ISSN 1674-1056
http://cpb.iphy.ac.cn
http://iopscience.iop.org/cpb
中国物理 B
Chinese Physics
中国物理
Chinese Physics B

Serverless distributed learning for smart grid analytics

Gang Huang(黄刚), Chao Wu(吴超), Yifan Hu(胡一帆), and Chuangxin Guo(郭创新)
Citation: Chin. Phys. B, 2021, 30 (**8**): 088802. DOI: 10.1088/1674-1056/abe232
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Uncovering offline event similarity of online friends by constructing null models

Wenkuo Cui(崔文阔), Jing Xiao(肖婧), Ting Li(李婷), Xiaoke Xu(许小可)
Chin. Phys. B, 2019, 28 (**6**): 068901. DOI: 10.1088/1674-1056/28/6/068901

Accomplishment and challenge of materials database toward big data

Yibin Xu(徐一斌)
Chin. Phys. B, 2018, 27 (**11**): 118901. DOI: 10.1088/1674-1056/27/11/118901

Generation of optimal persistent formations for heterogeneous multi-agent systems with a leader constraint

Guo-Qiang Wang(王国强), He Luo(罗贺), Xiao-Xuan Hu(胡笑旋)
Chin. Phys. B, 2018, 27 (**2**): 028901. DOI: 10.1088/1674-1056/27/2/028901

Link prediction in complex networks via modularity-based belief propagation

Darong Lai(赖大荣), Xin Shu(舒欣), Christine Nardini
Chin. Phys. B, 2017, 26 (**3**): 038902. DOI: 10.1088/1674-1056/26/3/038902

Statistical physics of hard combinatorial optimization： Vertex cover problem

Zhao Jin-Hua, Zhou Hai-Jun
Chin. Phys. B, 2014, 23 (**7**): 078901. DOI: 10.1088/1674-1056/23/7/078901

Serverless distributed learning for smart grid analytics*

Gang Huang(黄刚)[1,†], Chao Wu(吴超)[2], Yifan Hu(胡一帆)[3], and Chuangxin Guo(郭创新)[4]

[1] *Zhejiang Lab, Hangzhou* 311121, *China*
[2] *School of Public Affairs, Zhejiang University, Hangzhou* 310058, *China*
[3] *Polytechnic Institute, Zhejiang University, Hangzhou* 310015, *China*
[4] *College of Electrical Engineering, Zhejiang University, Hangzhou* 310027, *China*

(Received 13 December 2020; revised manuscript received 8 January 2021; accepted manuscript online 2 February 2021)

The digitization, informatization, and intelligentization of physical systems require strong support from big data analysis. However, due to restrictions on data security and privacy and concerns about the cost of big data collection, transmission, and storage, it is difficult to do data aggregation in real-world power systems, which directly retards the effective implementation of smart grid analytics. Federated learning, an advanced distributed learning method proposed by Google, seems a promising solution to the above issues. Nevertheless, it relies on a server node to complete model aggregation and the framework is limited to scenarios where data are independent and identically distributed. Thus, we here propose a serverless distributed learning platform based on blockchain to solve the above two issues. In the proposed platform, the task of machine learning is performed according to smart contracts, and encrypted models are aggregated via a mechanism of knowledge distillation. Through this proposed method, a server node is no longer required and the learning ability is no longer limited to independent and identically distributed scenarios. Experiments on a public electrical grid dataset will verify the effectiveness of the proposed approach.

Keywords: smart grid, physical system, distributed learning, artificial intelligence

PACS: 88.80.hh, 88.80.H–, 89.20.Ff **DOI:** 10.1088/1674-1056/abe232

1. Introduction

As a typical physical system, the power grid is data rich,[1] especially with the recent provision of advanced information and communication technologies (ICTs). However, the power industry is not yet so much data driven,[2] and we have to rely on traditional physics-based methods to analyze the grids most of the time. Due to the fast development of digital innovations and technology-enabled model innovations, artificial intelligence (AI), mainly driven by deep learning research,[3] is important to power the smart grid because it promises to take full advantage of the historical data and revolutionize the century-old industry from the ground.[4] As a consequence, AI has recently attracted much attention from academia, industry, and government in the field of power engineering.[5]

Having gone through several ups and downs, the AI's most recent resurgence is powered by deep learning, which belongs to a broader family of machine learning methods based on the concept of artificial neural networks. After the huge success of deep learning in fields such as computer vision and natural language processing,[6–8] scholars have tried to apply the method of deep learning to problems of smart grid analytics as well,[9–11] with the hope to tackle the issues that remain unsolved by traditional methods. In fact, an increasing number of data are generated in the power industry due to the wide spread of advanced ICTs such as phasor measurement units (PMU), micro phasor measurement units (μPMU), and advanced meter infrastructures. Deep learning can handle the big data situation[12] while traditional machine learning methods often fail.

By default, deep learning technologies are developed based on the aggregation of big data.[13] In other words, huge amounts of data collected from end users or various sensors must be transmitted to and then stored in a central server or a cluster of servers for modeling and analysis. This will cause technology issues in power systems because real-world electric power data are generally distributed among a large number of regions,[14] and it will also be cost-ineffective to collect, transmit, and store so much data. What is worse, data regulations for security and privacy protection[15] will become a critical issue that forbid the data aggregation process.

To overcome the above limitations, AI technologies that support distributed machine learning and data security and privacy protection are highly required. A promising solution is federated learning, which was proposed by Google[16,17] with the original aim of fixing the regulation issue arose from improving keyboard query suggestions. This technique is quite different from the distributed computing,[18] which is a more traditional concept in the field of distributed machine learning and does not deal with the issue of security and privacy protection. Federated learning has now been generalized to sce-

*Project supported by the National Natural Science Foundation of China (Grant Nos. 52007173 and U19B2042), Zhejiang Provincial Natural Science Foundation of China (Grant No. LQ20E070002), and Zhejiang Lab's Talent Fund for Young Professionals (Grant No. 2020KB0AA01).

†Corresponding author. E-mail: huanggang@zju.edu.cn

© 2021 Chinese Physical Society and IOP Publishing Ltd

narios, where a machine learning model is trained while raw data remain distributed, and Ref. [19] extended the concept of federated learning to horizontal federated learning, vertical federated learning, and federated transfer learning, according to the characteristics of different datasets. Recently, federated learning has also been applied to the field of smart grid. For example Ref. [20], proposed a federated learning approach to predict energy demand for electric vehicle charging stations Ref. [21], started the first use of federated learning to forecast electrical load Ref. [22], provided a distributed framework based on federated learning to train power-related model for smart grid metering system.

The security and privacy of electric power data can be protected by the above federated learning approach. However, federated learning can be regarded as a partial solution just because a server node is required. The server-based distributed learning method suffers from the risk of single point of failure when the central server gets damaged. In addition, as Refs. [23–25] pointed out, the above federated learning method will have unacceptable performance for scenarios where data are not independent and identically distributed (IID), which are often the cases we meet in the real-world physical systems.

Regarding the above challenges, we propose a serverless distributed machine learning platform for smart grid analytics in this paper. In the proposed platform, we deploy an initial model to a decentralized blockchain network with the form of smart contracts[26] and use the blockchain network as communication infrastructure. The initial model is trained individually by different grids or different parts of a grid, and then gets aggregated by all the training participants through a mechanism of knowledge distillation. Thus, a server node is no longer required to derive a global model. Moreover, the mechanism of knowledge distillation takes better advantage of electric power data and makes the learning of smart grid deeper, which could improve the performance of distributed learning for non-IID scenarios. The major contributions of this article are twofold: (1) we present a serverless distributed learning platform that will have wide applications in smart grids; (2) we propose an effective algorithm to deal with the non-IID issue during distributed learning process.

The remainder of this article is organized as follows. In Section 2, we will propose the components and the designed workflow of smart grid distributed learning platform. In Section 3, we will propose our serverless algorithm for distributed learning. Experiments for smart grid will be given in Section 4. Finally, we will conclude this article in Section 5.

2. Smart grid distributed learning platform

In this section, we will present our distributed learning platform for smart grid analytics. Both the components and workflow of this platform will be introduced.

2.1. Components

In general, three roles are involved in a machine learning process, namely data provider, model provider, and computation provider.[15] For smart grid applications, however, independent computation provider is seldom involved due to considerations of asset security. Thus, we focus on data provider and model provider in the distributed learning platform, as follows:

1. Data provider Data providers can be at different granularity levels. For example, different grids that are independent from each other can be viewed as different data providers, and end users or various sensors in a grid can also be viewed as different data providers. The latter is made possible due to the advance of ICTs, which empower terminal nodes whose original responsibility is to collect data the ability of doing computation. Thus, distributed learning is made possible. Before the data are contributed to learning a machine learning model, data providers need to evaluate the quality of data and provide data validation as well as its schema. We need to emphasize that when we say "data providers provide their data", we mean the data are granted for being used rather than being given out to others.

2. Model provider The model provider can be anyone that has the desire to start a machine learning task. For example, if a dispatch center wants to have a power grid stability prediction model, he may need to collect as much data as possible and train a machine learning model from the perspective of traditional deep learning method, but he needs only to release his need from the perspective of distributed learning methods. To be more specific, the model provider acts as a training task provider who initializes a model training task with data schema for required training data and validation data to evaluate the aggregated model. In traditional paradigms, the model provider has to set up training infrastructures for models and collect training data; in our distributed machine learning paradigm, it needs only to provide an initial model as smart contracts to data providers. Note that the initial model here can be a blank model with random parameters or a pretrained model saved by the model provider.

In addition to the above components, there are three other nodes supporting the platform in the blockchain network: 1) distributed file system, which provides a generic interface to access intermediate data during the machine learning process; 2) blockchain node, which records the addresses of participating nodes and the model/data; 3) notification node, which takes the responsibility of communication and event triggers.

2.2. Workflow

The whole workflow of the smart grid distributed learning platform is illustrated in Fig. 1, where all tokens used

within the workflow are transferred according to smart contracts. Whisper,[27] providing communications to two correspondents that know a hash only, is used to signal to each other in order to ultimately collaborate between nodes.

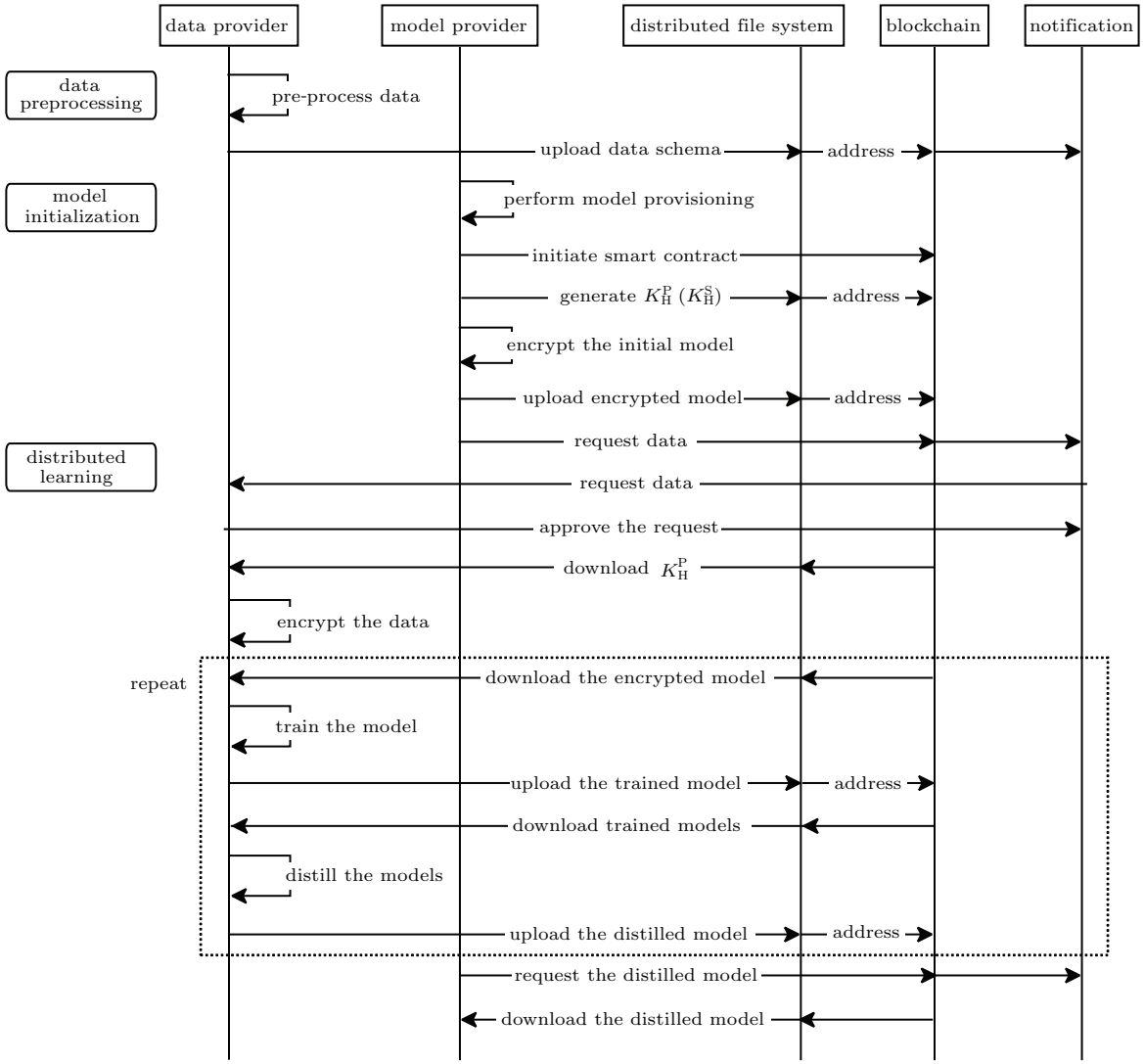


Fig. 1. An illustration of the workflow of smart grid distributed learning platform. Herein, K_H^P (K_H^S) represent the pair of homomorphic encryption keys.

The workflow consists of the following steps:

1. Data preprocessing This step is performed offline locally by data providers. There are several sub-steps involved in this step:

Sharding might be performed to divide data into small portions, as it is often difficult to scale with big chunks of data and some model providers might only wish to leverage a small subset of data instead of everything. Data quality scores are calculated using a commonly agreed quality measurement function. Metadata (e.g., device id, data description) are also prepared.

Data providers store data schema (including quality scores, metadata) on the distributed file system. The schema addresses are broadcasted to the blockchain network. Any node then can locate and view schema using the corresponding address.

2. Model initialization The model provider initiates a distributed learning task by performing the following steps:

As discussed in Section 2, the model provider performs model provisioning and deploys the training task as a smart contract. The smart contract is then initiated on the blockchain.

The model provider locally generates a pair of homomorphic encryption keys[28] (K_H^P/K_H^S), which is partially opened to the blockchain network (i.e., the public key K_H^P is written into smart contracts and broadcasted to all nodes). Note that the model provider should never reveal the private key K_H^S.

The model provider encrypts the initial model parameters M^0 using the public key K_H^P. Then, the encrypted model $f_H(M^0)$ is stored on the distributed file system with the hash uploaded to the blockchain.

3. Distributed learning The following sub-steps are executed in sequence, and the training process will be repeated until the distilled model is converged or the maximum learning

round is met.

The model provider determines the active data providers for training and then asks for approval from corresponding data providers.

When the data providers accept the request, they download the public key $K_{\mathrm{H}}^{\mathrm{P}}$. Training data ($D_i$) are encrypted using the public key, and encrypted data $f_{\mathrm{H}}(D_i)$ will be used for the training process:

(a) Data providers download the encrypted model $f_{\mathrm{H}}(M^0)$ and perform the training. The results ($f_{\mathrm{H}}(M_i^0)$) will be saved in the distributed file system and uploaded to the blockchain.

(b) Data providers download the trained models ($f_{\mathrm{H}}(M_{j\neq i}^0)$). Then, the mechanism of knowledge distillation is utilized to distill the knowledge of models into the local one. The details will be introduced in Section 3.

When the distilled model is converged or the maximum learning round is met, the model provider can be notified and will request the distilled model. The serverless distributed learning process is complete after the distilled model is downloaded.

3. Serverless distributed learning algorithm

In the last section, we have introduced our distributed learning platform for smart grid analytics. This platform is based on blockchain, and the machine learning task is performed according to smart contracts, thus no server is required. In this section, we will further present the details of our serverless distributed learning algorithm. We note that our proposed algorithm differs from current federated learning method by requiring no server, and the model aggregation process is inspired by the concept of knowledge distillation.[29]

3.1. The mechanism of knowledge distillation

The most representative method of model aggregation for distributed machine learning is FederatedAveraging.[17] This method, however, requires a central server. Moreover, this type of algorithms oversimplifies the model aggregation procedure as an average calculation or a weighted average calculation. The general process of server-based distributed learning algorithm is presented in Algorithm 1.

In Algorithm 1, we assume there are N data providers, which are denoted as $D_1,\ldots,D_i,\ldots,D_N$. Each data provider generates an initial local model M_i^k by training the global model M^k on the local dataset D_i. Then, the server receives the trained models and computes an updated global model M^{k+1} through average calculation or weighted average calculation. The updated global model is used to start the next round of distributed learning process, and the algorithm will stop when the global model gets converged or the maximum round of distributed learning is arrived.

We note that a server is of necessity in the above server-based distributed learning algorithm, which will suffer from the risk of single point of failure when the server gets damaged. In order to remove the dependence upon servers, we here introduce the mechanism of knowledge distillation, which requires no server and could leverage the power of data during model aggregation process. The distillation process happening at data provider i during aggregation round k is illustrated in Fig. 2. Here, local data is leveraged to calculate not only the local loss of local model M_i^k but also the distilled loss of other trained models $M_{j\neq i}^k$. Then, the total loss is used to update the local model to M_i^{k+1}.

Algorithm 1 Server-based distributed learning algorithm

```
The model provider initiates a distributed learning task
The server initiates a global model M^0
repeat
    The server determines the active data providers in round k
    The server sends the global model M^k to active data providers
    repeat
        Data provider i receives global model M^k
        Data provider i computes a local model M_i^k with its own data D_i
        Data provider i sends the trained model M_i^k to the server
    until Every data provider is traversed
    The server downloads the trained models M_i^k
    The server computes an updated global model M^{k+1} through (weighted) average calculation
until The global model converged or maximum learning round repeated
The model provider requests the learned model
The learned model is downloaded for use
```

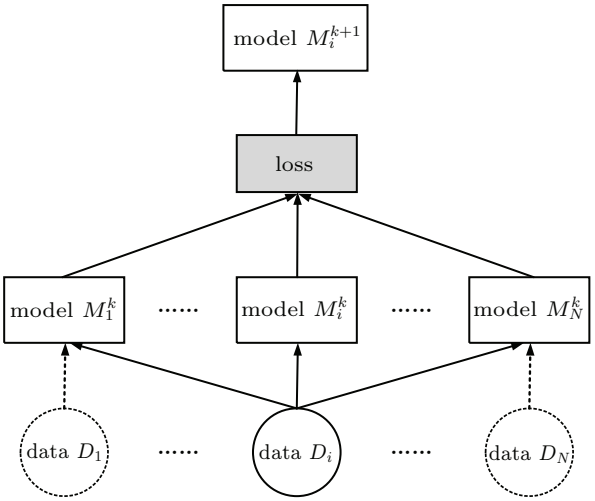


Fig. 2. An illustration of the mechanism of knowledge distillation.

While the calculation of local loss is straightforward, the calculation of distilled loss can be implemented via a measurement of difference between different models (e.g., the relative entropy). After the local model is updated from M_i^k to M_i^{k+1}, the one with the best validation accuracy will be picked up for the next round of model aggregation.

3.2. Distributed learning algorithm

Based on the above mechanism of knowledge distillation in Subsection 3.1, we here present the whole process of our serverless distributed learning method in Algorithm 2.

Similar to that in Subsection 3.1, we also assume there are N data providers (i.e., $D_1,\ldots,D_i,\ldots,D_N$). Data provider i generates an initial local model M_i^k by training the global model M^k on the local dataset D_i. Then, the trained models are aggregated via the mechanism of knowledge distillation proposed in Subsection 3.1. By this step, an updated global model M_i^{k+1} will be derived, and it is used to start the next round of distributed learning process at data provider i. The major difference between our serverless distributed learning algorithm and traditional server-based distributed learning algorithm is the introduction of the mechanism of knowledge distillation. This innovation removes the dependence upon servers and further leverages local data to derive the global model for each data provider, which could deal with the non-IID issue during distributed learning process. The effectiveness of the proposed distributed learning algorithm will be verified in the next section.

Algorithm 2 Serverless distributed learning algorithm

1: The model provider initiates a distributed learning task
2: A global model M_0 is stored on the distributed file system with its hash uploaded to the blockchain
3: **repeat**
4: The model provider determines the active data providers in round k
5: **repeat**
6: Data provider i downloads global model M^k
7: Data provider i computes a local model M_i^k with its own data D_i
8: Data provider i saves the trained model M_i^k to the distributed file system
9: Data provider i downloads the trained models $M_{j\neq i}^k$
10: Data provider i updates the global model to derive M^{k+1} through the mechanism of knowledge distillation
11: **until** Every data provider is traversed
12: **until** The global model converged or maximum learning round repeated
13: The model provider requests the distilled model
14: The distilled model is downloaded for use

4. Experiments

To show the effectiveness of our proposed method, experiments on the task of grid stability analysis are given here as an example. Our experiments are based on the Electrical Grid Stability Simulated Dataset[30] hosted in the UCI Machine Learning Repository. The task is actually a two-class classification problem, where the input shape is 12-dimensional and the output is one-dimensional. The original dataset has 10000 samples, and we randomly divide them into two subsets with the assumption that these data are collected from two grids with similar structures. The subsets can be independent and identically distributed (IID), which is an ideal assumption for academic research. However, the subsets can also be non-IID, which is more realistic. In this paper, we will investigate the

performance of our proposed method in both scenarios. We will compare the serverless distributed learning method with traditional distributed learning method via the FederatedAveraging algorithm. As a baseline, centralized learning results will also be provided.

4.1. Baseline

Centralized learning is the traditional deep learning method with the assumption that all the training data can be collected to form a centralized dataset. This assumption, however, is too strong, which will violate the regulation of data and cause privacy and security issues. Nevertheless, this method could provide a baseline for the performance evaluation of distributed learning methods, thus we present it firstly here.

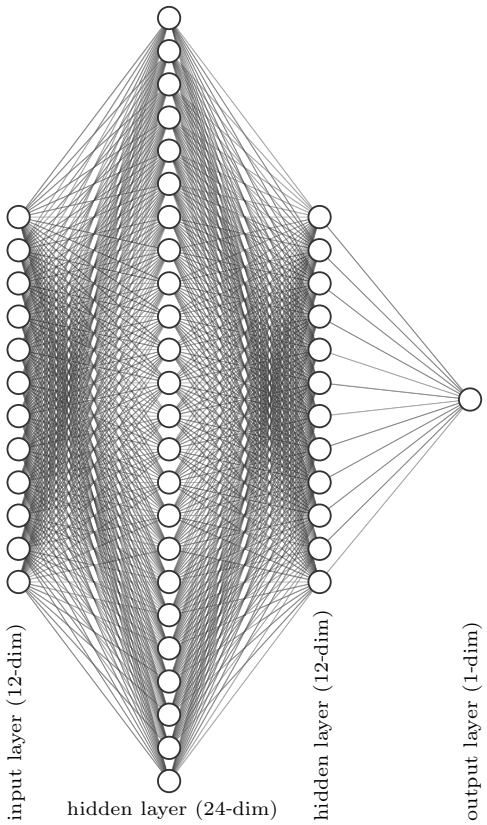


Fig. 3. The network architecture of the learning model.

For the smart grid stability prediction task, we use a four-layer neural network as the learning model. The network structure of the learning model is shown in Fig. 3, which includes one 12-dimensional input layer, two hidden layers (24-dimensions and 12-dimensions, respectively), and one 1-dimensional output layer in sequential order. The activation function for the output layer is Sigmoid and the others are rectified linear units (ReLU). We note that better neural network structure might exist, but this is beyond the scope of this article. The experiments are carried out on a 2.60-GHz Intel(R) Xeon(R) CPU E5-2690 based workstation, and PyTorch[31] is used to implement the deep learning method. Adaptive moment estimation (Adam)[32] is used as the solution method with learning rate set as 0.05. L2 regularization is also included in order to avoid the overfitting issue during training process, and the regularization factor is set as 0.0001.

We randomly divide the original data into training dataset, validation dataset, and testing dataset with the ratio of 8:1:1. As the centralized dataset is assumed to be available in the baseline, training dataset, validation dataset, and testing dataset will include 8000 samples, 1000 samples, and 1000 samples, respectively. While the training dataset is used to perform machine learning task, the validation dataset is used to select the trained model whose performance is the best. We use the independent testing dataset to evaluate the quality of the final model, thus reflects the performance of different machine learning methods. By setting the maximum training epoch as 500, the above deep learning method can reach an accuracy of 90.3% when predicting the stability of smart grid. Table 1 shows the detailed testing results.

Table 1. Testing results of centralized learning method.

	Predicted unstable	Predicted stable
Actual unstable	528	96
Actual stable	1	375

From the above results, we can see that the deep learning method works well for smart grid analytics. Only one stable state and 96 unstable states within the 1000 samples are wrongly predicted to be unstable or stable, and the rest samples (903 samples in total) are all correctly identified. To have a closer look at the training process, the centralized learning method accuracy on the validation dataset during the training process is shown in Fig. 4.

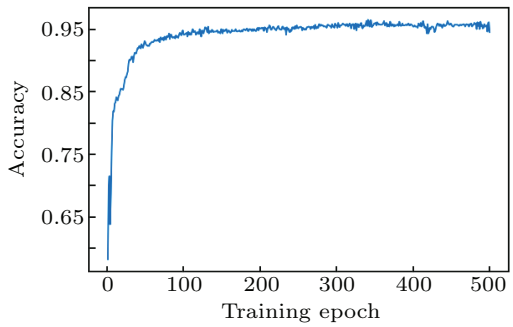


Fig. 4. Accuracy of centralized learning method on the validation dataset.

4.2. The IID scenario

The IID assumption is common in the field of machine learning, which indicates that all the training data are independent and identically distributed. Following this assumption, we randomly divide the training dataset and validation dataset into two subsets, represented as grid 1 and grid 2, respectively. Thus, each grid will include 4,500 samples in total, and the ratio of training dataset and validataion dataset is kept as 8:1. The testing dataset is kept exactly the same as that in Subsection 4.1.

Running the same neural network model in grid 1 and grid 2 separately, we apply the traditional distributed learning method to aggregate the trained model in each training epoch. Through these efforts, we get the accuracy of 90.2% on the testing dataset. The detailed results for stable states and unstable states are as follows:

Table 2. Testing results of traditional distributed learning method in the IID scenario.

	Predicted unstable	Predicted stable
Actual unstable	526	98
Actual stable	0	376

The above results verify the effectiveness of the traditional distributed learning method under the IID assumption. In fact, the traditional distributed learning method performs almost the same as that via centralized learning method. 902 samples of the testing dataset among total 1000 samples are correctly predicted. The traditional distributed learning method accuracy on the validation dataset during the training process is shown in Fig. 5.

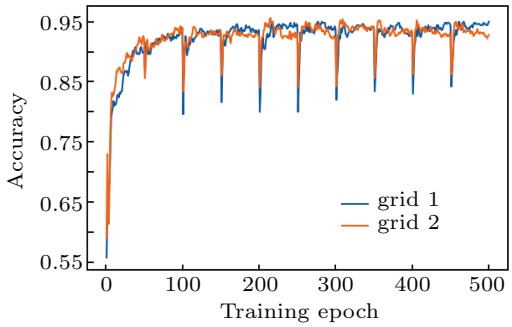


Fig. 5. Validation accuracy of traditional distributed learning method in the IID scenario.

As a comparison, we also run the serverless distributed learning method proposed in Section 3. Grid 1 and grid 2 keep the same subsets of the original dataset as that for the traditional distributed learning method. Running the same aggregation round and the same training epoch in each round, we finally get the accuracy of 90.5% on the same testing dataset, with the following detailed results.

Table 3. Testing results of serverless distributed learning method in the IID scenario.

	Predicted unstable	Predicted stable
Actual unstable	546	78
Actual stable	17	359

We can see from Table 3 that the proposed method performs slightly better than the traditional distributed learning method. For the 1000 testing samples, 905 samples are predicted correctly. This performance is almost the same as that via the centralized learning method and traditional learning method. In fact, the performance of our proposed method and that of the traditional learning method are quite similar, which could be further verified by the validation accuracy of serverless distributed learning method during the training process (see Fig. 6).

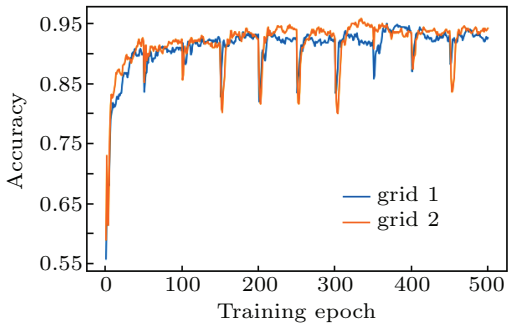


Fig. 6. Validation accuracy of serverless distributed learning method in the IID scenario.

4.3. The non-IID scenario

While the IID assumption is often made for machine learning research, it indeed deviates from the reality. In real-world applications, there is significant difference in data distribution across different data holders.[33] For example, some grid might be much less resilient than others, thus the ratio of unstable states will be much greater. To reflect the non-IID scenario, we keep one grid the same as that in Subsection 4.2 and change the other one to have a different ratio of unstable states. This indicates a non-IID scenario between grid 1 and grid 2.

We run the exactly same traditional distributed learning method as Subsection 4.2. This time, surprisingly, the accuracy on testing dataset reduces from 90.2% to 78.8%. This performance is much worse than the baseline setting in Subsection 4.1, which indicates that the IID assumption has a great impact on the effectiveness of traditional distributed learning method. Table 4 shows the detailed testing results.

Table 4. Testing results of traditional distributed learning method in the non-IID scenario.

	Predicted unstable	Predicted stable
Actual unstable	623	1
Actual stable	211	165

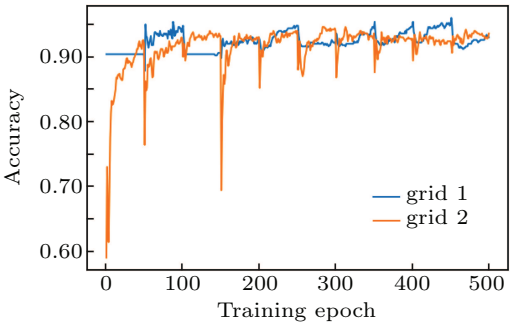


Fig. 7. Validation accuracy of traditional distributed learning method in the non-IID scenario.

To understand why the traditional distributed learning method fails in the non-IID scenario, we also present the validation accuracy during the training process in Fig. 7. From this figure, we see that the validation process has a much more violent shaking phenomenon than that in the IID scenario. This

might be caused by the huge difference between the data characteristics owned by grid 1 and grid 2: Due to the non-IID effect, the local models trained by grid 1 and grid 2 will be quite different in every aggregation round, thus the updated global model in every round will deviate much from both of them. This side effect makes the wrong predictions increase from 98 samples to 212 samples in the non-IID scenario.

With the same dataset for grid 1 and grid 2, we run our proposed serverless distributed learning method as well. The detailed testing results are given in Table 5, and the accuracy of our proposed method on the testing dataset is 90.6%.

Table 5. Testing results of serverless distributed learning method in the non-IID scenario.

	Predicted unstable	Predicted stable
Actual unstable	551	73
Actual stable	21	355

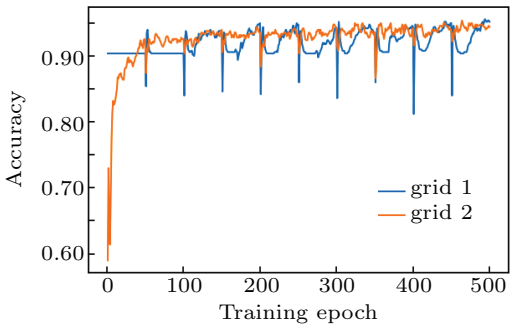


Fig. 8. Validation accuracy of serverless distributed learning method in the non-IID scenario.

The above results verify the effectiveness of our serverless method even in the non-IID scenario. While the results of traditional distributed learning method change a lot from the IID scenario to the non-IID scenario, the performance of our proposed method is quite stable. Actually, we can correctly predict 906 samples from the 1000 testing samples, which is at the same accurate level as the baseline in Subsection 4.1 and the IID scenario in Subsection 4.2. The validation accuracy of our proposed method during the training process (shown in Fig. 8) further proves the above observation. We can see that the accuracy change shares the similar pattern as those in Figs. 5 and 6, which are more stable than that in Fig. 7.

5. Conclusion

We propose a serverless distributed learning method for smart grid analytics in this article. Both the blockchain-based distributed learning platform and the distillation-driven serverless distributed learning algorithm are provided. Due to the introduction of smart contracts in blockchain, no server is required to communicate between different data providers. With the help of the mechanism of knowledge distillation, our distributed learning method will not be limited to non-IID scenarios which are more common in real-world physical systems. Experiments on the electrical grid stability simulated dataset have verified the effectiveness of the proposed method. We hope this article will encourage more research into the development of smart grid and other physical systems from the perspective of distributed machine learning.

References

[1] Bhattarai B P, Paudyal S, Luo Y, Mohanpurkar M, Cheung K, Tonkoski R, Hovsapian R, Myers K S, Zhang R and Zhao P 2019 *IET Smart Grid* **2** 141
[2] Akhavan-Hejazi H and Mohsenian-Rad H 2018 *Energy Rep.* **4** 91
[3] LeCun Y, Bengio Y and Hinton G 2015 *Nature* **521** 436
[4] Warren C 2019 *EPRI J.* 1
[5] Duchesne L, Karangelos E and Wehenkel L 2020 *Proc. IEEE* **108** 1656
[6] Goodfellow I, Bengio Y, Courville A and Bengio Y 2016 *Deep learning*. MIT press Cambridge
[7] Guo Y, Chen W and Jing S W 2020 *Chin. Phys. B* **29** 054203
[8] Wu F, Lu C, Zhu M, Chen H, Zhu J, Yu K, Li L, Li M, Chen Q and Li X 2020 *Nat. Mach. Intell.* **2** 312
[9] Pan X, Zhao T, Chen M and Zhang S 2020 *IEEE Trans. Power Syst.* **36** 1725
[10] Yin L, Gao Q, Zhao L and Wang T 2020 *Energy* **191** 116561
[11] Zhang D, Han X and Deng C 2018 *CSEE J. Power Energy Syst.* **4** 362
[12] Kezunovic M, Pinson P, Obradovic Z, Grijalva S, Hong T and Bessa R 2020 *Electr. Power Syst. Res.* **189** 106788
[13] Li T, Sahu A K, Talwalkar A and Smith V 2020 *IEEE Signal Process. Mag.* **37** 50
[14] Ghorbanian M, Dolatabadi S H and Siano P 2019 *IEEE Syst J.* **13** 4158
[15] Wu C, Xiao J, Huang G and Wu F 2019 *arXiv:1905.00753*
[16] Konevćný J, McMahan H B, Yu F X, Richtárik P, Suresh A T and Bacon D 2016 *arXiv:1610.05492*
[17] McMahan H B, Moore E, Ramage D, Hampson S and y Arcas B A 2017 *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282
[18] Liu T Y, Chen W, Wang T and Gao F 2018 *Distributed Machine Learning, Theories, Algorithms, and Systems* (China Machine Press)
[19] Yang Q, Liu Y, Chen T and Tong Y 2019 *ACM Trans. Intell. Syst. Technol.* **10** 1
[20] Saputra Y M, Hoang D T, Nguyen D N, Dutkiewicz E, Mueck M D and Srikanteswara S 2019 *Proceedings of the 2019 IEEE Global Communications Conference*, pp. 1–6
[21] Taïk A and Cherkaoui S 2020 *Proceedings of the 2020 IEEE International Conference on Communications*, pp. 1–6
[22] Zheng K, Xiao Y, Wang X and Chen W 2020 *Proc. CSEE* **40** 1
[23] Zhao Y, Li M, Lai L, Suda N, Civin D and Chandra V 2018 *arXiv:1806.00582*
[24] Liu L, Zhang J, Song S and Letaief K B 2019 *arXiv:1905.06641*
[25] Sattler F, Wiedemann S, Muller K-R and Samek W 2020 *IEEE Trans. Neural Netw. Learn. Syst.* **31** 3400
[26] Wood G 2014 *Ethereum Yellow Paper* **151** 1
[27] Zhang Z, Zhang L, Rasheed W, Jin Z, Ma T, Chen H and Xu G 2020 *Proceedings of the International Conference on Frontiers in Cyber Security*, pp. 215–230
[28] Gentry C 2009 *A Fully Homomorphic Encryption Scheme* (Ph.D. Dissertation) (Stanford: Stanford University)
[29] Gou J, Yu B, Maybank S J and Tao D 2020 *arXiv:2006.05525*
[30] Arzamasov V, Böhm K and Jochem P 2018 *Proceedings of the 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*, pp. 1–6
[31] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N and Antiga L 2019 *Proceedings of the 33rd Conference on Neural Information Processing Systems*, pp. 8026–8037
[32] Kingma D P and Ba J 2015 *Proceedings of the 33rd International Conference on Learning Representations*, pp. 1–15
[33] Hsieh K, Phanishayee A, Mutlu O and Gibbons P 2020 *Proceedings of the 37th International Conference on Machine Learning*, pp. 4387–4398