

<https://doi.org/10.48047/AFJBS.6.14.2024.1870-1878>



African Journal of Biological Sciences

Journal homepage: <http://www.afjbs.com>



Research Paper

Open Access

Intrusion detection in cyber-physical systems systems based on a federated learning approach

Muhammed Almendli¹, Jamshid Bagherzadeh Mohasefi¹

¹Department of Computer Engineering, Faculty of Electrical and Computer Engineering, Urmia University, Urmia, Iran

Volume 6, Issue 14, Aug 2024

Received: 01 June 2024

Accepted: 11 July 2024

Published: 29 Aug 2024

[doi: 10.48047/AFJBS.6.14.2024.1870-1878](https://doi.org/10.48047/AFJBS.6.14.2024.1870-1878)

Abstract- Cyber-physical systems (CPSs), a new computing system to control industrial infrastructures, are widely used in many key areas such as manufacturing, energy, and safety management systems. The rapid involvement of CPS in industry has considerably expanded the range of cyber threats. Many machine-learning methods have been employed for the design of effective anomaly detectors in CPSs. Currently, federated learning methods have been applied to distributed machine learning. The distributed nature of some CPS systems creates a potential to use federated learning in this ecosystem. In this paper, we have applied three federated learning methods, in various scenarios, over three datasets obtained from SWaT (Secure Water Treatment) data. This dataset is obtained from an operational water treatment testbed. We have done a sort of preprocessing and feature selection on SWaT, to get a better and clean dataset. Then federated and non-federated (central based) methods are applied in various scenarios. The results of using federated-based methods are very promising and in many cases even better than central-based methods.

Keywords: Cyber-physical systems, Intrusion detection, Deep learning, Federated learning, Network security.

1. INTRODUCTION

Cyber-Physical Systems (CPSs) are specifically designed for controlling industrial systems that work in critical environments. These systems are usually comprised of industrial equipment monitored by many sensors and controlled using many actuators. These sensors and actuators are controlled by Supervisory Control and Data Acquisition (SCADA) systems. Sensor networks are usually deployed in the interior or on the borders of plants to gather various critical information with the purpose of making correct and safe operation of the physical plants. With the help of this information, plant staff have the opportunity to do real-time reactions to some changes in physical plants manually or automatically using actuators [1]. CPSs are typically large and geographically dispersed, hence they are network-connected for remote monitoring and control.

Such network connectivities open up the likelihood of cyber-attacks. Such possibilities make it necessary to develop techniques to defend CPSs against cyber or physical attacks. Among other security mechanisms, Intrusion Detection Systems (IDSs) are the most important defense tools against network attacks [2]. The main purpose of IDS is to detect and prevent intrusions within an IT infrastructure, and then alert the relevant people. All intrusion detection systems analyze some sort of data to detect attacks from current data gathered online from various sources. To detect intrusion, they need to analyze current data against previous information (stored in a database or a model). Intrusion detection in CPS includes many challenges [3]. These challenges include data gathering, keeping data private, adaptability to specific CPS, performance, coping with new zero-day attacks, and many other issues.

Many AI-based methods are proposed in the literature for intrusion and attack detection tasks in CPSs [4], [5], [6], [7], [8], [9], [10] etc. They usually use classification (such as convolutional neural networks [6]) or clustering methods (such as generative adversarial networks [4], [5]). Most of these methods, for CPS intrusion detection, use general-purpose datasets available for intrusion detection, such as NSL-KDD [11], and UNSW-NB15 [12]. However, numerous current studies showed that for the current network threat environment, these data sets do not reflect network traffic and modern low-footprint attacks on CPSs. There are also some IDS systems [13] implemented and tested using specific data sets available for CPSs such as SWaT [14] and WDT [15].

Most of the existing IDSs are based on conventional centralized Machine Learning (ML) methods. These centralized ML methods, not only confront difficulties in collecting and managing data across heterogeneous sources but also have always come with privacy risks to personal data leakage, misuse, and abuse. Centralized ML algorithms require training data to be collected in a data server. Collecting, aggregating, and integrating high-volume data dispersed over various data sources as well as managing data privacy are challenging tasks.

To overcome such challenges, Federated Learning (FL), proposed by Google researchers in 2016, has appeared as a promising solution and attracted attention from both industry and academia [16] [Mc Mahan]. Generally, FL is a technique to implement an ML algorithm in distributed collaborative learning settings wherein the algorithm is executed on multiple local datasets stored at isolated data sources (i.e., local nodes) such as smartphones, tablets, and PCs, without the need for collecting and processing the training data at a centralized data server. FL allows local nodes to collaboratively train a shared ML model while keeping both dataset and computation at local nodes. Only the results of the training (i.e., parameters) are exchanged at certain times. The orchestration and management of nodes can be done by a central server to coordinate the training process (centralized FL) or can be done in a peer-to-peer way (i.e., decentralized FL) to aggregate the training results and calculate the global model.

The natural advantage of FL compared to the traditional ML approaches is the ability to ensure data privacy because personal data is stored and processed locally, and only model parameters are exchanged. This capability of FL could potentially inaugurate new opportunities to implement some sorts of ML algorithms for applications and services without acquiring clients' private data. Consequently, FL has emerged as a prospective solution that facilitates distributed collaborative learning without disclosing original training data. The geographically dispersed nature of CPS nodes along with the need for data privacy provides a high potential to apply FL-based IDS systems in CPS environments. There is some research done in this regard.

In 2020, Li et al. [17] proposed an FL framework to collaboratively construct the CPS intrusion detection model. Chatterjee et al. [18] proposed an FL-based intrusion detection system in which federated average (FEDAVG) and noise tolerance are used to address tag noise. Nguyen et al. [19] used FL to collect aggregate behavior profiles to build anomaly detection systems.

SWaT is a dataset extracted from an experimental water treatment system. There are two types of data extracted in the SWaT testbed: sensory dataset that includes the information read or written to sensors or actuators of the CPS system, and network dataset that represents the packets transferred in HTTP/HTTPS protocol between network nodes. Several machine-learning techniques have been employed for the design of an effective anomaly detection system on the SWaT dataset.

Figure 1: The overall structure of the proposed method

All of the IDS

research done on the SWaT dataset considers only sensory dataset [13]. Moreover, there are no FL-based IDSs applied to SWaT dataset. In favor of these gaps, in this paper, we contribute the following novelties:

- 1- We implemented three FL-based algorithms including FL-based Artificial Neural Network (FL-ANN), FL-based Deep Neural Network (FL-DNN),

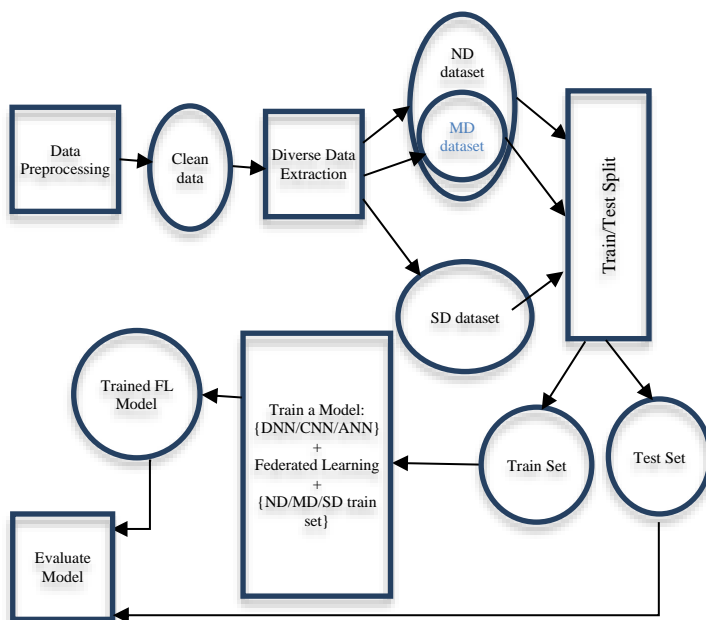
and FL based Convolutional Neural Network (FL-CNN) on Sensory dataset as well as Network data set.

- 2- We extracted Modbus protocol data from the Network data and implemented the above FL-based methods on Modbus data.
- 3- We have implemented centralized ML methods and compared FL-based methods with them.

This paper is organized as follows. In section two, we explain the data preparation and preprocessing operations. Then in section three, we explain the proposed methods including various federated learning methods. Then we show the results of the implemented methods in section four. Finally, section five presents the discussion and concludes the paper.

2. METHOD

Figure 1 shows the steps of the proposed method. We use the SWaT dataset for our Federated Learning based Intrusion Detection System (FL_IDS). This dataset goes under rigorous preprocessing. Then it is divided into 3 sub-datasets called MD, ND, and SD datasets. These three datasets are then split into train and test datasets (80% train, 20% test) and the various models are trained in the training part with various combinations of methods (DNN, CNN, or ANN) on selected datasets (SD, MD, or ND). The trained models are then evaluated using test datasets (MD, ND, or SD test sets). In the following subsections, we describe each step in more details.



2.1. Data Preprocessing

A representative dataset of intrusions in CPSs would give an invaluable resource for the development of new intrusion detection methods. Currently, only a limited number of datasets are available to evaluate ML-based

anomaly detection in CPSs [20]. However, some of these datasets are based on unrealistic implementations. Usually, data sets for CPSs consist of many information including Network traffic gathered using devices installed in the network to catch packets transferred in the network. There could be another type of data that represents snapshot of sensors and actuators values gathered during time steps. There might be other forms of data depending on the types of CPSs.

Mathur and Tippenhauer created a small-scale, but fully operational water treatment system (SWaT) testbed for cybersecurity research [14]. They released the final dataset containing process state variables, selected packet features, and logs of performed attacks. SwaT is an operational laboratory-size water treatment plant with a small footprint, producing 5 gallons/minute of doubly filtered water. Its main purpose is to generate data for research in the design of secure and safe CPS. Data from the sensors is available to the SCADA system and recorded for subsequent analysis. In the subsequent sections, we call this a Sensory Dataset (SD). There is a star network that allows the SCADA system to communicate with the PLCs dedicated to processes of SWaT. The data gathered from sensors and actuators are encapsulated in TCP packets which are transferred through TCP/IP protocol to the SCADA system. The TCP packets gathered from the communication of PLCs and SCADA system are called Network Dataset (ND) in the subsequent sections. Inside the Network dataset, there is a value known as Modbus_value which contains the encoded information of sensors and actuators communicated between SCADA and PLCs. This invaluable feature of every packet is extracted from the packets and gathered in a set called Modbus Dataset (MD). As per our information, there are no papers that have analyzed the SD, ND, and MD datasets altogether.

SWaT dataset includes two folders: sensory dataset and network dataset. We performed preprocessing on both datasets separately. Then we extracted a subset of the network dataset called the Modbus dataset which is explained in the following. We checked the datasets for missing values, and unknown values first and resolved them. Then we checked outliers in various ways, including histogram plots, boxplots, etc, and smoothed them. To normalize the features, we used MinMaxScaler from sklearn. This converts all the values to the range of [0, 1]. At the last step of data preprocessing, we performed a feature selection using a decision tree and correlation analysis. After careful analysis, we selected features with the most direct or inverse relation with the target label.

The network dataset (ND) includes features with different types which need transformations. First, we removed the features like ID, Date, Time, and Modbus_transaction_id which are not important in attack detection. For further processing, we considered only the features with more than one possible value. Next, we

converted categorical features (like IP addresses, i/f_dir, ...) to numeric values for further processing. This is done using `ce.BinaryEncoder()` function from `category_encoders` package. There exists an important attribute, `Modbus_Value`, which contains a list of 38 hexadecimal numbers, showing the values read or written to sensors and actuators using the Modbus protocol. Modbus protocol is usually used for communication in PLC devices. We converted this to a list of decimal numbers. Finally, ND contained 20 numeric features representing TCP features and 38 features for `Modbus_Value` total 58 features. The 38 features extracted from `Modbus_Value` construct our MD dataset.

2.2. Training FL-based models

The industrial control systems may include data gathered from various geographically dispersed sensors and data collecting nodes. In such environments, there are two challenges to storing whole data in a central server or cloud. The first problem is data privacy and security issues, where data owners (end nodes) do not like to share their private data in a central shared server, i.e. they might not trust the server. The other problem is data communication overload especially when the data generation rate and volume are high. As we explained in the previous sections, FL is proposed to tackle these challenges. In this section, we apply the FL-based machine/deep learning methods on different datasets to evaluate the effectiveness and performance of FL based approach compared to the centralized machine learning approach.

In this research, the centralized FL method is used to train an FL-based model. We assume there is a centralized server that coordinates the whole training process. The server first determines a global model to be trained. Then it starts a repeated process where in each repetition it performs the following process. It selects local nodes (participants) for each training round and requests the selected local nodes to locally compute the local gradients using their optimization methods and local datasets. The server next acquires and aggregates local training results sent by the local nodes and updates the global model based on the aggregated results. Then it disseminates the updated model to the local nodes. This process repeats in some rounds until the server decides to terminate the training process when the global model satisfies some requirements (e.g., it reaches a higher accuracy).

Local nodes, on the other hand, train the local model over their local dataset as requested, and send the training results back to the server. The algorithm of this centralized FL approach is illustrated in Figure 2. It includes the following steps:

1. Local Node Selection and Global Model Dissemination: The server selects a set of Local nodes to be involved in the training process. It then informs the global model parameters (or the global model updates) to the local nodes for the next training round.

2. Local Computations: Once receiving the global model parameters from the server, the local nodes update their current local model and then train the updated model using the local datasets that exist in the nodes. This step is performed at local nodes, and it requires local nodes to perform training algorithms and send the local model parameters to the server. (client side)
3. Local Models Aggregation: The server receives and aggregates the locally trained models in order to update the global model. This aggregation mechanism is required to integrate some privacy-preserving techniques such as secure aggregation, differential privacy, and advanced encryption methods to prevent the server from inspecting individual ML model parameters.
4. Global Model Update: The server performs an update on the current global model based on the aggregated model parameters obtained in step 3. Then the updated global model is disseminated to local nodes for the next training round. This 4-step algorithm is repeated until the global model has reached sufficient accuracy.

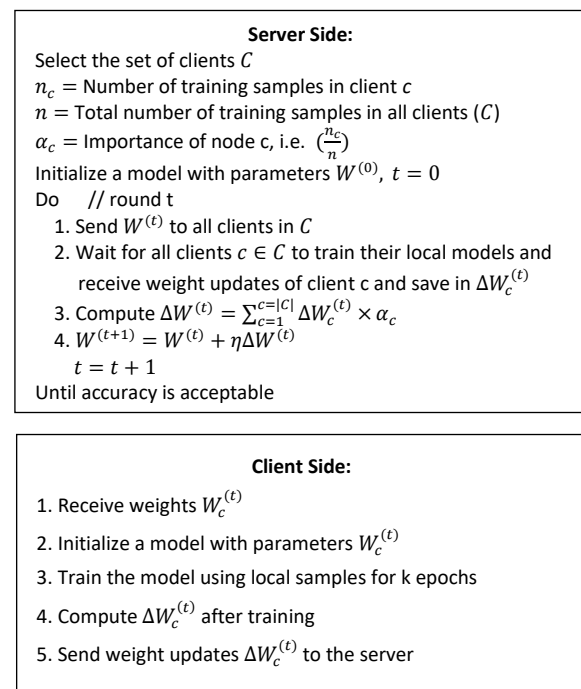


Figure 2: Algorithms of the server and local clients in federated learning approach

2.2.1. Federated Learning Using Sensory Dataset

The sensory dataset includes 51 features representing data gathered by sensors and sent to actuators in the water treatment factory testbed. We have applied 3 different methods, Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Deep Neural Networks (DNN), on a Sensory Dataset (SD), using a federated learning framework. The structure of ANN,

DNN, and CNN are represented in **Error! Reference source not found.** with hyper-parameters learning_rate=0.01, comm_rounds=30, and num_clients=10 in all models.

2.2.2. Federated Learning using Modbus and Network Datasets

SWaT network dataset includes features related to network traffic (TCP/IP traffic). There is an important feature called Modbus_value in the features of this dataset. We extracted three types of datasets from the network dataset: a) All features including Modbus_value, b) Only TCP/IP features without Modbus_value, and c) only the Modbus_value dataset. The only TCP/IP features without Modbus_value (part b) produced low accuracy in our experiments. The first and third datasets (Complete data, and only Modbus data) produced good results in our experiments. So in this research we only present the results of algorithms on the complete network dataset (called network dataset or ND as short) and only Modbus dataset (called MD as short). The network dataset included 43 features after all sorts of preprocessing, while Modbus dataset included 22 features.

Three methods, ANN, CNN, and DNN are applied on ND and MD datasets. These methods are applied in two ways, FL-based approach and single centralized

Accuracy/Loss	Fed. Learning 1 epoch (Final result)	Fed. Learning 6 epochs (Final result)	A single global model without FL
ANN			
Accuracy	0.9822524004	0.9922430654	0.991598506
Loss	0.0647162720	0.0226363576	0.026588220
DNN			
Accuracy	0.9819985775	0.9919096728	0.9954325213
Loss	0.0539590927	0.0234058070	0.0126319304
CNN			
Accuracy	0.9878756223	0.9933988264	0.9947323968
Loss	0.0422660410	0.0181997884	0.0148833394

model. The structure of the all models are similar to those in **Error! Reference source not found.**, with slight difference in the number of units and different input_shapes, which is (22,) for MD, and (42,) for ND datasets. For the sake of space we do not mention the structure of these models here.

```

ANN
model = Sequential()
model.add(Dense(25,
    input_shape=(51,)))
model.add(Activation("relu"))
model.add(Dense(20))
model.add(Activation("relu"))
model.add(Dense(1))
model.add(Activation("sigmoid"))
    
```

```

DNN
model = Sequential()
model.add(Dense(30,
    input_shape=(51,)))
model.add(Activation("relu"))
model.add(Dense(25))
model.add(Activation("relu"))
model.add(Dense(20))
model.add(Activation("relu"))
model.add(Dense(15))
model.add(Activation("relu"))
model.add(Dense(10))
model.add(Activation("relu"))
model.add(Dense(1))
model.add(Activation('sigmoid'))
    
```

```

CNN
model = tf.keras.Sequential()
model.add(Conv1D(filters=32, kernel_size=6, strides=1, activation='relu',
    input_shape = (51,1)))
model.add(MaxPooling1D(pool_size=2))
model.add(Conv1D(filters=64, kernel_size=3, strides=1, activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(10, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
    
```

Figure 3: Structure of ANN, CNN, and DNN on sensory dataset

3. RESULTS AND DISCUSSION

The ANN, DNN, and CNN of the structure of **Error! Reference source not found.** is implemented in the global model (server) as well as local nodes (clients). In our federated learning experiments we have used ten clients. To compare the results of the FL based methods with the single traditional method, we conducted three types of experiments: a) FL method where each local node uses one epoch in each round, b) FL method where local nodes apply six epochs in each round, and c) traditional centralized learning method using 100 epochs.

3.1. ANN, CNN, DNN on sensory dataset

The network structure used in server, local nodes, and single global model is the same as those mentioned in the **Error! Reference source not found.** In this structure the federated learning method with six epochs has mentioned promising results compared to the centralized method. Both the accuracy and Loss of the FL based method with six epochs has got the best accuracy and lowest loss in ANN structure, however, in DNN and CNN they are slightly weaker than centralized one. **Error! Reference source not found.** represents the evaluation of FL based method using ANN, DNN, and CNN on the sensory dataset. Many experiments are done using FL-based scenarios and here we show the average of those experiments.

Table 1: Accuracy and Loss of methods on the sensory dataset

In all FL-based methods we have used 30 rounds of communication. In each round the server sends the current parameters of its aggregated model to 10 local clients and each client starts a local training using its local dataset for number of epochs mentioned (one or six) after which the client returns back the updates needed of parameters to the server. Then server updates the current parameters by averaging the parameter updates collected from 10 clients. This process is repeated 30 times. **Error! Reference source not found.** illustrates this 30 rounds in detail and how the accuracy and loss change during the system's evolution for FL_CNN method. It is seen that after 20th round the improvement is very slight and tiny. So the improvement after 30th round is negligible. The results show that federated learning has very good results. In the literature, usually central model gets better results than FL as it has access to whole data in one place. In this framework we have got very promising results for FL. It

is because the local nodes have many samples with IID (Identical and Independent Distribution) distributions.

3.2. ANN, CNN, DNN on network and Modbus datasets

Error! Reference source not found. represents the evaluation of FL based methods using ANN, DNN, and CNN on network and Modbus datasets. The network structure used in server, local nodes, and single global model is similar to those mentioned in the **Error! Reference source not found.**. In general, IDS system gets lower results on ND and MD datasets compared to SD dataset. However here we have got better results for FL based methods in many situations compared to centralized method, in overall. Especially CNN structure with six epochs has got best results compared to the centralized method. In DNN structure with MD dataset the accuracy is higher than others and best in whole table. The numbers in table are the average of 5 runs for each scenario.

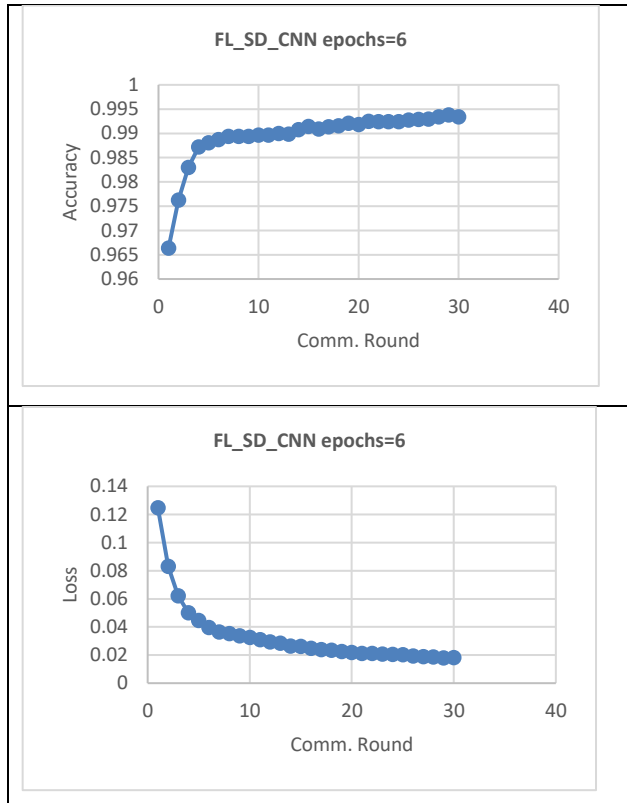
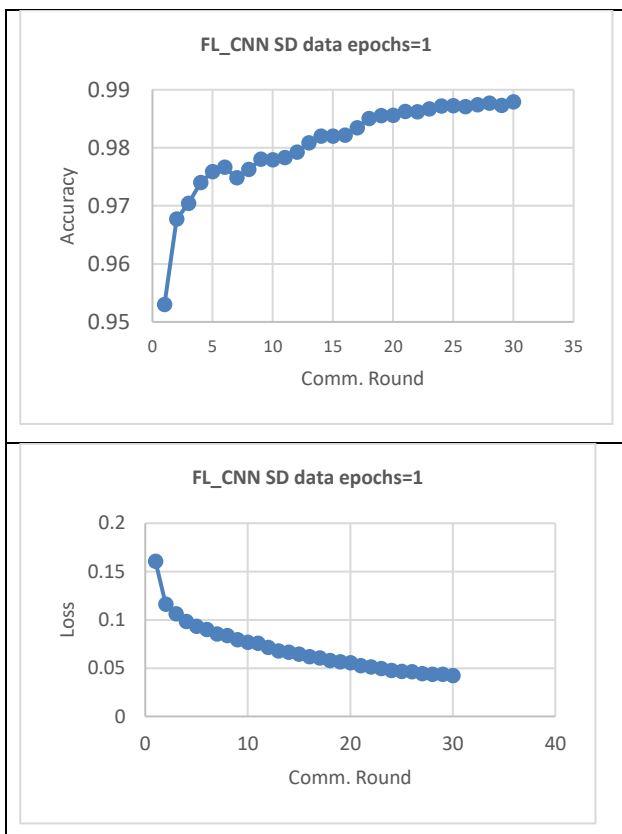


Figure 4: Accuracy and Loss of FL based CNN method on sensory dataset on 30 rounds

In the **Error! Reference source not found.**, we illustrate execution of FL_DNN on Modbus dataset, during 30 rounds of communication and it is seen that how the accuracy and loss change during the system's evolution.

Table 2: Accuracy and loss of methods on network and Modbus dataset

	Network dataset		
	FL one epoch (Final result)	FL six epochs (Final result)	A single global without FL
	ANN		
Accuracy	0.9048230331	0.9332540771	0.9292638661
Loss	0.2281062304	0.1582868695	0.1554569751
	DNN		
Accuracy	0.9293523972	0.9355748507	0.9404250741
Loss	0.1606968190	0.1421228350	0.1282967924
	CNN		
Accuracy	0.9286378271	0.9416012700	0.9328493647
Loss	0.1616785976	0.1313606650	0.1530912518

Modbus dataset			
	FL one epoch (Final result)	FL six epochs (Final result)	A single global without FL
ANN			
Accuracy	0.896589666	0.930079615	0.934702188
Loss	0.221318066	0.155525118	0.142489969
DNN			
Accuracy	0.918753992	0.944819998	0.939141377
Loss	0.167381153	0.125792593	0.133489876
CNN			
Accuracy	0.927689282	0.942935556	0.918633842
Loss	0.156594709	0.125022545	0.185667455

Federated learning methods implemented in this research use IID distribution. The dataset is divided randomly between 10 clients. Client use their own local data to build their local models and send the resulted parameters (parameter updates) to the central server. The method used for model aggregation in the central server is FEDAVG method. However we can used other aggregation methods as well. The good results obtained here is the high accuracied obtained from FL based methods compared to the centralized ML models (without using FL based approach). The main reason that we think has caused such a result is the IID nature of dataset plus high number of local data in local nodes. Currently we are working on different versions of this research considering non-IID dataset and few-shot dataset in local nodes. Moreover we are applying other aggregation methods as well. In our opinion this is the first work which has analyzed SWaT dataset from this point of view and so we are not able to compare the work with similar FL-based methods on SWaT dataset. For comparison we have mentioned a work done centrally on SWaT dataset which has very low accuracy compared to ours as mentioned in the **Error! Reference source not found.**

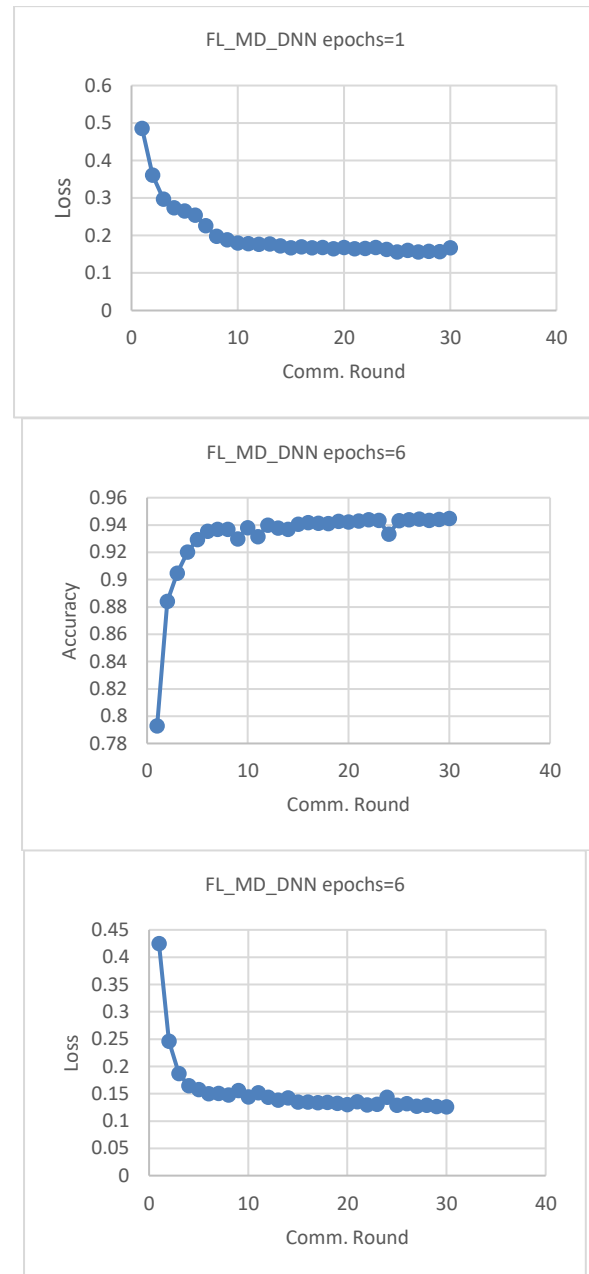


Figure 5: Evolution of FL based DNN method in 30 rounds on Modbus dataset

Table 3: Comparison of our method with the existing methods

Method	Accuracy	Precision	Recall	F1_score
Our best FL-based method SD dataset	99.3399%	99.8727%	98.2341%	99.0466%
Our best FL-based method ND dataset	94.1601%	95.2541%	93.8711%	94.5575%
Our best FL-based method MD dataset	94.4819%	94.2381	93.5671%	93.9014%

SVM+ (LUPI) [21]	74.2534%	77.251%	74.1692%	73.4782%
------------------	----------	---------	----------	----------

4. CONCLUSION

CPSs are specifically designed for controlling industrial systems that work at critical environments. CPSs are typically large and geographically dispersed, thus they open up the likelihood of cyber-attacks. To detect attacks and intrusion, we need to analyze current data against previous information using IDS systems. IDSs in CPS include face many challenges including data gathering and keeping data privacy, and many other issues. Existing IDSs based on conventional centralized Machine Learning (ML) methods confront with difficulties in collecting and managing data and come with privacy risks to personal data leakage, misuse, and abuse. To overcome such challenges, Federated Learning (FL) has appeared as a promising solution and attracted attention from both industry and academia. FL allows local nodes to collaboratively train a shared ML model while keeping both dataset and computation at local nodes. The geographically dispersed nature of CPS nodes along with the need for data privacy provides a high potential to apply FL based IDS systems in CPS environments. In this paper we implemented FL based neural network and deep learning methods for anomaly detection in CPSs. We used dataset obtained from a water plant testbed called SWaT, which included 2 databases: network dataset, and sensory dataset. Many steps of preprocessing are done in the datasets. The so called Modbus_value was extracted from network dataset which is treated separately. Finally the experiments mentioned that FL based ANN, DNN, and CNN produces very well results on sensory dataset with more than 99% of accuracy. On the network and Modbus datasets our method performed significantly better than previous methods in the literature. Our experiments in this paper used IID data. In the future, we have plan to experiment more scenarios using non-IID and multiclass situations along with diverse aggregation methods in the same dataset. We also plan to perform other advanced methods like continual learning to completely simulate a real environment of an IoT and cloud based CPS anomaly detection system.

REFERENCES

[1] X. Jiang and S. Li, "Plume front tracking in unknown environments by estimation and control," *IEEE Trans. Ind. Informat.*, DOI 10.1109/TII.2018.2831225.

[2] S. S. S. Sindhu and S. Geetha, *Network Intrusion Detection System using Machine Learning Techniques: A Quick Reference*, LAP LAMBERT Academic Publishing, 2013.

[3] K. Zhou, T. Liu and L. Liang, "Security in cyber-physical systems: Challenges and solutions,"

International Journal of Autonomous and Adaptive Communication Systems, vol. 10, no. 4, pp. 391-408, 2017.

[4] W. Liang, K. C. Li, J. Long, X. Kui and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2063-2071, 2020.

[5] P. F. de Araujo-Filho, G. Kaddoum and D. R. Campelo, "Intrusion detection for cyber-physical systems using generative adversarial networks in fog environment," *IEEE Internet Things J.*, vol. 8, no. 8, p. 6247-6256, 2021.

[6] Y. Lai, J. Zhang and Z. Liu, "Industrial anomaly detection and attack classification method based on convolutional neural network," *Secur. Commun. Netw.*, vol. 2019, pp. 1-11, 2019.

[7] Y. Yang, K. McLaughlin, S. Sezer and e. al., "Multiattribute SCADA-specific intrusion detection system for power networks," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1092-1102, 2012.

[8] Y. Xiao, J. Liu and L. Zhang, "Cyber-Physical System Intrusion Detection Model Based on Software-Defined Network," in *IEEE 12th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2021.

[9] B. Tang, Y. Lu, Q. Li and Y. Bai, "A Diffusion Model Based on Network Intrusion Detection Method for Industrial Cyber-Physical Systems," *Sensors*, vol. 23, no. 3, 2023.

[10] S. Adepu, K. N. Kandasamy and A. Mathur, "EPIC: An Electric Power Testbed for Research and Training in Cyber Physical Systems Security," in *Lecture Notes in Computer Science*, vol 11387, 2019.

[11] M. Tavallaee, E. Bagheri, W. Lu and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the IEEE International Conference on Computational Intelligence for Security & Defense Applications.*, Ottawa IL, USA, 2009.

[12] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *In Proceedings of the Military Communications and Information Systems Conference*, Cracow, Poland, May 2015.

[13] M. G. Raman, N. Somu and A. Mathur, "A multilayer perceptron model for anomaly detection in water treatment plants," *International Journal of Critical Infrastructure Protection*, vol. 31, 2020.

[14] N. O. Tippenhauer and A. P. Mathur, "Swat: A water treatment testbed for research and training on ics security," in *in 2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 2016.

[15] L. Faramondi, F. Flammini, S. Guarino and R. Seto,

"A hardware-in-the-loop water distribution testbed dataset for cyber-physical security testing," *IEEE Access*, vol. 9, pp. 385-396, 2021.

- [16] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artificial Intelligence and Statistics*, p. 1273–82, 2017.
- [17] B. Li, Y. Wu, J. Song, R. Lu, T. Li and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, p. 5615–5624, 2021.
- [18] S. Chatterjee and M. K. Hanawal, "Federated learning for intrusion detection in IoT security: A hybrid ensemble approach," *arXiv,2106.15349.*, 2021.
- [19] T. Nguyen, S. Marchal, M. Miettinen and H. Fereidooni, "DIoT: A federated self-learning anomaly detection system for IoT," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, Jul. 2019.
- [20] A. P. Gómez, L. F. Maimó, A. H. Celdrán and e. al., "On the generation of anomaly detection datasets in industrial control systems," *IEEE Access*, vol. 7, pp. 460-473, 2019.
- [21] M. Pordelkhaki, S. Fouad and M. Josephs, "Intrusion Detection for Industrial Control Systems by Machine Learning using Privileged Information," in *2021 IEEE International Conference on Intelligence and Security Informatics (ISI)*, San Antonio, TX, USA, 2021.



Muhammed Almendli    received the B.sc Eng. degree in technical computer engineering from Al-Rafidain University College, Iraq-Baghdad, in 2011 and the M.S. Information Technology Engineering-Information Systems Management from Imam Reza International University and researcher PhD degrees in Computer engineering-Computer Networks, he accepted for the 2021 academic year at Urmia University, Iran, and in 2012, he worked in Al-Rafidain University College as a Lecturer, and in 2022 he worked in Al-Mustafa University college as a Eteach Assistant .he teach the following material (computer Organization, AutoCAD, Electronic, Computer application , AI, Real Time System ,Computer Architecture). He can be contacted at email: muhammedalmendli@gmail.com



Jamshid Bagherzadeh Mohasefi    holds a PhD in Computer Science from Indian Institute of Technology, India. He is currently Professor of the Department of Computer Science at the Urmia University. His research topics include Artificial Intelligence, Machine Learning, and Network Security. His research has been funded by the Urmia University, Iranian Telecom Ministry, and Iranian Ministry of Science, Research, and Technology. He can be contacted at email: j.bagherzadeh@urmia.ac.ir.

BIOGRAPHIES