi.org/10.1109/GLOBECOM42002.2020.9348249.
[15] R. Lazzarini, H. Tianfield, V. Charissis, Federated learning for IoT intrusion detection, AI 4 (2023) 509–530, https://doi.org/10.3390/ai4030028.
[16] M.M. Alani, An explainable efficient flow-based Industrial IoT intrusion detection system, Comp. Electr. Eng. 108 (2023) 108732, https://doi.org/10.1016/j.compeleceng.2023.108732.
[17] Dina, A.S., Siddique, A.B., Manivannan, D., 2023. A deep learning approach for intrusion detection in Internet of Things using focal loss function. Internet of Things 22, 100699. https://doi.org/10.1016/j.iot.2023.100699.
[18] N.A. Jalali, H. Chen, Security issues and solutions in federate learning under IoT critical infrastructure, Wirel. Pers. Commun. 129 (2022) 475–500, https://doi.org/10.1007/s11277-022-10107-3.
[19] F. Ilhan, G. Su, Q. Wang, L. Liu, Scalable federated learning with system heterogeneity, in: 2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS). Presented at the 2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS), 2023, pp. 1037–1040, https://doi.org/10.1109/ICDCS57875.2023.00113.
[20] E. Diao, J. Ding, V. Tarokh, Heterofl: computation and communication efficient federated learning for heterogeneous clients, in: Presented at the ICLR 2021 - 9th International Conference on Learning Representations, 2021.
[21] D.N. Sachin, B. Annappa, S. Hegde, C.S. Abhijit, S. Ambesange, FedCure: a heterogeneity-aware personalized federated learning framework for intelligent healthcare applications in IoMT environments, IEEE Access 12 (2024) 15867–15883, https://doi.org/10.1109/ACCESS.2024.3357514.
[22] B. Li, W. Gao, J. Xie, M. Gong, L. Wang, H. Li, Prototype-based decentralized federated learning for the heterogeneous time-varying IoT systems, IEEE Int. Things J. 11 (2024) 6916–6927, https://doi.org/10.1109/JIOT.2023.3313118.
[23] K.M. Ahmed, A. Imteaj, M.H. Amini, Federated deep learning for heterogeneous edge computing, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA). Presented at the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 1146–1152, https://doi.org/10.1109/ICMLA52953.2021.00187.
[24] M.M. Rashid, S.U. Khan, F. Eusufzai, M.A. Redwan, S.R. Sabuj, M. Elsharief, A federated learning-based approach for improving intrusion detection in industrial Internet of Things networks, Network 3 (2023) 158–179, https://doi.org/10.3390/network3010008.
[25] D. Rani, N.S. Gill, P. Gulia, J.M. Chatterjee, An ensemble-based multiclass classifier for intrusion detection using Internet of Things, Comput. Intell. Neurosci. (2022) e1668676, https://doi.org/10.1155/2022/1668676, 2022.
[26] Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H.B., Arcas, B.A. y, Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., Diggavi, S., Eichner, H., Gadhikar, A., Garrett, Z., Girgis, A.M., Hanzely, F., Hard, A., He, C., Horvath, S., Huo, Z., Ingerman, A., Jaggi, M., Javidi, T., Kairouz, P., Kale, S., Karimireddy, S.P., Konecny, J., Koyejo, S., Li, T., Liu, L., Mohri, M., Qi, H., Reddi, S.J., Richtarik, P., Singhal, K., Smith, V., Soltanolkotabi, M., Song, W., Suresh, A.T., Stich, S.U., Talwalkar, A., Wang, H., Woodworth, B., Wu, S., Yu, F.X., Yuan, H., Zaheer, M., Zhang, M., Zhang, T., Zheng, C., Zhu, C., Zhu, W., 2021. A field guide to federated optimization. https://doi.org/10.48550/arXiv.2107.06917.
[27] M. Aledhari, R. Razzak, R.M. Parizi, F. Saeed, Federated learning: a survey on enabling technologies, protocols, and applications, IEEE Access 8 (2020) 140699–140725, https://doi.org/10.1109/access.2020.3013541.
[28] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Gen. Comp. Syst. 115 (2021) 619–640, https://doi.org/10.1016/j.future.2020.10.007.
[29] N. Abosata, S. Al-Rubaye, G. Inalhan, Customised intrusion detection for an industrial IoT heterogeneous network based on machine learning algorithms called FTL-CID, Sensors 23 (2023) 321, https://doi.org/10.3390/s23010321.
[30] M.A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, H. Janicke, Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning, IEEE Access 10 (2022) 40281–40306, https://doi.org/10.1109/ACCESS.2022.3165809.
[31] R.K. Bondugula, S.K. Udgata, N.S. Bommi, A novel weighted consensus machine learning model for COVID-19 infection classification using CT Scan images, Arab. J. Sci. Eng (2021), https://doi.org/10.1007/s13369-021-05879-y.
[32] B.S. Bhati, G. Chugh, F. Al-Turjman, N.S. Bhati, An improved ensemble-based intrusion detection technique using XGBoost, Transact. Emerg. Telecommun. Technol 32 (6) (2020) e4076, 1-15, Aug.
[33] S. Seth, G. Singh, K.K. Chahal, A novel time efficient learning-based approach for smart intrusion detection system, J. Big Data 8 (111) (2021) 1–28.
[34] Dorogush, A.V., Ershov, V., Gulin, A., 2018. CatBoost: gradient boosting with categorical features support. https://doi.org/10.48550/arXiv.1810.11363.
[35] M.M. Rashid, S.U. Khan, F. Eusufzai, M.A. Redwan, S.R. Sabuj, M. Elsharief, A federated learning-based approach for improving intrusion detection in industrial Internet of Things networks, Network 3 (2023) 158–179, https://doi.org/10.3390/network3010008.
[36] Ayesha S. Dina, A.B. Siddique, D. Manivannan, A deep learning approach for intrusion detection in Internet of Things using focal loss function, Int. Things 22 (2023) 100699, https://doi.org/10.1016/j.iot.2023.100699. ISSN 2542-6605.
[37] I. Priyadarshini, Anomaly detection of IoT cyberattacks in smart Cities Using Federated Learning and Split Learning, Big Data Cogn. Comput 8 (2024) 21, https://doi.org/10.3390/bdcc8030021.
[38] D.C. Attota, V. Mothukuri, R.M. Parizi, S. Pouriyeh, An ensemble multi-view federated learning intrusion detection for IoT, IEEE Access 9 (2021) 117734–117745, https://doi.org/10.1109/access.2021.3107337.
[39] X. Wang, S. Garg, H. Lin, G. Kaddoum, J. Hu, M.M. Hassan, Heterogeneous blockchain and AI-driven hierarchical trust evaluation for 5G-enabled intelligent transportation systems, IEEE Transac. Intell. Transpor. Syst 24 (2) (Feb. 2023) 2074–2083, https://doi.org/10.1109/TITS.2021.3129417.
[40] F. Leon, S.-A. Floria, C. Bădică, Evaluating the effect of voting methods on ensemble-based classification, in: 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA). Presented at the 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), 2017, pp. 1–6, https://doi.org/10.1109/INISTA.2017.8001122.
[41] X. Wang, S. Garg, H. Lin, J. Hu, G. Kaddoum, Md. Jalil Piran, M.S. Hossain, Toward accurate anomaly detection in industrial internet of things using hierarchical federated learning, IEEE Int. Things J. 9 (2022) 7110–7119, https://doi.org/10.1109/JIOT.2021.3074382.
[42] N. Moustafa, M. Keshky, E. Debiez, H. Janicke, Federated TON_IoT windows datasets for evaluating AI-based security applications, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). Presented at the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, Guangzhou, China, 2020, pp. 848–855, https://doi.org/10.1109/TrustCom50675.2020.00114.
[43] X. Wang, et al., QoS and privacy-aware routing for 5G-enabled industrial Internet of Things: a federated reinforcement learning approach, IEEE Transact. Indus. Infor 18 (6) (June 2022) 4189–4197, https://doi.org/10.1109/TII.2021.3124848.
[44] X. Wang, W. Liu, H. Lin, J. Hu, K. Kaur, M.S. Hossain, AI-empowered trajectory anomaly detection for intelligent transportation systems: a hierarchical federated learning approach, IEEE Transac. Intell. Transport. Sys 24 (4) (April 2023) 4631–4640, https://doi.org/10.1109/TITS.2022.3209903.
[45] D. Rani, N.S. Gill, P. Gulia, F. Arena, G. Pau, design of an intrusion detection model for IoT-enabled smart home, IEEE Access 11 (2023) 52509–52526, https://doi.org/10.1109/ACCESS.2023.3276863.
[46] Anupma Sangwan, Anju Sangwan, R.P. Singh, A classification of misbehavior detection schemes for VANETs: a survey, Wirel. Pers. Commun. 129 (2022) 285–322, https://doi.org/10.1007/s11277-022-10098-1.
[47] S. Devi, Anju Sangwan, Anupma Sangwan, M.A. Mohammed, K. Kumar, J. Nedoma, R. Martinek, P. Zmij, The use of computational geometry techniques to resolve the issues of coverage and connectivity in wireless sensor networks, Sensors 22 (2022) 7009, https://doi.org/10.3390/s22187009.