Proceedings of the 43rd Chinese Control Conference
July 28-31, 2024, Kunming, China

Privacy-preserving decentralised federated learning for short-term load forecasting

Yushi Wang[1], Yang Lu[1], Zhen Dong[2] and Yi Dong[3*]
1. Department of Computer Science, InfoLab21, Lancaster, UK
2. SEEEx TECH Co. Ltd, China
3. Department of Electronics and Computer Science, The University of Southampton, UK
E-mail: y.wang216, y.lu44@lancaster.ac.uk, z.dong@seeextech.com, yi.dong@soton.ac.uk

Abstract: With the increasing complexity of short-term load forecasting in the power system, exacerbated by the integration of renewable energy sources and the influx of data from smart meters, traditional centralized load forecasting methods are grappling with challenges in scalability, data privacy, and security. The adoption of decentralised federated learning frameworks enables peer-to-peer load forecasting but increases attack surface and raises new privacy concerns, particularly the threat of inference attacks from collaboration of neighboring devices during model training. To address these challenges, we have integrated Shamir's secret sharing scheme within the decentralised federated learning framework, ensuring the preservation of privacy during the learning process. This method effectively safeguards individual user data without compromising the accuracy or efficiency of the forecasting models. Experiments based on real-world load datasets validate the effectiveness of the proposed algorithm.

Key Words: Short-term load forecasting, federated learning, privacy preservation, decentralised learning

1 Introduction

The integration of renewable energy sources and more data from smart meters have significantly increased the complexity of electrical load forecasting, especially for the Short-term Load Forecasting(STLF). STLF is crucial for the effective operation and planning of electricity markets [1, 2]. Traditional centralised load forecasting methods are facing substantial challenges related to scalability, data privacy, and security, particularly as the volume of data and the number of connected devices continues to surge. Moreover, centralised data storage and processing are becoming increasingly untenable due to the growing public and regulatory demand for data privacy and system security.

Tracing back to eight years ago, the concept of Federated Learning(FL) has emerged as a promising paradigm, allowing multiple stakeholders to collaboratively train a shared model while keeping their training data local [3–6]. Start from 2015, Konevcny *et. al* introduced Federated Optimization as a new scenario for distributed machine learning optimization, developing a novel algorithm that addresses the challenge of efficiently communicating across a vast number of devices with unevenly distributed data, showing promising experimental results and setting a direction for future research in this field [3]. Then, McMahan *et. al* advocated FL, which leaves training data on mobile devices and learns a shared model by aggregating locally-computed updates [4]. However, FL typically assumes the presence of a central server, which poses a potential bottleneck and a single-point failure, both in terms of performance and privacy.

To address this gap, several researchers discovered different decentralised approaches that eliminate the central server, allowing for a fully peer-to-peer FL environment. While decentralisation reduces the risk of centralised data breaches, the transition to a decentralised framework introduces new challenges, notably in the realm of privacy. More specifically, without coordination and orchestration by central server of data sharing, nodes have to share models with multiple neighboring nodes, which increases attack surface and may suffer from inference attacks from collaboration of neighboring attackers. To tackle the privacy issues, several privacy preserving methods are proposed. For example, Wang *et. al* proposed electricity consumption prediction models based on peer-to-peer FL, which enhance prediction accuracy while protecting user privacy by clustering households for personalized model training and analyzing Deep Leakage from Gradients attacks, demonstrating significant scalability and robustness against missing data [7]. Dong *et. al* introduced a Markovian Switching-based distributed training framework that effectively counters gradient leakage and poisoning attacks through the use of a Secure-Aggregation algorithm and Distributed Markovian Switching topology, achieving load forecasting with reduced communication complexity and high accuracy while protecting data privacy [8]. Nevertheless, the above distributed solutions cannot guarantee accuracy compare to FL frameworks.

The contributions of the paper are threefold:

1) This paper develops the first consensus-based framework that supports FL over fully decentralised load forecasting. That is, during the FL-enabled load forecasting process, the communications between the users are fully peer-to-peer (each user can only communicate with its neighbours) without a central server.
2) This paper investigates the privacy issue of decentralised load forecasting. The technique of Shamir's secret sharing is integrated with the consensus framework to facilitate privacy preservation in load forecasting.
3) This paper conducts comprehensive experiments over real-world load forecasting datasets. A publicly accessible repository of our method with all source code, datasets, and real-world experiments is provided.

To this end, we explore the integration of Shamir's secret sharing scheme within the FL framework in this paper, ensuring that individual user data are not exposed during the consensus-based learning process. This method guarantees

* corresponding author.

that privacy is maintained without compromising the accuracy or efficiency of the forecasting models.

2 Related Works

Current research in STLF adopts centralized FL frameworks such as Fed-SGD [9, 10] and Fed-Avg [11, 12]. One research direction is how to improve model performance and prediction accuracy by adjusting the number of users and training iterations [11]. Another mainly focuses on the application of different clustering techniques, such as k-means [9] and clustering based on socio-economic factors [12, 13]. These studies examine the impact of various clustering methods and data grouping on the outcomes.

Compared with traditional distributed machine learning where users directly upload their data to a central server, centralized FL enhances privacy by enabling collaborative training global model among nodes without the need to directly share their private raw data. In this approach, each edge node trains its local model with local dataset, and only the local parameters are shared across multiple clients and form the global model on a central server, thus preserving the local data from being shared with other nodes. However, while FL provides a certain level of privacy protection, it does not entirely guarantee data privacy. For example, attackers may infer original data by analyzing shared gradients, model weights, or update information, posing a threat to privacy [14]. In STLF, Zhao et al.[15] design FL with Differential Privacy (DP) to enhance privacy protection. Specifically, they introduce additive noise into the training process of the Fed-Avg model to obfuscate data trend information. Based on this, DP is widely used in various FL in load prediction, e.g. FedSGD [16], clustered variant FL [17]. However, the addition of interfering noise to protect privacy reduces the accuracy of the model's predictions.

For distributed system, many research [18–20] proposed to apply STLF to discrete systems to handle such massive and high dimensional load data increasing the timeliness of STLF. Distributed systems, by dispersing data across multiple nodes, theoretically offer potential advantages in improving computational efficiency, enhancing system reliability and fault tolerance, and bolstering security and privacy protection. This approach reduces the risk of centralized data breaches due to the distributed storage of data. Specifically, in STLF applications, timely and accurate predictions are crucial for operational efficiency and system stability. However, in practical applications, without the implementation of appropriate security and privacy protection measures, direct peer-to-peer communication between nodes may leave the system vulnerable to attacks by malicious nodes. These nodes can disguise themselves as normal ones to receive or send information by creating one or several fake identities. In particular, when there is cooperation between malicious attackers, it is possible to reconstruct rare information from the network. Therefore, while distributed systems could enhance privacy protection due to their design, it also requires careful consideration and implementation of effective privacy protection mechanisms in system design.

In this paper, we consider the privacy presevering of FL for load prediction in the complete absence of central node coordination. In particular, a decentralised network model is privacy-protected using consensus algorithm and Shamir secret sharing [21]. The method is effective for convergence and encryption in problems such as regression analysis. We apply the distributed privacy preserving model proposed in [21] to STLF application. Using real-world load data, with a particular focus on innovative privacy-preserving mechanisms and scalability of distributed models.

3 Problem Statement and Preliminaries

Consider a network of N devices, denoted by $\mathcal{V} \triangleq \{1, \cdots, N\}$. The devices are connected by a communication graph $\mathcal{G}$. Assume that $\mathcal{G}$ is time-varying, undirected and connected. Each device can measure its past power consumption. The N devices aim to collectively use their past power consumption profiles to learn a load pattern for a future time instant [22]. Nonlinear mapping models utilize past data for time series forecasting, i.e., the model only forecasts one hour in advance using electricity consumption data from the previous p hours. The inputs to the time series model are the consumption values for the previous p time steps and the outputs are the consumption values for time t in user i:

$$Y_i^{(t)} = f(\sum_{l=1}^{p} \phi_t Y_i^{(t-l)}, z) + \varepsilon^{(t)} \tag{1}$$

where $Y_i^{(t)}$ is the prediction value at time t, ϕ_t is the influence parameters of the model, z is other variable that are correlated to $Y_i^{(t)}$, k is the set time step and $\varepsilon^{(t)}$ represents the error term. The learning problem is formulated as follows

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} (Y_i^{(t)} - f(\sum_{l=1}^{p} \phi_t Y_i^{(t-l)}, z))^2 \tag{2}$$

The power consumption profiles are confidential and cannot be disclosed to untrustworthy devices. This paper aims to develop a novel framework such that the devices can implement load forecasting subject to the underlying communication topology while preserving privacy of individual devices' power consumption profiles.

3.1 Attacker Model

Assuming attackers are semi-honest, meaning they adhere to all protocol rules without injecting, tampering with, or in any way compromising the integrity of exchanged data. However, they attempt to record and analyze exchanged private data [23]. Additionally, attackers can collaborate without violating the protocol, pooling their knowledge and data. This attacker model has been widely applied in various applications, such as privacy-preserving linear programming, distributed optimization, dataset processing, and consensus [24–27]. We assume that the communication links between users are secure.

4 Algorithm Design

This section provides the proposed privacy-preserving decentralised FL algorithm for STLF. The overall implementation framework is first introduced. Then, details of consensus design and privacy design are illustrated.

4.1 Implementation Framework

To protect privacy, Shamir's Secret Sharing algorithm [28] is used to aggregate users private model data without disclosing personal data. In our problem description the model for

Algorithm 1: Privacy-preserving algorithm design

```
for t = 1; t ≤ T; t = t + 1 do
    foreach i ∈ V do
        User i trains θ_i^(t);
        User i generates shares for all of its neighbors;
        User i encrypts the shares by the corresponding
          neighbors' keys, and sends encrypted data to its
          neighbors;
    end
    foreach i ∈ V do
        User i computes s_i^(t)(0) and sends it to all of its
          neighbors;
    end
    for k = 0; k < K; k = k + 1 do
        foreach i ∈ V do
            User i updates its state;
        end
    end
    foreach i ∈ V do
        User i transforms the state back to signed real
          number; User i updates its local model.
    end
end
```

local load data is represented as single-user residential level and aggregated data is represented as substation level model prediction. Specifically, for each user's secret s, it is divided into n shares using Shamir secret sharing and each user assigns one unique share of the secret to each neighbor. This algorithm ensures that the original information cannot be reconstructed as long as the number of colluding attackers is less than the quantity of secret shares.That is, as long as a node has a neighbor who is a benign user, the node's secret will not be compromised. For the reconstruction in each user, collecting the shares from neighbor users to reconstruct the shared secret. Within the decentralised learning framework, Metropolis-Hastings algorithm facilitates consensus for model update without a central authority, by employing secure communication links between users, thus ensuring the correctness of the aggregated model.

4.2 Consensus Design

To implement the basis for decentralised model aggregation, the states of all participants (i.e., model parameters θ) are converged to the average of all initial states through an iterative process. For each round t of model aggregation, the state of each participant is updated using a weighted adjacency matrix $A(t)$. The update rule can be simplified as

$$\theta_i^{(t+1)} = \sum_{j \in N_i(t)} a_{ij}^{(t)} \theta_j^{(t)} \tag{3}$$

where $\theta_i^{(t)}$ is the state of participant i at time t, $N_i^{(t)}$ is the set of i neighbours at time t, and a_{ij} is the weight in the weighted adjacency matrix indicating the strength of the connection between participants. The weighted matrix is constructed using the Metropolis-Hastings method to ensure that the matrix is doubly stochastic

$$a_{ij}^{(t)} = \begin{cases} \frac{1}{\max(|N_i^{(t)}|,|N_j^{(t)}|)+1} & \text{if } j \in N_i^{(t)} \\ 1 - \sum_{j \in N_i^{(t)}} \frac{1}{\max(|N_i^{(t)}|,|N_j^{(t)}|)+1} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $|N_i^{(t)}|$ is the size of the set, i.e. the number of neighbours of node i. For all i and j, it holds $\sum_j a_{ij} = \sum_i a_{ij} = 1$. The eigenvector ν and the corresponding eigenvalue λ for a matrix $A^{(t)} = [a_{ij}^{(t)}] \in \mathbb{R}^{N \times N}$ satisfy

$$A\nu = \lambda\nu \tag{5}$$

Due to the nature of the double stochastic matrix:

$$\begin{cases} \lambda_1 = 1 & \text{if } n = 1 \\ |\lambda_n| < 1 & \text{otherwise} \end{cases} \tag{6}$$

For iteration $k \to \infty$:

$$\begin{cases} \lambda_1^k = 1 & \text{if } n = 1 \\ \lambda_n^k \to 0 & \text{otherwise} \end{cases} \tag{7}$$

After a sufficient number of iterations, only the influence of the eigenvector corresponding to λ_1 remains, while the influence of the other eigenvectors vanishes. The matrix $(A^{(t)})^k$ converges to a particular steady-state distribution.

4.3 Privacy Design

Shamir's Secret Sharing (SSS) utilises Lagrange interpolation, which allows a secret to be split into multiple parts. Only when a sufficient number of secret shares have been collected can the original secret be recovered. For each receiving node (neighbouring user in this model), the formula for generating share $\sum_{i=1}^{m} S_i^{j(t)} = SSS_{alg}(x)$ is as follows

$$SSS_{alg}(x) = s + c_1 x + \cdots + c_{m-1} x^{m-1} \mod p \tag{8}$$

where the original sharing secret $\sum_{i \in V} s_i^{(t)}(0) = \sum_{i \in V} 10^{\sigma}\theta(t)$ in this model, p is a sufficiently large prime number, c_m is random coefficients and within a limited field of size p.Each user i send $S_i^{j(t)}$ to its neighbor and receive $\sum_{j \in |N_i^{(t)}|} S_j^{i(t)}$ from its neighbor. The reconstructed new state $s_i^{(t)}(0)$ shown as:

$$s_i^{(t)}(0) = SSS_{alg}(0) = \sum_{i=1}^{m} \sum S_j^{i(t)} \cdot l_i(0) \mod p \tag{9}$$

where $l_i(0)$ is Lagrange basis polynomials

$$l_i(0) = \prod_{1 \le j \le m, j \ne i} \frac{-x_j}{x_i - x_j} \mod p \tag{10}$$

The process of rebuilding secrets requires at least m sharing.

5 Simulation

5.1 Data Sources

Our case study focuses on the Smart Metering Electricity Customer Behavior Trials, as referenced by [29]. This study encompasses more than 5,000 participants from Irish homes and businesses throughout 2009 and 2010. Their power usage was recorded in half-hour increments by smart meters. The most extensive period of data collection stretched from January 1, 2009, to June 30, 2010. Following the processes of data cleansing and grouping, we chose 30 households to represent an ideal energy community. This selection was narrowed down from 30 homes that consistently clustered together across various methods and achieved a high rating.

In this case study, we employ a conventional non-Independently Identically Distributed (non-IID) dataset involving a considerable number of agents. Directly implementing FL on the unprocessed dataset is neither feasible nor effective. Consequently, we utilized the K-means algorithm to segment the dataset into smaller clusters as outlined by [30]. The clustering outcomes are depicted in Figure 1.

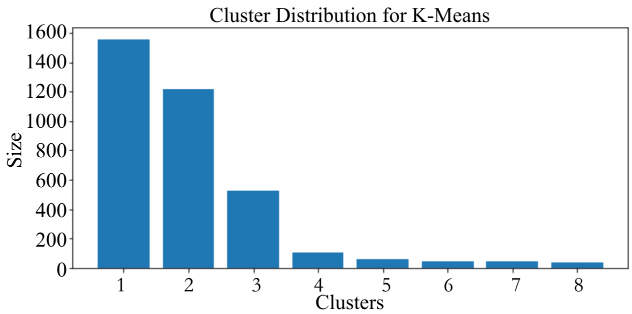

Fig. 1: The clustering result with K-means.

5.2 Experiment Setup

Hardware Environment The simulation environment is as follows. On the hardware side, the simulation is performed on a Lenovo ThinkPad laptop computer with Intel® Core™ i7-1360P CPU at 2200 MHz. On the software side, the simulation is performed on MATLAB R2021b.

Network for local training For load forecasting algorithm, we use time series algorithm. Time series algorithms applied to describe the time and load values were initially proposed by Box et al [31], and are widely used in deep learning prediction models for big data such as CNN [32], LSTM [33], and DBN [18, 34]. The structure begin with a 48 dimensional feature input layer. The core of the network consists of a sequence of fully connected layers paired with ReLU activation functions. The first combination includes a fully connected layer with 100 neurons, followed by a ReLU layer, introducing non-linearity and enabling complex pattern recognition. The second set repeat this module, where the fully connected layer contains 50 neurons, followed by another ReLU layer. A final fully connected layer with a single neuron compiles the features into a singular output.

5.3 Consensus Results

Between the decentralised network nodes, the locally trained network model layer parameters is recorded as information exchange. The local parameter $\theta_i^{(t)}$ of node i as:

$$\theta_i^{(t)} = \sum_{l=1}^{L} \{W_i^{[l]}, b_i^{[l]}\} \tag{11}$$

L is the total number of layers in the network. For global model aggregation FL, theoretic global model is recorded as θ and practical global model with private exchange in Alg. is recorded as $\tilde{\theta}_i^{(t)}$.

Time-Varying dense topology structure By convex combination from permutation matrices, we generate a set of dense doubly stochastic matrices as time varying simulation of node connectivity, and apply the above STLF data to validate the convergence of the algorithm. Fig 2 shows the trajectories of $\max_{i\in V}\left|\tilde{\theta}_i^{(t)} - \theta^{(t)}\right|$ for $t = 1, ..., 6$. **Case**

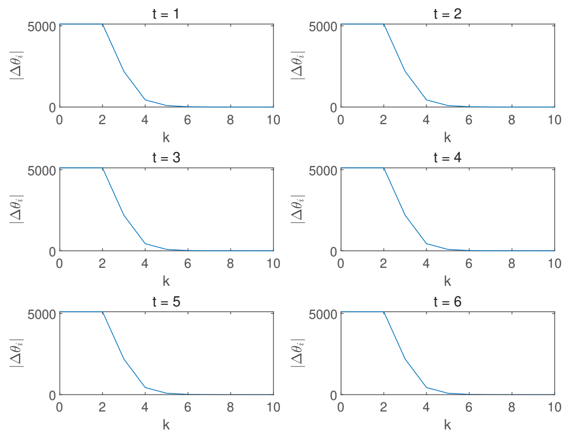

Fig. 2: Training performance on DNN models

IEEE 37 Bus test system IEEE 37 bus system[35], represents a typical medium voltage suburban distribution system. The connection situation is shown in Fig3. In this scenario,

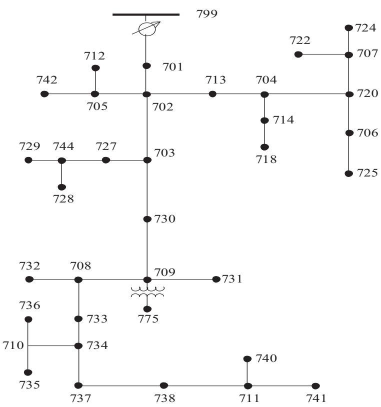

Fig. 3: connection situation of IEEE 37 Bus

each bus acts as a user, conducting local data processing, model training and secret exchange. The system encompasses both residential and commercial loads, providing a diverse set of load connection for FL model predictions. From Fig 4, case IEEE 37 bus system exhibits a relatively slow model convergence, requiring about 2000 training rounds to converge, due to its sparse communication topology. In this connection structure, only a few nodes have direct communication links, with most elements in the weighted adjacency matrix $A^{(t)}$ being zero. Regarding 4.2, the convergences of model depends on the spectral radius of $A^{(t)}$. Specifically, $(A^{(t)})^k$ describes the effect of information propagation be-

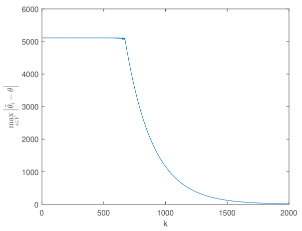

Fig. 4: Absolute difference trajectories in Case IEEE 37 Bus

tween nodes after k rounds of communication. If $A^{(t)}$ converges to a stable state for a sufficiently large k, it implies that the global model converge. The results show slow diffusion of information in sparse network,which leads to slower updating of the global model. In contrast, the speed of information propagation and model convergence is faster in dense communication topology structure.

5.4 Correctness and Time Overhead

The canonical training algorithms use FedAvg [4]. In the centralized case, the central server receives the weights and it aggregates and averages them to reach new consensus. In our algorithm, each node decrypts and averages the received secret share $s_i^{(t)}$ after receiving it. It can tell from Fig. 5 that

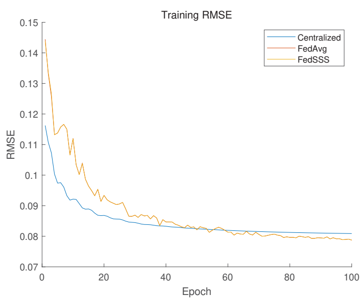

Fig. 5: Training performance of different algorithms

the results of decentralised FL encryption using Metropolis-Hastings and Shamir's Secret Sharing are almost equal to the correctness of the model with weighted aggregation by a central server. This shows that in terms of the correctness of the predicted results, our model shows exhibits the same high rate of correctness with the baseline FedAvg. In contrast to FedAvg, our model can be applied to a wide range of topologies without a central node, with the additional protection of model privacy. Compared to non-FL models, FedAvg and FedSSS models have difficulty learning common features during initial training due to distributed data heterogeneity. The generalization ability gradually increases in the later stages of training, but this is not the main part of our discussion.

The total number of parameters of neural network in the three fully connected layers in 5.2 is $n = (48+1)*100+(100+1)*50+(50+1)*1 = 10001$. We simplify by reducing number of neurons in each fully connected layer to test the operation of Alg. with different number of parameters. From the results in table 1, it can tell that the algorithm is still efficient in high level training network and with slightly difference from simplified network.

In another dimension, we compared the running time per node per round for different number of iterations. The number of iterations taken depends on densy of topology, for example, in Case IEEE 37, K=2000 applied. It shows that the algorithm is suitable for simulated topological and real power systems. Therefore, it is computationally efficient for sparse networks that may require a large number of consensus iterations. We also recorded the training time of baseline FedAVG application on our load prediction data, which was approximately 316.35s. The time for encryption and consensus is almost negligible compared to the training process.

Table 1: Breakdowns of computational overhead under different communication topologies and layer neurons number, where n is the number of model features, K is the number of consensus iterations, $t1$ represents the time of constructing the weighted adjacency matrix, $t2$ represents the time for generating and sharing the secret shares, and $t3$ represents the time for reaching the consensus process. All computed per user per round.

n	t	$K=10$	$K=150$	$K=2000$
2501	$t1$	0.0001	0.0002	0.0002
	$t2$	0.0005	0.0006	0.0005
	$t3$	0.0009	0.0090	0.1082
5421	$t1$	0.0002	0.0002	0.0002
	$t2$	0.0008	0.0010	0.0009
	$t3$	0.0026	0.0186	0.1823
6381	$t1$	0.0002	0.0002	0.0002
	$t2$	0.0010	0.0014	0.0013
	$t3$	0.0028	0.0316	0.2780
10001	$t1$	0.0002	0.0002	0.0002
	$t2$	0.0014	0.0022	0.0017
	$t3$	0.0051	0.0426	0.3931

6 Conclusion

This paper tackles the challenges of decentralised learning and data privacy in STLF by incorporating Shamir's secret sharing scheme into the FL framework. This innovative approach safeguards user data privacy during the forecasting process without sacrificing model accuracy or efficiency. Validated by experiments on real-world datasets, our findings highlight the potential of integrating cryptographic techniques with machine learning to enhance privacy in distributed systems. The effectiveness of our proposed solution not only addresses current concerns within the power system domain but also sets a precedent for future advancements in privacy-preserving technologies across various sectors. Moving forward, this work lays the foundation for exploring more sophisticated cryptographic methods and optimizing FL models to further balance privacy with analytical performance, promising a more secure and efficient data-driven future.

References

[1] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, "Designing microgrid energy markets: A case study: The brooklyn microgrid," *Applied energy*, vol. 210, pp. 870–880, 2018.

[2] A. Heydari, M. M. Nezhad, E. Pirshayan, D. A. Garcia, F. Keynia, and L. De Santoli, "Short-term electricity price

and load forecasting in isolated power grids based on composite neural network and gravitational search optimization algorithm," *Applied Energy*, vol. 277, p. 115503, 2020.

[3] J. Konečnỳ, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, 2015.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[5] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[6] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[7] Y. Wang, F. Zobiri, M. A. Mustafa, J. Nightingale, and G. Deconinck, "Consumption prediction with privacy concern: Application and evaluation of federated learning," *Sustainable Energy, Grids and Networks*, vol. 38, p. 101248, 2024.

[8] Y. Dong, Y. Wang, M. Gama, M. A. Mustafa, G. Deconinck, and X. Huang, "Privacy-preserving distributed learning for residential short-term load forecasting," *IEEE Internet of Things Journal*, 2024.

[9] Y. He, F. Luo, G. Ranzi, and W. Kong, "Short-term residential load forecasting based on federated learning and load clustering," in *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2021, pp. 77–82.

[10] J. Lin, J. Ma, and J. Zhu, "Privacy-preserving household characteristic identification with federated learning method," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1088–1099, 2021.

[11] J. Li, Y. Ren, S. Fang, K. Li, and M. Sun, "Federated learning-based ultra-short term load forecasting in power internet of things," in *2020 IEEE International Conference on Energy Internet (ICEI)*. IEEE, 2020, pp. 63–68.

[12] M. Savi and F. Olivadese, "Short-term energy consumption forecasting at the edge: A federated learning approach," *IEEE Access*, vol. 9, pp. 95 949–95 969, 2021.

[13] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in *ICC 2020-2020 IEEE international conference on communications (ICC)*. IEEE, 2020, pp. 1–6.

[14] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[15] Y. Zhao, W. Xiao, L. Shuai, J. Luo, S. Yao, and M. Zhang, "A differential privacy-enhanced federated learning method for short-term household load forecasting in smart grid," in *2021 7th International Conference on Computer and Communications (ICCC)*. IEEE, 2021, pp. 1399–1404.

[16] J. D. Fernández, S. P. Menci, C. M. Lee, A. Rieger, and G. Fridgen, "Privacy-preserving federated learning for residential short-term load forecasting," *Applied energy*, vol. 326, p. 119915, 2022.

[17] V. Tudor, M. Almgren, and M. Papatriantafilou, "Employing private data in ami applications: Short term load forecasting using differentially private aggregated data," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE, 2016, pp. 404–413.

[18] Y. Dong, Z. Dong, T. Zhao, Z. Li, and Z. Ding, "Short term load forecasting with markovian switching distributed deep belief networks," *International Journal of Electrical Power & Energy Systems*, vol. 130, p. 106942, 2021.

[19] S. Li, L. Goel, and P. Wang, "An ensemble approach for short-term load forecasting by extreme learning machine," *Applied Energy*, vol. 170, pp. 22–29, 2016.

[20] Q. Huang, J. Li, and M. Zhu, "An improved convolutional neural network with load range discretization for probabilistic load forecasting," *Energy*, vol. 203, p. 117902, 2020.

[21] Y. Lu, Z. Yu, and N. Suri, "Privacy-preserving decentralized federated learning over time-varying communication graph," *ACM Transactions on Privacy and Security*, vol. 26, no. 3, pp. 1–39, 2023.

[22] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016, pp. 7046–7051.

[23] C. Hazay and Y. Lindell, *Efficient secure two-party protocols: Techniques and constructions*. Springer Science & Business Media, 2010.

[24] J. Dreier and F. Kerschbaum, "Practical privacy-preserving multiparty linear programming based on problem transformation," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 2011, pp. 916–924.

[25] Y. Lu and M. Zhu, "Privacy preserving distributed optimization using homomorphic encryption," *Automatica*, vol. 96, no. 10, pp. 314–325, October 2018.

[26] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 1–19.

[27] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, 2012, pp. 81–90.

[28] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[29] I. Commission for Energy Regulation (CER), "CER smart metering project - gas customer behaviour Trial,2009-2010," 2012.

[30] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv prepr. arXiv:2002,10619*, 2020.

[31] M. Geurts, "Book review: time series analysis: forecasting and control," 1977.

[32] L. Yin and J. Xie, "Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems," *Applied Energy*, vol. 283, p. 116328, 2021.

[33] J. Q. Wang, Y. Du, and J. Wang, "Lstm based long-term energy consumption prediction with periodicity," *Energy*, vol. 197, p. 117197, 2020.

[34] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, "Deep belief network based electricity load forecasting: An analysis of macedonian case," *Energy*, vol. 115, pp. 1688–1700, 2016.

[35] W. H. Kersting, "Radial distribution test feeders," in *2001 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 01CH37194)*, vol. 2. IEEE, 2001, pp. 908–912.