IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 3, 2021, accepted August 17, 2021, date of publication August 24, 2021, date of current version August 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3107337

An Ensemble Multi-View Federated Learning Intrusion Detection for IoT

DINESH CHOWDARY ATTOTA, VIRAAJI MOTHUKURI, (Member, IEEE),
REZA M. PARIZI, (Senior Member, IEEE),
AND SEYEDAMIN POURIYEH, (Associate Member, IEEE)
College of Computing and Software Engineering (CCSE), Kennesaw State University (KSU), Marietta, GA 30060, USA

Corresponding author: Reza M. Parizi (rparizi1@kennesaw.edu)

This work was supported by U.S. SunTrust Fellow in Cybersecurity/Information Security Research Funding Program under Grant ST20-01.

ABSTRACT The rise in popularity of Internet of Things (IoT) devices has attracted hackers to develop IoT-specific attacks. The microservice architecture of IoT devices relies on the Internet to provide their intended services. An unguarded IoT network makes inter-connected devices vulnerable to attacks. It will be a tedious and ineffective process to manually detect the attacks in the network, as the attackers frequently upgrade their attack strategies. Machine learning (ML)-assisted approaches have been proposed to build intrusion detection for cybersecurity automation in IoT networks. However, most such approaches focus on training an ML model using a single view of the dataset, which often fails to build insightful knowledge and understand each feature's impact on the ML model's decision-making ability. As such, the model training with a single view may result in an incomplete understanding of patterns in large feature-set datasets. Moreover, the current approaches are mainly designed in a centralized manner in which the raw data is transferred from the edge devices to the central server for training. This, in turn, may expose the data to all kinds of attacks without adhering to the privacy-preserving of data security. Multi-view learning has gained popularity for its ability to learn from different data views and deliver efficient performance with more distinguished predictions. This paper proposes a federated learning-based intrusion detection approach, called MV-FLID, that trains on multiple views of IoT network data in a decentralized format to detect, classify, and defend against attacks. The multi-view ensemble learning aspect helps in maximizing the learning efficiency of different classes of attacks. The Federated Learning (FL) aspect, wherein the device's data is not shared to the server, performs profile aggregation efficiently with the benefit of peer learning. Our evaluation results show that our proposed approach has higher accuracy compared to the traditional non-FL centralized approach.

INDEX TERMS Internet of Things, IoT security, federated learning, neural networks, multi-view classification, intrusion detection system.

I. INTRODUCTION

There has been a sharp growth in the usage of Internet of Things (IoT) devices in recent years. They interconnect with other digital and physical devices, which enables information exchange and service delivery [1]. Flexible IoT devices are currently used for inter-connecting knowledge of cyber-physical systems in healthcare [2], [3], transportation [4], smart homes [5], and smart cities [6], to name a few. IoT-aided devices are very ubiquitous nowadays due to their massive adoption in various sectors. Even though these devices play a prominent role in our everyday lives, many security complications are involved while using these devices. The security measures [7] play a key role in the trustworthiness of these IoT devices and their services. Poor security measures will make these devices vulnerable to a variety of cyber-attacks such as data leakage, Denial of Service (DoS) [8], which may disrupt the normal functionality of the end device. Several lightweight protocols have been introduced [9] for effective communication within networks, one such protocol is Message Queuing Telemetry Transfer protocol (MQTT) [10]. MQTT protocol is

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

mainly designed for communication between devices with low bandwidth, making it an ideal solution for communication between IoT devices. The MQTT protocol's publish and subscribe communication style helps to exchange information in client-server communication through messaging. The network data collected during information exchange from IoT devices using MQTT protocol can be used to detect intruders in the network.

Intrusion detection is crucial in an IoT network as the intruders can attack and take over the IoT devices and the other devices connected to the IoT network. Intrusion Detection System (IDS) techniques can be classified into main categories: Signature-based and Anomaly-based Intrusion Detection. In the signature-based intrusion detection, there are a set of pre-defined malicious patterns of attacks, the device detects an attack when it encounters any of the known patterns, whereas the IDS using anomaly-based depends on the deviations of normal behavior for detecting malicious activity. Signature-based detection performs better than anomaly-based in terms of efficiency due to known attacks' information availability but fails to detect new attacks. In contrast, anomaly-based detection is more capable of detecting new attacks based on the traffic deviations compared to normal behavior. Adapting machine learning algorithms for anomaly-based detection techniques will increase the self-learning process and help develop more intelligent systems for detecting attacks in IoT environments.

Recent advancements in network intrusion detection have shown that having knowledge on multiple view behavior of an attack will achieve better performance rather than a single view feature set [11]. *Multi-View learning* is a new paradigm in which a distinct function is used to model a particular view and combines all functions to exploit redundant views of input data. In multi-view, the data is trained alternately to increase mutual agreement on two distinct views of input data. Learning tasks in a multi-view strategy is done with abundant information. Learning from the training of multiple views can be combined to have an efficient outcome. As we can consider network data into multi-view form, the same way, the attack information can also be analyzed as multiple views. The behavioral information of attacks can be varied from one view to the other.

In the conventional ML-based approach, the entire data is considered a single feature set and it requires more training time to find any deflections in few features of input data for detecting an attack. Whereas in the multi-view approach, we consider a set of features in multiple views rather than a single feature set. As the features set can be reduced in multi-view, the training can be more efficient, which leads to detection of attack with more accuracy. Having such view-level intelligence can be helpful in the better learning process of attacks and detect abnormal behavior in any view of network data. Most of the existing works use centralized ML techniques for intrusion detection [12]–[14]. Typically, IoT devices in the real world are placed far away from the location of the central server. The traffic logs generated by these IoT devices are tremendous. Sharing network logs data of IoT devices with the central server for intrusion detection at every point of time is a cumbersome process and can be subjected to various attacks. Though the centralized approach can detect attacks with good accuracy, the main overhead is the cost associated with transmitting IoT network data to the server. The process takes more time because IoT devices and the central server (intrusion detection system) are geographically isolated. Moreover, there will be instances where the data contains sensitive information that needs to be secured; therefore, sharing such data over the network will make data prone to various attacks, leading to critical consequences. The centralized methodology does not ensure user data privacy, and the latency cost is high in this paradigm. The edge computing paradigm solves latency by bringing the data and computational resources close to the end device, but this does not provide any privacy-preserving methods for data. Knowledge sharing is not supported in such edge computing manner, i.e., information of new attacks or change of behavioral information of existing attacks that any device encounters is not shared with latter devices.

Federated Learning (FL) [15] is an ideal solution for performing on-device training while maintaining privacy-preserving methods using decentralized data. In recent years, FL became a widely adopted solution for ensuring privacy [16]–[19] of the end-user data and updates with low latency. This technology addresses the limitations of centralized and edge-computing paradigms and marks it as an outstanding ML technique for maintaining data privacy while sharing knowledge among peers. FL uses an exceptional strategy [20] in which a trained ML model will be shared between multiple devices in the network, and devices that download the shared base ML model will train it with its local data and computational resources that are available with that device. After training, the devices share the updated local model parameters back to the server for performing aggregation of their locally computed parameters. Considering privacy measures, the aggregation process is made, so the server has no access to devices' training data. Using this paradigm, in this work, we are proposing an intrusion detection technique called Multi-view Federated Learning based Intrusion Detection (MV-FLID) that uses decentralized data for performing training and inference procedures at the device's end. The device's data is not being shared with the central server or any other external devices, thereby maintaining the security and privacy of the device's data. FL aggregation [21], [22] will be done at the server end by collecting all trained models of devices that participate in the FL process. Following is a list of contributions we made in this research:

1) Proposing an intrusion detection approach with multi-view information of IoT network data using federated learning methodology.
2) Integrating Grey Wolves Optimization mechanism for extracting optimized feature sets in the proposed approach.

3) Devising an ensemble-based method for detecting attacks from multiple views in the proposed approach.
4) Obtaining high accuracy results in detecting attacks compared to traditional machine learning approaches that use a single feature-set of attack information in a non-decentralized manner.

The remaining of the paper is structured in the following manner. Section II gives the related work. Section III presents the proposed approach and illustrates the underlying architecture with implementation details. Section IV presents the dataset, metrics, and evaluation results and summarizes our findings. Finally, Section V concludes the paper.

II. RELATED WORK

Intrusion Detection has become a foremost thing to be considered while utilizing IoT devices [13]. Many research works have been carried out on Intrusion detection using ML methodologies. In this section, we discuss some of the recent advancements proposed for Intrusion detection techniques.

The researchers in [23] used a supervised artificial neural network which is based on multi-level perceptron for training and testing threat patterns in an IoT Network. They have achieved 99.4% accuracy in detecting Denial of Service (DoS) / Distributed denial of service (DDoS) attacks.

Similarly, authors in [24] proposed an intrusion detection approach using Feed-forward neural networks and multi-class classification for detecting attacks like DoS/ DDoS, reconnaissance, and information theft attacks in IoT networks. This method involves collecting data from raw network packets using an analyzer tool. The analyzer tool captures all the generic features of the raw network traffic. The Pre-processed data is then used to train a Deep Neural network, which is used as a classifier for new incoming packet data. The classifier then labels the malicious packet into a specific category of attack.

Authors in [22] presented a comprehensive design of the FL system. Unlike traditional ML architecture, FL utilizes a decentralized approach. A variety of FL techniques have been proposed in the field of cybersecurity and also for intrusion detection in IoT networks.

For example, the authors in [25] proposed a Self-learning anomaly detection system using an FL approach for detecting compromised IoT devices in the network. It utilizes Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU's) for building the proposed model. It constantly monitors the devices' network traffic and detects anomalous deviations from given communication profiles of devices' without human intervention. It has achieved an accuracy of 98.2% and can detect 95.6% of attacks in 257ms with less false alarm rate.

Likewise, research work in [26] presents a deep learning model by combining NIDS and HIDS to detect cyberattacks. They assessed multiple stages of attacks and evaluated machine and deep learning algorithms on various NIDS and HIDS datasets. The best-performed algorithm is chosen for the list, and it was further evaluated on other multiple datasets for efficiency.

Researchers in [27] presented an Intelligent Intrusion detection using FL approach and Long Short Term Memory (LSTM) recurrent neural network. LSTM networks have cell state and memory state within to hold required information on long inputs of data. This model does compare the efficiency of the proposed algorithm with conventional neural networks using Adam optimizer and achieves an accuracy of 99.21% and an F1 score of 99.21. It used the supervised learning mechanism while performing training procedures.

On the other hand, the authors in [28] put forward a deep anomaly detection framework for sensing time series data using the FL approach on distributed edge devices for Industrial IoT. It also incorporates a CNN-LSTM model for extracting fine-grained features of historical observations sensing time series data and uses LSTM modules for time-series predictions, thereby preventing memory loss and gradient-dispersion problems.

Most of the works were carried out using a single view of network data for detecting attacks of the aforementioned proposed approaches. There have been advancements in intrusion detection techniques using the multi-view information of attacks. Authors in [11] proposed a semi-supervised co-training approach using multi-view nature of attacks. In this approach, attack behavior will be maintained in multiple views, and attack detection will be done using the predictions done by ML models of multiple views of an attack. They have used a centralized approach for implementing their research and had an active labeling procedure for labeling unknown attacks by experts. Researchers in [10] have introduced multi-view features of MQTT data and evaluated features using centralized ML algorithms. Authors in these works have proposed their methodologies in a centralized approach. In our work, we proposed a Federated Learning methodology (MV-FLID) to train network data segmented into three views in IoT devices. We used an ensembler at the end of deep learners for detecting attacks based on their occurrence.

To abridge, most of the research works have been carried out on a single feature set of data using a centralized mechanism and lags in using decentralized approaches for effective communication and intrusion detection in IoT environment. Our approach considers the limitations of existing works and proposes a multi-view decentralized approach for intrusion detection.

III. PROPOSED APPROACH

This section presents our proposed approach called MV-FLID, which integrates base classifiers trained on multiple views of network data to detect various attacks in IoT devices. The list of acronyms used is given in Table 1.

A. ARCHITECTURE

Our proposed approach is depicted in Figure 1 in which multiple virtual IoT instances are connected to Security Gateways,

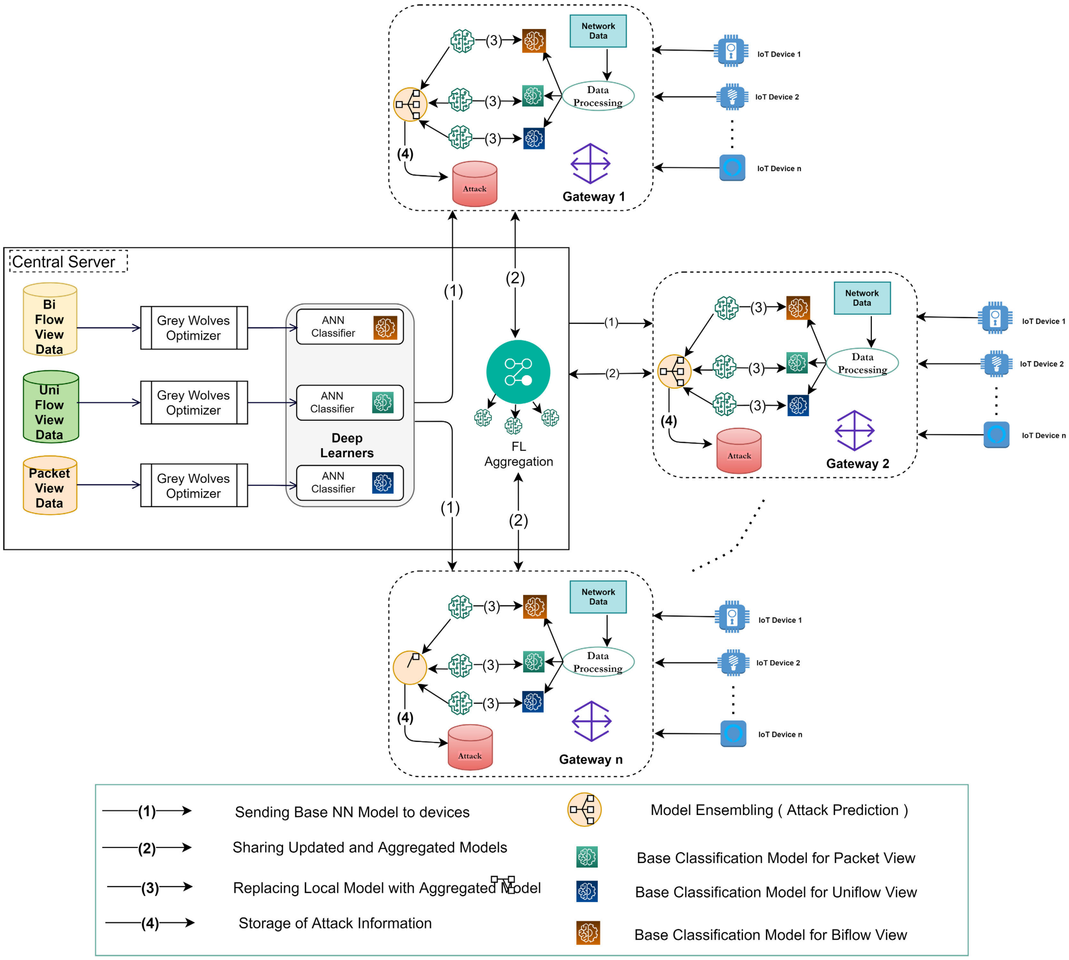

Network Data
Data Processing
(3)
(4)
Attack
Gateway 1
IoT Device 1
IoT Device 2
IoT Device n
Central Server
Bi Flow View Data
Uni Flow View Data
Packet View Data
Grey Wolves Optimizer
ANN Classifier
Deep Learners
FL Aggregation
(1)
(2)
Gateway 2
Gateway n
(1) Sending Base NN Model to devices
(2) Sharing Updated and Aggregated Models
(3) Replacing Local Model with Aggregated Model
(4) Storage of Attack Information
Model Ensembling (Attack Prediction)
Base Classification Model for Packet View
Base Classification Model for Uniflow View
Base Classification Model for Biflow View

FIGURE 1. High level architecture of the proposed approach.

TABLE 1. Acronyms.

Acronym	Description
FL	Federated Learning
ML	Machine Learning
CSV	Comma Separated Values
$sigma/\sigma$	Sigmoid
$tanh$	Tangent hyperbolic
rfc	Random forest classifier
IDS	Intrusion Detection System
GWO	Grey Wolves Optimization
MQTT	Message Queuing Telemetry Transfer Protocol
TCP	Transmission Control Protocol
ANN	Artificial Neural Networks

which are distributed at different sites. The MQTT protocol in the session layer of these IoT devices will share network data with the security gateway (middle agent) for further processing. Published traffic data of these devices will be monitored by an intrusion detector installed on the security gateway. Having such on-site intelligence will make gateways independent from the traditional server-device mechanism, in which the traffic data will be shared with the server every time for detecting intrusions. Such on-site intelligence will also expedite detection time by performing analysis on shared network data. The gateways are placed in the federated setup. For every selected amount of time, the ML models of multiple views that get trained on local network data of IoT devices participate in a server-based aggregation process. The aggregation process creates a more sophisticated intrusion detection model with aggregated and optimized all device models' parameters.

Sharing intrusion detection ML models with the server and aggregation of the ML models enable generating a model with global intelligence perception that helps in the detection of attacks with higher amounts of accuracy. The server shares a global optimized model back to security gateways and assists in increasing knowledge among gateways. Sharing the aggregated model to security gateways leads to a better learning process. It empowers devices to detect intrusion based on behavior learned by other devices participating in the aggregation process.

1) DATA PRE-PROCESSING

The network connection between the source and destination consists of a sequence of TCP packets, which have a n number of features for each packet. As the packets contain IoT device network logs information, in this module, based on their nature we consider packet information into three views such as Bi-Directional Features (Bi-flow view), Uni-Directional Features (Uniflow view), and Packet-Features (Packet view). The multi-view features of packet are listed in Table 2.

2) GREY WOLVES OPTIMIZATION

Feature Selection using Grey Wolves Optimization (GWO) [29] technique is a hunting kind of mechanism which can be implemented for generating optimal feature set from the given list of features in the training data. This heuristic approach attains the best feature set based on the accuracy of the features obtained at multiple iterations of the features list. Therefore, the final features set obtained after implementing the GWO mechanism will have more accuracy. The GWO mechanism works on a strategy that has four different categories of wolves called alpha, beta, gamma, and delta. The wolves have a strict dominance hierarchy wherein Alpha (α) is considered as the leader of the group who is conscientious and responsible for making decisions. The next category of a wolf in the hierarchy is the beta (β) wolf which is subordinate to the Alpha (α). Sometimes beta (β) wolves will find the best fitness and acknowledge α about the positions. The other lower-ranking wolves are called omega (ω) and delta (δ). α is considered as the best fitness solution. β and δ are considered as the next fitness for the sub-optimal solution. All these categories of wolves will be scattered into n number of dimensions and collectively tries to find the best solution for the given features on multiple iterations.

The wolves hunting process at a given point of time can be described in the below equations.

$$G = |C.F_{prey}(t) - F_{wolf}(t)| \tag{1}$$

$$F(t+1) = F_p(t) - A.G \tag{2}$$

$$A = 2ar_1 - a \tag{3}$$

$$C = 2r_2 \tag{4}$$

In Equations 1 and 2 G is the wolves current position for the current time iteration t, whereas F represents the position vector and F_{prey} denotes current position vector of the prey($target$). A & C are the coefficients calculated using variable in Equations 3 and 4. $r1$, $r2$ are random vectors that lie in the interval $[-1.28, 1.28]$ and a is an arbitrary variable that linearly decreases from 2 to 0.

3) VIEW SPECIFIC ML MODELS

Referring to the architecture, in this module, a feed-forward Artificial Neural Network based on a back-propagation training algorithm is used for training procedures. The structure of the proposed neural network consists of an input layer, hidden layers, and an output layer. The number of inputs was determined by the input features set. As we are dealing with the multiple views of the network data, in this module, we design neural network ML models for the three views, i.e., for Uniflow features, Biflow features, and Packet features of data shared by MQTT protocol. The ML models will be trained on the patterns of different kinds of attacks and a soft-max function is employed at the end of the network to classify network traffic into attack categories. The feature sets extracted in the pre-processing data phase were given as input to the classification models wherein the ML models will learn the temporal information of these feature sets, and the ML models get trained on the given knowledge.

We consider this set of ML models as the base ML models of the corresponding views, and the trained ML models will be shared with IoT devices for inference of the attacks on the real-time network data of the device.

4) FL TRAINING PROCESS

The FL process is implemented with the available number of virtual instances. Each instance has its training data using which the shared view specific ML models get trained, and shares updated parameters with the central server for aggregation purposes. The information presents in network data of IoT devices is segmented into three views and segmented data is considered for training. In our work, we used three deep learning ML models for training data present in three views. The communication rounds in the FL process are the number of times the aggregation procedure is being implemented for locally trained ML models of three views in IoT devices. The steps underlying this process are as follows:

Step 1: The initial ML models of three views are shared with the virtual IoT devices.

Step 2: After getting base ML models of three views from the server, the ML models will undergo a re-training process using local network data of the device.

Step 3: View-specific features information of network data is given as input to Biflow, Uniflow, and Packet View ML models to predict the class of an attack. All three classification ML models will predict the attack class based on the training parameters set in the initial model.

Step 4: From the list of different probabilities of attack classes by output layer, the ML model gives an attack class that has the highest probability from the list of predicted attacks as output.

Step 5: After training the ML models for several epochs locally on the device data, view-classification models of all IoT devices are sent to the server for performing FL Aggregation of the corresponding view-classification ML models. All the parameters of the view-classification ML models of multiple IoT devices are aggregated, and a Global ML model is created for a corresponding view-classification model.

Step 6: After completion of FL Aggregation, Global models are shared back to IoT devices, and the current classification models in the IoT devices are replaced with shared global models, and the training and inference procedure continues on local data.

Step 7: An ensembler is engaged and the attack predictions of all view-classification models are sent to the ensembler for selecting the class based on the majority predicted class from the trained classifiers.

5) FL AGGREGATION

Federated Aggregation is a computation algorithm in which a group of devices connected to a network hold private network data and collaborate in computing an aggregated model without revealing any sensitive information of the device and sharing only the locally-computed parameters of the device models for performing aggregation. Each device trains its models of three views locally for a selected number of epochs before sharing its updated parameters with the central server. This process limits the communication overhead of the device because of sharing only the trained models of views rather than view specific information of local data.

$$F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w) \tag{5}$$

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{6}$$

In Equation 5 K denotes total number of clients, k is index of client, n_k is the number of data samples that are available during training of client k, taking $f_i(w) = l(x_i, y_i\ w)$ which is Loss of prediction of sample (x_i, y_i) and P_k is set of indices of data samples on client k. Equation 5 estimates the weight parameters of all devices based on the loss values that encountered across data points they have trained with. Equation 6 performs scaling of the parameters and adding up all in a component-wise manner.

6) ENSEMBLER

We have used an ensemble-based technique in our work to predict the attack based on the predicted values of classification models of three views. For achieving better accuracy rates, we used Random Forest Classifier [30] to perform an ensemble process on the model's outcomes. For the given input of network data, the models of three views will predict the probabilities of a possible attack. The outcomes of these models will be sent to ensembler, which combines the predictions of three views and gives a single attack based on its probability and its occurrence. The process of ensembler is outlined in Figure 2 and its functionality is illustrated in Algorithm 1.

The Algorithm 1 formalizes the whole procedure of our proposed approach. Splitting the dataset into train set for training procedures and test set for evaluation purposes. The order of functions in the algorithm specifies the functioning of our approach. Starting with Feature Engineering(*featureEngineering_GWO*) of data presented in views, using Grey Wolves Optimization technique and resultant data is being forwarded to training procedure(*flTraining*) for a selected number of rounds. Next, followed by the Averaging of computed parameters of models using $FL_{average}$ function. Finally, the *EnsemblerLogic* function consists of logic for predicting the attack using test data of views that are passed to corresponding trained view models.

The following steps will elucidate more about the working procedure of our architecture.

Algorithm 1: FL Process

```
Input: Train and Test Data of MQTT Dataset
Output: Intrusion Detection in Multi − view Network Data
Data: mqtt_dataset
$m_b, m_u, m_p$    /* ML models for three Views */
Reading Input data of three views
$Views = D_b, D_u, D_p$    /* data for three views */
$d = d_1, d_2 \ldots d_{10}$    /* Initiating Virtual Devices */
Function featureEngineering_GWO(Views):
    for view in Views do
        while max_iterations do
            /* Set Random Positions of Wolves */
            $\alpha, \beta, \gamma, \delta$ = random_positions
            update(current position of wolves)
        calculate fitness of $\alpha, \beta, \gamma, \delta$
        update($\alpha, \beta, \delta$)
    return $\alpha$ EndFunction
Function flTraining(communication_rounds):
    while $d_i$ in d do
        $m_{wb}$ = train(data in $D_b$)
        $m_{wp}$ = train(data in $D_p$)
        $m_{wu}$ = train(data in $D_u$)
    return $m_{wb}, m_{wp}, m_{wu}$
    EndFunction
Function flAverage():
    $M_{wb} = FL_{average}(m_{wb})$  /* Averaging parameters of Biflow view models of all devices */
    $M_{wp} = FL_{average}(m_{wp})$  /* Averaging parameters of Packet view models of all devices */
    $M_{wu} = FL_{average}(m_{wu})$  /* Averaging parameters of Uniflow view models of all devices */
Function ensemblerLogic($M_{wb}, M_{wu}, M_{wp}$):
    $biflow\_view_{data}, packet\_view_{data}, uniflow\_view_{data}$
      predictions = $M_{wb}(biflow\_view_{data})$,
      $M_{wp}(packet\_view_{data})$, $M_{wu}(uniflow\_view_{data})$
      $Attack_{prediction}$ = Ensembler(predictions)
```

1) The features extraction using Grey Wolves Optimization (GWO) [31] method for extracting the optimal feature set from the list of available features for three views, i.e., Uniflow, Biflow, and Packet features.
2) The server initializes the base Artificial Neural Network models for these three views using the optimal feature sets obtained for corresponding views in the feature extraction process. The structure of these ML models and their hyper-parameters are defined at this stage. The defined neural network models perform an initial training round with available training data with different categories of attacks.

3) The ML models produced in step (2) are shared with Security gateways that are participating in the FL process.
4) Once the security gateways download intrusion detection models of the server's corresponding view, the learning parameters of the ML models are enhanced based on IoT devices' local network data.
5) As the ML models were trained with a certain number of attacks at the server, they will detect and identify the abnormal behavior in network traffic of IoT devices that assists in detecting attacks.
6) Only parameters of updated models are shared to the server for aggregation process instead of sharing sensitive information of network traffic data and creating scope for data theft.
7) The server aggregates the weights obtained from different gateway models and creates new sophisticated and updated models for corresponding views, which are communicated back to security gateways after successful aggregation.
8) Each security gateway uses an updated model on new upcoming traffic data.

The above steps (5), (6), and (7) will be repeated to enhance the learning process, improve models' accuracy and keep the global ML up-to-date with the latest data.

IV. EVALUATION RESULTS

In this section, we discuss the environment setup, the dataset, and its views segmentation used for implementing our proposed approach. Then we walk through the evaluation metrics used for analyzing the performance of our proposed approach. In the final part, we present our experimental results and provide discussion.

A. EXPERIMENTAL SETUP

For evaluating our proposed approach MV-FLID, we have used the Tesla V100-SXM2-16GB GPU server that was hosted as a backend for the Google Colab Pro environment. We have implemented the FL approach using PySyft [32] deep learning framework, which is based on Pytorch deep learning framework.

B. DATASET

To evaluate our approach, we have used a lightweight MQTT protocol dataset [10], which simulates realistic IoT device communication. The MQTT dataset consists of five recorded scenarios of 1 normal operation and 4 attack scenarios. It has both common network scanning attacks and brute-force attacks. MQTT protocol communication datasets are widely adopted [33], [34] for building an effective Intrusion Detection model for IoT devices.

The processed features of the MQTT dataset can be categorized into Unidirectional, bi-directional, and packet-based features. The features of corresponding groups are listed in Table 2. The distribution of attacks in each view are illustrated in Figure 3

TABLE 2. Dataset features.

View	Description	Total
Bi-Flow	'ip_src', 'ip_dst', 'prt_src', 'prt_dst', 'proto', 'fwd_num_pkts','bwd_num_pkts', 'fwd_mean_iat', 'bwd_mean_iat', 'fwd_std_iat','bwd_std_iat', 'fwd_min_iat','num_psh_flags', and so on	27
Packet	'timestamp', 'src_ip', 'dst_ip', 'protocol', 'ttl', 'ip_len', 'ip_flag_df','ip_flag_mf', 'ip_flag_rb', 'src_port', 'dst_port', 'tcp_flag_res', 'tcp_flag_ack', 'tcp_flag_push', 'tcp_flag_reset', 'tcp_flag_syn', 'tcp_flag_fin'	17
Uni-Flow	'ip_src', 'ip_dst', 'prt_src', 'prt_dst', 'proto', 'num_pkts', 'mean_iat','std_iat', 'min_iat', 'max_iat', 'mean_pkt_len', 'num_bytes', 'num_psh_flags', 'num_rst_flags, 'num_urg_flags', 'std_pkt_len', 'min_pkt_len', 'max_pkt_len'	18

C. EVALUATION METRICS

In our approach, we use Accuracy, Precision, Recall, and F1-score as our metrics for measuring the performance of classification models.

True Positive (TP): The total number of attack records that were correctly classified as an attack.

True Negative (TN): Total number of normal records that are accurately classified as normal.

False Positive (FP): Total number of normal records that were incorrectly classified as an attack.

False Negative (FN): Total number of attack records that were incorrectly classified as benign.

1) PERFORMANCE METRICS

a: ACCURACY

It is defined as ratio of correctly classified records to the total number of records.

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

b: F1-SCORE

It is the Harmonic mean of Precision and Recall.

$$F1 - score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

c: PRECISION

The ratio of Truly Positive to the Total number of results predicted positive.

$$Precision = \frac{TP}{FP + TP}$$

d: RECALL

It is the percentage of predicted positive to the total positive. It is also called as *True Positive Rate (TPR)*

$$Recall = \frac{TP}{FN + TP}$$

D. RESULTS

We have conducted our experiments using the Pysyft deep learning framework with 10 virtual IoT devices in a Federated setting to evaluate our approach. We have implemented ten

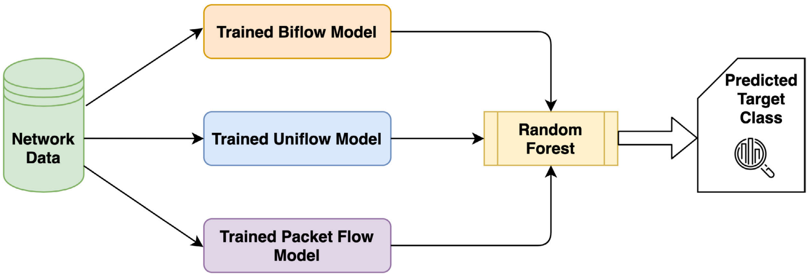


FIGURE 2. Prediction using Ensembled method.

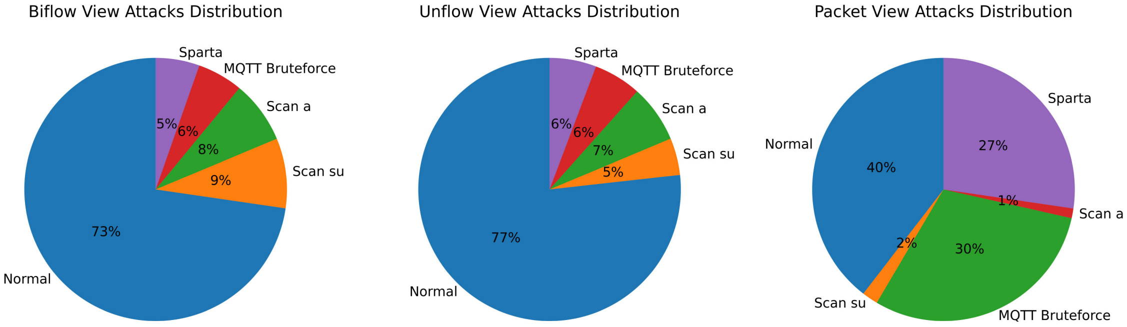


FIGURE 3. Distribution of attacks data in multiple views.

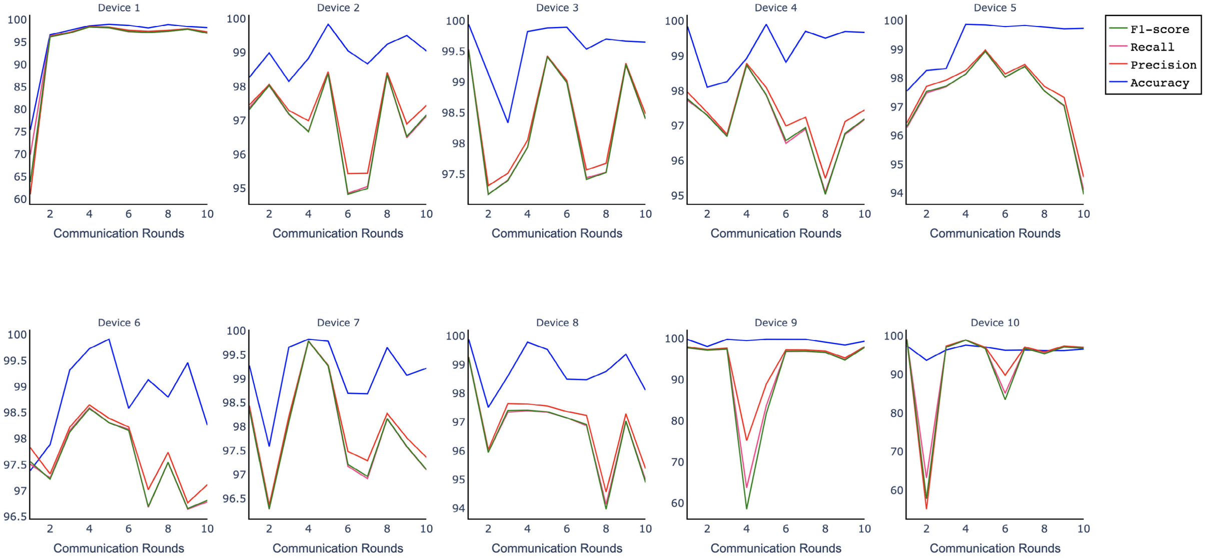


FIGURE 4. Evaluation metrics of training rounds in proposed approach.

rounds of aggregation of models that are trained using local data of the corresponding device. Figure 4 shows the device's accuracy trends after every round of Federated Aggregation. We have distributed training and test data with devices to enable a self-learning process. We have achieved significant accuracy (as shown in Figure 4) after each round of communication while preserving the privacy of the data. Federated Learning methodology has spurred the efficiency of the knowledge-sharing process regards to diverse behavior of attacks among devices. The deep learning models in these devices get trained with much information of attacks in every round of communication, increasing our accuracy.

1) COMPARISON WITH NON-FL APPROACH

We implemented the non-FL version of our proposed approach using the Pytorch deep learning framework to compare its evaluated results with the FL version.

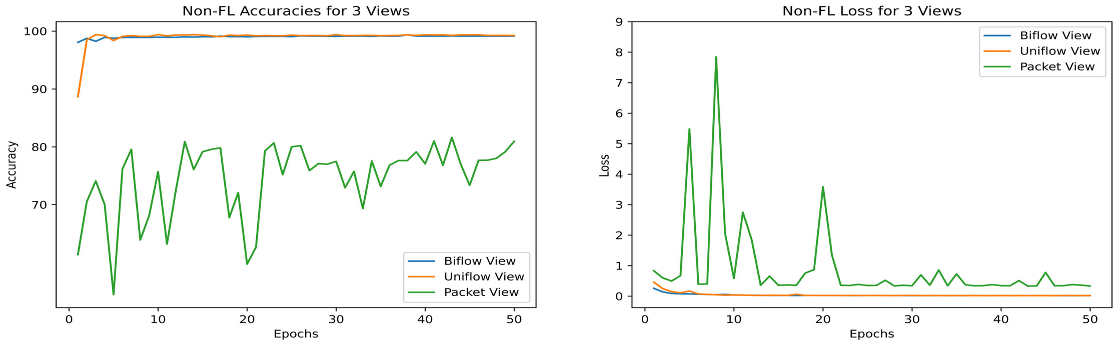


FIGURE 5. Training results for individual models in non-FL setup.

In the FL method, there are *ten* rounds of communication has been carried out. In every round of communication, any change in behavior of attack information encountered by any device during training is being shared with latter devices at the time of aggregation. Whereas in the Non-FL implementation, knowledge sharing is not happening. Also, the amount of data used for training in a centralized approach is huge compared to the amount of data used in the decentralized methodology. The procedure of implementing Non-FL version are as follows:

1) We performed Data Pre-Processing techniques on view datasets and extraction of valuable features set.
2) We have defined attack classification models for three views i.e., Biflow View, Packet View, and Uniflow View.
3) Next, we carried out training procedures for three-view for a given number of epochs.
4) As this procedure comes under a centralized approach, there is a single round of training for view classification models.

The results of the training round are shown in Figure 5. As can be seen from the figure, the Biflow and Uniflow view deep learners yielded more accuracy than that of the packet flow view. The same tendency continues with the loss values of models. The packet view deep learner has taken more steps for reaching an optimal minimum. The training accuracy in packet view is less in the Non-FL approach and required more information to identify abnormal behavior in packet view. Improving parameters of packet view deep learners by knowledge sharing technique enhances its efficiency in detecting abnormal behaviors in packet view.

In a side-by-side fashion, we compared the evaluation metrics of trained models of multiple views from the non-FL setup with the metrics of trained view-model instances after completing the 10^{th} round of FL averaging process of the proposed approach. Figure 6 illustrates the comparison of the evaluation metrics for FL and non-FL implementations. Compared to the results of the non-FL approach, our proposed approach using FL has achieved a higher amount of accuracy in detecting attack behavior in network data. The proposed approach has maintained a good amount of accuracy than the Non-FL approach. Using Federated Learning methodology, the network data of IoT devices will get trained after every round of aggregation process. As the aggregation includes merging deep learning models of multiple views, the knowledge of attack behavior across multiple devices is being shared, leading to an increase in the efficiency of detection of attacks. Whereas, In the Non-FL methodology, training procedure is implemented at the server end and a variety of attack behavior of IoT devices is not being shared to the server for global perception. Knowledge sharing, which the FL method supports, plays the most significant role in increasing the detection of attacks more efficiently.

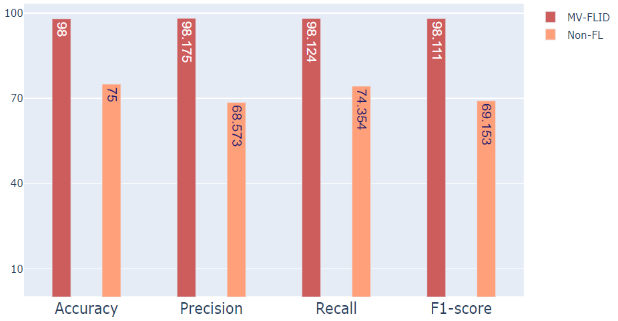


FIGURE 6. Evaluation metrics comparison of our proposed approach and non-FL approach.

2) SECURITY ANALYSIS

The previous results (presented in Figure 4) were measured with the premise that devices and their traffic data be legitimate. To further evaluate the security of our proposed approach in the presence of adversary/malicious traffic in the network, we considered three devices with malicious traffic for this analysis. In Figure 7, we have the accuracy trends of devices that contain legitimate and malicious traffic. We used devices 8, 9, and 10 with malicious traffic data. We implemented a poison attack by flipping the training data labels for infected nodes to analyze the efficiency of our approach. As shown in Figure 7, the accuracy trends of infected devices are low compared to other devices with

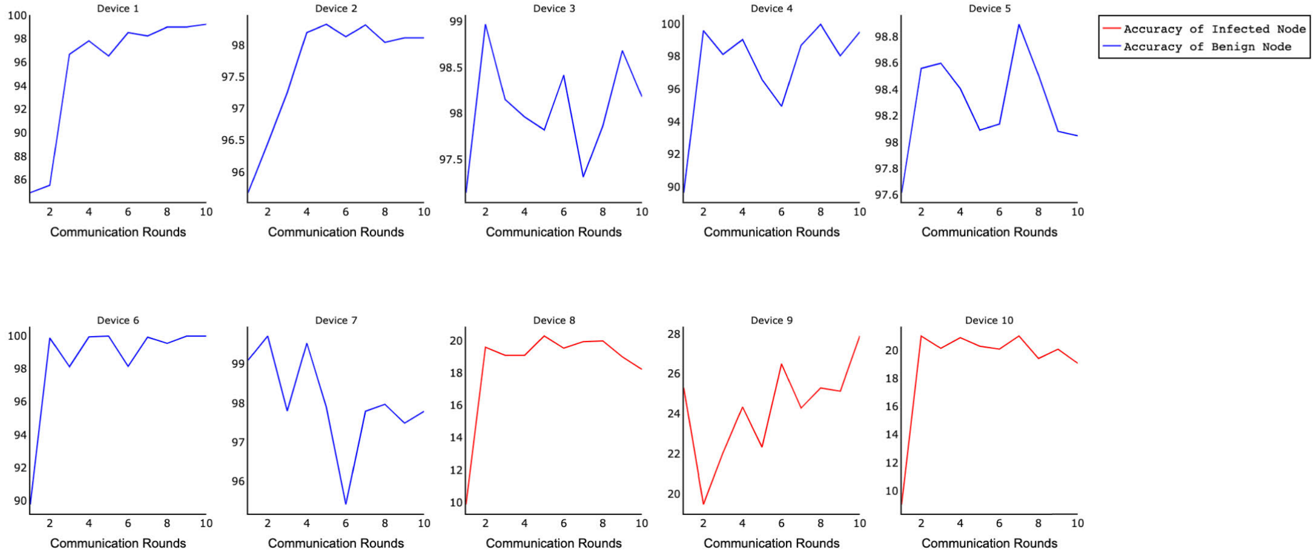


FIGURE 7. Accuracy trends of proposed approach with adversary nodes.

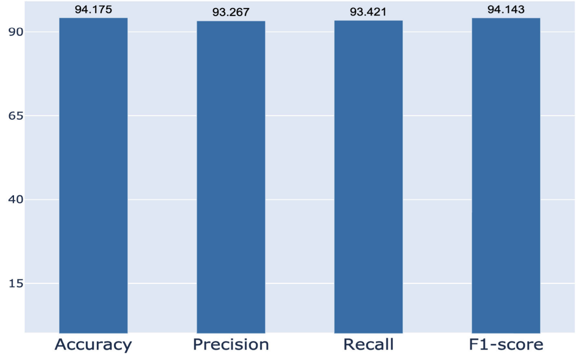


FIGURE 8. Evaluation metrics results of proposed approach with adversary nodes.

legitimate traffic. The training models of infected devices were also considered while performing the aggregation process. In the federated learning mechanism, parameters of on-device trained models of multiple devices will be combined in the aggregation process. It promotes knowledge sharing by averaging the parameters of trained models from all participating devices, thereby reducing the negative impact of adversary samples and generating a global optimized model as an outcome. Though the accuracy trends of infected devices are less, the global model that we generate after the federated aggregation process has maintained optimal performance. Figure 8 shows the evaluation metrics of our proposed approach with malicious traffic data. Our results show that the global model obtained after the 10th round of aggregation has maintained optimal accuracy by the use of federated learning in detecting attacks.

Considering the current extent of our approach, it is still possible for an attacker to perform poisoning attacks by injecting malicious traffic samples through compromised IoT devices. If the malicious samples were injected at a slow, gradual pace, deep learners might fails to detect those slight deviations of network behavior and considers them as normal traffic. To strengthen our approach, we plan to have a mechanism that can forestall poisoning attacks by implementing a sensitive outlier detection filtering technique. This method would analyze retrained models of multiple gateways and eliminate malicious/outlier models from participating in the aggregation process. Using this technique, we could identify the compromised devices in the network, and an alert can be made for intrusion.

E. DISCUSSION

This part discusses the computation capability of nodes used in the experiment and whether using the proposed approach with IoT devices in real-time may introduce any significant overhead interrupting their normal operations. Second, we discuss the possible application scenarios of our approach concerning device type and data.

1) COMPUTATION OVERHEAD

We have created virtual device instances using the Pysyft deep learning framework to demonstrate the functionality of our proposed approach. As these devices are virtual, the computational ability is the same as the runtime we used for executing our approach. On-device monitoring is hardly feasible in real-time IoT devices due to their resource limitation concerning computational ability, memory, and energy. Our proposed approach is based on the assumption that an intrusion detection system is installed on the security gateways and these gateways monitor the traffic of IoT devices that are connected to them. This approach supports most IoT devices as it is not required by the devices to be equipped with high computational power, and the work relying on computation power will be carried out at the security gateways. Thus, normal operations of IoT devices will unlikely be get affected in real-time scenarios.

2) APPLICATIONS

In this paper, we have provided a practical use case of multi-view learning in the IoT network domain. This approach can be implemented in the network architectures where MQTT protocol is used for communication with IoT devices and other communication protocols such as TCP and Modbus. This method can be applied to different types of devices rather than a single type because the device traffic is considered for performing intrusion detection rather than its specifications. Different devices have different functionality, so the data associated with the device and attacks corresponding to a device will also be different. The ideology of multi-view learning can be followed for most of the protocols. The network data of IoT devices that use any protocol for communication can be segmented based on the nature of features that corresponds to an attack. This approach can be implemented using the security service models that are available with service providers. FL helps in maximizing the training process with less data. It can be used irrespective of the communication protocol and end devices. Federated learning fosters the idea of Secure Multiparty computation. It prevents the end device data from being exposed to other parties and promotes differential privacy of data.

V. CONCLUSION

In this paper, we proposed a federated learning-based intrusion detection approach with multi-view ensemble learning. Our approach enhances the capability of identifying intrusions with a higher accuracy rate. Enhanced security and privacy levels are ensured as the data stays intact on the end devices. Multi-view learning enables the thorough analysis and understanding of attack patterns from multiple perspectives. The trained ML models learnings from multiple views are fused with an ensembler which increases the prediction accuracy of our approach. Our future research will focus on exploring unsupervised and reinforcement ML algorithms that can further enhance intrusion detection by identifying untrained attacks. In addition, we will focus on a mechanism that can forestall gradually injected poisoning attacks by implementing an outlier detection filtering technique. This method would analyze retrained models of multiple gateways and eliminate malicious/outlier models from participating in the aggregation process.

REFERENCES

[1] S. Kraijak and P. Tuwanut, "A survey on Internet of Things architecture, protocols, possible applications, security, privacy, real-world implementation and future trends," in *Proc. IEEE 16th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2015, pp. 26–31.

[2] R. K. Kodali, G. Swamy, and B. Lakshmi, "An implementation of IoT for healthcare," in *Proc. IEEE Recent Adv. Intell. Comput. Syst. (RAICS)*, Dec. 2015, pp. 411–416.

[3] B. S. Babu, K. Srikanth, T. Ramanjaneyulu, and I. L. Narayana, "Iot for healthcare," *Int. J. Sci. Res.*, vol. 5, no. 2, pp. 322–326, 2016.

[4] N. M. Kumar and A. Dash, "Internet of Things: An opportunity for transportation and logistics," in *Proc. 23rd Int. Conf. Inventive Comput. Informat. (ICICI)*, Nov. 2017, pp. 194–197.

[5] T. Malche and P. Maheshwary, "Internet of Things (IoT) for building smart home system," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Feb. 2017, pp. 65–70.

[6] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-Khah, and P. Siano, "Iot-based smart cities: A survey," in *Proc. IEEE 16th Int. Conf. Environ. Electr. Eng. (EEEIC)*, Jun. 2016, pp. 1–6.

[7] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of Things (IoT) security: Current status, challenges and prospective measures," in *Proc. 10th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Dec. 2015, pp. 336–341.

[8] A. Aris, S. F. Oktug, and S. B. O. Yalcin, "Internet-of-Things security: Denial of service attacks," in *Proc. 23nd Signal Process. Commun. Appl. Conf. (SIU)*, May 2015, pp. 903–906.

[9] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in *Proc. IEEE Int. Syst. Eng. Symp. (ISSE)*, Oct. 2017, pp. 1–7.

[10] H. Hindy, E. Bayne, M. Bures, C. R. Atkinson, C. Tachtatzis, and J. A. X. Bellekens, "Machine learning based iot intrusion detection system: An MQTT case study," 2020, *arXiv:2006.15340*. [Online]. Available: https://arxiv.org/abs/2006.15340

[11] C.-H. Mao, H.-M. Lee, D. Parikh, T. Chen, and S.-Y. Huang, "Semi-supervised co-training and active learning based approach for multi-view intrusion detection," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2009, pp. 2042–2048.

[12] B. B. Zarpelão, R. S Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.

[13] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019.

[14] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2661–2674, May 2013.

[15] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699–140725, 2020.

[16] Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen, and W. Pan, "Intrusion detection for wireless edge networks based on federated learning," *IEEE Access*, vol. 8, pp. 217463–217472, 2020.

[17] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[18] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, early access, May 5, 2021, doi: 10.1109/JIOT.2021.3077803.

[19] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[20] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[21] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federated learning on user-held data," in *Proc. NIPS Workshop Private Multi-Party Mach. Learn.*, Barcelona, Spain, 2016.

[22] K. A. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. M. Kiddon, J. J. Konečný, S. Mazzocchi, B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," in *Proc. Conf. Syst. Mach. Learn. (SysML)*, Stanford, CA, USA, 2019.

[23] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, May 2016, pp. 1–6.

[24] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep learning-based intrusion detection for IoT networks," in *Proc. IEEE 24th Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Dec. 2019, pp. 256–25609.

[25] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 756–767.

[26] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[27] R. Zhao, Y. Yin, Y. Shi, and Z. Xue, "Intelligent intrusion detection based on federated learning aided long short-term memory," *Phys. Commun.*, vol. 42, Oct. 2020, Art. no. 101157.

[28] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021.

[29] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[30] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. New York, NY, USA: Springer, 2012.

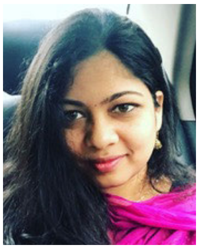
[31] H. Haddadpajouh, A. Mohtadi, A. Dehghantanaha, H. Karimipour, X. Lin, and K.-K.-R. Choo, "A multikernel and Metaheuristic feature selection approach for IoT malware threat hunting in the edge layer," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4540–4547, Mar. 2021.

[32] A. Ziller, A. Trask, A. Lopardo, B. Szymkow, B. Wagner, E. Bluemke, J.-M. Nounahon, J. Passerat-Palmbach, K. Prakash, N. Rose, T. Ryffel, Z. N. Reza, and G. Kaissis, *PySyft: A Library for Easy Federated Learning*. Cham, Switzerland: Springer, 2021, pp. 111–139.

[33] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, and C. Benavides, "Multiclass classification procedure for detecting attacks on MQTT-IoT protocol," *Complexity*, vol. 2019, pp. 1–11, Apr. 2019.

[34] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a new dataset for machine learning techniques on MQTT," *Sensors*, vol. 20, no. 22, p. 6578, Nov. 2020.

DINESH CHOWDARY ATTOTA received the B.Tech. degree in computer science from Jawaharlal Nehru Technological University, Kakinada, India, in 2018. He is currently pursuing the master's degree in computer science with Kennesaw State University (KSU), Marietta, GA, USA. He is currently working as a Research Assistant with the Decentralized Science Laboratory (dSL), KSU. His research interests include machine learning, IoT security, and computer vision.



VIRAAJI MOTHUKURI (Member, IEEE) is currently a Research Assistant with the College of Computing and Software Engineering (CCSE), Kennesaw State University (KSU), Marietta, GA, USA. She has several years of experience in Java and middleware technologies working with WIPRO and JP Morgan companies. She has a professional certification in machine learning to her credit. Her research interests include machine learning, hyperledger fabric, blockchain systems, and decentralized applications.

REZA M. PARIZI (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science, in 2005 and 2008, respectively, and the Ph.D. degree in software engineering, in 2012. Prior to joining Kennesaw State University (KSU), Marietta, GA, USA, he was a Faculty Member with New York Institute of Technology. He is currently the Director of the Decentralized Science Laboratory (dSL), KSU. He is also a consummate AI technologist and cybersecurity researcher with an entrepreneurial spirit. His research interests include research and development in decentralized AI, cybersecurity, blockchain systems, smart contracts, and emerging issues in the practice of secure software-run world applications. He is a Senior Member of the IEEE Blockchain Community and ACM.

SEYEDAMIN POURIYEH (Associate Member, IEEE) received the M.Sc. degree in information technology engineering from Shiraz University, in 2009, and the Ph.D. degree in computer science from the University of Georgia, in 2018. He is currently an Assistant Professor of information technology with Kennesaw State University (KSU), Marietta, GA, USA. His primary research interests include federated machine learning, blockchain, and cyber security.