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

Presenter: Seema G. Aarella

Seema G. Aarella[1], Venkata P. Yanambaka[2], Saraju P.Mohanty[3], Elias Kougianos[4]

University of North Texas, Denton, TX 76203, USA.[1,3,4]

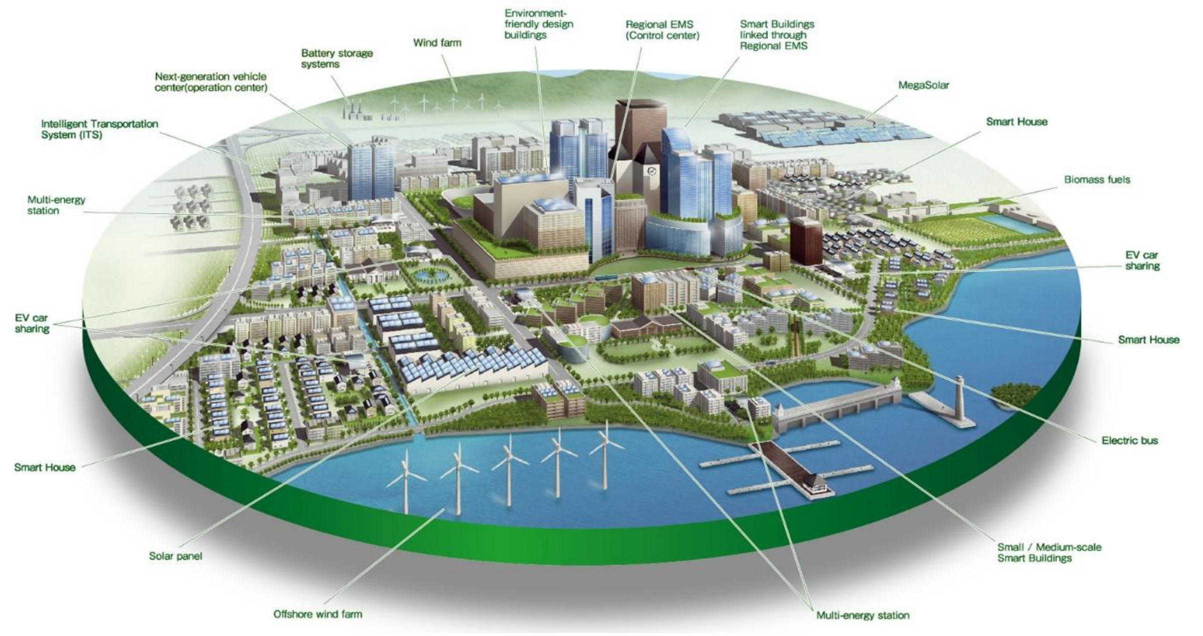
Texas Woman's University, Denton, TX, USA.[2]

Email: Seema.Aarella@unt.edu[1], vyanambaka@twu.edu[2], Saraju.Mohanty@unt.edu[3] and Elias.Kougianos@unt.edu[4]

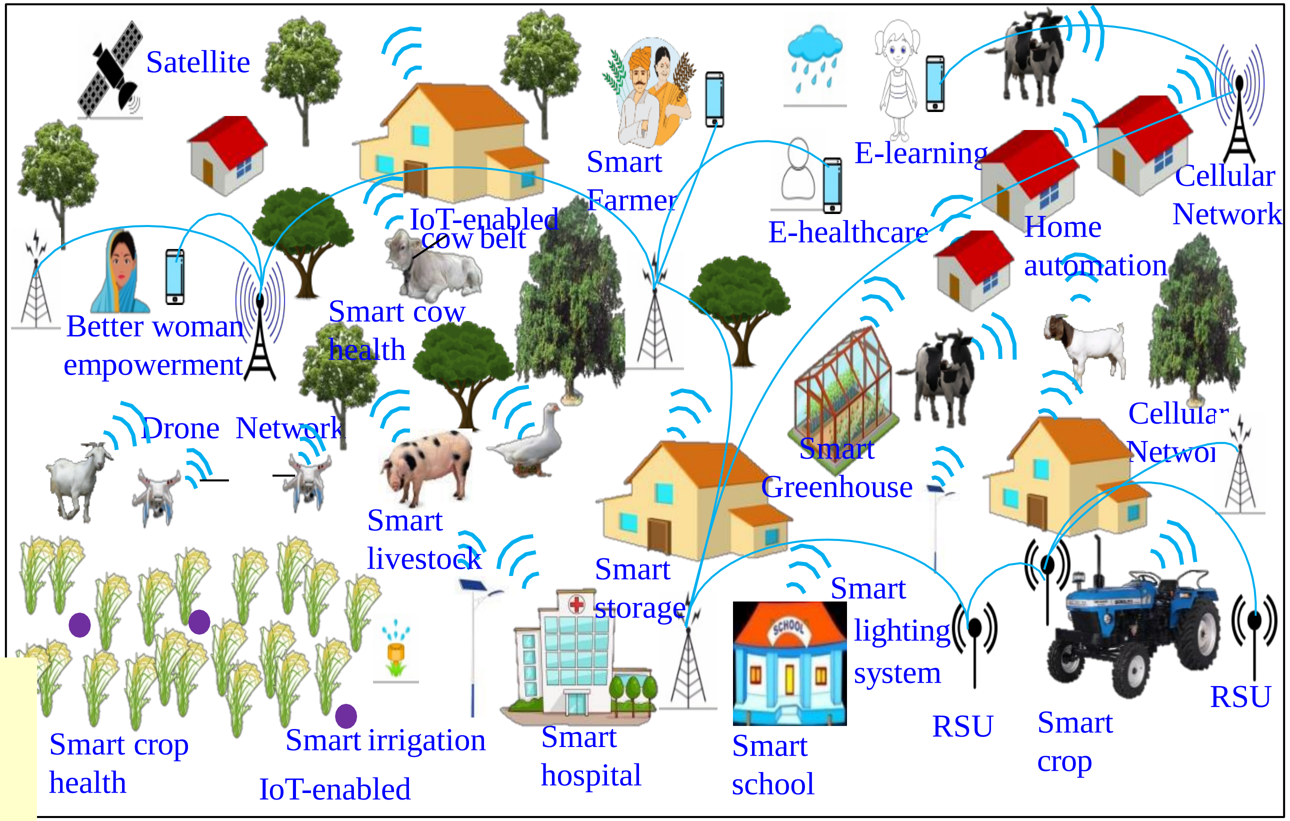
Outline of the Talk

- Introduction
- Collaborative Edge Computing
- Security Challenges and Motivation
- Fortified Edge Concept
- Federated Learning Framework
- Experimental Setup
- Results and Analysis
- Conclusion

Smart Cities Vs Smart Villages



Source: http://edwingarcia.info/2014/04/26/principal/



Source; P. Chanak and I. Banerjee, "Internet of Things-enabled Smart Villages: Recent Advances and Challenges," *IEEE Consumer Electronics Magazine*, DOI: 10.1109/MCE.2020.3013244.

Smart Villages

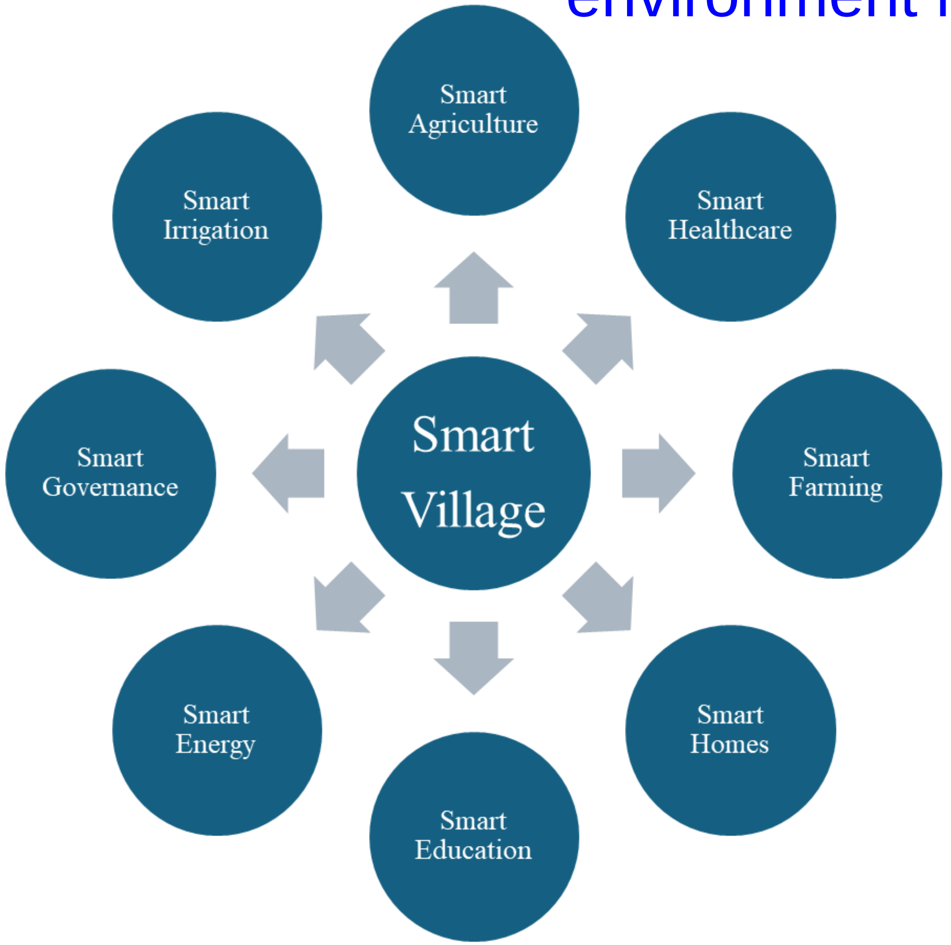
- CPS Types - Less
- Design Cost - Low
- Operation Cost – Low
- Energy Requirement - Low

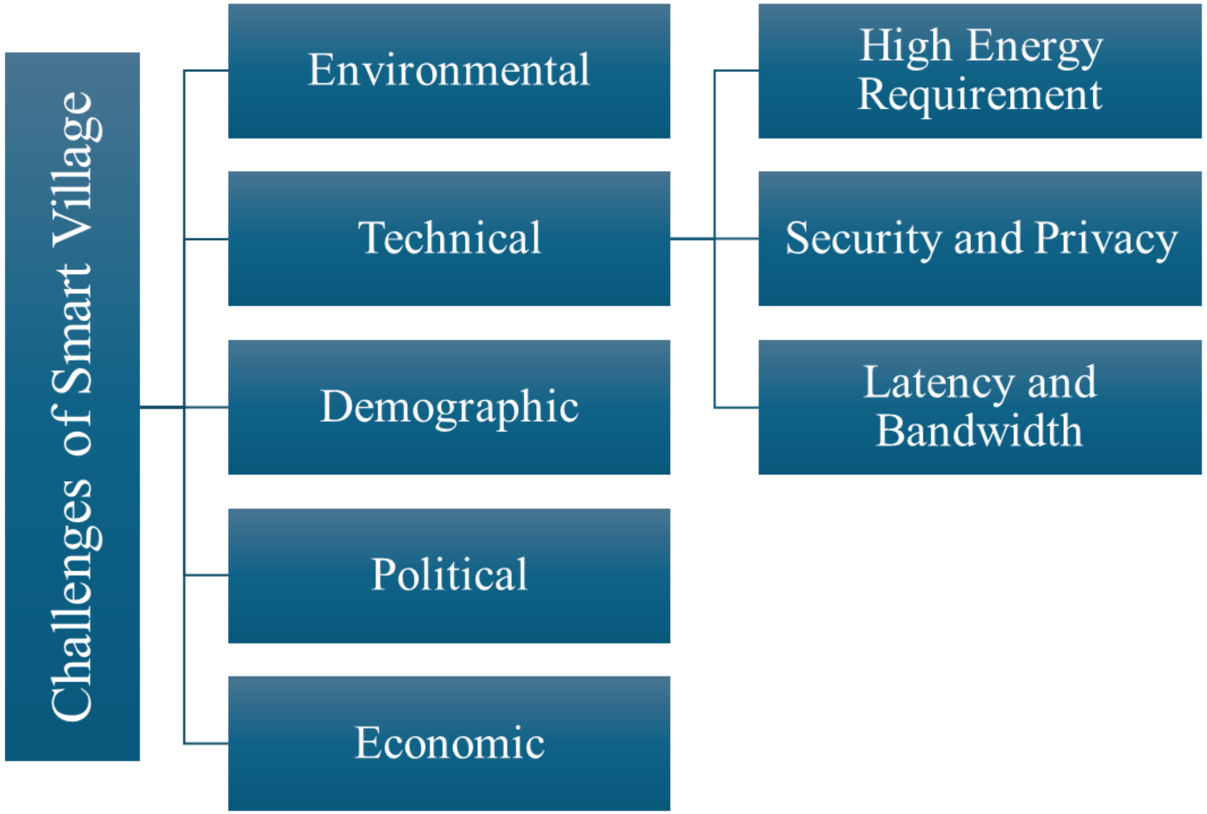
Smart Cities

- CPS Types - More
- Design Cost - High
- Operation Cost – High
- Energy Requirement - High

Introduction

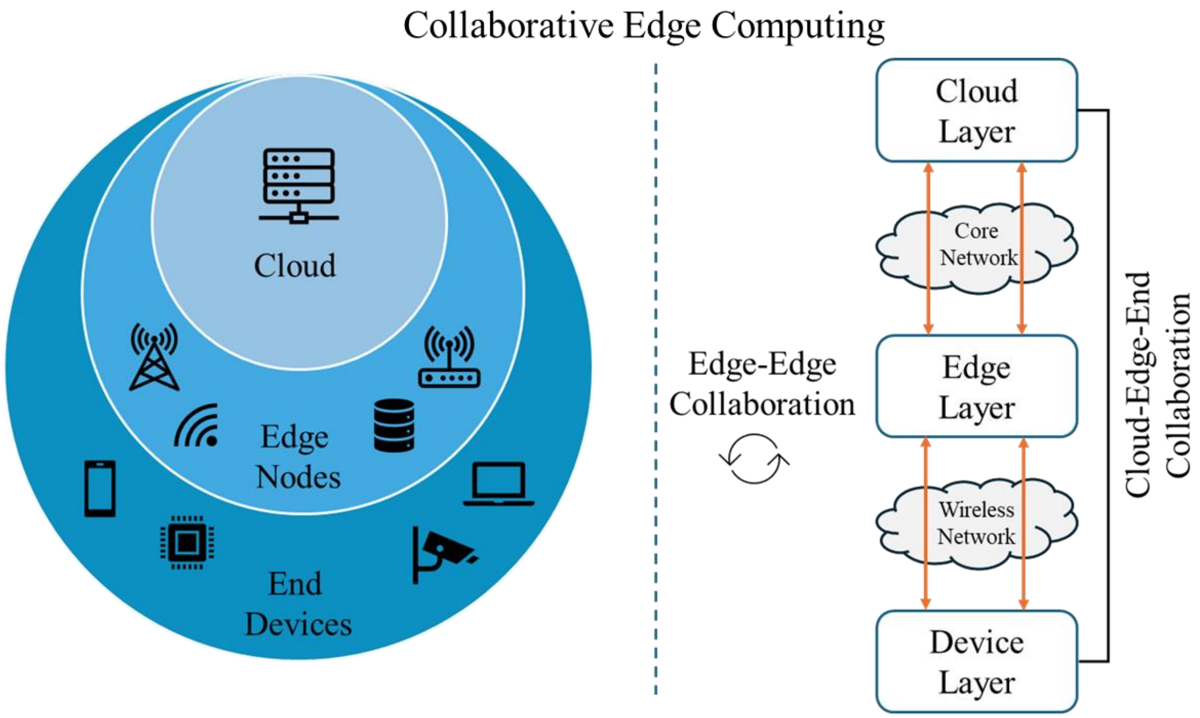
A Smart Village unlike a Smart City is a resource-constrained environment involving several challenges





Collaborative Edge Computing (CEC)

Collaborative Edge Computing enables IoT in smart villages



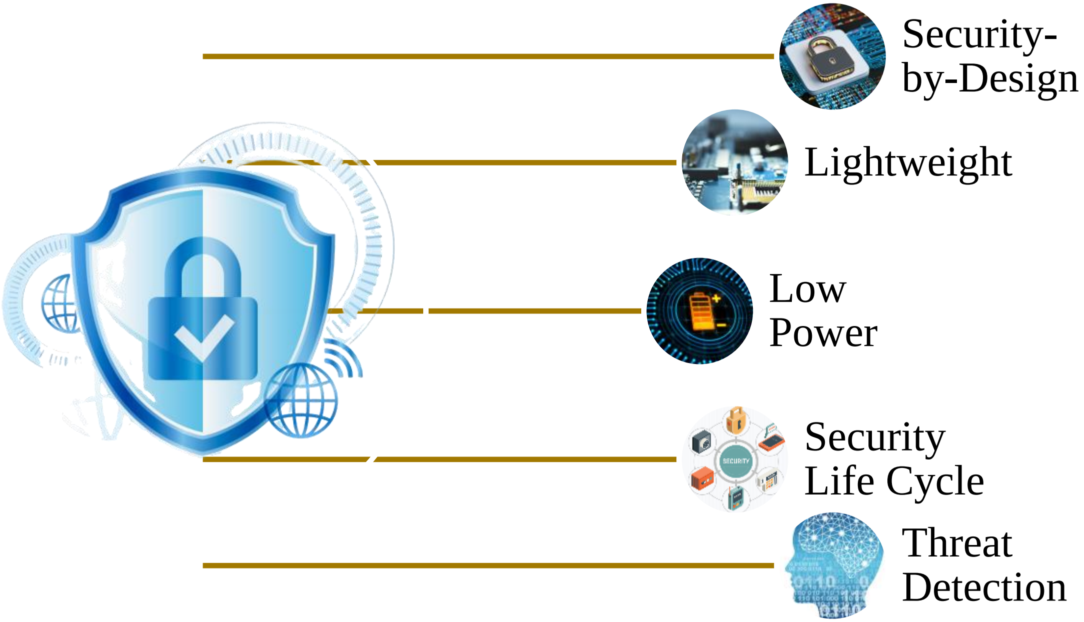
Collaborative Edge Computing

- Distributed processing environment
- Collaboration of distributed edge
- Smart control of heterogeneous network
- Reduced Bandwidth and Transmission costs
- Seamless processing through load balancing

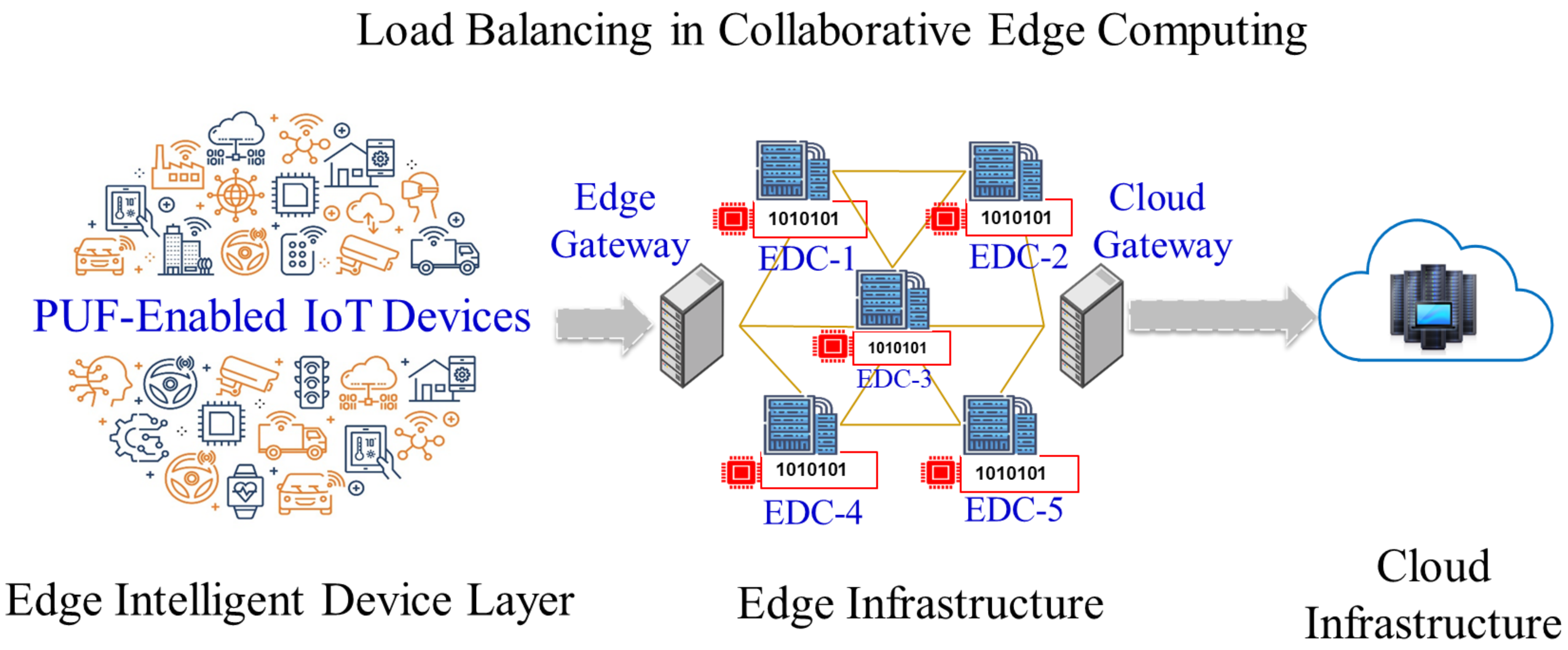
Motivation

To develop a secure authentication system for Edge Data Center in a collaborative environment

- Challenges of CEC
  - Heterogeneity of Devices
  - Scalability and Reliability
  - Resource Constraints
    - Load Balancing
    - Task offloading and Scheduling
  - Security and Privacy
    - Secure Authentication and Authorization
  - Network Latency
  - Energy Efficiency
  - Fault Tolerance and Resilience

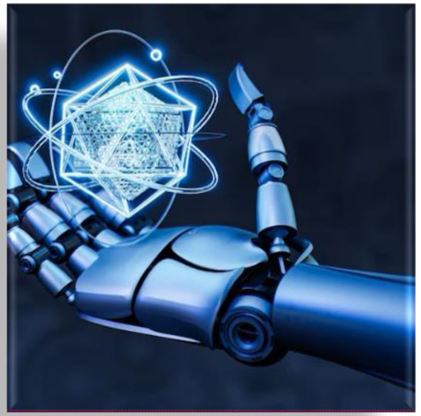


Secure Authentication of EDC in CEC



Long-Term Vision

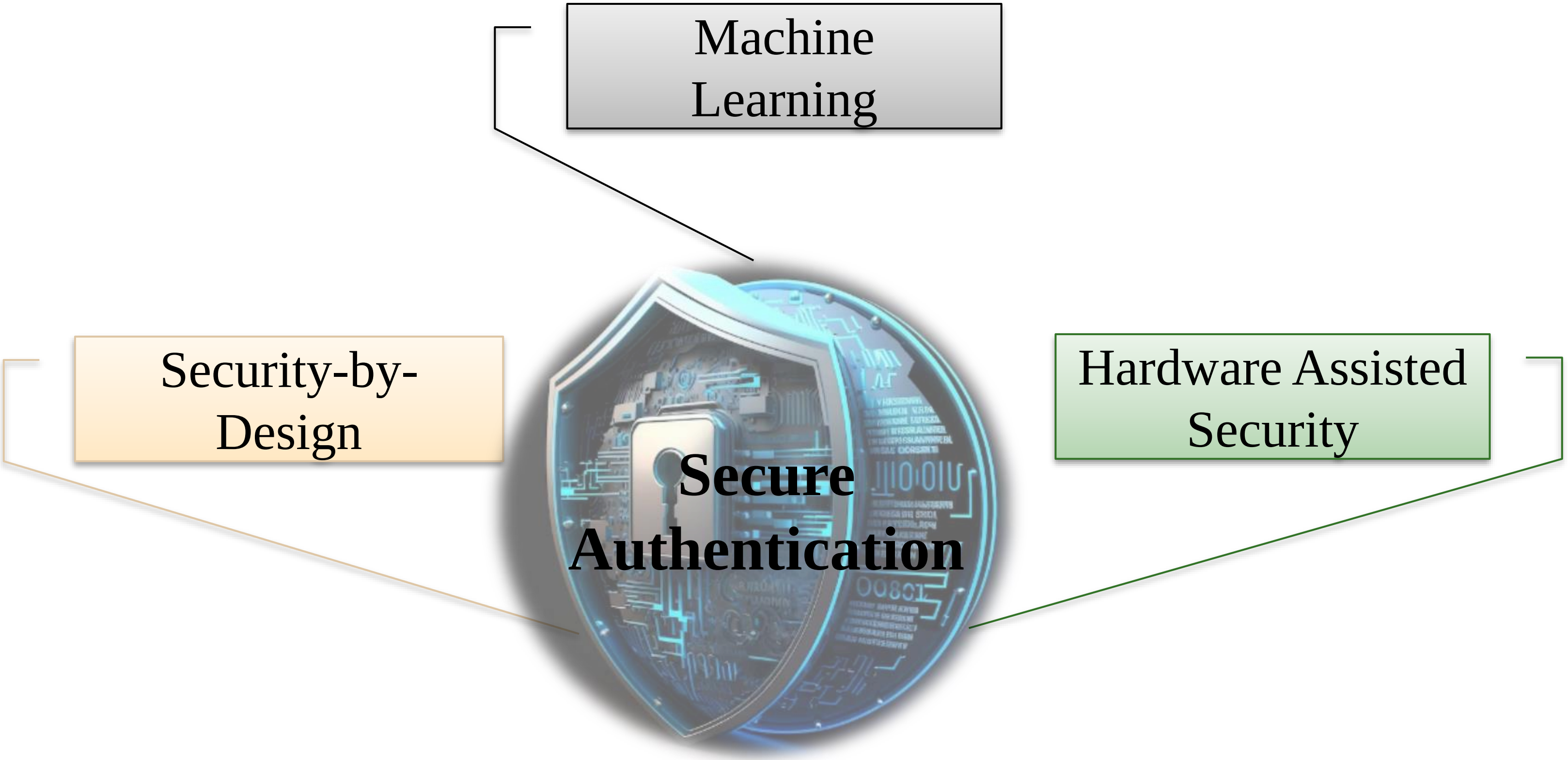
Cybersecurity for smart villages based on the SbD principles for secure resource sharing in the CEC environment



AI/ML for Cybersecurity in Smart Villages

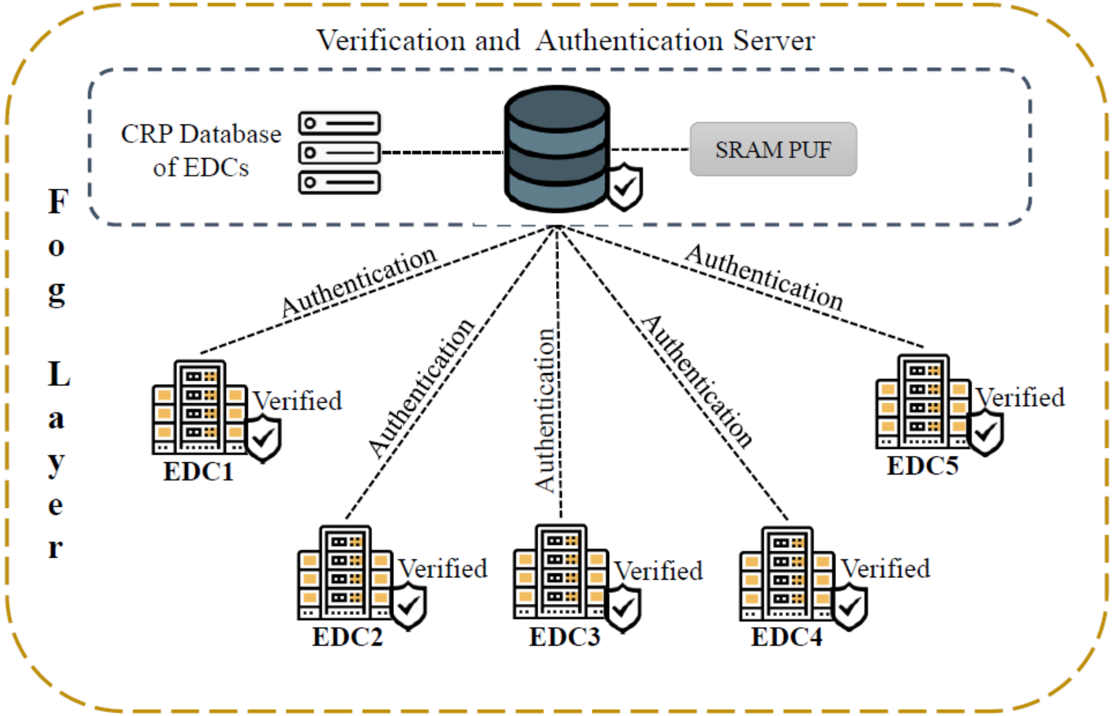
Our Fortified-Edge: The Key Idea

- A lightweight and Secure Authentication scheme for EDCs during load balancing in the CEC environment of smart villages



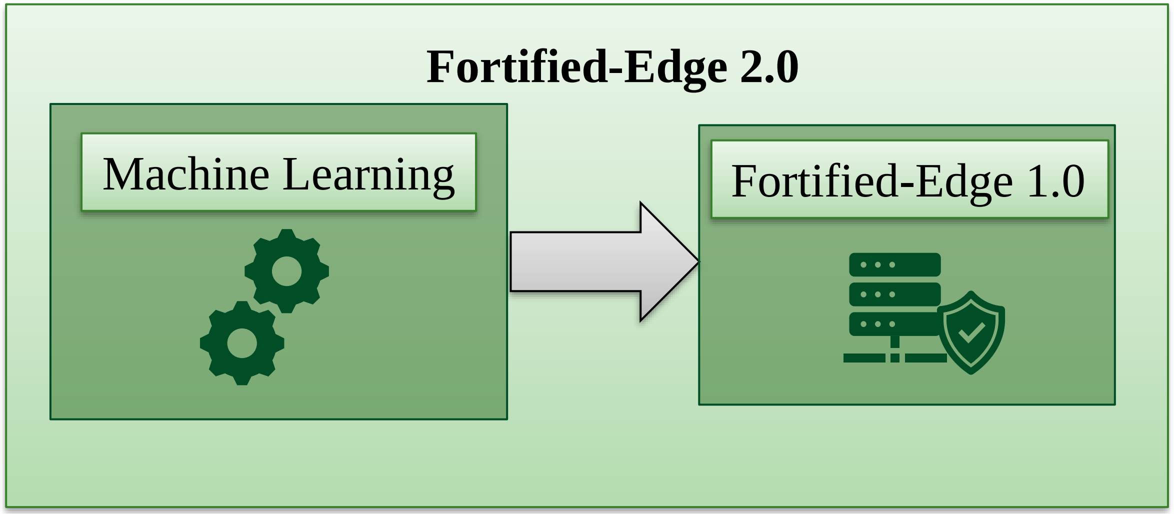
Fortified-Edge 1.0 - The Idea

- CEC enables applications in smart villages through load balancing
- To develop a secure authentication protocol for Load balancing
- Suitable for a smart village environment
- Incorporate Security-by-Design for smart and sustainable security



Source: 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 ACM Great Lakes Symposium on VLSI (GLSVLSI)*, 2023, pp. 249--254, DOI: https://doi.org/10.1145/3583781.3590249

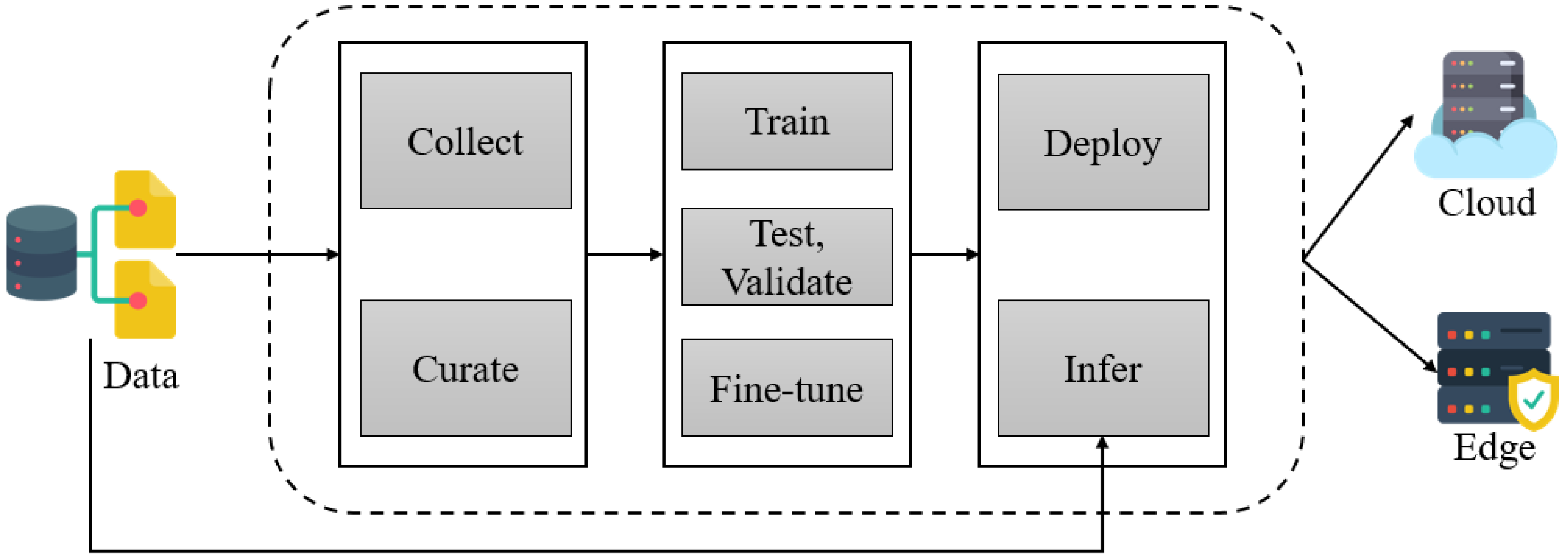
Fortified-Edge 2.0 - The Idea



Features

- Secure, Low Latency Authentication
- Device identification
- Intrusion detection
- Attack Prevention
- EDC Monitoring
- Resilient against malicious Requests
- ML model suitable for a smaller dataset

Fortified Edge 3.0 Machine Learning for Edge



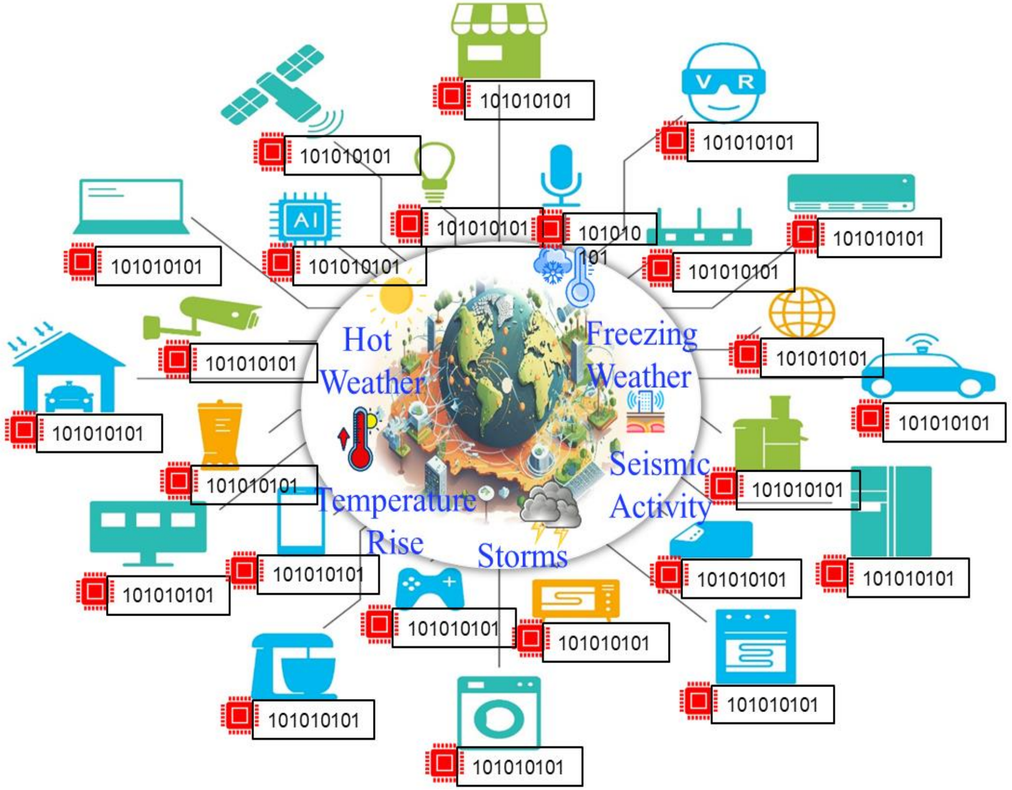
Develop
- Security System

Train
- ML Models

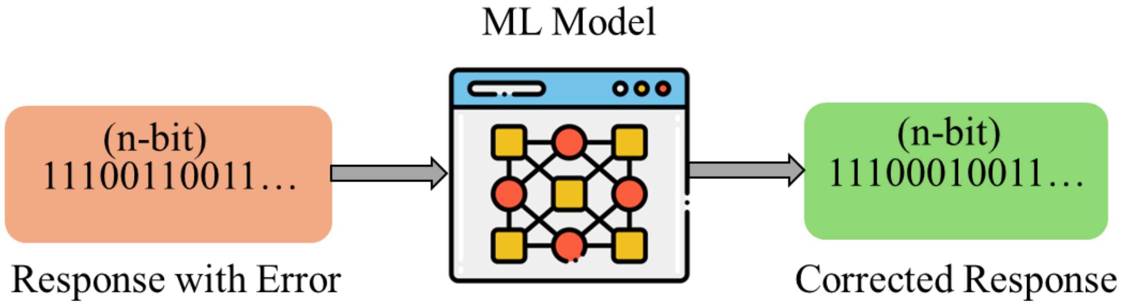
Deploy
- Cloud or Edge

Fortified- Edge 4.0

Environmental factors cause bit flips in the PUF response



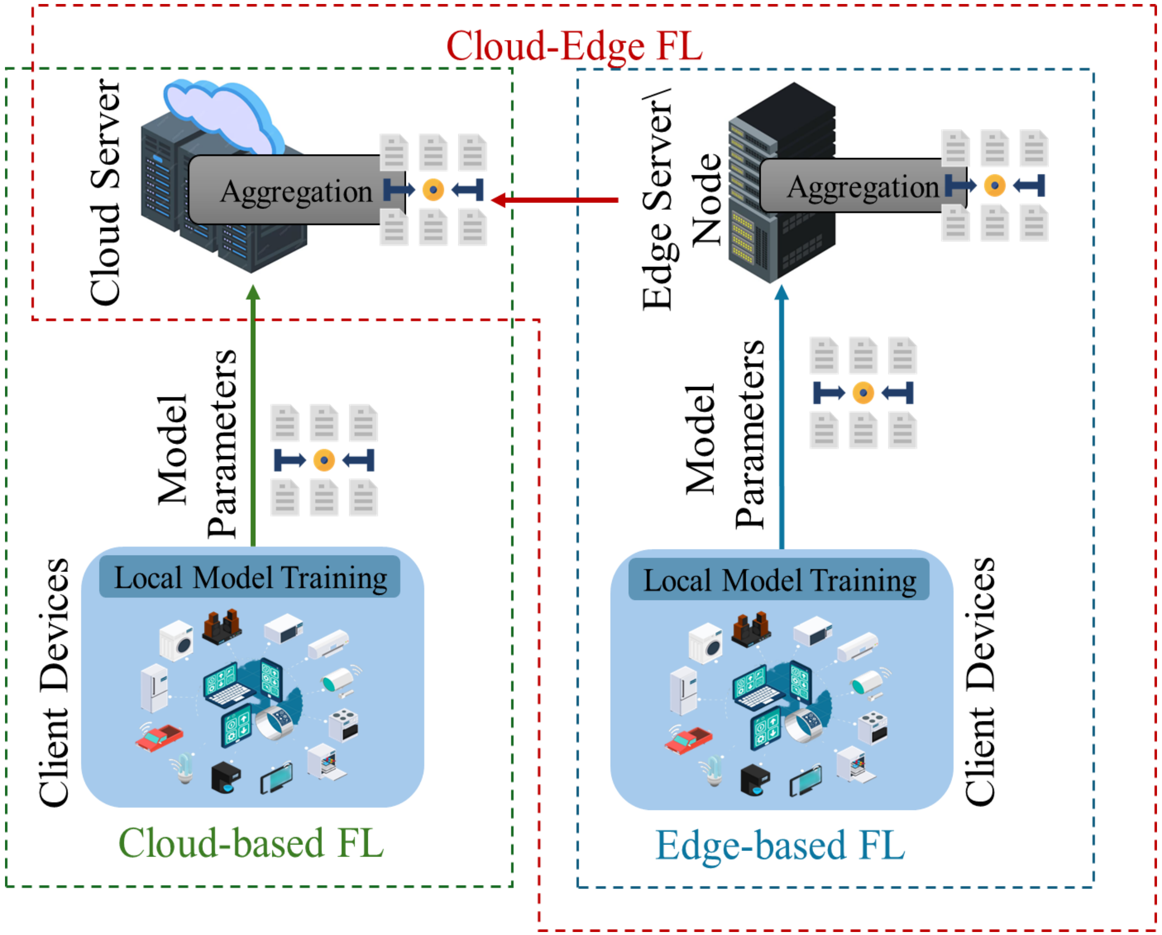
Machine Learning model to detect and correct the flipped bits



Fortified-Edge Research

Research	Algorithm	Application	Accuracy
Fortified-Edge 1.0	SRAM PUF-based Certificate	EDC Authentication	NA
Fortified-Edge 2.0	SVM	ML-based Authentication & Monitoring	100.0
Fortified-Edge 3.0	Lightweight ML models	Anomaly & Intrusion detection	99.33
Fortified-Edge 4.0	K-mer Sequence	PUF Response Bit Error Correction	99.74
Current Research Fortified-Edge 5.0	**Federated Learning**	**PUF Response Bit Error Correction**	**99.00**

Fortified-Edge 5.0 Motivation



- Improve reliability of PUF
- Bit error correction using Machine learning
- Federated Learning (FL) framework for distributed ecosystem
- The key aspects of FL are decentralized training, privacy preservation, and collaborative learning

Related Prior Research

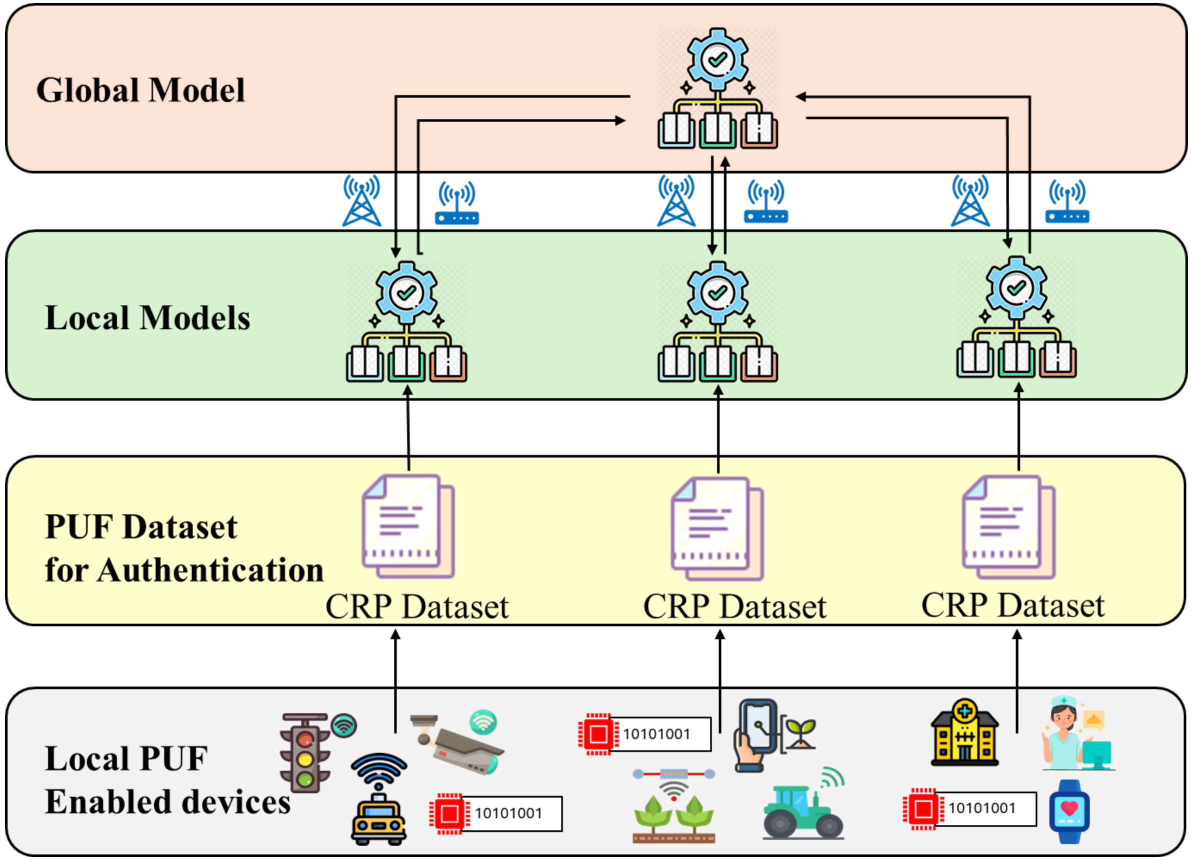
Research	Year	ML Algorithm	Dataset	Metrics
Karim et. al [10]	2023	RainForest	WEKA-Hypothyroid	Accuracy, Precision Recall, F1 Score
Jain et. al. [11]	2023	SGD	Adobe Stock	Accuracy
Korkmaz et. al. [12]	2022	Inception-v3	Medical Image Dataset	Accuracy
Chen et. al. [13]	2020	GRU and SVM	KDD CUP99	Accuracy, F1 Score
Mahadik et. al. [14]	2024	CNN	CIDDoS2019	Accuracy
Current Research Fortified-Edge 5.0	2024	K-mer Sequence	100k PUF Response Dataset	Accuracy

Novel Contributions of Current Research

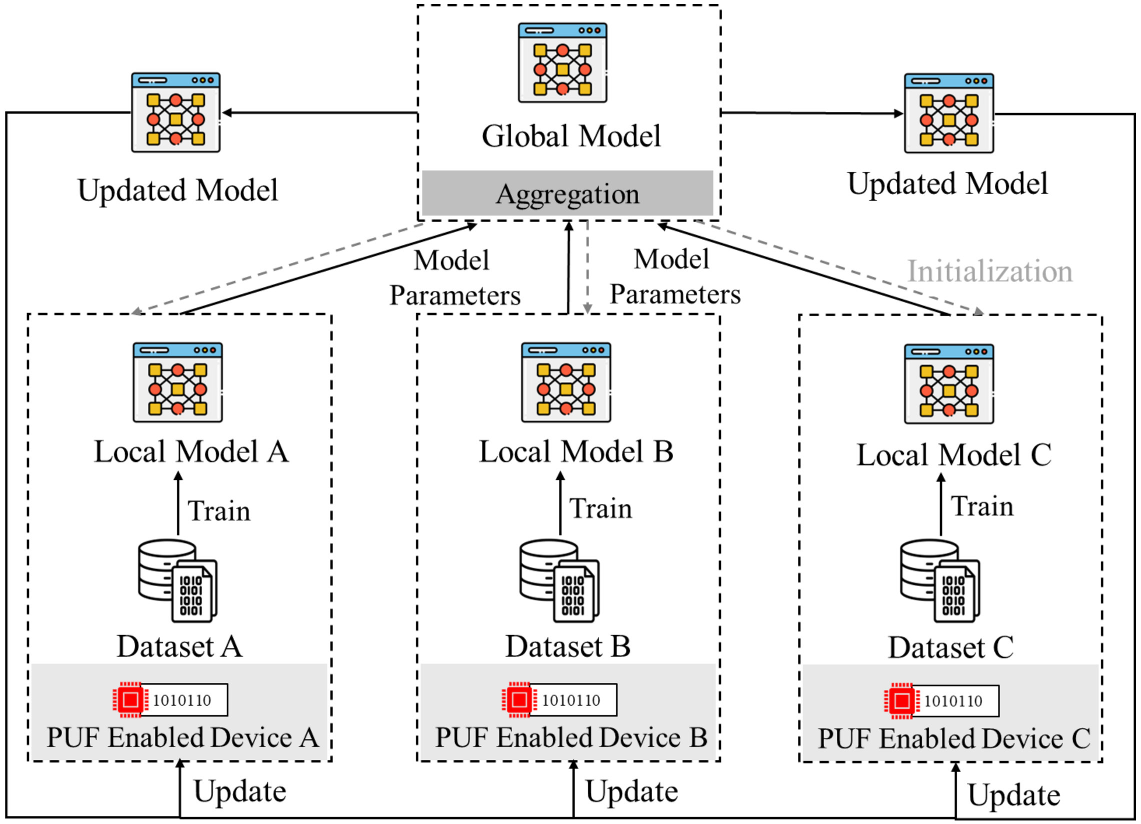
- Exploring FL for edge computing in a collaborative environment
- FL model training and deployment that is efficient in computation and power consumption
- Proposing an FL-based framework for PUF bit error detection that uses an ML algorithm
- ML model training using a Natural Language Processing (NLP)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

FL at Edge for PUF-based Authentication

FL enables access to diverse datasets without data sharing and with reduced data communication and storage requirements



Proposed FL Framework



- Each Client is trained using a local model ML model
- The local ML model is responsible for generating the vectors for the extracted features from the PUF response and classifying
- K-mer sequencing and Count Vectorization for feature generation
- MultinomialNB is used for Classification
- Flower Framework to implement the federated client-server model
- Federated Averaging (FedAvg) is used for model aggregation

Process Flow

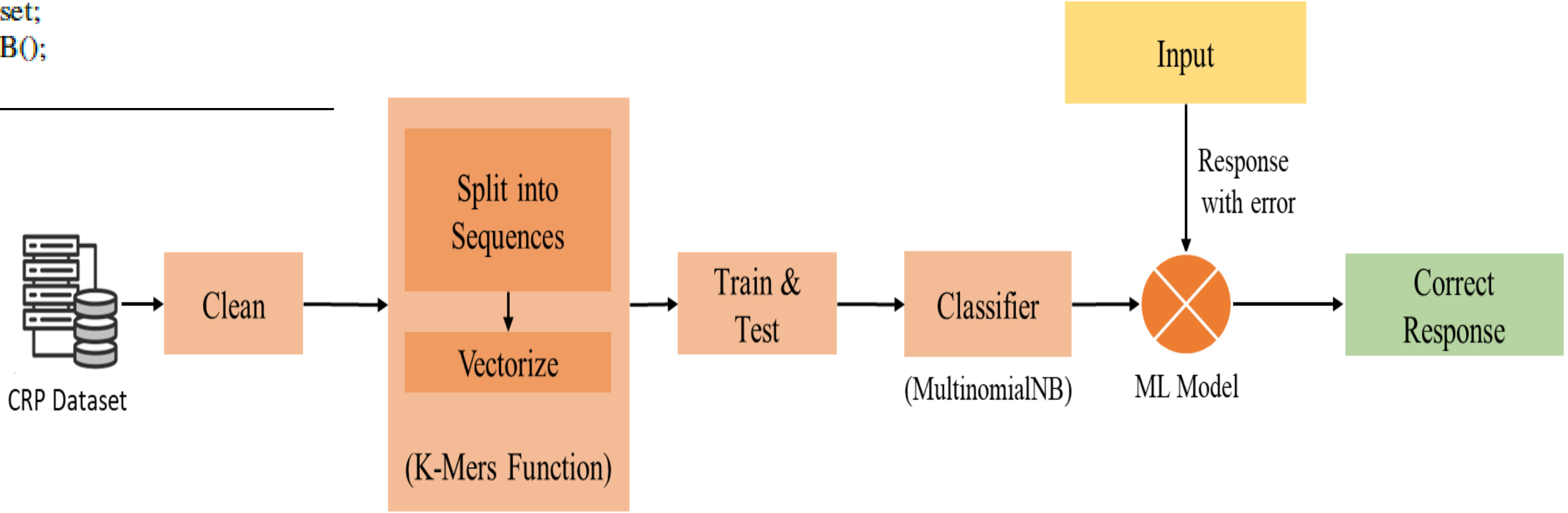
Algorithm 1: Local Model Training

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;

Local Model Training



Process Flow

Server Side Evaluation

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;

Client Side Evaluation

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;

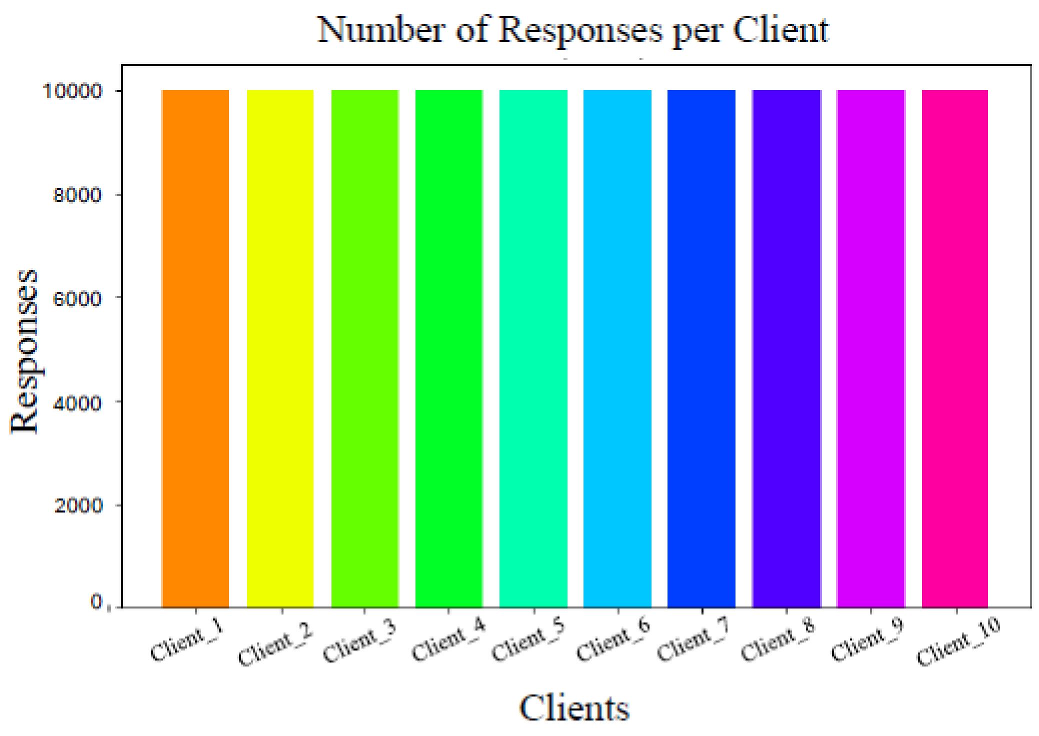
   end

Experimental setup

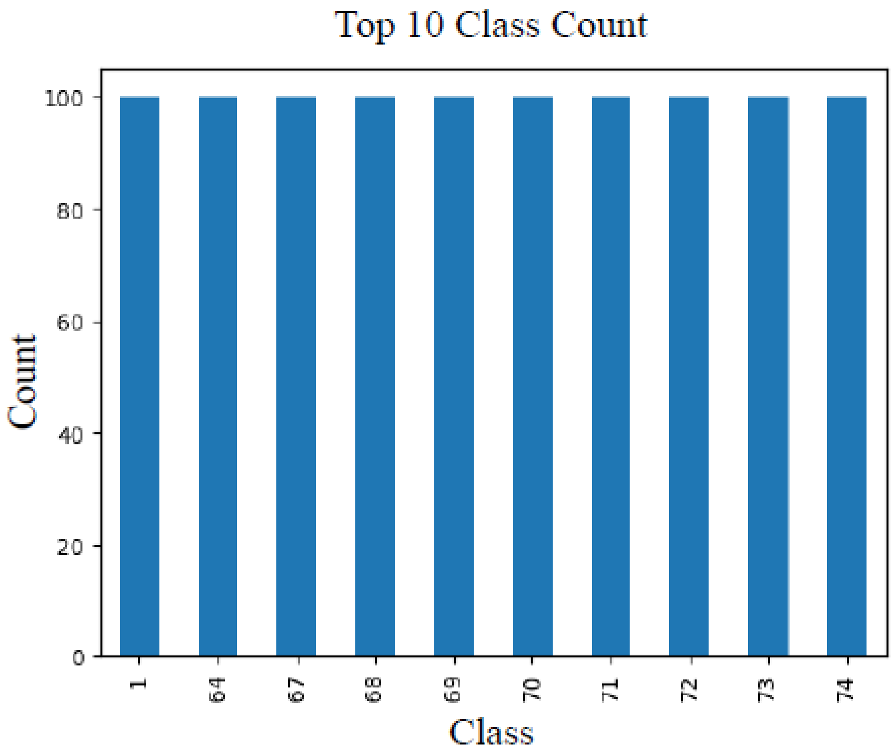
- This research uses the 64-bit Arbiter PUF architecture
- PUFs. PYNQ™ Z2 FPGA which is based on Xilinx Zynq C7Z020 SoC used for PUF implementation
- Xilinx BASYS3 FPGA used to build PUF
- Raspberry Pi 4 is used as the client (10) and Server(1)
- Flower AI framework for FL model training and testing
- Each client is trained on a 10K PUF response dataset

Results and Analysis

10K PUF Response Dataset



Classification of Data



Confusion Matrix

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

- MultinomialNB classifier is used for classification
- The binary sequences are classified based on features
- The feature space for the local model is 506
- Each of the 10 clients are trained on a 10K dataset

FL Model Analysis

- The local model is trained on a 10K dataset
- 80% of the data is used for training and 20% 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
- 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%.

Training Times & Power Consumption

- **The server-side evaluation:**
  - 3 server rounds are repeated in fitting the model parameters from 10 clients with 0 failures
  - The total time taken by the server to fit the global model is 154.62s
  - The server evaluation is increased for 10 rounds, the time taken to complete is 202.32s
- **The client-side evaluation:**
  - 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 = 3.7W,
- Average power consumed for local model training = 4.5W

Comparative Table 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 and SVM	0.99
Mahadik et. al. [14]	2024	CNN	0.99
Current Research Fortified-Edge 5.0	2024	K-mer Sequence	0.99

Conclusion

- 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
- Highly suitable for collaborative environment authentication system
- The CRP dataset need not be stored locally
- The accuracy and power consumption evaluations also prove that the model is suitable for edge deployment

Future Research

- The model can be further improved with secure ML model development strategies to preserve security and privacy
- The research can be taken forward to explore applications like Deepfake Detection and Data Forensics
- Secure Communication and Authentication with minimum data exposure
- Smart and sustainable security solutions

Thank you!