Fortified-Edge 5.0: Federated Learning for Secure and Reliable PUF in Authentication Systems

Seema G. Aarella
University of North Texas
Denton, Texas, USA
seema.aarella@unt.edu

Venkata P. Yanambaka
Texas Woman's University
Denton, Texas, USA
vyanambaka@twu.edu

Saraju P. Mohanty
University of North Texas
Denton, Texas, USA
saraju.mohanty@unt.edu

Elias Kougianos
University of North Texas
Denton, Texas, USA
elias.kougianos@unt.edu

***Abstract*—Physical Unclonable Functions (PUFs) are widely studied for the security of devices in the largely heterogenous Internet-of-Things ecosystem. The need for low-power and low-cost yet robust and reliable security systems is of prime importance in resource-constrained environments like smart villages. Using PUFs as a security primitive has the limitation of environmental effects that lead to bit flipping in the PUF response, the challenge in using PUFs is to overcome the bit errors without adding to the area overhead or computational overhead. This research proposes a novel bit error detection and correction algorithm implemented using Federated Learning (FL). The error detection and correction model uses the N-gram concept of Natural Language Processing (NLP). The FL model is implemented on Flower AI, the global model gets the locally trained model's parameters, updates itself, and shares the updated models with all the local models. At the edge, the use of FL for model training and updating enhances the efficiency of the authentication system that uses PUF Challenge-Response Pairs (CRPs), reduces the area overhead and power consumption, and improves the security of the PUF-based authentication system.**

***Index Terms*—Physical Unclonable Functions(PUFs), Smart Village, Collaborative Edge Computing, Federated Learning, Secure Authentication, Natural Language Processing**

I. Introduction

According to the statistics on IoT devices, it is expected that the number of IoT devices is expected to triple from 2020 to 2030, estimating the numbers to be in billions. Based on the emerging applications in areas such as smart healthcare, manufacturing, automotive industries, autonomous vehicles, gaming, and so on, there will be huge data generation that will exhaust the existing infrastructure. With the rise of time-sensitive applications, the processing is moving closer to the user edge. In the future, it is forecasted that the Edge Data Centers (EDC) could move closer to the data sources [1].

EDCs in a collaborative environment are a resourceful solution for faster computing with reduced latency and reduced load on the processors, through task offloading. Security of the EDC during load balancing requires a robust authentication and authorization system. The EDCs closer to user devices demand a security application that is low power consuming and computationally accommodatable. A suitable authorization and authentication system using PUFs was proposed in the research [2].

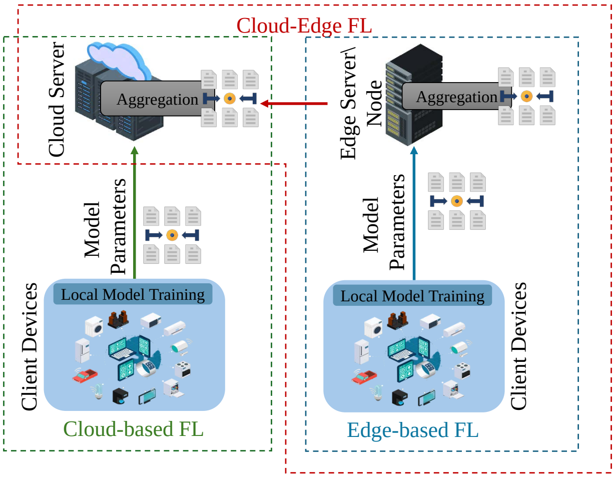


Fig. 1: Overview of Federated Learning Structure.

Although PUFs have been long studied as a robust solution for device identification and authentication in security applications, they are not without challenges. One of the prime challenges in implementing PUF is reliability, only the stable PUFs are considered secure, the main cause for instability in PUFs is environmental effects, the variations in the environment cause the bits in the PUF response to flip, thus reducing the stability and reliability of the PUF. To address this issue a novel ML-based PUF bit error detection and correction was proposed in the research [3]. This model is suitable for implementation at the edge of an IoT ecosystem that uses PUFs for authentication and authorization as it consumes low power, is computationally less intensive, and does not increase the area overhead.

Machine Learning has been a valuable enabling technology to use at the edge for developing security applications, it helps make intelligent and informed decisions concerning attack detection and prevention and mitigates security and privacy threats. ML can be leveraged to develop models to prevent or detect attacks like Distributed Denial-of-Service (DDoS), Intrusion, Eavesdropping, Malware, Man-in-the-Middle, Phishing, Spoofing, and so on [4]. Taking forward the capacity of ML is the emerging method FL that helps build models without exposing private data. FL with the capability of deploying and training on user devices where data is generated reduces the communication overhead and preserves privacy. FL allows the participation of a large number of devices enabling collaborative learning that will help in providing cloud-like computing capabilities at the edge [5].

FL structure is shown in Figure 1, there are 3 distinct frameworks for FL, it can be Cloud-based FL, where the model aggregation takes place at the cloud servers, Edge-based FL where the model aggregation is done at the edge servers without cloud dependency, and a hierarchical framework that involves both cloud and edge servers based on the distribution of aggregation. ML models that are low on processing and power requirements are most suitable for aggregation at the edge servers. The challenges in using FL at the edge lie in tackling the heterogeneity of the ecosystem, handling the volume of data, frequency of model updates, and so on. One must design the model in a way that avoids the communication overhead and uses sufficient data for model updates. This research uses the FL model for collaborative deployment and updation of the ML model for bit error correction, which overcomes the challenges.

II. RELATED PRIOR RESEARCH

The security and privacy of PUF CRPs need to be considered when designing PUF-based authentication system. The CRP dataset needs to be stored and the data is looked up when authentication requests come in. The dataset is a must for a verifier that identifies the corresponding response for a given challenge and rightly verifies the requesting device. The dataset needs to be secure and prevent any illegal access, that will compromise the security of the whole application. The CEC ecosystem that uses a PUF-based authentication system requires the CRP dataset to be stored in the participating servers, in scenarios like load balancing when a mutual authentication between servers or EDCs is needed on the go. However, the local availability of CRP datasets in multiple servers or EDCs could become a security threat. To address this issue research proposes the use of a Certificate Authority-based PUF authentication system that removes the need to store dataset in every EDC in CEC [6].

Error correction codes and Fuzzy extractors are largely used for bit error correction to increase the reliability of PUFs. These methods need publicly available helper data which is vulnerable to data manipulation attacks. A repetition code-like error correction protocol using machine learning is proposed in the research [7] that uses simulation challenges, the mechanism can predict corrected responses with 100% probability.

To improve the reliability of SRAM PUF, a novel lightweight one-layer convolution scheme is proposed in the research [8]. The scheme uses verification matrices and compares the verified and unverified matrix to generate PUF response. The reliability of the PUF responses reached upto 100%.

Studies have shown that ML techniques are effective in PUF-based authentication systems where they can be employed for error correction, simulated challenge, response generation, and so on. It can be said that ML can greatly contribute to enhancing security and privacy, reducing area overhead, and computation overhead. FL framework allows a network of devices to train a model without the need for centralized data, analysis of the performance of FL at the edge with various types of system heterogeneity, statistical heterogeneity, system statistical heterogeneity, and communication bandwidth, can be used to study the impact on model convergence, the knowledge of which can be used for optimizing the FL systems [9]. Combining the aspects of ML and using the FL framework the bit error correction model can be further enhanced to suit the needs of edge computing in a network while preserving data privacy, which is crucial for the security of the CRP dataset.

The following Table I shows the comparative study of various applications that use the FL framework. All these various research uses different datasets related to healthcare, medical, Image classification, and Intrusion detection, and apply FL for better results in detection and classification with enhanced data security and privacy.

III. NOVEL CONTRIBUTIONS OF CURRENT RESEARCH

Enhancing the reliability of the PUF-based secure authentication and authorization system is the prime focus of this research. This research uses a 64-bit Arbiter PUF for CRP generation [15]. The use of Machine Learning (ML) and Artificial Intelligence (AI) to support the security applications not only improves the security it also makes the system powerful against external attacks by providing attack detection and prevention options through various protocols across several levels of the IoT infrastructure. In the process of leveraging the benefits of ML and AI and reinforcing the PUF-based authentication system, the following are the novel contributions of the current research:

- Exploring FL for edge computing in a collaborative environment
- FL model with model training and deployment which is efficient in computation and power consumption
- Proposing a FL based framework for PUF bit error detection that uses ML algorithm
- ML model training using NLP approaches
- Global Model aggregation through parameters received from local models
- Global and Local Model training and testing on edge devices for computational efficiency

TABLE I: Comparative Table for State-of-the-Art Literature.

Research	Year	ML Algorithm	Dataset	Metrics
Karim et. al. [10]	2023	RainForest	WEKA-Hypothyroid	Accuracy, Precision, Recall, F1 Score
Jain et. al. [11]	2023	SGD	Adobe Stock	Accuracy
Korkmaz et. al. [12]	2022	Inception-v3	Medical Image Dataset	Accuracy
Chen et. al. [13]	2020	GRU (gated Recurrent Unit) and SVM	KDD CUP99	Accuracy, F1 Score
Mahadik et. al. [14]	2024	CNN	CICDDoS2019	Accuracy
Current Research Fortified-Edge 5.0	2024	K-mer Sequence	100K PUF Response Dataset	Accuracy

- Power analysis and computation analysis of the FL model

IV. Federated Learning at the Edge for PUF based Authentication

FL is a distributed approach used to train AI and ML models that allows collaboration while preserving data privacy and security. The key aspects of FL are decentralized training, privacy preservation, and collaborative learning. FL allows a baseline model to be shared with participating devices or clients and allows the clients to train on their local model with enhanced learning quality. Another advantage of FL is that it enables access to diverse datasets without data sharing with reduced data communication and storage requirements. Edge computing as we know is growing to get processing closer to the user device, and the attributes of FL are suitable for such an environment. Based on the data distribution FL is classified into Horizontal FL (HFL), Vertical FL (VFL) and Federated Transfer Learning (FTL) [16].

- *Horizontal FL:* The database has same feature space but different sample spaces, the clients can use the same AI or ML model to train data locally.
- *Vertical FL:* The database has different feature space but the sample space is same, enable shared AI or ML model training.
- *Federated Transfer Learning:* The database has different feature space and different sample space, enables to train data aggregated from multiple clients.

FL applications in an edge computing environment include computation offloading and content caching, Malware and Anomaly Detection, Task scheduling, and resource allocation. Some of the challenges that FL at Edge poses are communication and computation efficiency, heterogeneity management, privacy and security preservation, client selection, and resource allocation [17].

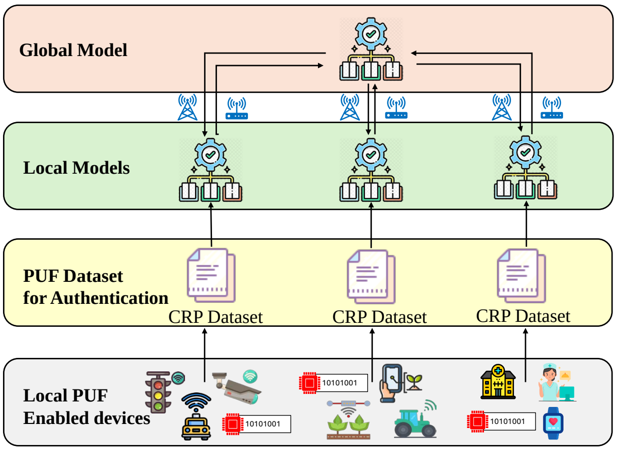


Fig. 2: Federated Learning Framework for Bit Error Correction in PUF enabled Authentication System.

An FL framework for edge that can be used for PUF bit error correction is shown in Figure 2. The data-generating devices are at the edge which are using PUF modules for authentication and authorization, in a CEC the EDCs can be replaced as the local devices. the local devices generate local data and save it locally as their dataset, each participant can train their local model on their data and send parameters to the global model, the global model aggregates the clients, updates the global model, and sends the updated model to local devices for updation.

V. Proposed Federated Learning Framework

The proposed FL framework is shown in Figure 3. The client devices are the participants that use PUF for device authentication and authorization. In the use case for this research, we consider EDCs as the participants in a collaborative environment. Each PUF module generates its own PUF responses for a given set of challenges, each module will have its own CRP dataset represented in the figure as Dataset A, Dataset B, and so on. Each dataset is trained using a local model using K-mer sequencing and Count Vectorization, for classification we are using MultinomialNB. The local ML model is responsible for generating the vectors for the extracted features from the PUF response and classifying them. The local model classifies the responses into unique classes based on which it is trained to predict the class of any new response. Flower Framework is used to implement the federated client-server model [18].

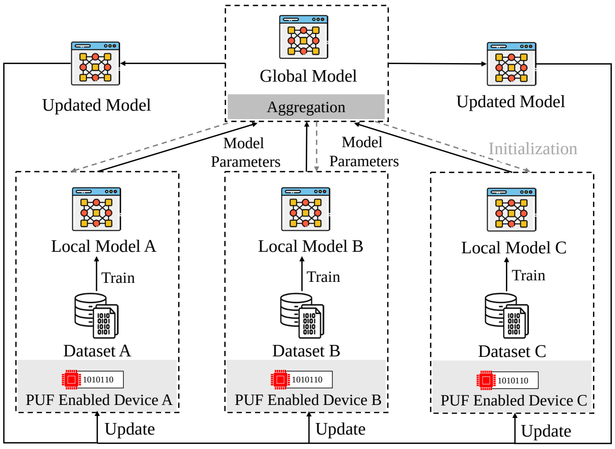


Fig. 3: Proposed Federated Learning Framework for Machine Learning based PUF bit Error Correction System.

The local model is further modified to detect responses with errors, predict the correct class, and correct the response as shown in Figure 4.

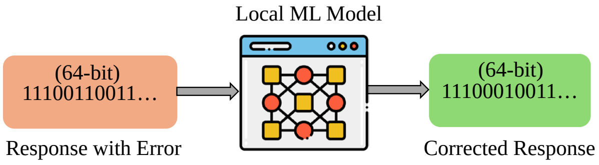


Fig. 4: Local Machine Learning Model for PUF Bit Error Correction.

Whenever a new challenge is used in any one of the clients, and a new response is generated the local parameters of the client are sent to the global model and the trained global model will send updates to all the local models. In this way, the other clients will be updated to correctly predict the class of the new response if the same challenge is given to them. The workflow of the proposed FL framework is as follows:

- *Initialization:* The Server initializes the global model hosted in CPU
- *Client Selection:* The Server can select to train on all available clients of a subset of the clients, maximum number of clients available in this research is 10.
- *Model Distribution:* The participating clients receive a global model update and the same is made available for other clients.
- *Local Training:* The local ML model is trained using the CRP dataset locally available for each participating client. The model is trained over certain epochs to improve performance and KFold cross-validation is done.
- *Model Aggregation:* The updated models from clients are sent to the server after local training.
- *Model Averaging:* The central server aggregates the model by averaging the parameters received from the clients.

The operation of the Client-Server model of the FL system is split into two parts, local training using the ML algorithm and Federated averaging using the FL framework for model updation. The local training steps are shown in Algorithm 1. Each participating client is trained on a local 10K response dataset.

Algorithm 1: Local Model Training

```
Input: 64-bit Binary Response Dataset stored in CSV
       file
Result: Trained model and predictions
1. Read CSV File;
2. Convert Binary data to string;
3. Label the data;
4. Apply K-mers of size 6;
5. Use CountVectorizer() for feature extraction;
6. Split data into train and test set;
7. Classify using MultinomialNB();
8. Predict;
```

The process of FL is discussed as the client-side and server-side processes. The Algorithm 2 shows the steps involved in the server-side model.

Algorithm 2: Server Side Evaulation

```
Input: Number of Clients, Model Parameters
Result: Aggregation and Averaging
1. Set the Number of clients;
2. Start flower server;
3. Request initial parameters from random client;
4. if received parameters then
     Evaluate initial global parameters;
     Evaluate loss and accuracy;
     Start fit;
end
5. update the global model;
6. Send updated global model to all clients;
```

The process involved in the client-side model training and evaluation is shown in Algorithm 3.

The process of local training, sending parameters to the server, aggregation, and new model updates are repeated over several counts until the model converges and performs well across all the clients.

```
Algorithm 3: Client Side Evaluation
  Input: Response dataset CSV file
  Result: Updated Model
  1. Load data;
  2. Preprocess data for client_$n$;
  3. Train Local model;
  4. if Trained then
      Start flower client;
      Send model parameters to server;
      Wait;
  end
  5. if received updated model from server then
      Start fitting;
      Evaluate model;
      End model update;
  end
```

VI. Experimental Setup

The FL framework proposed in this research is Horizontal Federated Learning (HFL), where all clients train a global FL model using their local dataset. The feature space of each dataset is the same, but the sample space is different. Flower provides the infrastructure to perform FL in an easy, scalable, and secure way, allowing federation, analytics, and evaluation of any ML framework.

A 64-bit Arbiter PUF architecture using PUFs. PYNQ™ Z2 FPGA which is based on Xilinx Zynq C7Z020 SoC, Xilinx BASYS3 FPGA was used to build the PUF. A 100K dataset of PUF responses is generated from this Arbiter PUF.

Each participating client is individually trained on 10K unique dataset of responses. The graph in Figure 5, shows the size of data consumed by each of the 10 clients. To suit the HFL model, the feature space for local models is the same, that is 506 features, but the dataset for each client is different. That is, each local model is trained with a different set of responses.

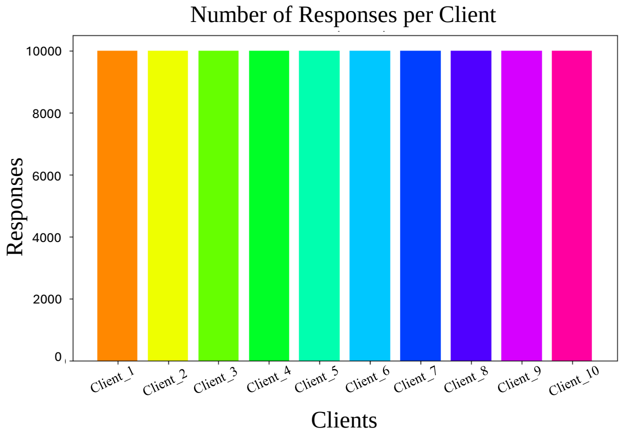


Fig. 5: Data Partitioning of 100K dataset.

To emulate the edge computing environment, Raspberry Pi 4 is used to act as the 10 clients and the server is the CPU, with a 64-bit Operating System, Intel i7 processor, 16GB RAM, and 2.80 GHz.

10K dataset is pre-processed, the binary data is converted to a string, and K-mers of size 6 are applied to convert the data to sequences, the classifier groups the sequences into unique classes. The classification of the sequences for the 10K dataset for Client_1 is shown in Figure 6. The sequences are vectorized using the CountVectorizer class from the scikit-learn library in Python, the vector-matrix columns represent the unique n-gram.

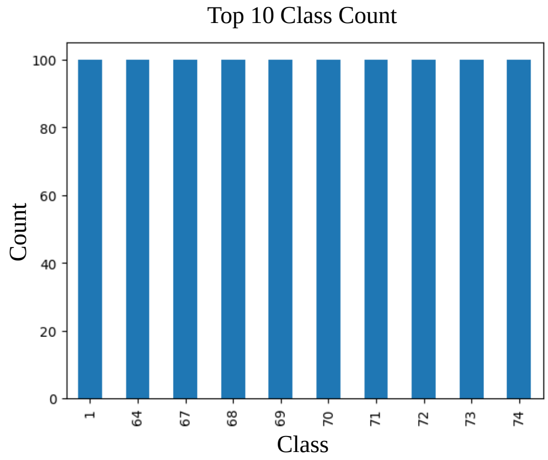
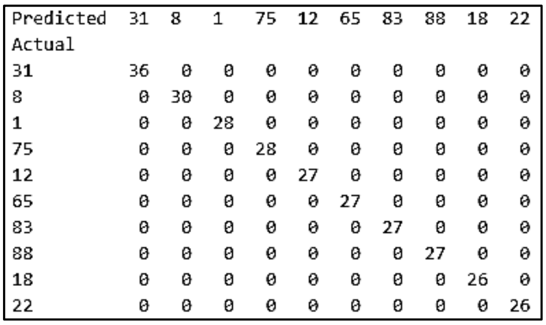


Fig. 6: Classification of sequences into classes.

MultinomialNB classifier is used for classification and the local model is fitted. The confusion matrix showing the actual and predicted class is shown in Figure 7.



```
Predicted  31   8    1   75   12   65   83   88   18   22
Actual
31         36   0    0    0    0    0    0    0    0    0
8           0  30    0    0    0    0    0    0    0    0
1           0   0   28    0    0    0    0    0    0    0
75          0   0    0   28    0    0    0    0    0    0
12          0   0    0    0   27    0    0    0    0    0
65          0   0    0    0    0   27    0    0    0    0
83          0   0    0    0    0    0   27    0    0    0
88          0   0    0    0    0    0    0   27    0    0
18          0   0    0    0    0    0    0    0   26    0
22          0   0    0    0    0    0    0    0    0   26
```

Fig. 7: Confusion matrix.

VII. Results and Analysis

The local model is trained on a 10K dataset, 80% of the data is used for training and 20% is used for testing. The model can efficiently predict classes of new responses and be ready for client-side evaluation. To test for overfitting of the local model KFold cross-validation is done to study the performance over multiple folds. The accuracies obtained over 5-fold cross-validation are 98.75%, 99.3%, 99.65%, 99.8%, and 99.35%, with a mean accuracy of 99.37%.

TABLE II: Comparative Table of Results for State-of-the-Art Literature.

Research	Year	ML Algorithm	Accuracy(%)
Karim et. al. [10]	2023	RainForest	0.99
Jain et. al. [11]	2023	SGD	0.94
Korkmaz et. al. [12]	2022	Inception-v3	0.8-0.99
Chen et. al. [13]	2020	GRU (gated Recurrent Unit) and SVM	0.99
Mahadik et. al. [14]	2024	CNN	0.99
Current Research Fortified-Edge 5.0	2024	K-mer Sequence	0.99

After training the local model the client and server are set up for FL using the Flower FL system.

The server-side evaluation results show a total of 3 server rounds are repeated in fitting the model parameters from 10 clients with 0 failures. The total time taken by the server for fitting the global model is 154.62. The server evaluation is increased for 10 rounds, time taken to complete is 202.32s.

The client-side evaluation shows an average of 99.45% accuracy with 0.0 loss for all 10 clients. The total time taken for local model training with initial parameter update is 6s, total time taken for model update over 10 rounds is 130.42s.

The idle power of the Raspberry Pi was an average of 3.7W, and the average power consumed for local model training was 4.5W.

A comparison of the results from various research listed in the comparative table for state-of-the-art Literature is shown in Table II. The accuracy of the baseline models used in an FL framework is high and is used by a variety of IoT applications handling different datasets.

VIII. CONCLUSIONS

FL framework is easy, scalable, and secure and enables the use of any ML algorithms for local model training. The use of FL for PUF bit error correction has shown enhanced performance and prediction accuracy while providing data privacy and security. In a collaborative environment authentication system using PUFs can benefit from this technique where the CRP dataset need not be stored locally. The accuracy and power consumption evaluations also prove that the model is suitable for edge deployment. The model can be further improved with secure ML model development strategies and the research can be taken forward to explore applications like Deepfake detection, Secure Communication, and Secure Authentication protocols with minimum data exposure.

REFERENCES

[1] P. Arroba, R. Buyya, R. Cárdenas, J. L. Risco-Martín, and J. M. Moya, "Sustainable edge computing: Challenges and future directions," *Software: Practice and Experience*, vol. n/a, no. n/a, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.3340

[2] S. G. Aarella, S. P. Mohanty, E. Kougianos, and D. Puthal, "PUF-based Authentication Scheme for Edge Data Centers in Collaborative Edge Computing," in *IEEE International Symposium on Smart Electronic Systems (iSES)*, 2022, pp. 433–438.

[3] S. G. Aarella, V. P. Yanambaka, S. P. Mohanty, and E. Kougianos, "Fortified-Edge 4.0: A ML-Based Error Correction Framework for Secure Authentication in Collaborative Edge Computing," in *Proceedings of the Great Lakes Symposium on VLSI*, ser. GLSVLSI '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 639–644. [Online]. Available: https://doi.org/10.1145/3649476.3660384

[4] S. Singh, R. Sulthana, T. Shewale, V. Chamola, A. Benslimane, and B. Sikdar, "Machine-Learning-Assisted Security and Privacy Provisioning for Edge Computing: A Survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 236–260, 2022.

[5] Q. Duan, J. Huang, S. Hu, R. Deng, Z. Lu, and S. Yu, "Combining Federated Learning and Edge Computing Toward Ubiquitous Intelligence in 6G Network: Challenges, Recent Advances, and Future Directions," *IEEE Communications Surveys and Tutorials*, vol. 25, no. 4, pp. 2892–2950, 2023.

[6] S. G. Aarella, S. P. Mohanty, E. Kougianos, and D. Puthal, "Fortified-Edge: Secure PUF Certificate Authentication Mechanism for Edge Data Centers in Collaborative Edge Computing," in *Proceedings of the Great Lakes Symposium on VLSI*, ser. GLSVLSI '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 249–254. [Online]. Available: https://doi.org/10.1145/3583781.3590249

[7] A. Ali-pour, F. Afghah, D. Hely, V. Beroulle, and G. Di Natale, "Secure PUF-based Authentication and Key Exchange Protocol using Machine Learning," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2022, pp. 386–389.

[8] R. Cao, N. Mei, and Q. Lian, "Method for Improving the Reliability of SRAM-Based PUF Using Convolution Operation," *Electronics*, vol. 11, no. 21, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/21/3493

[9] P. K. Quan, M. Kundroo, and T. Kim, "Experimental Evaluation and Analysis of Federated Learning in Edge Computing Environments," *IEEE Access*, vol. 11, pp. 33 628–33 639, 2023.

[10] M. Karim, N. K. Kundu, D. Saha, S. Kabir, S. A. Mim, and D. Md. Farid, "Implementing Federated Learning based on RainForest Model," in *IEEE 8th International Conference for Convergence in Technology (I2CT)*, 2023, pp. 1–6.

[11] S. Jain and K. R. Jerripothula, "Federated Learning for Commercial Image Sources," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 6523–6532.

[12] A. Korkmaz, A. Alhonainy, and P. Rao, "An Evaluation of Federated Learning Techniques for Secure and Privacy-Preserving Machine Learning on Medical Datasets," in *IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2022, pp. 1–7.

[13] 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. 217 463–217 472, 2020.

[14] S. S. Mahadik, P. M. Pawar, and R. Muthalagu, "Edge-Federated Learning-Based Intelligent Intrusion Detection System for Heterogeneous Internet of Things," *IEEE Access*, vol. 12, pp. 81 736–81 757, 2024.

[15] P. K. Sadhu, V. P. Yanambaka, S. P. Mohanty, and E. Kougianos, "Easysec: Puf-based rapid and robust authentication framework for the internet of vehicles," 2022.

[16] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. Vincent Poor, "Federated Learning for Internet of Things: A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[17] H. G. Abreha, M. Hayajneh, and M. A. Serhani, "Federated learning in edge computing: A systematic survey," *Sensors*, vol. 22, no. 2, p. 450, Jan 2022.

[18] "Flower: A Friendly Federated Learning Framework," https://flower.ai/, accessed: 2024-07-31.