8-26-2022

FedDP: A privacy-protecting theft detection scheme in smart grids using federated learning

Muhammad Mansoor Ashraf

Muhammad Waqas
Edith Cowan University, m.waqas@ecu.edu.au

Ghulam Abbas

Thar Baker

Ziaul Haq Abbas

See next page for additional authors

10.3390/en15176241
Ashraf, M. M., Waqas, M., Abbas, G., Baker, T., Abbas, Z. H., & Alasmary, H. (2022). FedDP: A privacy-protecting theft detection scheme in smart grids using federated learning. Energies, 15(17), 6241. https://doi.org/10.3390/en15176241

energies

MDPI

Article

FedDP: A Privacy-Protecting Theft Detection Scheme in Smart Grids Using Federated Learning

Muhammad Mansoor Ashraf [1], Muhammad Waqas [2,3], Ghulam Abbas [1], Thar Baker [4,*], Ziaul Haq Abbas [5] and Hisham Alasmary [6,7]

1 Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Swabi 23460, Pakistan
2 Computer Engineering Department, College of Information Technology, University of Bahrain, Shakir 32038, Bahrain
3 School of Engineering, Edith Cowan University, Joondalup Perth, WA 6027, Australia
4 Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah 27272, United Arab Emirates
5 Faculty of Electrical Engineering, GIK Institute of Engineering Sciences and Technology, Swabi 23460, Pakistan
6 Department of Computer Science, College of Computer Science, King Khalid University, Abha 62529, Saudi Arabia
7 Information Security and Cybersecurity Unit, King Khalid University, Abha 62529, Saudi Arabia
* Correspondence: tshamsa@sharjah.ac.ae

Abstract: In smart grids (SGs), the systematic utilization of consumer energy data while maintaining its privacy is of paramount importance. This research addresses this problem by energy theft detection while preserving the privacy of client data. In particular, this research identifies centralized models as more accurate in predicting energy theft in SGs but with no or significantly less data protection. Current research proposes a novel federated learning (FL) framework, namely FedDP, to tackle this issue. The proposed framework enables various clients to benefit from on-device prediction with very little communication overhead and to learn from the experience of other clients with the help of a central server (CS). Furthermore, for the accurate identification of energy theft, the use of a novel federated voting classifier (FVC) is proposed. FVC uses the majority voting-based consensus of traditional machine learning (ML) classifiers namely, random forests (RF), k-nearest neighbors (KNN), and bagging classifiers (BG). To the best of our knowledge, conventional ML classifiers have never been used in a federated manner for energy theft detection in SGs. Finally, substantial experiments are performed on the real-world energy consumption dataset. Results illustrate that the proposed model can accurately and efficiently detect energy theft in SGs while guaranteeing the security of client data.

Keywords: federated learning; smart grids; federated voting classifier; privacy protection; theft detection

Citation: Ashraf, M.M.; Waqas, M.; Abbas, G.; Baker, T.; Abbas, Z.H.; Alasmary, H. FedDP: A Privacy-Protecting Theft Detection Scheme in Smart Grids Using Federated Learning. *Energies* **2022**, *15*, 6241. https://doi.org/10.3390/en15176241

Academic Editor: Ignacio Mauleón

Received: 9 July 2022
Accepted: 24 August 2022
Published: 26 August 2022

1. Introduction

Smart grids (SG) are among the most important form of the Internet of Things (IoT) network, which brings comfort to users with the easily managed production, dispensation, and utilization of energy [1]. An SG provides unwavering quality, adaptability, and proficiency of power systems to consumers [2,3]. With the auspicious development of SG, it has extracted expanding consideration from states, ventures, and researchers. As neoteric research indicated that the SG market increase from USD 23.8 billion in 2018 to USD 61.3 billion by 2023 [4]. SG can be enhanced further by integrating it with new technologies such as machine learning (ML) cloud computing and fifth generation (5G) cellular networks [5]. With the popularity of the IoT, appliances such as smart meters (SMs) can produce an enormous amount of data [6]. Data-driven AI technologies could benefit from this data to enhance the user experience with customized energy strategies.

These technologies also enable the service providers (SPs) to better predict the power consumption and increase profits [7].

Despite the conspicuous features of SMs, they endanger the user's privacy [8,9]. For instance, SPs can easily infer the consumer's routines and daily lifestyle from the real-time energy consumption data collected by SMs. Moreover, the knowledge of these trends even results in crimes such as energy theft. In particular, client energy consumption data needs to be transferred to the central server (CS) for knowledge extraction [10,11], which, as a result, compromises the security and confidentiality of the user energy data. Previous studies show significant annual financial losses due to energy theft, e.g., Canada faces a loss of USD 100 million [12], USD 170 million are lost in the United Kingdom [13], and in the United States, energy theft can cause losses yearly of USD 6 billion [14].

Federated learning (FL), also known as collaborative learning, is a novel ML approach that trains the ML model over different devices. These devices hold different local data samples and are placed at various locations. FL allows SPs to extract intuition from users' data while enabling clients to keep their private data on their respective devices [15]. Figure 1 depicts the generic framework for the FL-enabled SGs. In this, only the parameters of the ML model need to be shared with the CS while keeping the user data secure on the trusted theft detection station (TDS). TDS downloads the parameters from CS and performs training and evaluation of AI models using the local data. FL is an iterative process that is repeated for a specific number of iterations or until a predefined accuracy is achieved [16] or the loss of the ML model is minimized [17]. CS, instead of receiving all the user data, only aggregates the parameters of the model from various TDS, which enables collaborative learning.

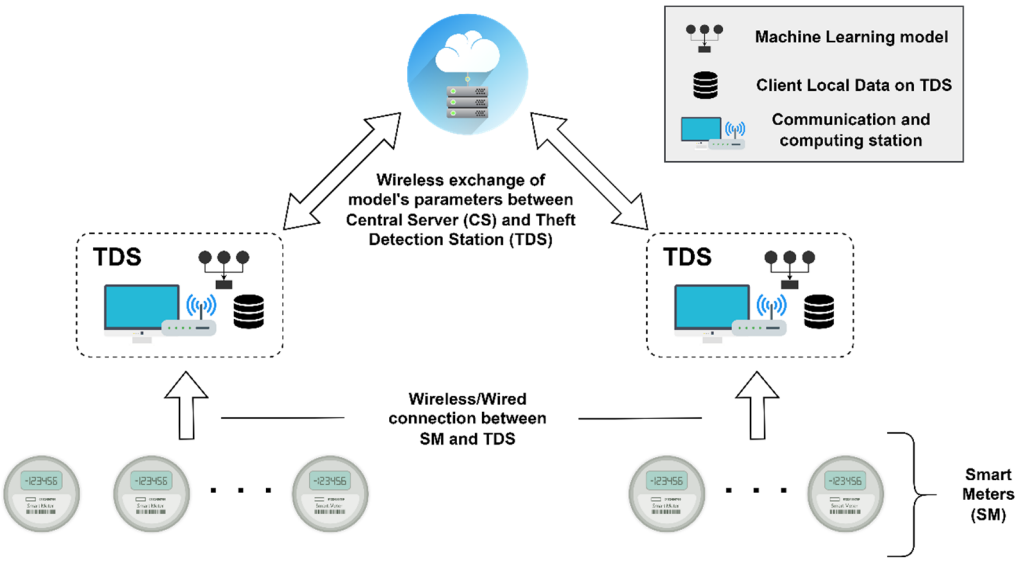

Figure 1. Federated learning enables smart grids networks for theft detection.

Despite the benefits of FL, it faces some basic challenges while implementing the ML models. For instance, cutting-edge deep neural networks (DNN) have extensively been used for identifying the energy theft in SGs. The problem with these models is that they require significant computing resources that are not a viable option for resource-constrained SGs. Additionally, these models have low accuracy and precision in energy theft detection [17–19]. Motivated by this, this research proposes the novel energy theft detection model for SGs which has relatively high accuracy and can preserve consumer data.

Contributions

The major contributions of this research are summed up below.

- First, a novel framework, federated data privacy (FedDP), is presented for energy theft detection in SGs.
- Secondly, to improve the accuracy and to avoid bias of any single ML algorithm, an ensemble learning classifier is proposed in a federated learning environment, named the federated voting classifier (FVC).
- FVC can identify the energy theft in SGs, even in the presence of highly unbalanced data, with an accuracy of 91.67%, which is a relatively better performance than the existing techniques.
- Moreover, FVC can also significantly outperform other state-of-the-art algorithms in terms of execution time when implemented on the same hardware.

More specifically, FedDP has significant characteristics. First, energy utilization data from SMs are placed in TDSs that can preserve their privacy. Second, all TDSs can cooperatively train and evaluate the ML model by applying the FL in which only the parameters of the ML model are shared with the CS for aggregation. FVC takes the consensus of traditional ML models, namely, random forests (RF), k-nearest neighbors (KNN), and bagging classifiers (BG). Experiments on the real energy usage dataset show that FVC could surpass the other advanced models concerning precision and log loss. The major contribution of this research is illustrated in Figure 2 below.

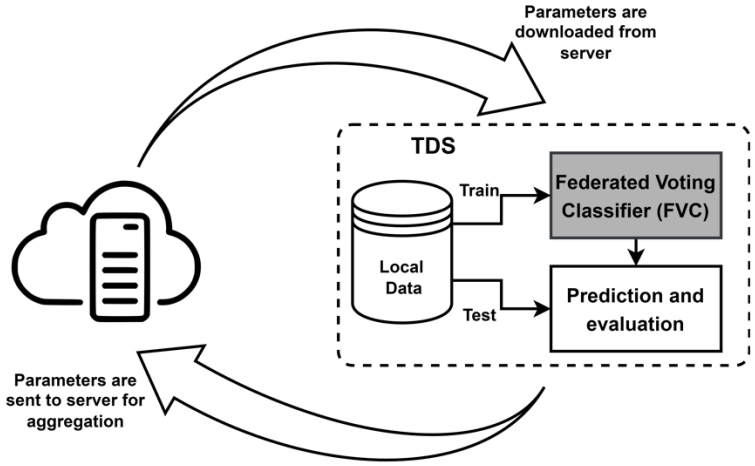

Figure 2. Abstract overview of the proposed framework. Highlighted area depicts the proposed novel federated voting classifier (FVC).

2. Literature Review

Reviewed literature in this section focuses on energy theft detection in SGs and FL schemes.

2.1. Energy Theft in SGs

Energy theft is the primary concern in SGs, and therefore a significant amount of the literature is based on proposing new ML models for this problem. To identify electricity theft, Zheng et al. [20] presented the novel wide and deep convolutional neural network (CNN). Moreover, a model established on the three gradients boosting was suggested by Punmiya et al. in 2019 [21]. Their proposed model initially extracted the intrinsic laws from the recorded energy utilization data. Later, the authors employed the gradient boosting, categorical boosting, and light gradient boosting methods to accurately detect the energy theft in smart grids. Results depict that the model has superior performance when compared with other ML models. To better detect electricity theft in smart grids, Li et al., 2019 [22] presented the three-stage models using ML and various statistical models

to better detect electricity theft in smart grids. The authors found out that the proposed technique can achieve high theft detection accuracy. These studies, however, focus on theft sensing in the context of power consumption, and Ismail et al., 2020 [23] emphasized theft detection in distributed power generation systems.

In addition to detecting electricity and power theft in SGs, the application of security and privacy is also gaining traction. This is because the established networks need to follow different laws and policies, such as the General Data Protection Regulation (GDPR) [24]. To this end, Yao et al. in 2019 [25] suggested the use of a CNN for detailed examination of consumption data along with the implementation of homomorphic encryption to secure the transmitting data. Furthermore, the use of functional encryption for information security and a feedforward neural network was proposed by Ibrahim et al. [26] in 2021.

All the mentioned approaches can accurately perceive energy theft in SGs, but this is at the cost of the privacy of user data. In other words, all the user data is sent to the central entity for analysis, which causes huge computational overheads and network congestion. Although some studies use encryption to protect the data before transmitting, this require additional computations at the edge node in IoT. As compared to the existing solutions, the proposed FedDP framework employs FL to amplify the privacy and security of consumers' data. Moreover, this research develops the novel FVC algorithm that can exceed the existing solutions for theft detection in SGs.

2.2. Applications of Federated Learning

The majority of the research focuses on applying encryption algorithms to the users' raw data, which is too arduous and laborious to implement [27]. To address this issue, recent studies have used FL to collectively train and test ML models while keeping data private on the trusted nodes. Only the specification of the ML algorithms is shared with the server, which can notably reduce the security and computational problems in centralized ML [28]. In [29], Li et al. exerted FL to examine the unusual actions of the user. In contrast, Sater et al. in [30] reduced the response delay using FL to detect anomalies in smart buildings. Moreover, a multi-layer perceptron (MLP) model was proposed in a federated manner for medical devices [31]. A federated CNN framework for wearable medical devices was presented by Chen et al. in 2020 [18]. In addition, Liu et al. proposed the attention-based long short-term memory (LSTM) for industrial IoT [32]. Furthermore, the security of FL in the application process was investigated by Li et al. [33] in 2020. However, the authors concentrated on malicious third parties rather than the security of customers' information.

2.3. Federated Learning in Smart Grids

Despite the advantages of FL in IoT, it has not been completely explored for distributed energy theft detection. For instance, Liu et al. [34] in 2022 proposed an FL-enabled framework for understanding the different patterns of power utilization in smart grids. Moreover, the use of FL in collaboration with edge cloud in SGs was proposed by Su et al. [17] in 2021. Only recently, a novel FedDetect approach was introduced by Wen et al. [19]. The authors proposed temporal convolutional networks (TCN) to distinguish between normal energy usage and energy theft in SGs. The problem with these DNN models is that they require high computational power and have an enormous number of parameters. These models also require a huge amount of data for training. Moreover, these models are less precise, have high training time, and may not be practicable in resource-constrained IoT networks. This study presents the novel FVC that can accurately and precisely detect theft in SGs environments with less computational overhead.

3. Methodology

In this subsection, the fundamental ideas of the proposed FedDP architecture are introduced. Mainly, the FedDP model is explained, and the proposed FVC model is elaborated.

3.1. System Model

As described earlier, federated data privacy (FedDP) aims to design a privacy-preserving FL framework that can exploit ML classifiers to identify energy theft in a distributed fashion. FL is the platform that supports ML classification algorithms over multiple decentralized clients that hold the local data samples. This provides data privacy by enabling on-device prediction without sharing the complete data. Moreover, for predicting the energy theft in SGs, FedDP proposes a voting classifier in a federated manner. The voting classifier is an ensemble learning classification model that operates by taking the consensus of the different ML classification models to predict the final class.

FedDP is a two-tier framework that has two major constituents.

(1) Theft detection station (TDS): A TDS can obtain real-time data on energy utilization from the group of SMs in its vicinity. This research assumes that (1) the wired or wireless connection between SM and TDS is secure. (2) TDS are low-powered devices but have sufficient storage and processing power that can store the data along with the training of the ML model. (3) Each TDS can automatically infer the data label that associates with previous data of an SM with electricity thievery. (4) TDS can also securely communicate with CS for the exchange of the ML model parameters. In FedDP, TDS is considered a federated client. During the training stage, TDSs download the model parameters (e.g., weights in the neural networks and number of neighbors, Leaf_size, etc., in KNN) from the server and evaluate them using local data.

(2) Central server (CS): It is the initialization of the FL process and is responsible for broadcasting default parameters and learning models to all TDS. In the FL process, the CS can receive the model parameters, accumulate them, and can broadcast the improved parameters to all TDSs.

3.2. Privacy-Preserving FedDP

As the energy-related data of each user is limited, TDS pivot on the assortment of large, best quality data from SMs. In each training round of TDS, it can determine $X_{train} \in X$ for training the global classification model CM. Additionally, X_{train} can be represented by the collection of input samples corresponding with their respective label $\{x_t, y_t\}_{t=1}^{X_{train}}$, where x_t is the single SM record and y_t is its corresponding label (i.e., theft or no theft). A fundamental task of any ML model is to learn the mapping of input samples to output labels and the specification of CM, which predicts the y_t relative to x_t while increasing the accuracy or mitigating the loss [17]. f_{loss} identifies the difference between the predicted and true labels of each training instance $\{x_t, y_t\}$. For all samples in X, f_{loss} is the mean loss of all instances in X_{train}. Therefore, for each TDS, the overall loss is the mean prediction loss of all instances in X_{train}. The notations used in this paper are listed in Glossary.

$$L(CM) = \frac{1}{X_{train}} \sum_{t=1}^{X_{train}} f_{loss}(x_t; CM) \tag{1}$$

The major purpose of the FL is to find the most advantageous parameters of the ML model that lessen the global loss function.

$$CM^* = argmin_{CM}(L) \tag{2}$$

FedDP proposes a novel framework for collaborative learning of TDS. It has various phases that are elaborated below and illustrated in Figure 3.

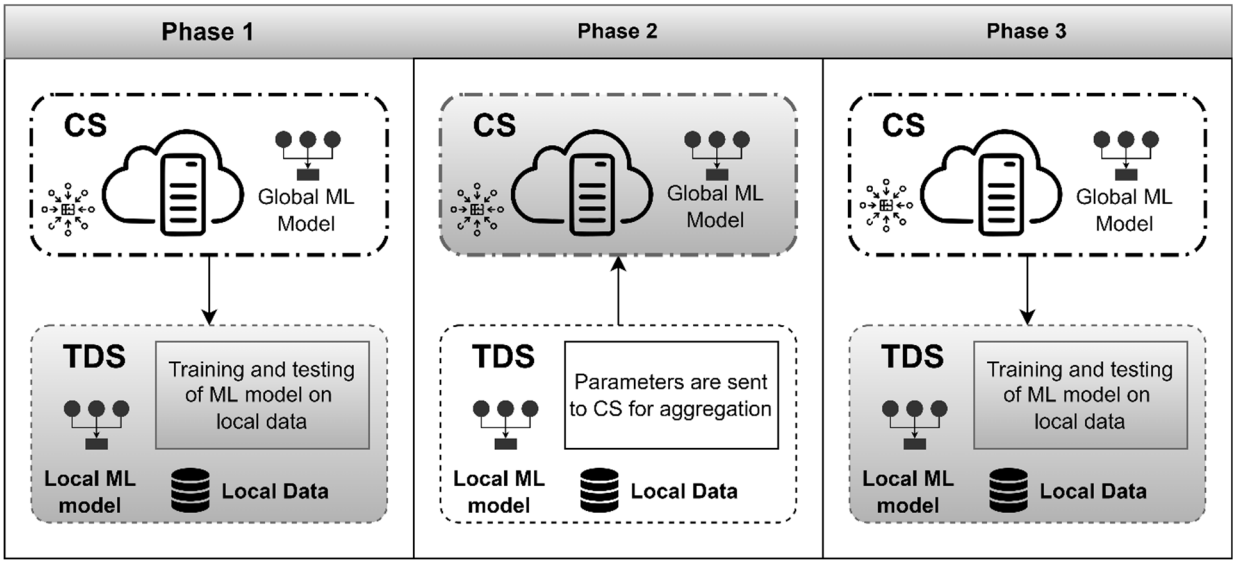

Figure 3. An illustration of the FedDP architecture. The shaded area represents the working in the specific phase.

Phase 1: In phase 1 of FedDP, CS initializes CM with any parameters that are necessary for the training of the ML model. Moreover, CS also disseminates these specifications to all the participating clients, i.e., TDS. Once each TDS receives the parameters from CS, the local model is trained on the local heterogeneous dataset to generate a new set of model parameters.

Phase 2: Once θ(T), a set of TDSs, has updated local parameters, then these characteristics are sent to CS for aggregation. CS executes the federated aggregation algorithm that computes the average of parameters received from individual TDS. In this way, a global and more accurate model is developed. The job of CS is to allow collective learning in such a way that each TDS learns from the experience of other detection stations and learns to build an accurate machine learning model. This enables the ML model to continually evolve and update itself.

Phase 3: Subsequently, CS transmits the aggregated parameters to each participating TDS, so that TDS again trains and evaluates CM on its local data by integrating these updated parameters. After the third phase in FedDP, the federated process again continues for a specific number of iterations.

3.3. Predictive Methods

Several classification algorithms are used to predict the energy theft in the network. These models are trained and evaluated on the training and test data. This research proposes a voting-based ensemble of RF, KNN, and BG in a federated manner. To the best of our knowledge, conventional ML algorithms have never been used in FL. These classifiers are briefly described below.

3.3.1. Random Forest (RF)

RF was inspired by decision-tree learning and was first proposed by Breiman in 2001 [35] as a classifier. It consists of an abundance of tree-structured classifiers so that each tree depends on the value of individualistic and same arbitrary vectors. In contrast, each tree chooses the most successive class in the dataset. In RF, every node is divided by utilizing the best attributes among the subspace of features, picked at random for that node. This technique is vigorous and performs well when contrasted with other regularly used ML models. However, its execution depends on identifying several trees to develop and the quantity of contenders arbitrarily selected at each stage. Generally, the user specifies the number of trees in RF, and this can be done by starting from a low number and slowly

increasing it. This study uses 10 trees to train the RF model. Moreover, the model uses the *Gini* index to split the whole dataset to create the subset of data for each tree.

3.3.2. k-Nearest Neighbors

The k-nearest neighbors (KNN) classification technique was proposed by Cover et al. in [36]. It is a classification method that classifies the test data by looking at the nearby set of predicted samples. Two choices should be remembered while executing the KNN. First, the value of k determines that many neighbors ought to be viewed to characterize the test sample and the distance metric to evaluate the distance between the test sample and previously classified samples (i.e., train samples). This study uses 3 as the value of k, and the *Euclidean distance* function calculates the interspace between X_{train} and X_{test}. To optimize the memory utilization, *leaf_size* is set to 5. These values are taken on test and trial basis. If E_{ij} denotes the distance between i^{th} and j^{th} datapoint and a_{ik} and a_{jk} constitutes the values of the k^{th} variable for sample i and j, respectively, then *Euclidean distance* can be given as:

$$E_{ij} = \sqrt{\sum_{i} \left(a_{ik} - a_{jk} \right)^2} \tag{3}$$

3.3.3. Bagging Classifier (BG)

In 1996, Breiman [37] first presented bagging predictors as a model for creating different indicators and utilizing them to make accumulated predictors. It is an ensemble model, that aims to boost further the strength and accuracy of the machine learning model. It accumulates the mean over the variations while predicting a numerical result and does a vote when predicting a class. The essential concept of the bagging classifier is that it generated many "weak learners" and utilized them to construct "strong learners". It builds numerous decision trees (DTs) that are weak learners and merges them to produce a strong learner. Each tree gives the decision in favor of a class, and the last expectation of the new class is acquired by the class that has the most votes. Using the decision tree in the bagging classifier shows that our dataset has a high disparity among classes. So, DTs offer good behavior by weighting the outputs of the trees and lessening the variance of the dataset and avoiding over-fitting. This study uses the ensemble of 10 DTs that can cast the vote of the majority class to give the final prediction.

3.4. Proposed FVC Algorithm

This research proposes a novel federated voting classifier (FVC) for energy theft detection. FVC is an ensemble technique that combines various ML models to make one optimal predictive model. The ensemble model combines the output of different ML models, regarding them as the "committee" of multiple decision-makers. FVC is similar to the bagging classifier, with the main difference being that the bagging classifier uses an ensemble of decision trees. In contrast, FVC takes the vote of RF, KNN, and BG, as shown in Figure 4. Every classifier in the voting classifier is run independently on the training data, and FVC uses the majority consensus of all the models in FVC. For example, let the individual model, RF, KNN, and BG in FVC be given by CL_i and their corresponding predictions are specified by y_i, y_j, and y_k, respectively. Moreover, $\hat{y}_{x_{test}}$ represents the final prediction of the FVC model, then the consensus of FVC can be given by:

$$\hat{y}_{X_{test}}(CM_{FVC}) = argmax CL_i(X_{test}) \tag{4}$$

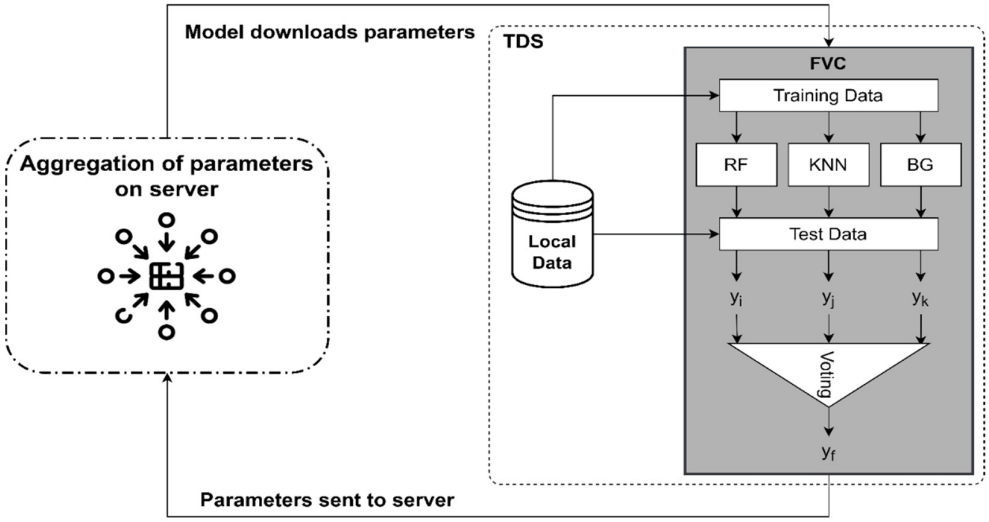

Figure 4. Functionality of the proposed FVC model.

FVC is trained on each client on the local data partitioned into X_{train} and X_{test}. Furthermore, an evaluation of the FVC model is performed on the TDS. After that, each client sends model parameters to the server to execute the aggregation of parameters from each TDS. These rounds are computed multiple times and optimized model CM^* is achieved.

4. Methodology

4.1. Computing Platform

To evaluate this research, FedDP is implemented by creating virtual FL clients and servers on the same machine. The system used has an Intel i7-6700 CPU with 8GB RAM. Regarding training and testing FL models, this research uses the Windows operating system, and *flower* [38] is used as a federated framework.

4.2. Dataset Description

Dataset used in this research is from State Grid Corporation of China (SGCC) [25], which recorded the energy utilization of 42,372 customers for 1035 days. SGCC data is binary data having labels theft or no theft. The dataset is divided randomly so each TDS can have a different dataset. This research divides the data set so that 80% of data are for training the ML model, and 20% of data are for testing the model.

4.3. Evaluation Metrics

Numerous performance metrics could be used to evaluate the ML model. To measure the execution of the various models, this research uses accuracy, precision, and F-measure. These criteria are measured based on standard indicators, true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*). For example, *TP* and *TN* decide the absolute number of accurately characterized tests, whereas *FP* and *FN* show the count where the ML model has been misclassified. Then *accuracy*, *precision*, *f_measur*, and s *recall* are given by:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$precesion = \frac{TP}{TP + FP} \tag{6}$$

$$f\_measure = \frac{2 * TP}{2 * TP + FP + FN} \tag{7}$$

$$recall = \frac{TP}{TP + FN} \tag{8}$$

This study also utilizes log loss and root-mean-squared error (RMSE) as performance criteria. If y_i represents the true label, $p(y_i)$ is the predicted probability of that respective label, and $\hat{y}_i$ is the predicted class for i^{th} sample, and n_{test} are the sample count in X_{test}, then log loss and RMSE can be given by:

$$l_{log} = -\frac{1}{n_{test}} \sum_{i}^{n_{test}} \widetilde{y}_i \,.\log(p(y_i)) + (1 - y_i) \,.\log(1 - p(y_i)) \tag{9}$$

$$RMSE = \sqrt{\sum_{i}^{n_{test}} \frac{(\hat{y}_i - y_i)^2}{n_{test}}} \tag{10}$$

Moreover, to illustrate the computation overhead of various ML models, federated time (FT) is used, which is the total time taken by one FL round [39].

$$FT = T(fc_i) + T(CS) + T(average) \tag{11}$$

where fc_i is the i^{th} federated client, $T(fc_i)$ is the complete time taken by any client, $T(CS)$ dis the time the server takes to communicate with the participating clients, and $T(average)$ represents the entire time taken for implementation of the FL averaging method.

5. Results and Discussion

The experimental setup for this study is aimed to substantiate the proposed model and measure its efficacy to the existing state-of-the-art schemes. In addition, the results are also compared with the centralized voting classifier (VC) by considering the performance measures discussed in Section 4.3.

5.1. Comparison with Centralized Voting Classifier

To compare the results of the FVC and centralized voting classifier, five clients are considered, each having an individualistic dataset. Five experiments are performed using five rounds of federated communication and in each test, and the last round's performance is reported. Accuracy, precision, F-measure, and recall are shown in Tables 1–4, respectively. In each of the comparisons, centralized voting classifiers have higher performance but with significant data privacy issues as all the data needs to be transferred to the CS. In comparison, FVC can achieve almost the same performance while assuring data privacy.

Table 1. Comparison of accuracy between different clients and centralized model.

After 5 Rounds of FL	Client 1	Client 2	Client 3	Client 4	Client 5	Centralized
Test 1	91.44	91.58	91.42	91.06	91.33	92.08
Test 2	91.25	91.61	91.41	91.05	91.35	91.91
Test 3	91.31	91.44	91.37	91.19	91.64	91.97
Test 4	91.56	91.51	91.39	91.33	91.54	91.91
Test 5	91.53	91.67	91.12	91.12	91.59	91.81
Avg	**91.42**	**91.56**	**91.34**	**91.15**	**91.49**	**91.93**

Table 2. Comparison of precision between different clients and centralized model.

After 5 Rounds of FL	Client 1	Client 2	Client 3	Client 4	Client 5	Centralized
Test 1	88.78	88.68	88.77	87.81	88.00	90.75
Test 2	88.00	88.75	88.63	88.07	87.86	90.45
Test 3	88.26	88.19	88.46	88.40	89.02	90.57
Test 4	89.32	88.22	88.49	88.99	88.47	90.41
Test 5	89.16	89.03	87.45	88.10	88.80	90
Avg	**88.70**	**88.57**	**88.36**	**88.27**	**88.43**	**90.43**

Table 3. Comparison of f_measure between different clients and centralized model.

After 5 Rounds of FL	Client 1	Client 2	Client 3	Client 4	Client 5	Centralized
Test 1	84.43	88.63	87.92	87.69	88.47	89.93
Test 2	88.04	88.86	88.03	88.25	88.31	89.16
Test 3	88.15	88.51	88.41	88.02	88.43	89.29
Test 4	88.49	88.21	88.05	88.24	88.17	89.19
Test 5	88.47	88.72	87.98	87.54	88.59	89.12
Avg	**87.51**	**88.58**	**88.07**	**87.94**	**88.39**	**89.33**

Table 4. Comparison of recall between different clients and centralized model.

After 5 Rounds of FL	Client 1	Client 2	Client 3	Client 4	Client 5	Centralized
Test 1	91.44	91.58	91.42	91.06	91.33	91.84
Test 2	91.25	91.61	91.41	91.05	91.35	91.87
Test 3	91.31	91.44	91.37	91.19	91.64	91.97
Test 4	91.56	91.51	91.39	91.33	91.54	91.90
Test 5	91.53	91.67	91.12	91.12	91.59	91.83
Avg	**91.41**	**91.56**	**91.34**	**91.15**	**91.49**	**91.88**

Additionally, an illustration of the voting classifier on a central server and FVC in terms of log loss and RMSE is illustrated in Figures 5 and 6, respectively. It is noticeable that centralized voting classifiers have comparatively less loss, but all of the data needs to be sent to CS in a centralized model, which compromises privacy. Moreover, different clients have varying log loss and RMSE, which shows that each TDS has a different dataset.

Figure 5. Comparison of log loss between different clients and centralized model.

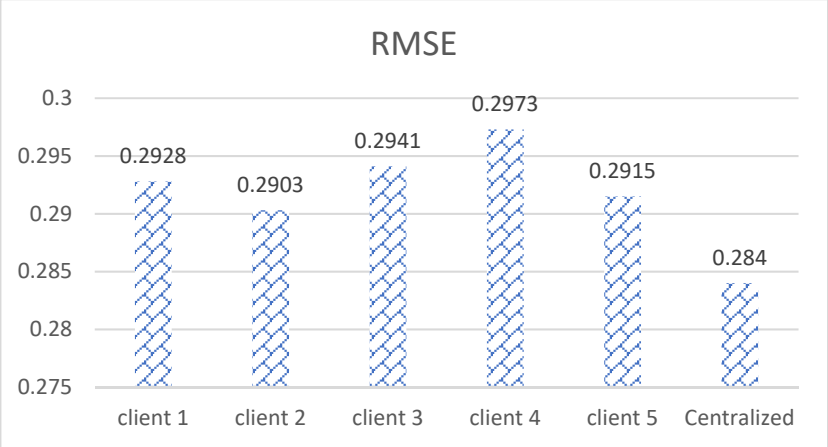

Figure 6. Comparison of average RMSE between different clients and centralized model.

5.2. Comparison with Other State-of-the-Art Models

Finally, the comparison of previous state-of-the-art models is given in this section. The accuracy, precision, F-measure, and recall findings are expressed in Table 5, whereas Figures 7 and 8 illustrate the log loss and RMSE. Two federated clients (i.e., TDSs) and one federated round are considered for these experiments. The neural networks used in this research in a federated manner are simple recurrent neural networks (RNN), simple long-short term memory (LSTM), and simple gated recurrent units (GRU). Each model used one input layer, one RNN/LSTM/GRU layer with 64 neurons and "*tanh*" as an activation function. These neural networks also have one output layer with "sigmoid" as an activation function. Moreover, results of FVC are also compared with the previous literature; specifically, FedCNN was proffered by Chen et al. [18] for abnormal detection in wearable health, FedMLP was presented by Schneble et al. [31] for identification of abnormalities in computer programs, and FedTCN was used by Wen et al. [19] for energy theft detection. It can be seen from Table 4 that the accuracy, precision, and F-measure of FVC are the highest among all the models. In addition, Figures 7 and 8 illustrate that the proposed FVC approach has the lowest log loss and RMSE.

Table 5. Comparison of previous literature with FVC in terms of accuracy, precision, F-measure, and recall.

Models	Accuracy	Precision	F-Measure	Recall
FedTCN [19]	91.54	83.79	87.50	91.28
FedCNN [18]	91.31	83.86	87.17	91.05
FedMLP [31]	89.14	86.37	87.48	89.14
Fed Simple RNN	91.50	83.73	87.45	88.10
Fed Simple LSTM	91.53	88.38	87.86	90.10
Fed Simple GRU	90.87	86.63	87.92	88.56
FVC	**91.67**	**89.03**	**88.72**	**91.67**

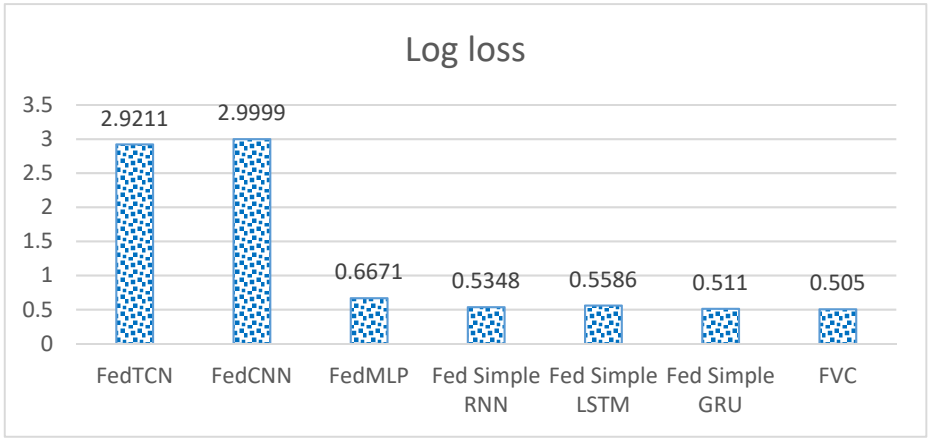

Figure 7. Comparison of previous literature with FVC in terms of log loss.

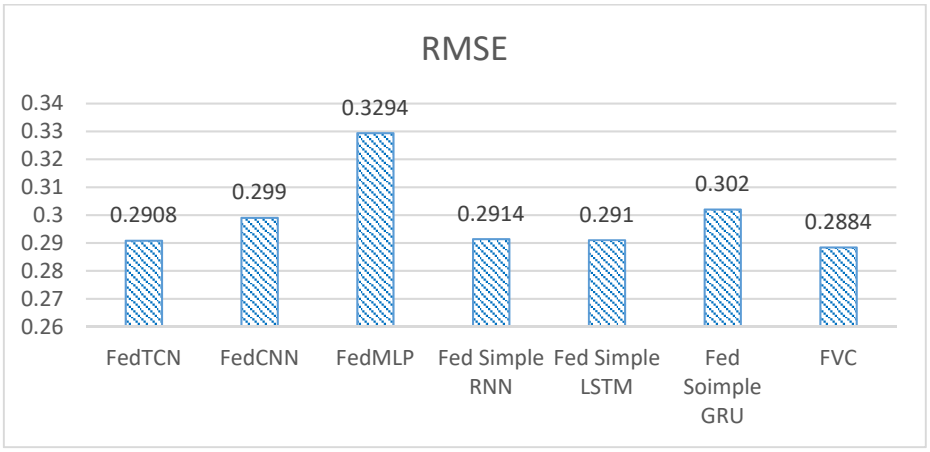

Figure 8. Comparison of previous literature with FVC in terms of RMSE.

Computation Overhead

This section compares computation overhead in terms of federated time (Equation (10)) of FVC with earlier research. It can be seen from Figure 9 that two clients and one round of communication between server and client is the fastest in FedMLP, but this model is less accurate (Table 4). Although the federated time for FVC is approximately 35 s, it is acceptable as it provides the highest accuracy and lowest loss.

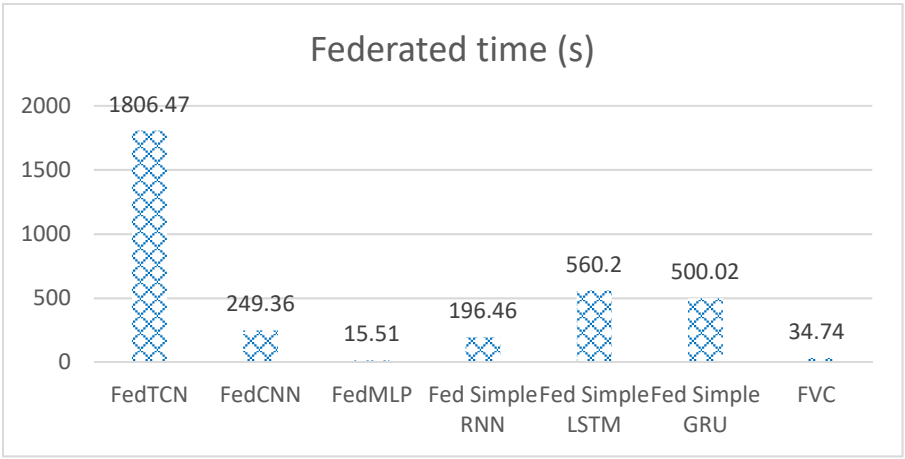

Figure 9. Federated time comparison of FVC with previous research.

6. Conclusions

This study investigates how to identify energy theft in SGs while ensuring the privacy of users' data. A novel federated framework, FedDP, is proposed that uses traditional ML algorithms to predict energy theft behaviors. FedDP first ensures the privacy of data sensed by the SMs, and to enable collaborative learning, a server-based approach is used to aggregate the parameters of the ML classifier with the computational overhead of approximately 35 s. This overhead is significantly better when compared with state-of-the-art models. Current research also proposes the use of a novel federated voting classifier (FVC) to detect energy theft while using the real-world dataset, accurately. Comparative results demonstrate that the proposed FVC performed better when compared with other models. Results show that FVC has the highest accuracy and precision of 91.67% and 89.03, respectively, when compared with the existing models. Moreover, slight improvements can also be seen in terms of F_measure and recall (Table 5). The efficacy of the FVC classifier was also estimated using RMSE and log loss. Results illustrate that the FVC has the lowest loss (i.e., 0.2884 RMSE and log loss of 0.5050). In future work, we may also want to include the security schemes to protect the parameters when they are exchanged between server and client. Moreover, it is also possible for CS to learn user behavior from the parameters it receives. To avoid this, the parameters could be encrypted using homomorphic encryption so that the CS cannot even look at the parameters.

Author Contributions: Conceptualization, M.M.A., M.W., G.A., Z.H.A.; methodology, M.M.A., M.W, G.A., Z.H.A.; software, M.M.A., M.W., G.A., Z.H.A.; validation, M.W., G.A., Z.H.A.; formal analysis, M.M.A.; investigation, M.W., G.A., Z.H.A.; resources, M.W., G.A., T.B., Z.H.A.; data curation, M.M.A.; writing—original draft preparation, M.M.A.; writing—review and editing, M.W., G.A., Z.H.A., T.B., H.A.; visualization, M.M.A., T.B., H.A.; supervision, M.W., G.A., Z.H.A.; project administration, T.B., H.A.; funding acquisition, T.B., H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://github.com/EngrMMansoor/FedDP-A-Privacy-Protecting-Theft-Detection-Scheme-in-Smart-Grids-Using-Federated-Learning.

Conflicts of Interest: The authors declare no conflict of interest.

Glossary

fc_i	i^{th} federated client
f_{loss}	Loss function of CM for a data sample
n_{test}	Total number of samples in testing data
$p(y_i)$	Predicted probability of the i^{th} label
CM	Classification model
CM^*	Classification model with optimal parameters
E_{ij}	Euclidean distance between i^{th} and j^{th} data point
FT	Total execution time of the federated process
L	Overall loss of each TDS
$T(average)$	Time taken by the averaging algorithm running on the server
$T(CS)$	Time taken by the server for communication
$T(fc_i)$	Time taken by i^{th} federated client
X	Whole dataset
X_{test}	Testing dataset on each client
X_{train}	Training dataset on each client
$\theta(T)$	Set of TDSs
$\hat{y}_i$	Predicted class for the i^{th} sample

References

1. Tariq, M.; Ali, M.; Naeem, F.; Poor, H.V. Vulnerability assessment of 6g-enabled smart grid cyber-physical systems. *IEEE Internet Things J.* **2021**, *8*, 5468–5475. [CrossRef]
2. Zhang, X.; Biagioni, D.; Cai, M.; Graf, P.; Rahman, S. An Edge-Cloud Integrated Solution for Buildings Demand Response Using Reinforcement Learning. *IEEE Trans. Smart Grid* **2021**, *12*, 420–431. [CrossRef]
3. Jiang, A.; Wei, H.; Deng, J.; Qin, H. Cloud-Edge Cooperative Model and Closed-Loop Control Strategy for the Price Response of Large-Scale Air Conditioners Considering Data Packet Dropouts. *IEEE Trans. Smart Grid* **2020**, *11*, 4201–4211. [CrossRef]
4. Cui, L.; Qu, Y.; Gao, L.; Xie, G.; Yu, S. Detecting false data attacks using machine learning techniques in smart grid: A survey. *J. Netw. Comput. Appl.* **2020**, *170*, 102808. [CrossRef]
5. Hussain, H.M.; Narayanan, A.; Nardelli, P.H.J.; Yang, Y. What is energy internet? Concepts, technologies, and future directions. *IEEE Access* **2020**, *8*, 183127–183145. [CrossRef]
6. Feng, C.; Wang, Y.; Zheng, K.; Chen, Q. Smart Meter Data-Driven Customizing Price Design for Retailers. *IEEE Trans. Smart Grid* **2020**, *11*, 2043–2054. [CrossRef]
7. Mohajeri, M.; Ghassemi, A.; Gulliver, T.A. Fast Big Data Analytics for Smart Meter Data. *IEEE Open J. Commun. Soc.* **2020**, *1*, 1864–1871. [CrossRef]
8. Zhao, S.; Li, F.; Li, H.; Lu, R.; Ren, S.; Bao, H.; Lin, J.H.; Han, S. Smart and Practical Privacy-Preserving Data Aggregation for Fog-Based Smart Grids. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 521–536. [CrossRef]
9. Alahakoon, D.; Yu, X. Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey. *IEEE Trans. Ind. Inform.* **2016**, *12*, 425–436. [CrossRef]
10. Pandey, S.R.; Tran, N.H.; Bennis, M.; Tun, Y.K.; Manzoor, A.; Hong, C.S. A Crowdsourcing Framework for On-Device Federated Learning. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3241–3256. [CrossRef]
11. Zhang, J.; Tao, D. Empowering Things with Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things. *IEEE Internet Things J.* **2021**, *8*, 7789–7817. [CrossRef]
12. Ahmad, T.; Chen, H.; Wang, J.; Guo, Y. Review of various modeling techniques for the detection of electricity theft in smart grid environment. *Renew. Sustain. Energy Rev.* **2018**, *82*, 2916–2933. [CrossRef]
13. Xia, X.; Xiao, Y.; Liang, W. SAI: A Suspicion Assessment-Based Inspection Algorithm to Detect Malicious Users in Smart Grid. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 361–374. [CrossRef]
14. McDaniel, P.; McLaughlin, S. Security and privacy challenges in the smart grid. *IEEE Secur. Priv.* **2009**, *7*, 75–77. [CrossRef]
15. Taik, A.; Cherkaoui, S. Electrical Load Forecasting Using Edge Computing and Federated Learning. In Proceedings of the 2020–2020 IEEE International Conference on Communications (ICC), Online, 7–11 June 2020; Volume 2020. [CrossRef]
16. Kang, J.; Xiong, Z.; Niyato, D.; Xie, S.; Zhang, J. Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory. *IEEE Internet Things J.* **2019**, *6*, 10700–10714. [CrossRef]
17. Su, Z.; Wang, Y.; Luan, T.H.; Zhang, N.; Li, F.; Chen, T.; Cao, H. Secure and Efficient Federated Learning for Smart Grid With Edge-Cloud Collaboration. *IEEE Trans. Ind. Inform.* **2021**, *18*, 1333–1344. [CrossRef]
18. Chen, Y.; Qin, X.; Wang, J.; Yu, C.; Gao, W. FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare. *IEEE Intell. Syst.* **2020**, *35*, 83–93. [CrossRef]
19. Wen, M.; Xie, R.; Lu, K.; Wang, L.; Zhang, K. FedDetect: A Novel Privacy-Preserving Federated Learning Framework for Energy Theft Detection in Smart Grid. *IEEE Internet Things J.* **2022**, *9*, 6069–6080. [CrossRef]
20. Zheng, Z.; Yang, Y.; Niu, X.; Dai, H.N.; Zhou, Y. Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1606–1615. [CrossRef]
21. Punmiya, R.; Choe, S. Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Trans. Smart Grid* **2019**, *10*, 2326–2329. [CrossRef]
22. Li, W.; Logenthiran, T.; Phan, V.T.; Woo, W.L. A novel smart energy theft system (SETS) for IoT-based smart home. *IEEE Internet Things J.* **2019**, *6*, 5531–5539. [CrossRef]
23. Ismail, M.; Shaaban, M.F.; Naidu, M.; Serpedin, E. Deep Learning Detection of Electricity Theft Cyber-Attacks in Renewable Distributed Generation. *IEEE Trans. Smart Grid* **2020**, *11*, 3428–3437. [CrossRef]
24. Hoenkamp, R.; Huitema, G.; Moor, A.D. The neglected consumer: The case of the smart meter rollout in the Netherlands. *Renew. Energy Law Policy Rev.* **2011**, *2*, 269–282. [CrossRef]
25. Yao, D.; Wen, M.; Liang, X.; Fu, Z.; Zhang, K.; Yang, B. Energy Theft Detection with Energy Privacy Preservation in the Smart Grid. *IEEE Internet Things J.* **2019**, *6*, 7659–7669. [CrossRef]
26. Ibrahem, M.I.; Nabil, M.; Fouda, M.M.; Mahmoud, M.M.E.A.; Alasmary, W.; Alsolami, F. Efficient Privacy-Preserving Electricity Theft Detection with Dynamic Billing and Load Monitoring for AMI Networks. *IEEE Internet Things J.* **2021**, *8*, 1243–1258. [CrossRef]
27. Nabil, M.; Ismail, M.; Mahmoud, M.M.E.A.; Alasmary, W.; Serpedin, E. PPETD: Privacy-Preserving Electricity Theft Detection Scheme with Load Monitoring and Billing for AMI Networks. *IEEE Access* **2019**, *7*, 96334–96348. [CrossRef]
28. Hao, M.; Li, H.; Luo, X.; Xu, G.; Yang, H.; Liu, S. Efficient and Privacy-Enhanced Federated Learning for Industrial Artificial Intelligence. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6532–6542. [CrossRef]
29. Li, S.; Cheng, Y.; Liu, Y.; Wang, W.; Chen, T. Abnormal Client Behavior Detection in Federated Learning. *arXiv* **2019**, arXiv:1910.09933.

30. Sater, R.A.; Hamza, A.B. A Federated Learning Approach to Anomaly Detection in Smart Buildings. *ACM Trans. Internet Things* **2020**, *2*, 1–23. [CrossRef]
31. Schneble, W.; Thamilarasu, G. Attack detection using federated learning in medical cyber-physical systems. International Conference on Computer Communications and Networks. In Proceedings of the 28th International Conference on Computer Communications and Networks (icccn), Valencia, Spain, 29 July–1 August 2019; pp. 1–8.
32. Liu, Y.; Garg, S.; Nie, J.; Zhang, Y.; Xiong, Z.; Kang, J.; Hossain, M.S. Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach. *IEEE Internet Things J.* **2021**, *8*, 6348–6358. [CrossRef]
33. Li, B.; Wu, Y.; Song, J.; Lu, R.; Li, T.; Zhao, L. DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5615–5624. [CrossRef]
34. Liu, H.; Zhang, X.; Shen, X.; Sun, H. A Federated Learning Framework for Smart Grids: Securing Power Traces in Collaborative Learning. *arXiv* **2019**, arXiv:2103.11870.
35. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
36. Cover, T.M.; Hart, P.E. Nearest Neighbor Pattern Classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
37. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
38. Beutel, D.J.; Topal, T.; Mathur, A.; Qiu, X.; Parcollet, T.; de Gusmão, P.P.; Lane, N.D. Flower: A Friendly Federated Learning Research Framework. *arXiv* **2020**, arXiv:2007.14390. [CrossRef]
39. Mothukuri, V.; Khare, P.; Parizi, R.M.; Pouriyeh, S.; Dehghantanha, A.; Srivastava, G. Federated-Learning-Based Anomaly Detection for IoT Security Attacks. *IEEE Internet Things J.* **2022**, *9*, 2545–2554. [CrossRef]