



OPEN

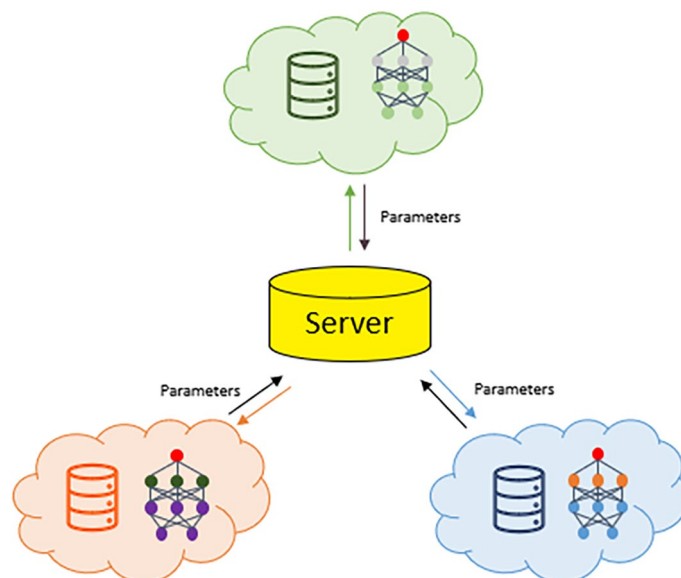
## Neural network training method for materials science based on multi-source databases

Jialong Guo<sup>1,2</sup>, Ziyi Chen<sup>1,2</sup>, Zhiwei Liu<sup>1,2</sup>, Xianwei Li<sup>3</sup>, Zhiyuan Xie<sup>4</sup>, Zongguo Wang<sup>1,2</sup>✉ & Yangang Wang<sup>1,2</sup>

The fourth paradigm of science has achieved great success in material discovery and it highlights the sharing and interoperability of data. However, most material data are scattered among various research institutions, and a big data transmission will consume significant bandwidth and tremendous time. At the meanwhile, some data owners prefer to protect the data and keep their initiative in the cooperation. This dilemma gradually leads to the “data island” problem, especially in material science. To attack the problem and make full use of the material data, we propose a new strategy of neural network training based on multi-source databases. In the whole training process, only model parameters are exchanged and no any external access or connection to the local databases. We demonstrate its validity by training a model characterizing material structure and its corresponding formation energy, based on two and four local databases, respectively. The results show that the obtained model accuracy trained by this method is almost the same to that obtained from a single database combining all the local ones. Moreover, different communication frequencies between the client and server are also studied to improve the model training efficiency, and an optimal frequency is recommended.

Materials are closely related to technological innovation and economic development. Government have always paid more attention to the competition in the field of materials. In order to promote the development of materials, the United States announced the “Materials Genome Initiative” in 2011, to shorten the research cycle of new materials and decrease the cost of material research through the collaborative cooperation of advanced experimental, computation and database technologies. China, Japan and European Union have also launched similar research plans to seize the opportunity in a new round of competition in materials science<sup>1</sup>. During this period, high-throughput experiment, high-throughput computing and material database technologies have achieved rapid development and extensive application, and some representative computational material databases, such as Materials Project<sup>2</sup>, the Open Quantum Materials Database (OQMD)<sup>3</sup> have been established. In addition, the development of artificial intelligence technology has further accelerated the transformation of scientific paradigm with “Data-Driven” mode in material field. Among them, support vector machine, random forest and neural network algorithms have become common data processing and analysis tools in the field of materials, and they have achieved remarkable results in the prediction of properties<sup>4</sup>, phase diagram<sup>5</sup> and structures<sup>6,7</sup>. Data is an important factor of the data-driven scientific paradigm<sup>8</sup>, and its quality determines the upper limit of the predictive power of the model, while the improvement of model is only to constantly approach this upper limit. In recent years, many material databases have been established with a large number of data with the help of high-throughput technologies. However, the “data island” problem in materials science is very serious, due to lacking of security sharing strategy of material data. Firstly, a number of high-quality material data which are difficult to obtain are still stored locally in different institutions, and collecting these data needs to consume significant network bandwidth and tremendous time. Secondly, the data opening and sharing strategy is still imperfect. Researchers in special material area who own the raw data have the absolutely initiative in the cooperation, and they usually refuse to share the raw data to maintain their advantageous. To best of our knowledge, there is no study on how to use the data located in different institutions without getting the raw data, although data protection in material science has been paid more and more attention. The suggestion, providing the access permission of database for their trust users<sup>9</sup>, has given a data sharing way but failed to protect the local data of the client

<sup>1</sup>Computer Network Information Center, Chinese Academy of Sciences, Beijing 100083, China. <sup>2</sup>University of Chinese Academy of Sciences, Beijing 100049, China. <sup>3</sup>China Petroleum Pipeline Engineering Co., Ltd., International, Langfang 065000, Hebei, China. <sup>4</sup>Department of Physics, Renmin University of China, Beijing 100872, China. ✉email: wangzg@cnic.cn



**Figure 1.** Schematic diagram of model training framework with multi-source databases. This framework contains three clients and one “Server”. Each client has one local dataset in this illustration, and the “Server” is used to receive and send the model parameters.

from disclosure. Therefore, to explore the value-added potential of material data and improve the innovation ecosystem in material science, it is urgent to develop a new neural network training method based on multiple databases without remote raw data access for material researchers.

The successful application of federated learning strategy on medical<sup>10–13</sup>, finance, IoT<sup>14–17</sup> and other fields<sup>18</sup> provides a new important enlightenment to realize the secure sharing of material data. Federated learning<sup>19</sup> is an emerging technology in the field of artificial intelligence. This technology was proposed to solve the problem of protecting users’ private data<sup>20,21</sup>, and it has developed a “Federated Transfer Learning” solution in the financial field combined with transfer learning<sup>22</sup>. Federated learning shows obvious advantages in effectively solving “data island” problem. It realizes the data sharing of each user while ensuring the user’s data security and personal data privacy, and carries out efficient machine learning among multiple participants or multiple computing nodes. As a machine learning framework, federated learning can be applied to neural networks and other algorithms. It is expected to become the basis of the next generation of artificial intelligence cooperative algorithms and networks.

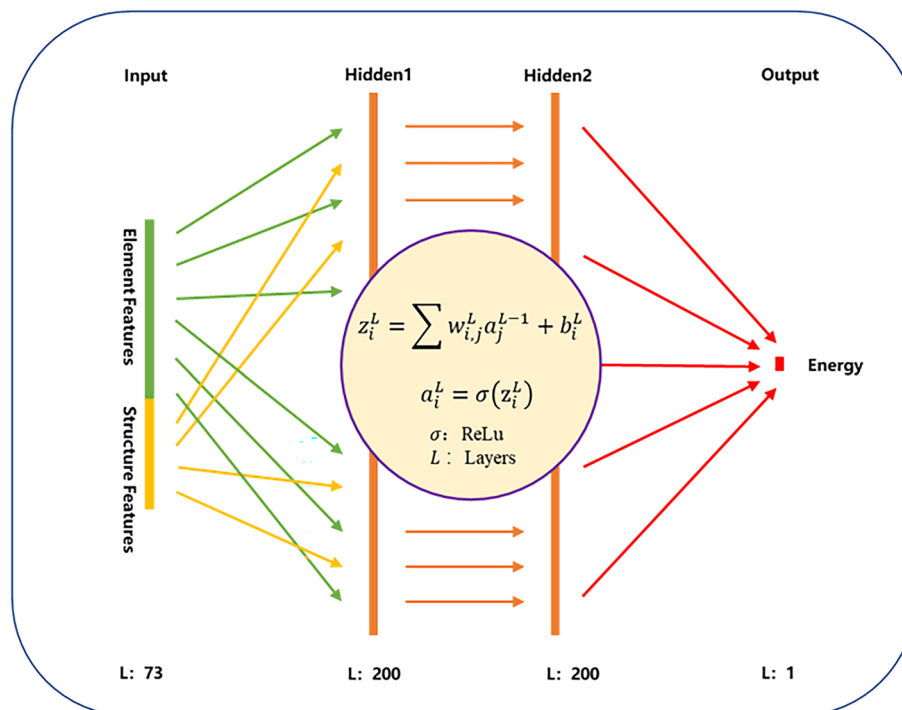
Horizontal federated learning scheme is suitable for the situation that the features of material data in different datasets have a lot of overlap but the types of materials have no overlap<sup>23,24</sup>. According to the characteristics of target material dataset we investigate, horizontal federated learning scheme is adopted to realize the material data sharing and model training. Under this strategy, each user can get and use the global model without accessing others’ local datasets, so as to reduce the risk of user information leakage and achieve the purpose of data sharing.

In this paper, we have proposed a new neural network training method based on multi-source database which combines the federated learning schema and characteristics of material data. The illustration of model training framework with multi-database used in this paper is shown in Fig. 1. We use this method to predict the relationship between structure and its formation energy with two clients, the accuracy of predicted model is similar to that trained from all the training data directly. To investigate the influence of client numbers and data distribution on the model accuracy, four clients with different data distribution are used to train a neural network model. Different communication frequencies are studied and a suitable communication frequency that balances time and model performance is provided. This work supplies an important way and universal workflow to promote data sharing and improve the accuracy of material property prediction in the filed of materials science.

## Results

**Formation energy prediction from two databases.** Neural network is a mathematical model that imitates the behavior characteristics of animal neural network. It is based on artificial neurons, and each neuron has an associated weight and threshold which are considered as the parameters of network used in materials science. Our aim is to train a shared model based on multi-source databases by sharing the parameters. The material dataset used in this work is randomly divided into training set and test set, training set has 10,000 terms of data and test set has 2,897 terms of data. The training set is equally divided into two subsets, each subset contains 5000 terms of data, and the two datasets are respectively placed on two clients located in two institutions. Using these two clients to train the relationship between structures and their formation energies.

Both structure and element features are extracted and recalculated to as the input of the training network. The structure feature includes lattice constant, unit cell volume and space group number. The element feature includes row, column and block in periodic table, atomic mass, atomic radius, Mendeleev number, resistivity, speed of sound, thermal conductivity, melting point, Young’s modulus, and coefficient of linear thermal expansion. The



**Figure 2.** Neural network structure diagram, the neural network is a four layer fully connected network, the activation function is ReLu, the input includes element and structure features, and the output is formation energy of structures.

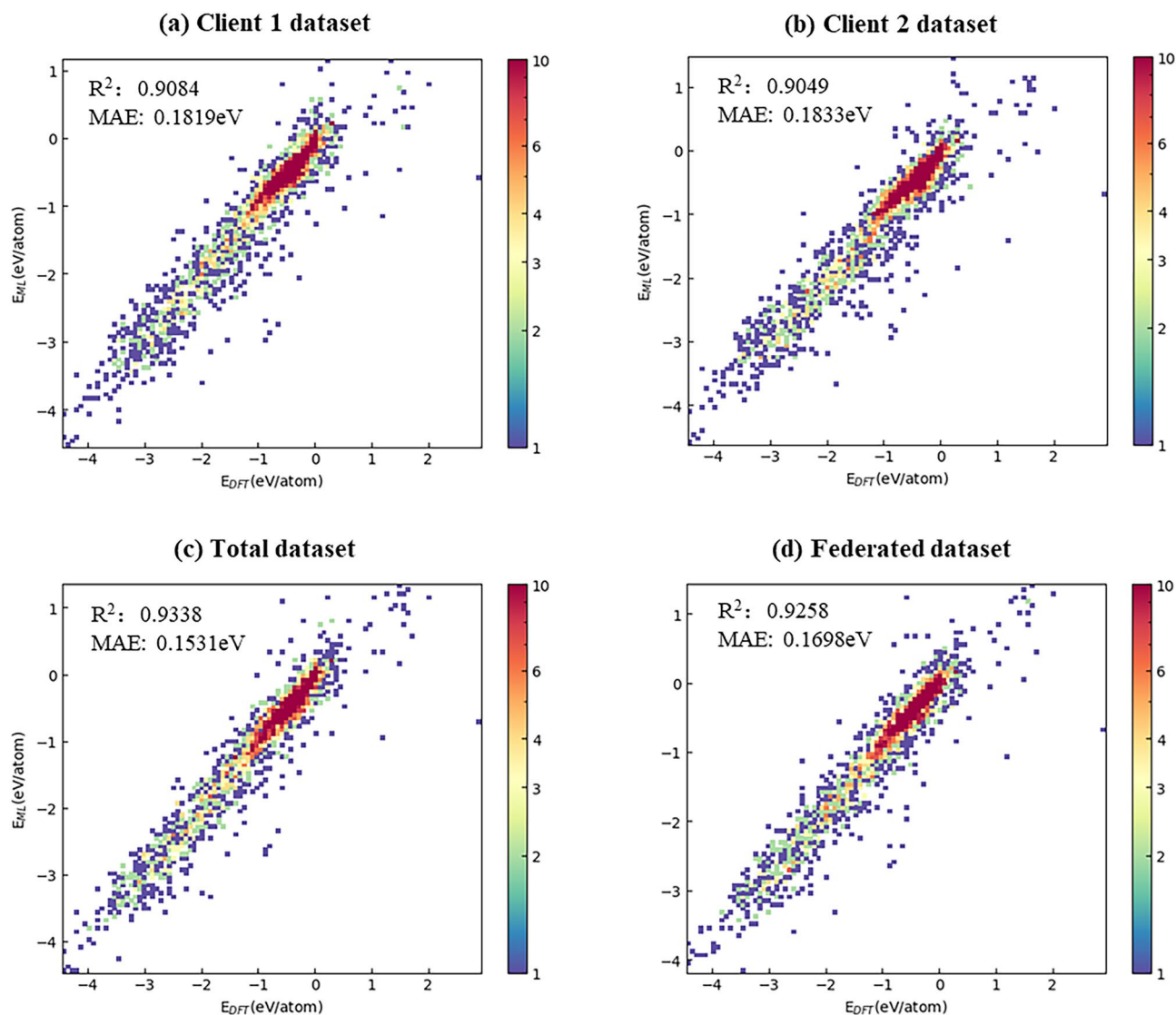
recalculated method includes five types of statistical information: minimum, maximum, range, mean and variance. The formation energies are extracted to as the output of the network. The schematic diagram of the neural network used in this work is shown in Fig. 2. The dimension of input layer is 73, the dimension of the two hidden layers is 200, the learning rate is set to 0.001, the batch size is set to 200, the loss function is set to MSELoss function, and the total number of iteration steps is set to 100. The structure features and their formation energies that can be directly used in the training process for neural network are provided in Supplementary information.

Using the same neural network, four models are trained based on different datasets. Two models are respectively trained based on the dataset of each client, the third one is trained based on the total training set (which merges the two datasets in client 1 and 2 together), and the fourth one is also trained based on the total training set but the two datasets are still respectively located in the different clients, and this type of dataset is called as federated dataset and the fourth model is named as shared model in the following sections. The four models are verified on the same test set, and the coefficient of determination ( $R^2$ ) and the Mean Absolute Error (MAE) of all models are shown in Fig. 3.

It can be seen that the prediction power of shared model is significantly improved than that of model trained from client 1 or client 2. In the training process, only the network parameters information is transmitted, and it is also able to encrypt data transmissions to improve the safety of data. Therefore, using the new neural network training method we proposed to train a shared model on multiple datasets can prevent the local data of the client from being directly obtained by others. At the same time, we should see the model performance is a little bit worse than that directly trained from total training set, but this can be further improved by optimizing the algorithms.

**Training model based on four databases with different data distribution.** In the previous section, two clients with the same size of training sets are used to construct the model training framework and train a shared model. However, the situation in materials science is more complex, which usually has multiple databases from different research groups or institutions in the same field. It is necessary to consider the applications of multiple clients with different sizes of datasets. The shared model trained from four clients with different sizes of datasets is further investigated by the training framework. The sizes of total training set and test set are still 10,000 and 2,897, respectively, but the training data set is scattered into four subsets and stored into four different clients (labeled 1, 2, 3, and 4). The local dataset sizes are respectively set to 1000, 2000, 3000, and 4000, forming a total training set containing 10,000 terms of data. The same network structure and parameters shown in Fig. 2 are used. The shared model is trained based on four datasets located in four clients, the  $R^2$  and MAE are shown in Fig. 4.

It can be seen that when each client uses its local dataset to train a model, the model trained by more data generally has a better performance on the test set. The shared model trained from multi-source datasets still performs more better than that from individual dataset. And compared with the case of two clients with same

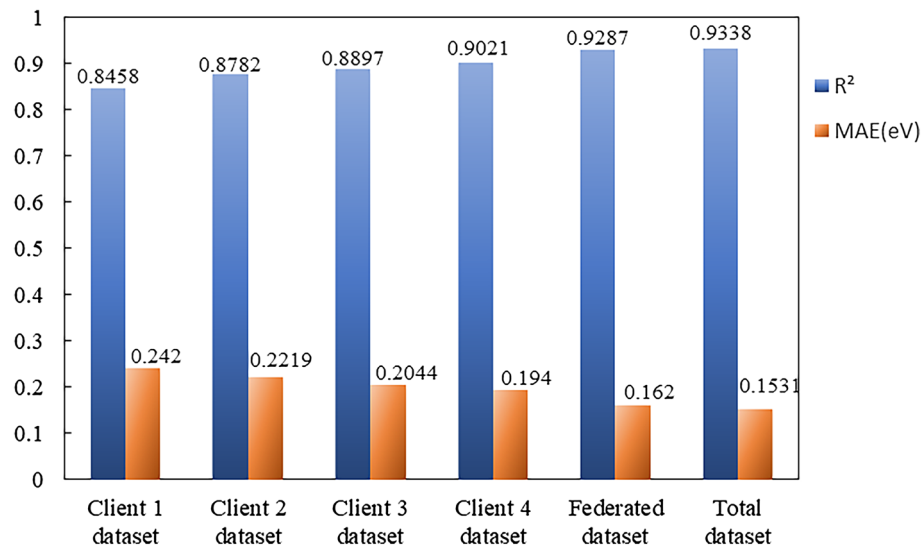


**Figure 3.** Prediction power of models trained on different datasets and method. **(a)** is for the model trained on the local dataset of client 1, **(b)** is for the model trained on the local dataset of client 2, **(c)** is for the model trained on the total training set, and **(d)** is for the shared model based on federated dataset (client 1 and 2) by using the method we proposed. Among them, different colors indicate aggregation degrees of points, red indicates that the point distribution is clustered, and blue indicates that the point distribution is sparse.

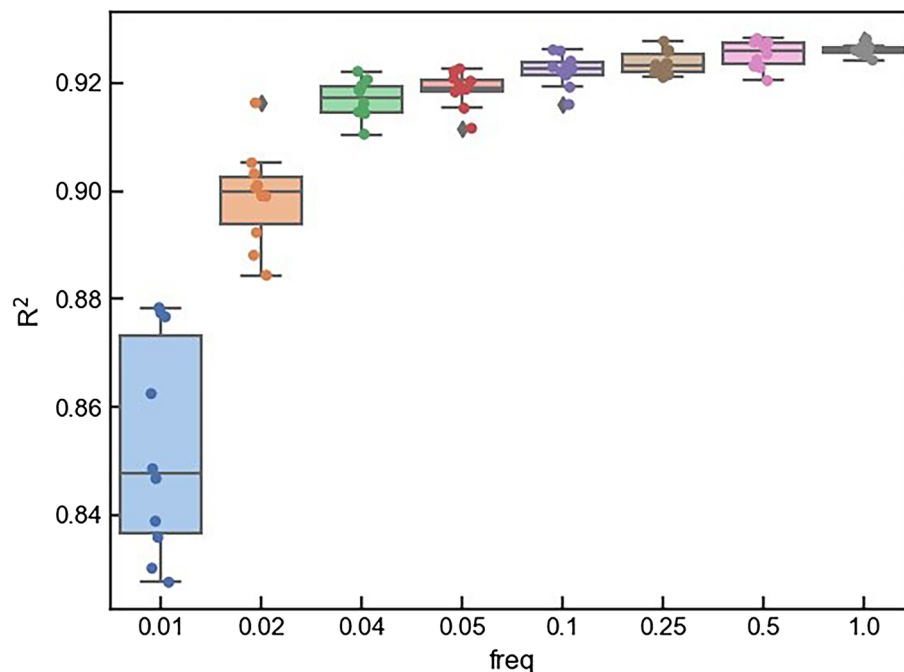
size of dataset, the prediction power of model does not decrease. This shows that the model training framework we used still has a stable performance in the case of more clients and uneven data distribution.

It is noticed that the choice of batch size for the case based on the total dataset is different with that based on multi-source datasets. When the model is training based on total training set, each batch size is randomly selected from the dataset, while each batch is a proportional combination of those extracted from each client when the model is training based on multi-source datasets located in different clients. Comparing these two methods, the former is more flexible, but this cannot be used as the absolute advantage for model training. Depending on the characteristics of dataset, the model training based on multi-source datasets may have a better performance than that based on the total training set.

**Communication frequency optimization.** During the training process, a communication is that network parameters between server and client have one exchange. Training a shared model from two clients and four clients in the above two experiments, there will be a communication between the server and the client once the network parameters of the local model are updated. When the size of client's local dataset is 5000, if both the batch size and iteration steps are set as 100, the number of communications will reach to 5000 in a local model training process and more time is need to finished a training process. In order to reduce the communication cost and improve the training efficiency, an in-depth study is carried out on the variation of prediction power of the training model with different communication frequencies.



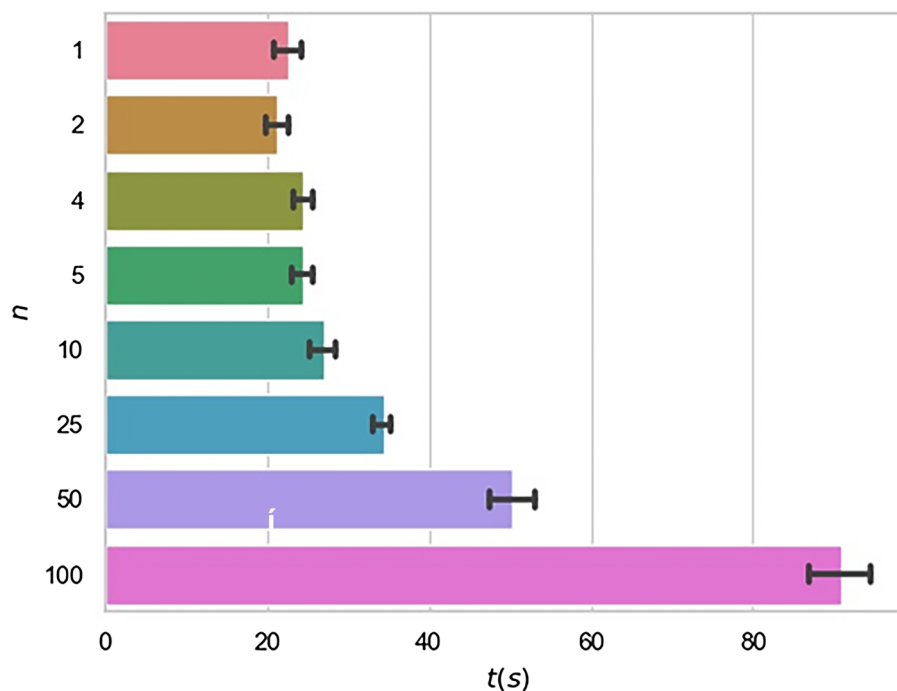
**Figure 4.** Column chart of shared model and individual trained model on each local dataset.



**Figure 5.** Prediction power of shared model based on four datasets with different communication frequencies from 10 times of experiments. Colors present different communication frequencies. The two ends of the rectangular box are respectively the upper quartile and lower quartile of  $R^2$  according to current communication frequency, the horizontal line in the middle of the rectangle is the median, and the short horizontal lines outside of the rectangular box are the largest and smallest values of  $R^2$  of ten data.

In this experiment, shared model is trained based on four clients and eight cases are considered, i.e., the communication frequency is respectively set to 0.01, 0.02, 0.04, 0.05, 0.1, 0.25, 0.5 and 1.0. When the total number of iteration steps is 100, the corresponding times of model training communication are 1, 2, 4, 5, 10, 25, 50 and 100, respectively. Take 10 times as example, the client will update its network parameters to the server when ten iteration steps are completed in the client. We have repeated the experiments ten times for all these eight cases. The performances of the trained model on the test set for each communication are shown in Fig. 5.

It can be seen that the prediction power of the model on the test set gradually increases with the increase of communication frequency. When the communication frequency is low, the shared model is obviously unstable, and it has a terrible performance than that from a single client dataset. With the increase of communication frequency, the prediction power of shared model rapidly increases, and the trend of increase will be slow when



**Figure 6.** Bar chart of training time consuming with different communication times for model trained based on four clients.  $t$  is the time spent on model training,  $n$  is the communication times.

it reaches a desired level. Meanwhile, the fluctuation range of the  $R^2$  in ten times of training processes generally decreases, and the shared model tends to be stable. It also can be seen that the model training framework can choose an appropriate communication frequency to balance the model performance and communication cost. In this experiment, when the model frequency is 0.25, it can both greatly reduce the communication cost and ensure the high prediction accuracy of model. In addition, the fewer communication times decreases the risk of data privacy leakage and increases the privacy-preserving capability in the model training process.

The training time of model with different communications has been also analyzed and the results are shown in Fig. 6. Among them, the training time have been mainly spent on communication between the client and the server and model gradient update calculation in the local client. It can be seen from the Fig. 6 that when the communication times is less than 10, the time of model gradient update calculation is dominant and the total training time keeps generally stable. With increasing the times of communications, the time spent on communications will be significantly longer than that on the model gradient update calculation. It is worth noting that the time up to 2710.45s (not shown) for the case that a communication will occur once the local model is updated in each client, which is longer than that in any case shown in Fig. 6, however, the prediction power has not been significantly improved. From Fig. 6, when the communication frequency is 0.25, both the accuracy and training efficiency of model have higher performance. In this way, the network bandwidth consumption has been reduced effectively.

## Discussion

In this paper, a new neural network training method for multi-source databases is proposed, and shared models between structures and their formation energies have been respectively trained based on two and four clients by using this method. Both of uniform and uneven data distributions in each client are investigated, and the results show that, the coefficient of determination  $R^2$  of model trained on two clients with uniform data is closer to that trained on four clients with uneven data, and the model biases from both is controlled below 1% comparing to the model trained based on total training set. Through the study of different communication frequencies, a reasonable communication frequency has been recommended, and the prediction power is kept excellent as the same time as communication costs are decreased.

One of important goals of this method we proposed is to provide a way to protect the data privacy, and make researchers take the initiative to sharing their valuable data for model training in materials science. In the future work, technologies such as blockchain and differential privacy will be combined to obtain stronger privacy protection effects. Neural network training framework we used can effectively ensure the security and traceability of data by storing model parameters as transaction records in the blockchain. In addition, to get a more precise model, some technologies such as sample alignment, stack pollution intelligent detection<sup>25</sup>, and gradient screening and shielding<sup>26</sup> methods will also be used to preprocess material data and reduce the interference of “noisy” data in the process of training universal model<sup>27</sup> for different material properties.

The neural network training method this work proposed performs a success in the structure-formation prediction of materials. The accuracy and prediction power of models are improved with the participation of

multi-source databases while safeguarding the data privacy. This work provides a new data application method and cooperation strategy for researchers to voluntarily share scientific data. It also supplies an important pathway to construct big data ecosystem and accelerate the transformation of scientific research paradigms in materials science.

## Methods

The neural network training framework includes the server and the client (shown in Fig. 1). The server collects, processes and shares the model parameters to all clients, the client uses the local dataset to train a model and updates parameters of the model. The client re-uploads the model parameters to the server when new model parameters are obtained from local dataset. And then, the server receives the model parameter data, calculates the weighted average of the model parameters, and distributes them to the client to start a new round of model training. The network parameters returned to each client by the server is calculated by Eq. (1):

$$w = \sum_i \alpha_i w_i \quad (1)$$

where  $w_i$  is the network parameter uploaded by client  $i$ ,  $\alpha_i$  is the weight of the client's local dataset in the total dataset.

**Server.** The server is used to realize the communication between different datasets. At the beginning, the server will connect with the clients of each institution or group who owns the dataset, and unify the initial model parameters trained on each client, such as iteration step, batch size, learning rate, optimizer, loss function and network structure. Then, the server will initialize the network model parameters and distribute the parameters to each client. After that, the server will wait for all clients to upload the updated network parameters. After all client network parameters are uploaded to the server, the server will perform a weighted average operation on all network parameters, and redistribute the operated network parameters to each client. Iterate continuously until the global model converges. The algorithm for sever used in this method is shown in Algorithm 1.

---

### Algorithm 1: Server

---

**Input** : Communication times:  $n$ , Proportion of data in client  $c$ :  $\alpha_c$

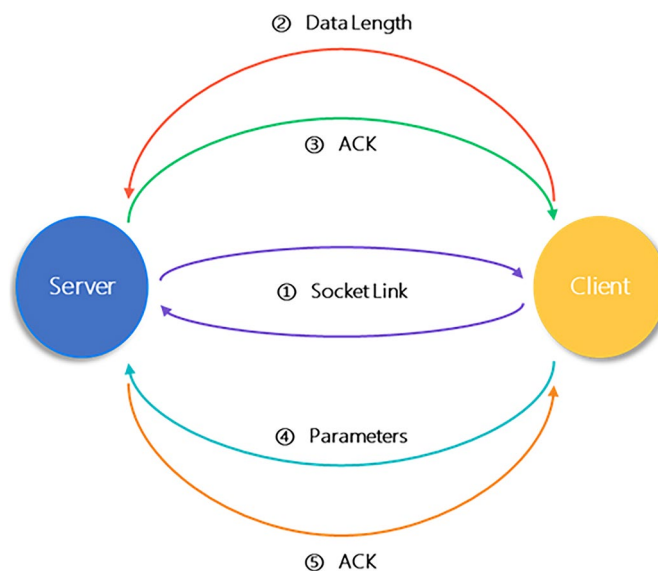
**Result** : The trained model

```

1 Initialize network parameters, and share them with clients;
2  $i = 0$ ;
3 while  $i < n$  do
4   foreach client  $c$  do
5      $w_c = \text{ReceiveParameters}()$ ;
6   end
7    $w = \sum \alpha_c w_c$ ;
8   foreach client  $c$  do
9      $\text{ShareParameters}(c, w)$ ;
10  end
11   $i = i + 1$ ;
12 end
```

---

**Client.** The client is used to implement the training process of the neural network model on the local dataset. First, the client will wait for the initial model parameters distributed by the server. Every client receives the model parameters and load them to the local network model. The local network model will read data from the local dataset, perform feedforward and backpropagation operations of the neural network, and update the weight parameters of the network. When the local model parameters are updated, the client will upload the local model parameters to the server and wait for the next round of local network model parameters' update. The algorithm for client is shown in Algorithm 2.



**Figure 7.** Diagram of the communication strategy between the server and the client.

---

#### Algorithm 2: Client

---

**Input** : Communication times:  $n$

**Result** : The trained model

```

1 Receive network parameters from server, and initialize local model parameters;
2  $i = 0$ ;
3 while  $i < n$  do
4    $w_c = \text{ReceiveParameters}()$ ;
5    $\text{UpdateParameters}(w_c)$ ;
6    $\text{TrainModel}()$ ;
7    $w_c = \text{GetParameters}()$ ;
8    $\text{ShareParameters}(\text{server}, w_c)$ ;
9    $i = i + 1$ ;
10 end
11  $\text{SaveModel}(\text{path})$ 

```

---

**Parameters transfer.** In this method, socket is used to realize the communication between the server and the client. The transmission process of the network parameters includes the model parameters serialization and deserialization. In addition, in order to avoid missing the received parameters data, the length of communication data will be firstly sent, and the model parameters will be sent when the acknowledge character (ACK) returned by receivers is received. The model parameters are received according to the data length. After the data is completely received, the acknowledge character is returned again, and the sender completes the transmission process after receiving the acknowledge character. The communication process is shown in Fig. 7.

**Dataset.** The data used in this work is from The Open Quantum Materials Database (OQMD), and 12,897 terms effective material data containing composition, unit cell, sites and formation energy are collected. Each data and its expanding information are extracted and stored in the database for model training. All data information can be seen from the supplementary information.

#### Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 4 June 2022; Accepted: 29 August 2022

Published online: 12 September 2022



## References

1. Yanjing, S., Huadong, F., Yang, B., Xue, J. & Jianxin, X. Progress in materials genome engineering in China. *Acta Metall. Sin.* **56**, 1313–1323 (2020).
2. Jain, A. *et al.* Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002 (2013).
3. Saal, J. E., Kirklin, S., Aykol, M., Meredig, B. & Wolverton, C. Materials design and discovery with high-throughput density functional theory: The open quantum materials database (oqmd). *JOM* **65**, 1501–1509 (2013).
4. Perdew, J. P. Artificial intelligence “sees” split electrons. *Science* **374**, 1322–1323 (2021).
5. Kusne, A. G. *et al.* On-the-fly closed-loop materials discovery via Bayesian active learning. *Nat. Commun.* **11**, 1–11 (2020).
6. Fung, V., Zhang, J., Hu, G., Ganesh, P. & Sumpter, B. G. Inverse design of two-dimensional materials with invertible neural networks. *NPJ Comput. Mater.* **7**, 1–9 (2021).
7. Vasylenko, A. *et al.* Element selection for crystalline inorganic solid discovery guided by unsupervised machine learning of experimentally explored chemistry. *Nat. Commun.* **12**, 1–12 (2021).
8. Pan, M. *et al.* DHPA: Dynamic human preference analytics framework: A case study on taxi drivers’ learning curve analysis. *ACM Trans. Intell. Syst. Technol.* **11**, 1–19. <https://doi.org/10.1145/3360312> (2020).
9. Pizzi, G., Cepellotti, A., Sabatini, R., Marzari, N. & Kozinsky, B. Aiida: Automated interactive infrastructure and database for computational science. *Comput. Mater. Sci.* **111**, 218–230 (2016).
10. Huang, L. *et al.* Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *J. Biomed. Inform.* **99**, 103291 (2019).
11. Peiffer-Smadja, N. *et al.* Machine learning for covid-19 needs global collaboration and data-sharing. *Nat. Mach. Intell.* **2**, 293–294 (2020).
12. Warnat-Herresthal, S. *et al.* Swarm learning for decentralized and confidential clinical machine learning. *Nature* **594**, 265–270 (2021).
13. Xu, J. *et al.* Federated learning for healthcare informatics. *J. Healthc. Inform. Res.* **5**, 1–19 (2021).
14. Shahid, O., Mothukuri, V., Pouriyeh, S., Parizi, R. M. & Shahriar, H. Detecting network attacks using federated learning for iot devices. In *2021 IEEE 29th International Conference on Network Protocols (ICNP)*, 1–6 (IEEE, 2021).
15. Yang, K. *et al.* Federated machine learning for intelligent iot via reconfigurable intelligent surface. *IEEE Netw.* **34**, 16–22 (2020).
16. Zhao, Y. *et al.* Privacy-preserving blockchain-based federated learning for iot devices. *IEEE Internet Things J.* **8**, 1817–1829 (2020).
17. Hu, N., Tian, Z., Lu, H., Du, X. & Guizani, M. A multiple-kernel clustering based intrusion detection scheme for 5g and iot networks. *Int. J. Mach. Learn. Cybern.* **12**, 3129–3144. <https://doi.org/10.1007/s13042-020-01253-w> (2021).
18. Li, L., Fan, Y., Tse, M. & Lin, K.-Y. A review of applications in federated learning. *Comput. Ind. Eng.* **149**, 106854 (2020).
19. McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282 (PMLR, 2017).
20. Zhu, H., Zhang, H. & Jin, Y. From federated learning to federated neural architecture search: A survey. *Complex Intell. Syst. J.* **7**, 639–657 (2021).
21. Lu, H. *et al.* Autod: Intelligent blockchain application unpacking based on jni layer deception call. *IEEE Netw.* **35**, 215–221. <https://doi.org/10.1109/MNET.011.2000467> (2021).
22. Liu, Y., Kang, Y., Xing, C., Chen, T. & Yang, Q. A secure federated transfer learning framework. *IEEE Intell. Syst.* **35**, 70–82 (2020).
23. Liang, T., Bi, Z. & Chen, G. Federated learning survey: Concept, technology, application and challenge. *J. Comput. Appl.* **8**, 1–13 (2020).
24. Yang, Q., Liu, Y., Chen, T. & Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.* **10**, 12 (2019).
25. Lu, H. *et al.* Research on intelligent detection of command level stack pollution for binary program analysis. *Mob. Netw. Appl.* **26**, 1723–1732. <https://doi.org/10.1007/s11036-019-01507-0> (2021).
26. Gu, Z. *et al.* Gradient shielding: Towards understanding vulnerability of deep neural networks. *IEEE Trans. Netw. Sci. Eng.* **8**, 921–932. <https://doi.org/10.1109/TNSE.2020.2996738> (2021).
27. Liang, Y. *et al.* A universal model for accurately predicting the formation energy of inorganic compounds. *Sci. China Mater.* **66**, 1–9. <https://doi.org/10.1007/s40843-022-2134-3> (2022).

## Acknowledgements

The authors thank Prof. Zhongyi Lu and Prof. Qun Hui for discussions. This work was supported by the Informatization Plan of Chinese Academy of Sciences (Grant No. CAS-WX2021PY-0102), Key Research Program of Frontier Sciences, CAS (Grant No. ZDBS-LY-7025), and Youth Innovation Promotion Association CAS.

## Author contributions

Z.W. and Y.W. provided the idea and conception, J.G. designed the experiments and wrote the article, J.G., Z.C. and Z.L. analyzed and interpreted the experimental results, X.L. and Z.X. revised article. All authors contributed to the general discussion and approved the submitted version.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-022-19426-8>.

**Correspondence** and requests for materials should be addressed to Z.W.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher’s note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022