susFL: Federated Learning-based Monitoring for Sustainable, Attack-Resistant Smart Farms

Dian Chen*, Paul Yang*, Dong Sam Ha†, Jin-Hee Cho*
* Department of Computer Science, Virginia Tech, USA
†Department of Electrical and Computer Engineering, Virginia Tech, USA

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10825116

***Abstract*—We propose a sustainable federated learning (FL)-based monitoring system, namely `susFL`, for smart animal farms to address the challenge of inconsistent health monitoring due to fluctuating energy levels of solar sensors. This system equips animals, such as cattle, with solar sensors with computational capabilities, including Raspberry Pis, to train a local deep-learning model on health data. These sensors periodically update Long Range (LoRa) gateways, forming a wireless sensor network (WSN) to detect diseases like mastitis. Our proposed `susFL` system incorporates a game-theoretic approach, called *mechanism design*, to select intelligent clients to optimize monitoring quality while minimizing energy use. This strategy ensures the system's sustainability and resilience against various adversarial attacks, including data poisoning and privacy threats, that could disrupt FL operations. Our work in smart farm technologies sets a new standard by developing an animal monitoring system that is both energy-adaptive and resistant to attacks. Through extensive experiments, we demonstrate that our FL-based monitoring system significantly outperforms existing methods in prediction accuracy, operational efficiency, system reliability (i.e., mean time between failures or MTBF), and social welfare maximization by the mechanism designer. Our experimental results show that `susFL` significantly outperforms the state-of-the-art counterparts, including a 10% reduction in energy consumption, a 15% increase in social welfare, and a 34% rise in Mean Time Between Failures (MTBF) while maintaining the global model's prediction accuracy.**

***Index Terms*—Smart farm, energy-aware, federated learning, deep learning, solar sensors, sustainability.**

I. Introduction

In modern agriculture, solar sensor-based smart farm technologies have revolutionized how farm production is monitored and managed. By harnessing the power of these technologies, farms can achieve higher productivity and efficiency [1]. Integrating solar energy with sensor technology supports sustainable agricultural practices. In addition, it ensures continuous collection of a large volume of data and real-time monitoring, enhancing operational effectiveness and efficiency [2]. Despite such advantages, deploying solar sensors in smart farming raises significant challenges, particularly regarding energy consumption. Energy efficiency becomes paramount as these sensors must operate autonomously over extended periods. Therefore, developing energy-efficient approaches while maintaining continuous monitoring capabilities is essential for the sustainability of smart farming solutions. Moreover, as the scale and sophistication of smart farming systems increase, the concerns surrounding the large volume of data, and their security and privacy (e.g., farm operations, employee information, and financial data) have been raised. The consequences of failing to protect this data adequately are profound, ranging from economic losses due to operational disruptions to severe breaches of privacy in smart farm systems [3]. However, implementing robust security measures often comes at a high cost, presenting a substantial challenge for sustainable smart farming operations.

In response to these challenges, our work develops a monitoring system resistant to cyber and adversarial attacks as well as energy-efficient. **The aim of this work** is to provide a reliable framework that upholds both the operational integrity and the privacy of the data to build an attack-resistant, sustainable monitoring system for smart farm environments.

Federated learning (FL) is a suitable approach to address these multifaceted challenges. Unlike traditional centralized learning models, FL enables data to be processed locally at the sensor level, drastically reducing the amount of data that needs to be transmitted and thereby conserving energy. Additionally, by decentralizing the data processing, FL inherently enhances data security and privacy, as sensitive information is not required to be sent over the network. This methodology not only aligns with the energy efficiency goals but also fortifies the system against potential data breaches and cyber-attacks, making it an ideal choice for our sustainable and attack-resistant smart farm monitoring system. Our proposed approach is named `susFL`, representing a sustainable FL system in the presence of cyber and adversarial threats.

Our work made the following **key contributions**:

1) **Sustainable FL with energy-efficient client selection via mechanism design**: We utilize a game-theory-based *mechanism design* strategy to enhance the sustainability of smart farming systems by energy-adaptively selecting clients (i.e., sensor-equipped animals). This method's effectiveness is quantitatively assessed using the reliability metric, called *Mean-Time-Between-Failures (MTBF)* representing the sum of a system's uptime, to build a smart farm that conserves energy while maintaining high operational reliability, addressing gaps in existing energy-efficient solutions [4, 5].
2) **Pioneering FL for disease detection in smart farm animals**: Our work is the first to explore FL-based monitoring systems for smart farms by identifying livestock illnesses. Unlike previous studies [6–9], which did not apply FL for animal disease detection, we leverage comprehensive experiments with data from the Internet of Animal Health

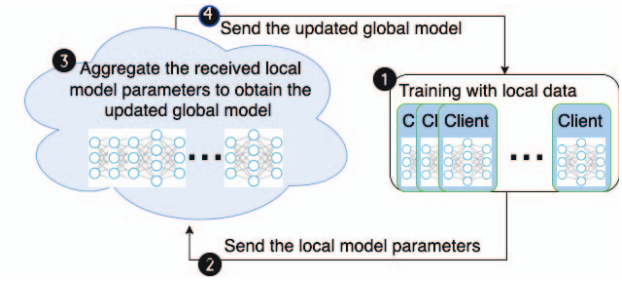


Fig. 1. Key steps in FL training process

Things (IoAHT) [10], focusing on clinical mastitis in cows, providing a solid foundation for validating FL in real-world agricultural settings.

3) **Robust hierarchical FL under adversarial attacks**: We address adversarial attacks in smart farm systems using a hierarchical FL framework to maintain high global prediction accuracy through data quality-aware aggregation. Unlike [11–13], we evaluate the effect of attacks on prediction accuracy and sensor node energy efficiency in resource-constrained environments, using real-world data from Virginia Tech's smart farm.
4) **Experimental Validation of** `susFL`: Our results demonstrate `susFL`'s enhanced performance, achieving a 10% decrease in energy use, 15% boost in social welfare, 34% higher MTBF, and slightly improved prediction accuracy in the global model.

II. Background & Related Work

A. Federated Learning

FL emerges as a cutting-edge machine learning (ML) paradigm that facilitates collaborative model training across multiple data providers, aiming to construct a high-quality, centralized model without compromising data privacy [14]. As illustrated in Fig. 1, the FL framework encompasses a central server hosting the global model and numerous client devices, each maintaining a local model. Within this ecosystem, we consider N distinct data providers, denoted as $\{C_1, \ldots, C_N\}$, each possessing a unique dataset $\{D_1, \ldots, D_N\}$. The training of an ML model M_{FED} under the FL protocol involves the collective effort of all participating data providers. Here, each provider C_i exclusively accesses its dataset D_i to contribute to the global learning process [15].

B. FL-based Smart Farms

Idoje et al. [1] leveraged FL for smart agriculture, focusing on crop classification and time series forecasting, respectively, with [13] exploring its use in agricultural risk management and milk quality prediction. They emphasized FL's benefits but lacked real-world data evaluation and relied on federated averaging. Friha et al. [11] introduced an FL-based intrusion detection system that enhances data privacy in agricultural IoT systems with multiple datasets. Praharaj et al. [12] proposed a hierarchical federated transfer learning framework for cybersecurity in smart farming without experimental validation.

Unlike the above works [1, 11–13], our work pioneers an FL-based system for monitoring animal health in smart farming, a novel application in the sector. It shifts focus from conventional energy-efficient FL methods to robust, secure services suited for solar sensor-equipped farms, filling a significant gap in the current literature.

C. FL-based Monitoring Systems

Sun et al. [16] utilized FL frameworks to enhance IoT security monitoring, with the former identifying anomalies and the latter adapting models within LANs, though both lacked comprehensive performance metrics. Wu et al. [8] introduced *FedHome*, an FL-based health monitoring system using a generative convolutional autoencoder to optimize communication. Khoa et al. [17] developed an autoencoder model to personalize FL applications efficiently. Elayan et al. [6] proposed a deep FL framework specifically for IoT healthcare, focusing on accuracy and privacy across three operational phases. Fan et al. [9] extended FL application to the Internet of Medical Things with the *FLDIoMT* architecture, incorporating data reputation to enhance global model updates and address security concerns.

The above studies [6, 8, 9, 16, 17] leveraged FL to significantly enhance monitoring system security and efficiency. However, achieving comprehensive security and privacy in FL deployments remains significantly less explored particularly in smart farm settings.

III. Problem Statement

We employ FL to accurately predict animal disease risks while prolonging system longevity. We conceptualize this system as a *hierarchical FL structure*, as depicted in Fig. 2. This structure features a global model hosted on a cloud server, with local models operating on LoRa gateways (termed as *edge devices*) and Raspberry Pis mounted on animals (i.e., clients). In this work, models on gateways are designated as *edge models*, and those on clients as *local models*, with the central server running the *global model*.

Our work focuses on the edge level, where gateways execute global models and clients manage local models. Each gateway communicates with a specific set of sensor clients within its communication range. Clients update their models and decide whether to transmit their model updates to the gateway based on their estimated utility. We detailed how to estimate a client's utility in Section V-B. Further, Section IV-A described our network model, depicted in Fig. 2.

Each gateway in our system seeks to enhance the performance of its edge model by aggregating learning parameters from local models of a carefully chosen set of clients. This selection process prioritizes security, energy efficiency, and fairness. Conversely, clients strive to conserve energy to prolong their operational lifespan while supplying essential updates to improve the edge model's effectiveness. The proposed `susFL` aims to achieve the following objectives:

$$\text{maximize} \quad \mathcal{ACC}(M(s^*)), \quad \text{subject to } \mathcal{EC}(s^*) \leq \varepsilon. \qquad (1)$$

In our system, $M(s^*)$ denotes the edge model trained using a selected set of sensor nodes s^* for the FL aggregation process. s^* will be selected by our proposed client selection mechanism in Section V. The $\mathcal{ACC}(M(s^*))$ measures the prediction accuracy of $M(s^*)$, and $\mathcal{EC}(s^*)$ quantifies the energy consumption of these selected sensor nodes. Both accuracy $\mathcal{ACC}(M)$ and the target accuracy threshold ε are normalized within the range $[0, 1]$. Our objective is to maximize $\mathcal{ACC}(M(s^*))$ while ensuring that $\mathcal{EC}(s^*)$ does not exceed ε. This approach underlines our commitment to developing a sustainable smart farm through hierarchical FL, i.e., `susFL`. One significant concern with such an FL framework is the high energy consumption of Raspberry Pi devices when training local models. Therefore, an energy-adaptive `susFL` is proposed to maintain sufficient energy levels of solar-powered clients while achieving high system performance.

The proposed `susFL` incorporates a mechanism design-based client selection for FL aggregation to withstand cyber and adversarial attacks in Section IV-C. We will elaborate on the system's design and components to attain its sustainability in Section V.

IV. System Model

A. Network Model

Our smart farm system employs a network model that integrates solar-powered sensors, wearable Raspberry Pis (R-Pis), Long-Range (LoRa) gateways, and a cloud server, as illustrated in Fig. 2. Each animal, such as a cow, is fitted with solar sensors on their ears to monitor body conditions, with the data transmitted to nearby R-Pis. A selected group of animals equipped with R-Pis act as computational clients, processing their data and that from others to train local models constituting step 1. LoRa gateways, equipped with edge servers, receive these local updates, refine the edge model, and forward the refined model parameters to the cloud server, which then updates the global model in steps 2 and 3. Our approach, `susFL`, focuses on optimizing these processes, particularly for energy-limited devices (i.e., clients). In step 4, the cloud server dispatches the latest model parameters back to the gateways, distributing them to the clients within range in step 5. This configuration capitalizes on LoRa technology to boost IoT connectivity, minimizing costs while extending the range. A deep learning (DL) model, deployed across local servers on client devices, edge servers, and the central server, is designed to assess the risk of mastitis in animals. It outputs a binary classification: 0 indicates a healthy cow, while 1 signifies a cow diagnosed with mastitis.

B. Node Model

In this network, sensors periodically transmit data to nearby clients, enabling the training of local models with freshly sensed data. Given these sensors are solar-powered, their energy levels naturally fluctuate due to various environmental influences, including the animals' locations, weather conditions, and seasonal variations in sunlight exposure. Additionally, designing FL systems faces significant challenges, such as high communication and computational costs, along with the need to ensure data privacy. Consequently, the system needs to be resilient, adapting to the dynamic and energy-variable environment of the smart farm, and fortified against potential adversarial attacks.

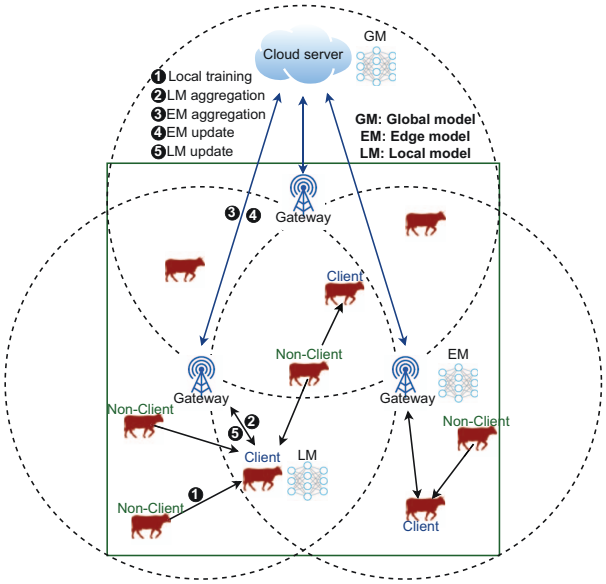


Fig. 2. Hierarchical FL-based network architecture designed for a wireless solar sensor-based smart farm.

The sensors use Bluetooth Low Energy (BLE) to send data to proximal clients, and these clients, in turn, forward their local updates to LoRa gateways. This setup employs the LoRa protocol for long-range communications, effectively covering distances between 5 to 15 km with a data transfer speed of 27 $kbps$. For shorter distances, up to 100 meters, the BLE protocol is used, for a faster transfer speed of 2 $Mbps$.

For energy consumption, the LoRa radio of SAM R34/35 expends about 170 mW during data transmission, while the BLE radio has a lower consumption rate of approximately 11 mW [18]. A Raspberry Pi's power usage is 0.117 W per second when idle, increasing to 0.172 W per second under load [19]. Sensor nodes, once fully charged, hold an initial energy reserve of 5 kW. The charging efficiency for solar-powered sensors varies with light exposure—about 10 mW/cm^2 in outdoor settings and 0.1 mW/cm^2 indoors.

The two sensor node types are as follows:

- *Normal sensor node* [18]: This node lacks the computational resources for local model training, instead periodically sending its data to a Raspberry Pi (R-pi)-based node via BLE.
- *R-pi-based sensor node* [2, 18, 19]: This node gathers data from normal nodes within its range and trains local models. It decides on its participation in the FL aggregation by sending its local model parameters to the edge model, acting as a client within the FL framework.

This model demonstrates our key motivation for developing `susFL` that trains DL models on solar-powered sensors instead of on each gateway. Training on each gateway requires

frequent raw data transmission from sensor nodes, increasing data exposure risk and energy consumption. As described above, normal sensor nodes transmit data to nearby clients (i.e. R-pi-based sensor nodes) via BLE protocol, consuming 15 times less energy than direct LoRa communication. Thus, sensor nodes only send local model parameters to LoRa gateways, reducing data volume and transmission frequency.

C. Threat Model

To understand the vulnerabilities within FL systems, we examine the following types of adversarial attacks: (1) ***Byzantine attacks*** disrupt the FL training process by injecting arbitrary metrics via Stochastic Gradient Descent (SGD) updates [20]. These attacks primarily target local devices or clients, prolonging their learning duration or leading to model divergence. (2) ***Backdoor attacks*** compromise the integrity of edge and global models through malicious clients that submit altered local model updates [21]. The objective of backdoor attackers is to preserve high prediction accuracy during testing to evade detection while causing the model to incorrectly classify a specific target class. (3) ***Collaborative attacks*** involve multiple compromised clients working together to degrade the global model's accuracy [22]. This type of attack affects both the global model on the central server and the edge models on gateways. Attackers may adjust training hyperparameters or alter model weights before these are sent to the edge model. The success rate of backdoor attacks increases with the proportion of attacker-controlled clients, surpassing the effectiveness of conventional data poisoning strategies.

To assess the impact of these attacks, we analyze the attack success probability, denoted as P_A, representing the likelihood of an attacker successfully executing an attack at any given time t. Our primary aim is to create a sustainable FL-based monitoring system ensuring its resilience and reliability of functioning effectively under Byzantine, backdoor, and collaborative attacks. To be clear, our work does not develop specific defenses against these attacks. Instead, we emphasize the selection of trustworthy clients and the secure aggregation of local model parameters to ensure the system's tolerance under threats and robustness against such cyber and adversarial threats. We investigate the system's resilience under such attacks, showing the impact of different attack frequencies on prediction accuracy, as other existing approaches [22–24] have shown in the literature.

V. Proposed Approach: susFL

A. Key Processes of the FL Model Aggregation

In the given network, we consider a group of n clients, denoted by $N = \{1, \ldots, n\}$, each possessing a local model eligible for selection. The cost of including client i's local model in the aggregation process is represented by c_i, a known value publicly. With a total budget constraint of B, we ensure that the selected clients maintain adequate energy reserves after completing a given FL task. To this end, we propose to develop a client selection mechanism, $\mathcal{M}$, to identify an optimal subset of clients. We adopted the game theory called

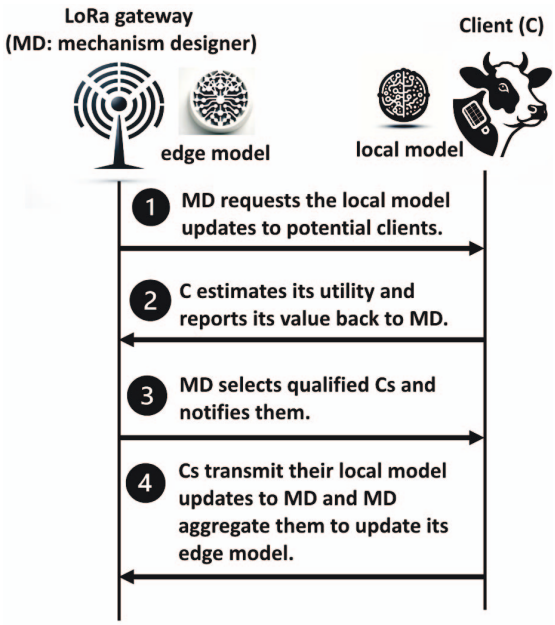


Fig. 3. Key steps for an edge server (i.e., mechanism designer) to aggregate local model updates from selected clients.

mechanism design to employ a centralized entity aiming to optimize multiple objectives in the proposed smart farm system. While centralized systems often suffer from scalability, data security, and resilience, we leverage FL to combine centralized optimization benefits with FL's robust, scalable, secure, and privacy-preserving distributed intelligence. This approach contributes to the FL model aggregation, effectively balancing between achieving the desired accuracy of the edge model and minimizing energy consumption.

Fig. 3 describes the FL model aggregation process in four key steps in our proposed `susFL` system: ❶ *Requesting local model updates*: Each gateway's server requests local updates from clients within its operational area. ❷ *Utility estimation and response*: Clients calculate their utility based on the anticipated energy expenditure for participating in the FL process, as defined in Eq. (2). Clients with utilities equal to or larger than a predefined threshold θ communicate their values, as in Eq. (4), back to the gateway. ❸ *Client selection and notification*: The mechanism designer, operating at the gateway, selects clients for the FL process based on their values and notifies the chosen clients. ❹ *Local update transmission and model aggregation*: Selected clients transmit their local updates to the gateway's edge server, where they are aggregated into an enhanced edge model focused on high-quality data (Section V-D). This model is then used to assess the health conditions of cows on the `susFL`-based smart farm.

After completing each FL cycle, the gateways upload their updated models to the central server, consolidating them into a new global model. This updated global model is then disseminated back to the gateways, ensuring continuous improvement and accuracy of the system's predictive capabilities.

B. Clients' Utility Estimation

Upon receiving a local model update request from its gateway, a client assesses the task's utility to decide on its participation in the aggregation process. The utility for client i upon receiving this request from gateway j at time step t, $u_t(i,j)$, is calculated by:

$$u_t(i,j) = e_t(i) - ec_t(i,j). \tag{2}$$

The $e_t(i)$ refers to the current energy level of client i, and $ec_t(i,j)$ is the expected energy consumption of client i when participating in the FL aggregation process with gateway j, where both are normalized to the range $[0,1]$ as real numbers. The expected energy consumption, $ec_t(i,j)$, is determined by:

$$ec_t(i,j) = \underbrace{|D_i^t| \times r \times E_{\mathrm{R-pis}}}_{\text{for training}} + \underbrace{\mathrm{Dis}(p_i^t, p_j^t) \times E_T}_{\text{for transmission}}. \tag{3}$$

The expected energy consumption, $ec_t(i,j)$, is calculated based on $|D_i^t|$, the volume of data client i uses for local training at time step t, and r, the rate at which training time extends per additional data sample. It also considers $E_{\mathrm{R-pis}}$, the energy required to process the specified workload per second. The energy model for training the models is measured by the energy consumed by raspberry-pi to train the model in the training time estimated by the amount of data sample because the volume of data is the main factor influencing the computational cost of model training [25]. The distance between client i and gateway j during the FL process at time step t, denoted as $\mathrm{Dis}(p_i^t, p_j^t)$, and E_T, the energy for transmitting local updates via LoRa, are also factored into the calculation. Our energy model for transmission, as defined in Eq. (3) above, aligns with the standard energy consumption metrics for sensor nodes widely referenced in existing literature and hardware manual [18, 26].

The utility function enables client i to evaluate the benefit of participating in the aggregation process at time step t. If participating would result in the client's energy being entirely depleted from transmitting data to the edge model, leading to a utility of $u_t(i,j) \leq 0$, the client will opt out of the aggregation. This decision adheres to the *individual rationality* property within mechanism $\mathcal{M}$, ensuring clients participate only when the utility is positive, thus preserving their normal operation until a more opportune moment arises. If the client's utility is overestimated, causing its energy to be fully depleted, the client will be excluded from the selection process until its utility becomes positive again through solar recharging. Upon opting to contribute, client i communicates its value v_i to the edge server, which then assesses client selection. The value v_i reflects the data quality of client i's model update [27] and is formulated by:

$$v_i = \frac{\Psi}{\log(\frac{1}{\varepsilon_i})}. \tag{4}$$

The Ψ is a pre-defined parameter as a coefficient related to the number of local model iterations impacted by local data accuracy. The $\log(\frac{1}{\varepsilon_i})$ denotes the iteration count for a local model update to maintain a constant global accuracy. This assessment is supported by theoretical guarantees based on empirical evaluations rather than hypothetical expectations [27, 28]. Importantly, v_i assumes a negative value if $\Psi > 0$ and $\varepsilon_i > 1$, indicating that the closer v_i is to zero, the higher its perceived value.

To address the challenges of attaining multiple objectives and reducing the solution search space, gateways initially identify clients whose participation in the selection process could ensure a minimum accuracy of ε. Consequently, only clients with a value of $v_i \geq \theta$ qualify as candidates for aggregation. Here, θ represents the threshold to determine the subset of clients eligible for further consideration in the client selection phase.

C. Client Selection Mechanism

Mechanism $\mathcal{M}$ is designed to fulfill specific properties, simplifying our discussion by excluding the notation j. This approach generalizes the utility of client i across any scenario where it receives a request for a local model update to participate in the FL aggregation of gateway j, as depicted in Fig. 3. The mechanism designer (MD) enforces the following properties to achieve desirable outcomes in this strategic setting. For clarity, the client's utility, $u_t(i,j)$, from Eq. (2) is simplified to $u(i)$ in the explanations below.

- *Truthfulness* [29]: $\mathcal{M}$ is truthful if each client i's dominant strategy is to report true information when

$$u(i) \geq u(i^{'}), \tag{5}$$

  where $i^{'}$ is any client's information that $i^{'} \neq i$. Since the utility function is independent of the information disclosed to the gateway (e.g., data quality), clients have no incentive to misreport their values, leading to truthful reporting.
- *(Weak) Budget balance* [30]: $\mathcal{M}$ is budget-balanced when

$$\sum_{i}^{N} \mathcal{COM}_{\mathcal{E}}(i) \leq B, \tag{6}$$

  where $\mathcal{COM}_{\mathcal{E}}(i)$ represents the communication cost to integrate client i into the aggregation. LoRa gateways, not limited by energy constraints for model training, could theoretically accept clients' local updates immediately. However, to minimize energy consumption on the client side and preserve the operational longevity of sensor nodes, the number of communication rounds should be limited by the global model's perspective while the clients should ensure energy efficiency.
- *Individual rationality* [31]: $\mathcal{M}$ is individual rational if

$$\forall i \quad u(i) \geq 0. \tag{7}$$

  The utility function, $u(i)$, is designed to assess the variance between a client's present energy status and the energy expenditure contributing to aggregation j. Thus, clients will partake in aggregation j only if they can sustain their energy levels post-participation and not deplete their energy by engaging in the FL process with gateway j.

The mechanism designer (MD: LoRa gateway) aims to maximize ***social welfare*** by selecting an optimal set of clients via mechanism $\mathcal{M}$, as

$$\max \sum_{i}^{n} u(i), \tag{8}$$

while meeting Eqs. (5) – (7) to ensure that clients can participate without depleting their energy reserves.

The MD's objective to maximize the system's social welfare is challenging due to the nature of optimizing multiple objectives. This challenge can be modeled as a 0/1 knapsack problem, using a dynamic programming technique [32], guaranteeing an optimal solution. The time complexity for the dynamic programming solution is $O(N \times W)$, where N is the number of clients and W represents the gateway's budget constraints. Dynamic programming resolves complex problems by dividing them into smaller, overlapping subproblems, addressing each once. For each of the W possible budgets, we determine the optimal value for every client, resulting in $N \times W$ subproblems. Each subproblem is solvable in constant time, or $O(1)$.

D. Quality-Aware Parameter Aggregation

Upon receiving local updates from the chosen clients, the gateways proceed to aggregate these updates to train new edge models. To achieve high model accuracy, the strategy employs FedAvg [33], focusing on utilizing high-quality data. This approach involves a weighted parameter aggregation, where the weighting is determined by the reported data quality of each client. The process is formulated by:

$$w_{new} = \frac{\sum_{i}^{k} \frac{v_i}{v_{\max} - v_{\min}} w_i}{k}, \tag{9}$$

where the weighting of each local parameter, w_i, is determined by $\frac{v_i}{v_{\max} - v_{\min}}$, whose range is given within a $[0, 1]$ in the overall distribution, with k representing the total number of participating clients. Aggregation occurs both at the edge server level, involving gateway and client updates, and at the cloud server level, with updates from the edge. This process ensures the data quality values reported by clients to improve the process. Following aggregation, the gateways forward the updated edge model parameters to the central server, where edge updates are aggregated to construct the global model. During training, local devices employ a loss function to assess model performance on their datasets, with gradients informing local updates. We evaluate the effectiveness of our `susFL` system by the global model's performance in Section VII.

VI. Experimental Setup

A. Parameterization

This work leverages clinical mastitis data in cows, captured via IoT sensors on the udder, to detect the disease [10]. The dataset consists of 6,600 entries, with three records per cow, featuring 15 attributes monitored by flex and temperature wireless sensors connected to Raspberry Pis and digital-to-analog converters. This dataset is collected using the Internet of Things (IoT), thus ideally suitable for evaluating the proposed IoT-based smart farm system. Our simulation utilizes semi-synthetic data from Virginia Tech's SmartFarm Innovation Network$^{\text{TM}}$ (College of Agriculture and Life Sciences), incorporating the effects of adversarial attacks (Section IV-C) and the diseases detailed in [10]. The data contains the records of cow movement activities, and adversarial attacks are injected into the data for our attack resilience analysis. Although the experiments are not performed physically on the farm due to hardware constraints, all data is collected and transmitted by solar-powered sensors deployed in their smart farm network to simulate animal behaviors. This network serves as a hub for collecting and analyzing data across Virginia farms, indicating cows typically move at speeds within the range of $[1, 2]$ meters per second. Movement probability for cow i, denoted as P^i_{mv}, is modeled by a normal distribution with an average speed of 1.5 m/s and a standard deviation of 0.1 m/s. Our system, detailed in Table I, uses FL to diagnose animal diseases with sensor data. Serial numbers are omitted from the training datasets to avoid spurious correlations, ensuring that irrelevant data do not affect predictions.

TABLE I
Dataset Description

Metric	Description
Serial	A unique animal identifier
Size-udder	Size of an udder (udder front left, front right, rear left, and rear right) for inhale and exhale limit
Average temperature	Average body temperature in Celsius
Hardness	Hardness of an udder
Pain-level	Pain due to swelling of an udder
Average-activity	Average activity recorded by the number of steps taken
Battery-level	Residual battery life
Timestamp	Date and time of transmission

This work encompasses a farm spanning 40 acres, approximately 160,000 square meters, with each side measuring 400 meters. It focuses on monitoring 30 cows using three gateways to ensure efficient surveillance over the 48-hour simulation period. Each gateway implements an edge model, leveraging our `susFL` mechanism to predict animal diseases, to optimize system performance given the current conditions. We classify 60% of the cows as clients equipped with Raspberry Pi (R-pi)-based sensor nodes, with the remaining serving as standard sensor nodes, as in Section IV-B. Gateways solicit updates from these client nodes at 60-minute intervals, designated as T_u. All sensor nodes start with a random initial energy level, E_{init}, within the range of $[0.3, 0.8)$. Table II summarizes the key design parameters, their meanings, and default values. Initially, we posit that 30% of the sensors, denoted as P_C, are compromised at the system's onset. The model assumes full trust in both gateways and the cloud server, with attackers solely targeting sensor nodes.

B. Metrics

- **Prediction accuracy** measures how accurately the global model predicts animal diseases compared to actual out-

TABLE II
KEY DESIGN PARAMETERS, MEANINGS, & DEFAULT VALUES

Notation	Meaning	Value
n	Total number of sensors(cows)	30
N	Total number of clients	20
P_{mv}^{1}	Probability of cow i to move	[0.3,0.7]
P_A	Probability for an attacker or a compromised node to perform a certain attack (e.g.,)	0.1
P_C	Percentage of compromised clients in sensor network	0.3
T_s	Time interval for a sensor to send sensed data	30 s
T_u	Time interval for a gateway to request local updates	1 hr
T_g	Time interval for a gateway to report edge models to central server	1 hr
E_{init}	Initial energy level of sensors	$[0.3, 0.8)$
ε	Threshold for minimum energy level	0.15
B	Number of communication rounds budget for each gateway	5

comes. Specifically, it evaluates the system's ability to correctly identify mastitis in cows, calculated as the proportion of correct predictions out of the total predictions made during the simulation.

- **Energy consumption** ($\mathcal{EC}$) quantifies the overall energy usage by the system for communications and computations within the FL framework. The $\mathcal{EC}$ is formulated by $\mathcal{EC} = \mathcal{COM}_{\mathcal{E}} + \mathcal{COMP}_{\mathcal{E}}$, where $\mathcal{EC}$, the total energy consumption, comprises two main components: $\mathcal{COM}_{\mathcal{E}}$, the energy used for communications between clients and gateways, and $\mathcal{COMP}_{\mathcal{E}}$, the cumulative energy expended by sensor nodes for system operations, including model training and energy depletion over time. The $\mathcal{COM}_{\mathcal{E}}$ is calculated by: $\mathcal{COM}_E = \sum_{i=0}^{l} \mathcal{E}_{CR}^{i}$, where $\mathcal{E}_{CR}^{i}$ represents the energy consumed in a single communication round between a client and a gateway, and l signifies the total number of communication rounds throughout the simulation. The $\mathcal{COMP}_{\mathcal{E}}$ accounts for the energy utilized in model training and the natural energy drain experienced during the system's operational period. The $\mathcal{COMP}_{\mathcal{E}}$ is given by:

$$\begin{aligned} \mathcal{COMP}_{\mathcal{E}} &= \mathcal{E}_{TC} + \mathcal{E}_{\text{active}} + \mathcal{E}_{\text{sleep}} \\ &= \frac{|S^*| e_{TC}}{E_S} + \frac{T_u}{E_S}(d_{\text{active}} + d_{\text{sleep}}), \end{aligned} \tag{10}$$

where $\mathcal{COMP}_{\mathcal{E}}$ accounts for the energy dynamics of the participating client set $|S^*|$, including e_{TC}, the energy a client consumes to join the aggregation process, and E_S, the full charge energy level of a sensor. It further considers d_{active} and d_{sleep}, the energy depletion rates per second in active and sleep modes, respectively. T_u refers to the interval at which local updates are requested by the edge server. This model focuses solely on the energy consumption of sensor nodes, excluding the energy transactions between gateways and the central server, as they do not face energy limitations.

- **Mean Time Between Failures (MTBF)** [34, 35] quantifies the average duration of system reliability, calculated as:

$$MTBF = \frac{\sum_{f \in F}(u_{s,f} - d_{s,f})}{|F|}, \tag{11}$$

where $d_{s,f}$ marks the commencement of downtime, $u_{s,f}$ the onset of uptime, and F the collection of failure instances. A system is considered to have failed when the average energy level across all nodes falls below ε.

- **Social Welfare** encapsulates the collective utility of all sensor clients, as described in Eq. (8).

C. Comparing Schemes

We compare the proposed `susFL` against the following state-of-the-art (SOTA) FL schemes to evaluate its effectiveness: (1) **FedAvg** [36] employs a variant of Stochastic Gradient Descent (SGD) where clients independently execute SGD and the server averages these models to avoid aggregating the entire dataset. (2) **FedProx** [37] modifies FedAvg to accommodate client heterogeneity by incorporating a proximal term into the local optimization problems, allowing variable local updates and managing statistical heterogeneity. (3) **FLTrust** [38] enhances robustness against data poisoning and backdoor attacks by utilizing trusted execution environments to periodically verify client integrity. (4) **DivFL** [4] improves communication efficiency with a diverse, greedy client selection mechanism for each aggregation round to diversify the gradient space and expedite training. (5) **GreenFL** [5] introduces a green-quantized FL approach that uses stochastic quantization in both local training and data transmission, optimizing energy use and accuracy through precision adjustments in Quantized Neural Networks (QNN).

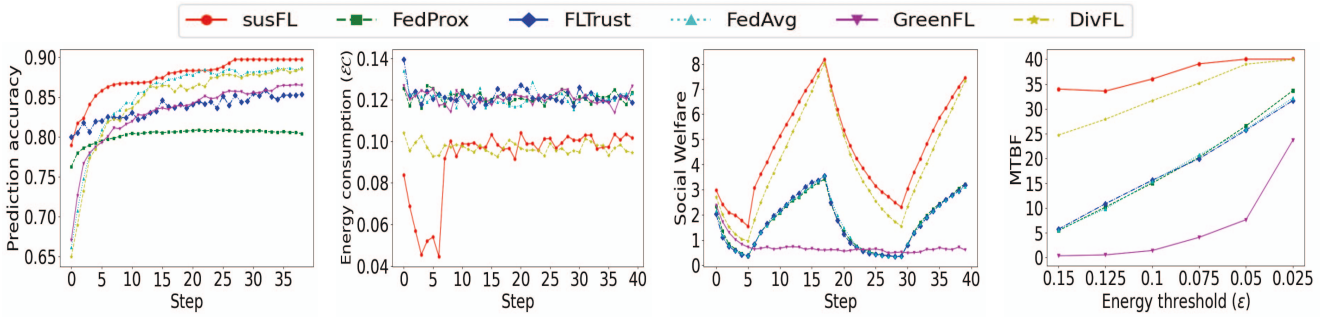
The source code for the implemented `susFL` will be released upon the paper's acceptance.

VII. RESULTS AND ANALYSES

To assess the effectiveness of the proposed `susFL` scheme alongside the five existing FL schemes, we conducted 50 simulation runs using the parameter configurations in Section VI. The results presented for each scheme are the average outcomes derived from these 50 simulations, ensuring valid evaluation and comparison.

A. Comparative Performance Analyses

Fig. 4 showcases the FL training progress across six schemes, as detailed in Section VI-C, including our proposed `susFL` plus the five SOTA schemes. Our `susFL` surpasses the five other schemes in prediction accuracy (Fig. 4(a)), energy consumption (Fig. 4(b)), social welfare (Fig. 4(c)), and MTBF (Fig. 4(d)). This superior performance underscores the efficiency of `susFL`'s quality-aware client selection strategy, which excludes clients with poor-quality data, thereby enhancing prediction accuracy. By allowing clients to assess their utility for participation, `susFL` achieves minimal energy usage throughout the simulation, leading to the highest social welfare and MTBF. `susFL`'s energy consumption is particularly low at the simulation's start, attributed to the initial absence of



(a) Prediction Accuracy (b) Energy Consumption (c) Social Welfare (d) MTBF

Fig. 4. Comparative performance analysis during training time

sunlight for charging sensors, demonstrating a strategic energy conservation for FL operations. The fluctuation in social welfare (Fig. 4(c)) mirrors the solar pattern, highlighting the dynamic nature of the smart farm environment. Among the SOTA schemes, `DivFL` demonstrates the lowest energy usage due to its efficient client selection method that bolsters learning effectiveness. Conversely, `GreenFL` exhibits the highest energy demand, attributed to the added computational load from managing a QNN. These findings emphasize the importance of minimizing computational demands on energy-limited devices in resource-constrained FL systems for smart environments.

B. Sensitivity Analyses

1) **Effect of Varying Attack Severity (P_A):** Fig. 5 illustrates the impact of different levels of attack frequency (P_A), where higher P_A triggers an attack more often, representing higher attack severity, on the performance metrics of FL schemes. An increase in P_A results in lower prediction accuracy due to the inclusion of more compromised sensors, undermining the model's performance. However, attack severity does not introduce a significant impact on energy consumption and MTBF, showing the system's robustness under attacks, because `susFL` considers energy-adaptive operations to tolerate energy drainage.

For `susFL`, increased attack severity leads to a decline in social welfare and MTBF as illustrated in Fig. 5(c) and (d). This decline necessitates greater client involvement in FL tasks to uphold prediction accuracy. Although this situation slightly raises energy consumption (Fig. 5(b)), the increase is marginal when compared to the gap between `susFL` and other SOTA schemes, rendering the consumption curve nearly flat. Overall, `susFL` remains superior to its peers, such as `DivFL` and `FedProx`, in handling varying P_A, showcasing robust performance across metrics.

2) **Effect of Node Density:** Fig. 6 examines how performance metrics respond to varying numbers of client sensor nodes in the network. We observe that `susFL` effectively manages FL operations in large sensor networks. The broader distribution of clients improves prediction accuracy by increasing the likelihood of high-quality data, giving gateways better selection options. This strategic selection reduces average energy consumption (Fig. 6(b)), enhances social welfare (Fig. 6(c)), and improves MTBF (Fig. 6(d)).

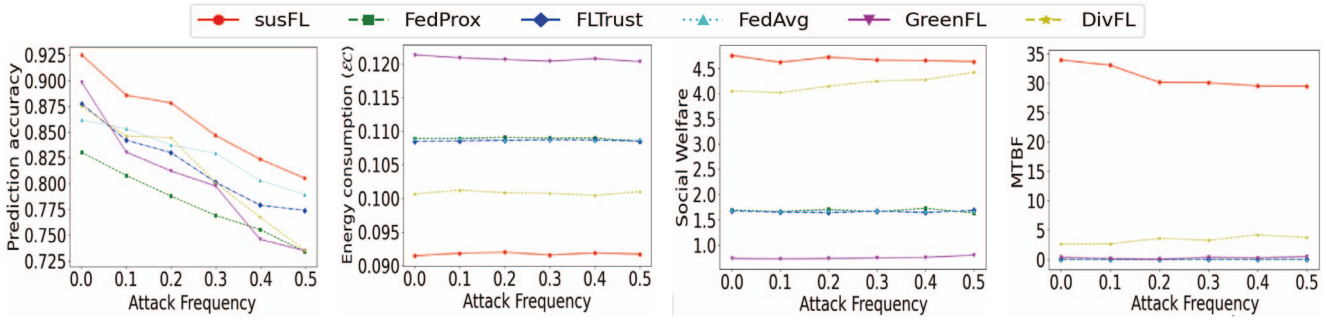
In contrast, the other schemes' performance appears relatively unaffected under varying node density across all metrics, except `GreenFL`'s prediction accuracy. As the system has more clients, `GreenFL`'s prediction accuracy diminishes, implying a potential shortfall with suboptimal local models. This highlights `susFL`'s superiority in prediction accuracy and energy efficiency in large-scale WSNs.

3) **Effect of Initial Energy Levels (E_{init}) on clients:** Fig. 6 demonstrates that increasing initial energy levels in client sensor nodes positively affects performance metrics, with `susFL` outperforming the considered SOTA models in prediction accuracy. This benefit is due to `susFL`'s strategy of selecting clients based on available energy, allowing more clients to participate in FL as initial energy increases, resulting in enhancing accuracy. In contrast, other schemes do not vary the number of participating clients with energy levels, so their prediction accuracy remains unchanged.

Higher energy in the sensors slightly increases energy consumption for the FL process as shown in Fig. 7(b). However, this improves social welfare and MTBF, as in Figs. 7(c) and (d). Unlike other models where client participation is fixed, `susFL` adapts to energy availability, showing its effectiveness in using energy resources for optimal FL operations.

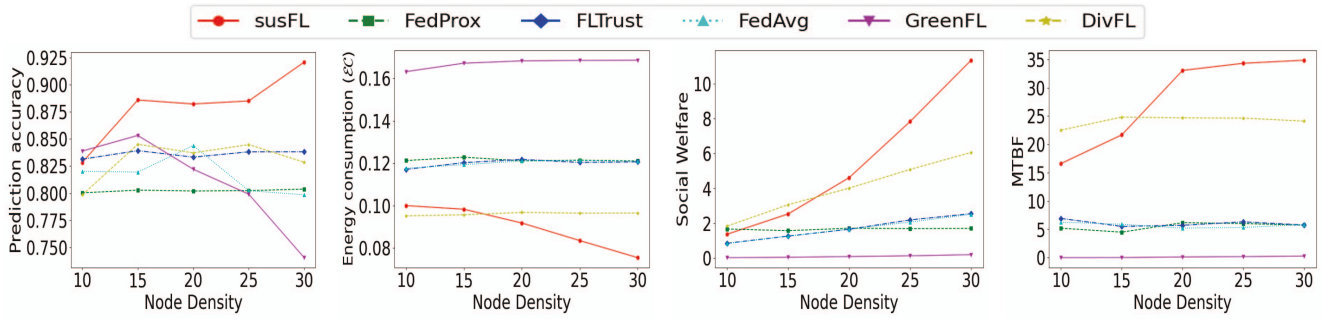
VIII. Conclusion & Future Work

From our research, we found the following. First, we observed that `susFL` demonstrates superior efficiency in global model training within FL operations, utilizing the least energy to attain the highest prediction accuracy compared to benchmark schemes. This efficiency is attributed to an energy-aware client selection mechanism, adeptly choosing an optimal set of clients to balance high accuracy with energy conservation in the sensor network. Second, we found that `susFL` excels in MTBF, enhancing system reliability amid energy variability and environmental dynamics in smart farming. This finding emphasizes the scheme's robustness and the necessity to minimize computational demands on sensor nodes, a lesson underscored by `GreenFL`'s relative underperformance. Lastly, In scenarios involving cyber and adversarial attacks, `susFL` significantly maintains high prediction accuracy with minimal energy use, ensuring ongoing system availability and sustainability. This resilience highlights `susFL`'s effectiveness in



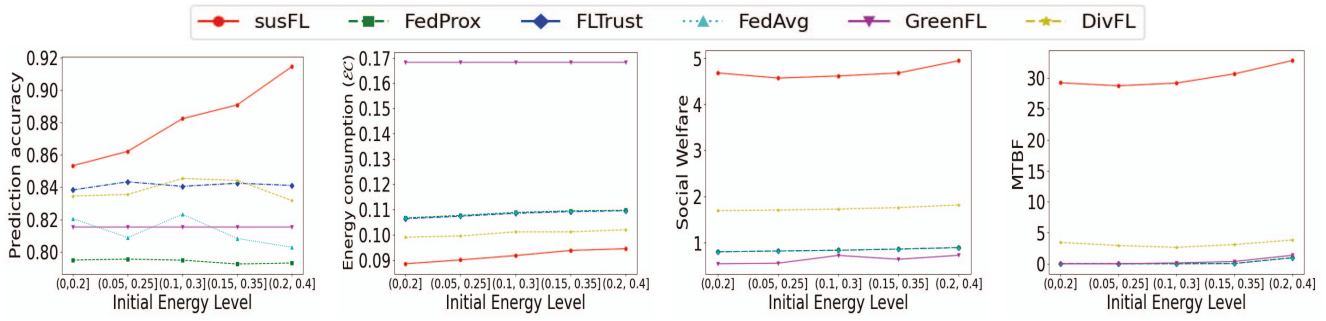
(a) Prediction Accuracy (b) Energy Consumption (c) Social Welfare (d) MTBF

Fig. 5. Effect of Varying Attack Frequency (P_A)



(a) Prediction Accuracy (b) Energy Consumption (c) Social Welfare (d) MTBF

Fig. 6. Effect of Varying Node Density



(a) Prediction Accuracy (b) Energy Consumption (c) Social Welfare (d) MTBF

Fig. 7. Effect of Varying Initial Energy Levels (E_{init})

safeguarding FL operations against potential security threats. These insights emphasize susFL's comprehensive approach to optimizing FL for energy-constrained environments, offering a scalable, secure, and efficient solution for smart agricultural practices.

For **future work**, we will take the following research directions. First, we will incorporate fairness and privacy preservation into client selection to ensure secure and equitable participation in the FL process. Second, we will enhance scalability by increasing the number of clients in the FL operations and expanding the system's capacity for larger and more diverse datasets. Lastly, we will further optimize performance by refining the client selection mechanism to improve prediction accuracy and energy efficiency, especially for large-scale operations.

ACKNOWLEDGEMENT

This work is partly funded by NSF Grants 2106987 and 2107450, the Commonwealth Cyber Initiative (CCI), and Virginia Tech's ICTAS EFO Opportunity Seed Investment Grant.

REFERENCES

[1] G. Idoje, T. Dagiuklas, and M. Iqbal, "Federated learning: Crop classification in a smart farm decentralised network," *Smart Agricultural Technology*, vol. 5, p. 100277, 2023.

[2] M. Caria, J. Schudrowitz, A. Jukan, and N. Kemper, "Smart farm computing systems for animal welfare monitoring," in *2017 40th Int'l Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2017, pp. 152–157.

[3] J. Kaur and R. Dara, "Ensuring privacy in smart farming: Review of regulations, codes of conduct and best practices," *Encyclopedia of Smart Agriculture Technologies*, pp. 1–16, 2023.

[4] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. Bilmes, "Diverse client selection for federated learning via submodular maximization," in *Int'l Conf. on Learning Representations*, 2022.

[5] M. Kim, W. Saad, M. Mozaffari, and M. Debbah, "Green, quantized federated learning over wireless networks: An energy-efficient design," *ArXiv*, vol. abs/2207.09387, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:250644501

[6] H. Elayan, M. Aloqaily, and M. Guizani, "Sustainability of healthcare data analysis iot-based systems using deep federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7338–7346, 2022.

[7] M. Genemo, "Detecting high-risk area for lumpy skin disease in cattle using deep learning feature," *Advances in Artificial Intelligence Research*, vol. 3, no. 1, pp. 27–35, 2023.

[8] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2818–2832, 2022.

[9] J. Fan, X. Wang, Y. Guo, X. Hu, and B. Hu, "Federated learning driven secure internet of medical things," *IEEE Wireless Communications*, vol. 29, no. 2, pp. 68–75, 2022.

[10] K. Ankitha, D. Manjaiah, and M. Kartik, "Data for: Clinical mastitis in cows based on udder parameter using internet of things (iot)," *Mendeley Data*, vol. 1, 2020.

[11] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K.-K. R. Choo, and M. Nafaa, "Felids: Federated learning-based intrusion detection system for agricultural internet of things," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 17–31, 2022.

[12] L. Praharaj, M. Gupta, and D. Gupta, "Hierarchical federated transfer learning and digital twin enhanced secure cooperative smart farming," in *2023 IEEE Int'l Conf. on Big Data (BigData)*. IEEE, 2023, pp. 3304–3313.

[13] D. Vimalajeewa, C. Kulatunga, D. P. Berry, and S. Balasubramaniam, "A service-based joint model used for distributed learning: Application for smart agriculture," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 838–854, 2021.

[14] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[15] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, jan 2019.

[16] Y. Sun, H. Ochiai, and H. Esaki, "Intrusion detection with segmented federated learning for large-scale multiple lans," in *2020 Int'l Joint Conf. on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[17] T. A. Khoa, D.-V. Nguyen, M.-S. Dao, and K. Zettsu, "Fed xdata: A federated learning framework for enabling contextual health monitoring in a cloud-edge network," in *2021 IEEE Int'l Conf. on Big Data (Big Data)*, 2021, pp. 4979–4988.

[18] *CC2640R2F SimpleLink™ Bluetooth® 5.1 Low Energy Wireless MCU*, Texas Instruments, 2016, rev. C. [Online]. Available: https://www.ti.com/product/CC2640R2F

[19] PiCockpit. (2023) How much does power cost for the pi 4?: Picockpit: Monitor and control your raspberry pi: Free for up to 5 pis! [Online]. Available: https://picockpit.com/raspberry-pi/how-much-does-power-usage-cost-for-the-pi-4/

[20] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[21] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 1505–1521.

[22] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third Int'l Conf. on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 2938–2948.

[23] J. So, B. Güler, and A. S. Avestimehr, "Byzantine-resilient secure federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2168–2181, 2020.

[24] C. Xie, M. Chen, P.-Y. Chen, and B. Li, "Crfl: Certifiably robust federated learning against backdoor attacks," in *Int'l Conf. on Machine Learning*. PMLR, 2021, pp. 11 372–11 382.

[25] C. Chen, P. Zhang, H. Zhang, J. Dai, Y. Yi, H. Zhang, and Y. Zhang, "Deep learning on computational-resource-limited platforms: a survey," *Mobile Information Systems*, vol. 2020, pp. 1–19, 2020.

[26] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, "Energy consumption model for sensor nodes based on lora and lorawan," *Sensors*, vol. 18, no. 7, p. 2104, 2018.

[27] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[28] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE Conf. Computer Communications*. IEEE, 2019, pp. 1387–1395.

[29] X. Bei, N. Chen, N. Gravin, and P. Lu, "Budget feasible mechanism design: from prior-free to bayesian," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 2012, pp. 449–458.

[30] S. Chawla, D. L. Malec, and A. Malekian, "Bayesian mechanism design for budget-constrained agents," in *Proceedings of the 12th ACM Conf. on Electronic commerce*, 2011, pp. 253–262.

[31] T. A. Gresik, "Ex ante efficient, ex post individually rational trade," *Journal of Economic theory*, vol. 53, no. 1, pp. 131–145, 1991.

[32] P. Toth, "Dynamic programming algorithms for the zero-one knapsack problem," *Computing*, vol. 25, no. 1, pp. 29–45, 1980.

[33] Z. Shi, L. Zhang, Z. Yao, L. Lyu, C. Chen, L. Wang, J. Wang, and X.-Y. Li, "FedFAIM: A model performance-based fair incentive mechanism for federated learning," *IEEE Transactions on Big Data*, 2022.

[34] A. Colombo and A. S. de Bustamante, "Systems reliability assessment," *Proceedings of the Ispra Course Held at the Escuela Tecnica Superior de Ingenieros Navales. In Collaboration with Universidad Politecnica de Madrid*, 1990.

[35] J.-H. Cho, S. Xu, P. M. Hurley, M. Mackay, T. Benjamin, and M. Beaumont, "Stram: Measuring the trustworthiness of computer-based systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–47, 2019.

[36] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 394–21 405, 2020.

[37] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[38] X. Zhang, F. Li, Z. Zhang, Q. Li, C. Wang, and J. Wu, "Enabling execution assurance of federated learning at untrusted participants," in *IEEE INFOCOM 2020-IEEE Conf. Computer Communications*. IEEE, 2020, pp. 1877–1886.