Energy Disaggregation with Federated and Transfer Learning

Qi Li, Jin Ye, Wenzhan Song
Center for Cyber-Physical Systems
University of Georgia
Athens, Georgia, 30602
Email: {qi.li, jin.ye, wsong}@uga.edu

Zion Tse
Department of Electrical Engineering
University of York
Email: zion.tse@york.ac.uk

***Abstract*—Energy disaggregation is the decomposition of a home or facility's total energy consumption into individual equipment-level readings. Currently, non-intrusive load monitoring (NILM) is performed by constructing a regression model for different equipment using either data-driven or statistical methods, which requires the collection of equipment consumption data to train the model. However, this could potentially compromise privacy since users' activity could be inferred by an adversary from the equipment working pattern. This paper proposes a new framework called Distributed Federated Non-Intrusive Load Monitoring (DFNILM) based on federated learning and transfer learning, which can achieve optimal user privacy while preserving good performance. Besides, the transferability of our framework between different dataset domains is explored using transfer learning. We implement our framework and conduct extensive experiments to evaluate the performance of two public NILM datasets. The results show that our framework can achieve comparable performance to state-of-the-art NILM algorithms while providing strong privacy protection, which is valuable for deploying the distributed algorithm to the IoT (Internet of Things) devices in a household.**

***Index Terms*—NILM, Energy disaggregation, Federated learning, Transfer learning**

I. Introduction

In today's world dominated by digital technology, the IoT plays a prominent role in our lives. It has created an ecosystem that links different systems to give us smart performances. Among all the IoT devices, the facility and appliances in residence are particularly highly related to our daily lives, reflecting the resident's indoor activity and energy consumption. An essential step of a facility or a house energy management is to identify the individual electrical equipment's power consumption pattern and anomalies. Optimization and maintenance decisions may be timely made.

In this work, we introduce a new Distributed Federated and Transfer Learning for Non-Intrusive Load Monitoring (DFNILM) framework that self-learns the equipment usages on the fly and self-adjusts the prediction model to accommodate new equipment and behaviors. Fig 1 shows the schematic diagram of our proposed DFNILM framework. The energy disaggregation will be trained in a distributed way, and only the parameters of the model will be transferred. In this way, the residents can reap the benefits of the NILM without worrying about putting their privacy at risk. To the best of our knowledge, DFNILM is the first framework dealing with the household energy disaggregation problem using federated learning. We implement and evaluate the proposed framework on three different public NILM datasets. We demonstrate the performance of DFNILM through comprehensive experiments. For example, we compared the performance under the framework with different learning models (i.e., long short-term memory (LSTM) [1], Convolutional neural network (CNN) and seq2point [2]). In summary, our main contributions are as follows:

- We develop a new DFNILM framework for training a joint energy aggregation model, which achieves the state-of-the-art NILM performance while preserving the optimal privacy protection of the data of residents in the house.
- We perform comprehensive experiments via different public NILM datasets, which demonstrate that our framework will not degrade the performance compared with the current NILM framework.
- Transfer learning is considered in our framework, which demonstrates the transferability of the proposed framework when dealing with different dataset domains (e.g., different countries, different areas).

This paper introduced a novel distributed approach of federated and transfer learning to learn individual equipment energy consumption on the fly. Federated learning allows the learning performed at the network edge to preserve the data privacy and hopefully bandwidth as well, while transfer learning has been used to test and deploy the method in different domains. To test the performance of a disaggregation algorithm for a specific country, it is important to have access to data from that country because electricity usage varies significantly between countries; both because different countries use different sets of equipment and also because different cultures show different usage patterns.

The rest of this paper is organized as follows: Section II describes the related works. In Section III, the system design of our proposed framework are discussed, and the structure and methodology are presented. In Section IV, we describe the experiments in detail, including the datasets, environment setup, data preprocessing, and experimental evaluation results.

And we conclude this paper in Section V.

II. Related Works

First proposed by Hart [3], energy disaggregation, also called non-intrusive load monitoring (NILM), is a computational technique aiming to decompose the total energy consumption of a residence into equipment-level signal. The ultimate goals of NILM are to help residents gain a full understanding of their daily energy consumption with respect to a specific equipment and create a plan to save their energy; help energy suppliers plan and operate power system networks more efficiently [4]; and identify faulty equipment by studying the equipment working patterns. Instead of installing a smart sensor for each equipment, which is expensive, NILM aims to recover the energy consumption of each equipment by only using the mains readings. Recently, this research field has been drawing more and more attention [5]. The early works focused on extracting electricity events rather than directly decomposing the energy consumption signals [3], [6]. Combinatorial optimisation (CO) and Factorial Hidden Markov Model (FHMM) were commonly used as the baseline. As the machine learning community has grown, a variety of data-driven or machine learning methods have been applied to NILM, and they have achieved good performance, such as artificial neural networks (ANN) [7], K-nearest neighbor (KNN) [8], and committee decision mechanisms (CDM) [9]. Among those algorithms, deep learning approaches have achieved the state-of-the-art performance. Particularly, it has been shown in the literature that seq2point which is based on convolutional neural networks (CNN), achieves the best performance [2]. As we know, data and their quality are critical to the performance of data-driven methods. However, in NILM, the collection process of equipment data for training is usually time consuming, generally ranging from a minimum of one week to more than one year. To facilitate research and establish a common baseline, various public datasets are available online for researchers to test their algorithms on, such as REDD [10], BLUED [11], iAWE [12], UK-DALE [13] and BLOND [14]. These datasets have different sample frequency, time range, and locations, which increases the variety of available data. Currently, there are more than 25 datasets publicly available [5]. To get the optimal performance in NILM, those data-driven methods depend on access to a sufficient amount of raw data. In regard to model training, the most common approach is to collect both the individual equipment consumption data and the aggregated household consumption data and send this data to a central server, which is responsible for training the energy disaggregation model. However, during the data transmission, the privacy-sensitive equipment data are exposed to the risk of hacking, which is a privacy concern since equipment data could be used to infer the user's activity in their residence. In fact, there is an attack called the NILM attack [15] that is commonly used to intrude into the privacy of smart meter users.

To achieve good performance and user privacy simultaneously, the Federated Learning (FL) framework is proposed

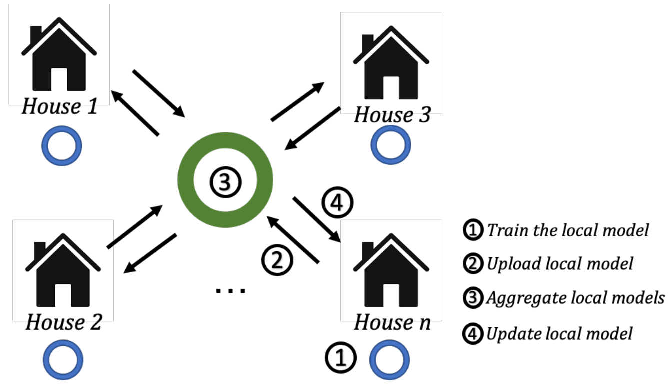

Fig. 1: Schematic diagram of the proposed DFNILM framework. In every training round, firstly the model will be trained locally at resident's house. Secondly, the local model will be uploaded to the parameter server at set time intervals. Thirdly, local models are aggregated in a parameter server. Lastly, the updated global model will be distributed to local residents' houses.

[16]–[18]. The main goal of FL is to keep datasets collected by distributed remote devices private by learning a model from remote devices and only sharing the learned model with others. In this case, a parameter server (PS) which aggregates the models from remote devices is used to maintain the global learning model, and it periodically updates the model to the remote devices based on a certain aggregation rule.

As mentioned earlier, user privacy is compromised when performing NILM. There are few studies considering applying FL in the NILM scenarios to achieve more protection of user data. The following are some works related to both NILM and FL: Kelly [19] firstly adapted three neural network architectures, long short-term memory (LSTM), denoising autoencoder (DAE), and CNN, to energy disaggregation. Compared with combinatorial optimisation or factorial hidden Markov models, all three neural nets achieved better performance. In [20], an open-source toolkit called NILMTK was designed to enable the comparison of energy disaggregation algorithms in a reproducible manner. It provides researchers with a powerful tool to evaluate their algorithms more efficiently. In [21], transfer learning was applied on their previous NILM algorithm seq2point [2]. The results demonstrated their proposed algorithm is transferable between different dataset domains. For FL, most of works are focused on improving the convergence rates of the algorithms. In [22], the convergence rate of the classical algorithm FedAvg [17] was theoretically derived for convex and non-convex loss functions respectively, while assuming the samples were iid (independent and identically distributed). In [23]–[26], the convergence rate of FedAvg with non-iid samples was studied. In [27]–[31], the improvements of FedAvg with heterogeneous and limited resources were discussed. But the combination of federated learning algorithm and household energy disaggregation has not been considered in any of the above literature.

III. System Design and Algorithms

In this section, we will introduce the design scope and overview of our proposed framework. Take the diagram in Fig 1 as an example. In recent years, more and more households are being equipped with various Internet of things (IoT) devices or smart sensors, which have access to the outside network via routers. With the rapid development of edge computing, the data generated from every home could be conveniently sent to the closest server. In our framework, we make the following reasonable assumptions:

- There is one parameter server responsible for collecting the model from the household involved.
- Every home has at least one computational unit that can handle the local energy disaggregation model's training task.
- The computational capability of the computational unit in different houses is the same.

The standard workflow of the proposed framework can be described as follows: (1) A random model is generated in the parameter server and then distributed to every local household. (2) Each local household trains the given model using its local equipment energy consumption data and mains reading data. (3) After a certain number of local model training rounds, the local model parameters are uploaded to the parameter server. (4) The parameter server performs a specific aggregate rule to generate a better model. Then the new model update is sent to the local households. Step (2), (3), (4) are repeated until the convergence condition is met.

A. Non-Intrusive Load Monitoring

The total aggregated measurement of a household at time t can be described in the following equation:

$$P(t) = \sum_{i=1}^{n} P_i(t) + e(t) \tag{1}$$

, where $P(t)$ is the aggregation of all the active power consumption of all equipment; P_i denotes the load of an individual equipment at time t that contributes to the total load; $e(t)$ represents the model noise, which often follows a Gaussian distribution with zero mean and variance σ_t (i.e., $e(t) \sim \mathcal{N}(0, \sigma_t^2)$). It is similar to a non-identifiable Blind Source Separation (BBS) problem. Adding features of the individual equipment, such as the rated power, working pattern, equipment power level, and ON-OFF state changes could overcome the identifiability problem. Based on this, some methods have been proposed, such as using equipment features [32]–[34], matrix factorization [35], [36], and clustering [37], [38]. However, with respect to the machine learning methods, by leveraging the dataset, the NILM is modeled as a supervised learning problem. If we use Y to represent the aggregated power consumption and X to represent the reading of a certain equipment, then the NILM problem turns into training a model f to represent the relationship between X and Y such that $X = f(Y)$, which can be treated as a regression problem. In this case, every equipment has its model and treats other equipment' signals as noise. Then the problem turns into extracting the specific equipment energy consumption from the noise. In [39], a denoising autoencoder was adopted. Besides, some state-of-the-art deep learning methods have been used, such as generative adversarial networks (GAN) [40], CNN [2], [41], [42], and RNN [19]. Among them, CNN achieved the best performance [2].

B. Federated learning

Since the seminal work of [16], [17], federated learning (FL) has drawn more and more attention. The most popular algorithm FedAvg [16] sheds the light on the basic idea of how federated learning works. It periodically updates the global model by aggregating local stochastic gradient descent (SGD) updates from remote clients with imbalanced and possibly non-IID datasets [43], [44]. The pseudo-code is given in Algorithm 1 [16]. From the algorithm, we can see the computation is mainly determined by three parameters: C, the fraction of remote clients that perform computation on each round; B, the local mini-batch size used for the client updates; and E, the number of training passes each remote client makes over its local dataset on each round. The goal is typically to minimize the following objective function:

$$\min_{w} F(w), where\ F(w) := \sum_{k=1}^{m} p_k F_k(w). \tag{2}$$

where m is the total number of devices; $p_k \geq 0$; $\sum_k p_k = 1$; and F_k is the local objective function for the kth device. Based on the FedAvg, various algorithms have been proposed by researchers to optimize the convergence rate or the communication cost. In this paper, we used FedAvg as our distributed algorithms because it is simple to implement and sufficient for the household dataset.

Algorithm 1 FederatedAveraging [16]. The K clients are indexed by k; B is the local mini-batch size; E is the number of local epochs; and ϵ is the learning rate.

SERVER EXECUTES:
initialize w_0
for each round t = 1,2,... **do**
  $m \leftarrow$max(C$\cdot K, 1$)
  $S_t \leftarrow$(random set of m clients)
  for each client $k \in$S$_t$ in parallel **do**
    $w_{t+1}^k \leftarrow ClientUpdate$(k, w$_t$)
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
  end for
end for
ClientUpdate(k, w)**:** //Run on client k
$\beta \leftarrow$ (split D$_k$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in B$ **do**
    $w \leftarrow w - \eta \bigtriangledown l(w; b)$
  end for
end for
return w to server =0

IV. Experiments and Evaluations

In this section, we will cover the details of the experiments including the environment setup, datasets, preprocessing, selection of hyperparameters, and so on.

A. Environment setup

We implemented our framework through Pytorch (1.3.1) and Sklearn (0.22.1) [45], [46] on an Ubuntu 16.04 server (CPU: i7-6850K, 3.60GHz; Memory: 64GB) armed with GPU (GeForce GTX 1080 Ti) for training. Specifically, the implementation of FL was realized by using the Python library called PySyft [47], which was developed by OpenMind for secure and private Deep Learning. To evaluate the performance, we mainly used the mean absolute error (MAE) as the metric, which evaluates the absolute difference between the prediction $\hat{x_t}$ and the ground truth x_t at every time point and calculates the mean value. MAE is frequently used in the evaluation of the regression problem, defined as follows:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|\hat{x_t} - x_t| \tag{3}$$

B. Datasets

As mentioned in the previous sections, there are various open-source household energy datasets available. These data were measured with different sampling frequencies in residences located in different countries. The sensors installed in these buildings recorded the active power data. In our experiments, the three following public household energy consumption datasets were used:

1) UK-DALE: The UK-DALE (U.K. Domestic Appliance-Level Electricity) contains data from five buildings in the U.K. from 2013 to 2015. The sampling intervals for mains and appliances were 1 s and 6 s, respectively [13].

2) REFIT: The REFIT data contains data from 20 buildings in England (Loughborough area) from 2013 and 2015. The sampling interval for mains and appliances is 8 s.

3) REDD: The Reference Energy Disaggregation Data Set (REDD) contains data in 6 buildings in U.S. [10] Measurements include mains with 1 s sampling period and several appliances with 3 s sampling period. The lengths of observations were between 3 and 19 days.

For easier comparisons between different datasets, the five most common appliances were considered in our experiment, which are kettle, microwave, refrigerator, dishwasher, and washing machine.

C. Data preprocessing

As the first step of the data-driven approach, data preprocessing was used to transform raw data into a more understandable format. The normalization is one of the most important steps before all the data were fed into the model, which is defined as follows:

$$\frac{x_t - \bar{x}}{\sigma} \tag{4}$$

TABLE I: Experimental distribution of the REFIT dataset. This table shows the data partition of chosen appliances in different households.

	Training	Validation	Test
Appliance	house	house	house
Kettle	3, 4, 6, 7, 8, 9	5	2
Microwave	10, 12, 19	17	4
Fridge	2, 5, 9	12	15
Dish washer	5, 7, 9, 13, 16	18	20
Washing machine	2, 5, 7, 9, 15, 16, 17	18	8

where x_t represents the data at time t, $\bar{x}$ denotes the mean value of an appliance or mains reading, and σ denotes the standard deviation of them. Before being fed into the models for training, these mean and standard deviation values should be calculated.

After exploring the three datasets, we partitioned the data into the training set and test set. Taking the REDD dataset as an example, data from buildings 1-5 comprise the training data, and data from building 6 comprise the test set. The tableII shows our training distribution of the REFIT dataset, where a set of houses act as local training nodes while the data of the other two random houses are used for validation and testing.

D. Performance evaluation of DFNILM

To evaluate our proposed method, we plan to develop comprehensive experiments under our DFNILM framework with the above three datasets. In this paper, we performed all the experiments on our DFNILM framework with Pytorch. We have implemented the FedAVG algorithms in Pytorch with the library PySyft, a Python library for secure, private machine learning. PySyft extends PyTorch, Tensorflow, and Keras with remote execution capabilities, federated learning, differential privacy, homomorphic encryption, and multi-party computation. It's noted that for performance evaluation, the NILM algorithm we adopted was seq2point [2], which achieved state-of-the-art performance. We used the REFIT dataset to implement our method and compare it with the original NILM method. Though the sampling interval for mains and appliances is 8s and there is some missing data problem because of outages in between, It has 20 houses data from 2013 and 2015 in total, which has more houses to act as local node so that it is better for us to implement a distributed algorithms than UK-DALE or REDD. To evaluate the performance, we applied the state-of-the-art algorithm Seq2Point on both the proposed and original method. Although every household in REFIT has different appliances, they have some in common. In our case, as shown in Table II, we chose kettle, microwave, fridge, dishwasher, washing machine, which are more widely used in the household. We do our training as the distribution in Table II. Take kettle as an example, for the proposed method, house 3, 4, 6, 7, 8, 9 would act as the nodes doing local training, then the local model would be aggregated in specific rounds. While for the original method, which is centralized, the data from those houses would be collected together then

TABLE II: The MAE value comparison between normal NILM framework with the proposed framework. (REFIT dataset)

	Proposed framework	normal framework
	MAE	MAE
Kettle	7.238	6.980
Microwave	14.529	12.770
Fridge	24.912	20.134
Dish washer	13.156	12.260
Washing machine	18.270	17.012

TABLE III: Transfer learning result (UK-DALE to REDD).The MAE value comparison between normal NILM framework with the proposed framework.

	Proposed framework	normal framework
	MAE	MAE
Fridge	41.475	24.212
Washing machine	31.555	27.824

do the centralized training. And house 5 and house 2 are for validating and testing, respectively.

Table II shows the MAE value comparison between the original NILM framework with our proposed framework. From the kettle to the washing machine, the performance decreased by 3.70%, 13.77%, 23.73%, 7.31%, 7.39%, which are acceptable considering our method can effectively protect user privacy. Besides, the more complex the working pattern of the appliance is, the more performance degrades. Among all tested appliances, the kettle got the least performance decreasing. Fig 2 shows a snippet of the disaggregation result of the fridge.

E. Transferability evaluation of DFNILM

To evaluate the transferability between our proposed method's different data domains, we transferred the model trained on the UK-DALE dataset to the REDD dataset and compared the performance to the traditional NILM framework. Table III shows the MAE value of the fridge and washing machine between the original method and the proposed method. From the table, we could see that for the transfer learning result of the washing machine, the performance decreased by 13.41%, which is acceptable. However, the result of the fridge is not so good, whose performance decreased by 71.30%. It demonstrates a big difference in the energy consumption data from different domains (UK-DALE is from the UK while REDD is from the USA), and vanilla transfer learning by directly transferring the trained model is not enough for good performance.

V. Conclusion and Future Plans

In the paper, we proposed a new framework, DFNILM, which combined FL with state-of-the-art NILM algorithms to achieve both good performances and increased privacy for the residents. Besides, the framework's transferability was also considered. The experiment result shows that our framework will not affect the model's transferability too much, and it could be improved as more training data involved. The experiment results show that our framework can achieve comparable performance to state-of-the-art NILM algorithms while providing strong privacy protection, which is valuable for deploying the distributed algorithm to the IoT devices in a household.

In our future plan, we will try to contribute to the open-source community through merging our work to the nilmtk toolkit [20], which is very useful but lacks the support of the distributed algorithm so far. Secondly, we plan to set up real experimental scenes in different residences, collecting more granular and complicated data than the public dataset used in this paper. The smart sensors and IoT devices such as Raspberry Pi will be deployed to implement the framework. Finally, the communication cost and the efficiency of the model will be considered and analyzed in details.

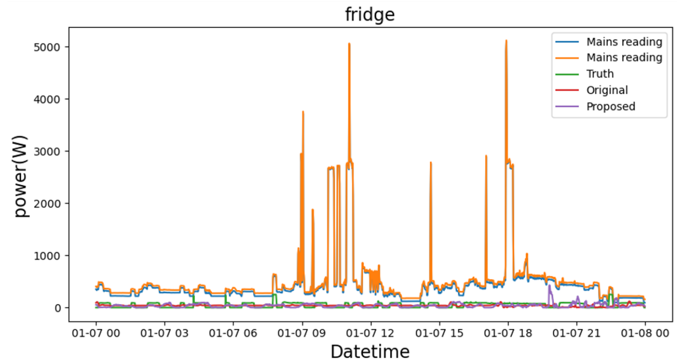

Fig. 2: A snippet of the disaggregation MAE result of both proposed method and original method of the fridge. The date is from 2008-01-07 to 2008-01-08.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for nonintrusive load monitoring," *arXiv preprint arXiv:1612.09106*, 2016.

[3] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[4] R. Arghandeh and Y. Zhou, *Big data application in power systems*. Elsevier, 2017.

[5] M. Sun, F. M. Nakoty, Q. Liu, X. Liu, Y. Yang, and T. Shen, "Non-intrusive load monitoring system framework and load disaggregation algorithms: A survey," in *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*. IEEE, 2019, pp. 284–288.

[6] S. Gupta, M. S. Reynolds, and S. N. Patel, "Electrisense: single-point sensing using emi for electrical event detection and classification in the home," in *Proceedings of the 12th ACM international conference on Ubiquitous computing*, 2010, pp. 139–148.

[7] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Information sciences*, vol. 295, pp. 395–406, 2015.

[8] S. Kang and J. W. Yoon, "Classification of home appliance by using probabilistic knn with sensor data," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–5.

[9] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, 2011.

[10] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, vol. 25, no. Citeseer, 2011, pp. 59–62.

[11] A. Filip *et al.*, "Blued: A fully labeled public dataset for event-based nonintrusive load monitoring research," in *2nd Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2011, p. 2012.

[12] N. Batra, M. Gulati, A. Singh, and M. B. Srivastava, "It's different: Insights into home energy consumption in india," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013, pp. 1–8.

[13] J. Kelly and W. Knottenbelt, "The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes," *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.

[14] T. Kriechbaumer and H.-A. Jacobsen, "Blond, a building-level office environment dataset of typical electrical appliances," *Scientific data*, vol. 5, p. 180048, 2018.

[15] D. Li and S. Dick, "Residential household non-intrusive load monitoring via graph-based multi-label semi-supervised learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4615–4627, 2018.

[16] 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.

[17] 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.

[18] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2019.

[19] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, 2015, pp. 55–64.

[20] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava, "Nilmtk: an open source toolkit for non-intrusive load monitoring," in *Proceedings of the 5th international conference on Future energy systems*, 2014, pp. 265–276.

[21] M. D'Incecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1419–1429, 2019.

[22] S. U. Stich, "Local sgd converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

[23] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[24] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[25] A. Khaled, K. Mishchenko, and P. Richtárik, "First analysis of local gd on heterogeneous data," *arXiv preprint arXiv:1909.04715*, 2019.

[26] L. Cai, D. Lin, J. Zhang, and S. Yu, "Dynamic sample selection for federated learning with heterogeneous data in fog computing," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[27] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[28] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[29] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "Performance optimization of federated learning over wireless networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[30] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1698–1707.

[31] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning based on over-the-air computation," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[32] A. L. Wang, B. X. Chen, C. G. Wang, and D. Hua, "Non-intrusive load monitoring algorithm based on features of v–i trajectory," *Electric Power Systems Research*, vol. 157, pp. 134–144, 2018.

[33] M. Berges, E. Goldman, H. S. Matthews, L. Soibelman, and K. Anderson, "User-centered nonintrusive electricity load monitoring for residential buildings," *Journal of computing in civil engineering*, vol. 25, no. 6, pp. 471–480, 2011.

[34] S. Du, M. Li, S. Han, J. Shi, and H. Li, "Multi-pattern data mining and recognition of primary electric appliances from single non-intrusive load monitoring data," *Energies*, vol. 12, no. 6, p. 992, 2019.

[35] Z. Zhang, J. H. Son, Y. Li, M. Trayer, Z. Pi, D. Y. Hwang, and J. K. Moon, "Training-free non-intrusive load monitoring of electric vehicle charging with low sampling rate," in *IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2014, pp. 5419–5425.

[36] N. Batra and Y. Jia, "Transferring decomposed tensors for scalable energy breakdown across regions," in *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

[37] K. S. Barsim and B. Yang, "Sequential clustering-based event detection for non-intrusive load monitoring," *Computer Science & Information Technology*, vol. 10, p. 5121, 2016.

[38] Q. Liu, K. M. Kamoto, X. Liu, M. Sun, and N. Linge, "Low-complexity non-intrusive load monitoring using unsupervised learning and generalized appliance models," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 1, pp. 28–37, 2019.

[39] R. Bonfigli, A. Felicetti, E. Principi, M. Fagiani, S. Squartini, and F. Piazza, "Denoising autoencoders for non-intrusive load monitoring: improvements and comparative evaluation," *Energy and Buildings*, vol. 158, pp. 1461–1474, 2018.

[40] K. Bao, K. Ibrahimov, M. Wagner, and H. Schmeck, "Enhancing neural non-intrusive load monitoring with generative adversarial networks," *Energy Informatics*, vol. 1, no. 1, pp. 295–302, 2018.

[41] C. Shin, S. Joo, J. Yim, H. Lee, T. Moon, and W. Rhee, "Subtask gated networks for non-intrusive load monitoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1150–1157.

[42] K. Chen, Q. Wang, Z. He, K. Chen, J. Hu, and J. He, "Convolutional sequence to sequence non-intrusive load monitoring," *The Journal of Engineering*, vol. 2018, no. 17, pp. 1860–1864, 2018.

[43] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[44] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[46] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[47] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, and J. Passerat-Palmbach, "A generic framework for privacy preserving deep learning," *arXiv preprint arXiv:1811.04017*, 2018.