Journal of Organizational and End User Computing
Volume 34 • Issue 1

A Hierarchical Clustering Federated Learning System Based on Industry 4.0

Chun-Yi Jiunn-Yin Lu, National Penghu University of Science and Technology, Taiwan
Hsin-Te Wu, National Ilan University, Taiwan*

ABSTRACT

This study proposes using dendrogram clustering as the basis to construct a federated learning system for A.I. model parameter updating. The authors adopted a private blockchain to accelerate downloads of the latest parameters corresponding to the computation results of an A.I. model. This study reduced the computational complexity of the backend server with the A.I. model to elevate backend server performance. Furthermore, the authors propose a hash function to determine whether the machines added new training data. The experimental results revealed that the proposed method could reduce the computational complexity of federated learning and that private blockchains can be applied to ensure parameter confidentiality and completeness. In summary, this research uses software computing methods to save machine learning data transmission, reduce network load, and provide privacy protection for parameter data without updating existing production equipment so that small-cost enterprises can import Industry 4.0.

KEYWORDS

Artificial Intelligence, Blockchain, Clustering, Federated Learning, Industry 4.0, Machine Learning, Network Security, Smart Factory

INTRODUCTION

Material requirement planning (MRP) emerged when people began using information computations to assist manufacturing and production operations. M.R.P. is a decision-making model for cost management (i.e., raw material purchase schedules and quantities) after manufacturers receive purchase orders. As computer performance evolves over time, MRP is gradually incorporated into the master production schedule (MPS.), rough-cut capacity planning (RCCP), and capacity requirement planning (CRP). It is combined with raw materials, human resources, equipment, and costs to form a manufacturing resource plan (MRP-II). Because material requirement planning and manufacturing resource planning share the same acronym, they are referred to as "MRP I" and "MRP II," respectively. Over time, enterprises began to integrate MRP II into accounting, finance, human resources, and other MRP II-related systems to develop contemporary enterprise resource planning (ERP). MRP II is associated with the effective use of resources in operational job scheduling on shop floors to decrease production costs and increase production capacity. Namely, it signifies that production lines on shop floors can comprise different workers and production equipment working together to

DOI: 10.4018/JOEUC.313194 *Corresponding Author

manufacture products. Over the past few decades, many studies regarding job sequencing theories and other applications have been conducted. (R. Ruiz &T. Stützle.,2007; Luo et al.,2013; Zhao et al.,2021). Commonly seen scheduling models include job sequencing, flow shop, job shop, mixed shop (D-Shop), and open shop (Noor, Sahar, et al., 2009).

The Internet of Things (IoT) has emerged following the confluence of Internet ubiquity and the minimization of the chip manufacturing process. The application of IoT in shop floors enables manufacturing equipment to display operating equipment-related information to on-site personnel in real-time or transmit related parameters to backend servers to perform related artificial intelligence (A.I.) computations that allow the optimal production parameters of various items to be obtained. On a shop floor, each production line is composed of a variety of machines and personnel. Before switching to different machines to produce different things, the machines may need to be turned off to allow parameters to be adjusted before restarting. Thus, parameters may need to adjust in response to various scheduling patterns, necessitating different settings. Industry 4.0 refers to transmitting production equipment parameters to backend computation systems to obtain optimal machine operation parameters under certain scheduling constraints. Many Industry 4.0 devices rely on A.I. models to conduct intelligent automated processes. Servers must constantly train the A.I. models to enhance identification accuracy and improve production yield. The concept of federated learning (F.L.) has been developed to help devices calculate relevant parameters on the client side before encrypting and sending the data to the server. After collecting the uploaded data, the server generates an optimal solution through gradient descent. Finally, the server sends the optimal solution back to the client side for updates, improving the A.I. model accuracy. The FL technique only needs to collect the parameters of the device without needing to collect client-side information, thus avoiding privacy concerns.

During FL computation processes, deep learning requires the transmission of a significant quantity of data to backend servers for training and testing. The term "client-side" refers to the production equipment and machines; all machines on shop floors can be linked to a network and exchange data with backend servers. Although these client-side devices use private Internet protocol (I.P.) addresses and both client-side and backend servers belong to an enterprise's intranet, information security and management problems may occur. For example, unlike general desktop computers, the computer operating systems for controlling client-side production equipment do not undergo security updates or version updates when their developers introduce updates. Because such production equipment and machines are customized devices closely associated with the kernel components of operating systems, unexpected updates of the computer operating systems controlling devices may cause the control interface of relevant devices to malfunction and production lines to shut down. Malfunctioning of the computers controlling production line machines results in serious enterprise losses. In addition, using computers that are too old entails possible encryption problems related to information transmission. When linked to networks, production line equipment may come under distributed denial-of-service (DDoS) attacks. In an enterprise intranet, if a trojan horse infects one employee's personal computer, the infection may spread to production machines. Moreover, the production machines may become part of the DDoS attack and attack the servers via the intranet. Firewalls that are used to prevent external hacker attacks do not provide protection against attacks from the intranet. This type of DDoS on backend servers can cause production line failure.

Production line failure is mainly caused by the large computational load of backend servers and DDoS. From the perspective of the backend servers, each production line on a shop floor is composed of different production machines. The backend servers must update the corresponding A.I. models. From the perspective of the frontend servers, if each production machine computes parameters independently and transmits the results to the backend servers for integrated computation, the backend servers may need to receive and process large quantities of data. In addition, among the production devices of an enterprise, there may be numerous machines that belong to the same model and code. Thus, when such machines transmit data, the computational complexity of backend servers may increase because they must identify the different frontend devices. The optimal production parameters

required for frontend devices of the same model may be identical. Separating such parameters to the backend servers may cause redundant computations and increased loading. Backend servers in the original F.L. application already collect massive computation data. The backend servers must compute existing parameters and update optimal parameters for each cline-side A.I. module. Thus, when the backend servers are under DDoS attacks, the overwhelming workload will terminate the client-side parameter updating services, causing production line failure. On the other hand, F.L. has the following four problems: P1. High computational complexity and time-consuming optimal parameter calculation; P2. A.I. models need to download the optimal parameters from the server, causing extra loading on the server; P3. The encryption technology must be lightweight in privacy protection to lower communication and calculation costs; P4. The server cannot continue updating the optimal parameters if it encounters attacks.

The study wants to achieve uses software computing methods to save machine learning data transmission, reduce network load and provide privacy protection for parameter data without updating existing production equipment so that small-cost enterprises can import Industry 4.0. Therefore, this research presents a dendrogram clustering F.L. system suited to Industry 4.0 that utilizes dendrogram clustering to group A.I. models, allowing the A.I. models with diverse requirements to integrate the optimal parameters. The features of this article are as follows:

1. Every parameter that a machine calculates goes through the hash algorithm, and the server will eliminate identical parameters to reduce calculation workload.
2. After the calculation, the server will redirect the signal to the blockchain for those devices with the same parameters to download updated data.
3. The blockchain parameters enable user devices to verify authenticity quickly.
4. The blockchain nodes will constantly update the optimal parameters and record the updating conditions of each machine.
5. Each machine will automatically verify the device identification by smart contracts and update the optimal parameter.

The experimental results are divided into two parts: 1. F.L. results; 2. Blockchain results. The experimental performance has shown that the approach can lower the calculation, improve the recognition rate of user devices, and enhance A.I. model accuracy in Industry 4.0, boosting yields and productivity.

BACKGROUND AND RELATED WORKS

This research employs the F.L. technique to protect user devices and privacy. The issues of decreasing F.L. workload and communication costs have been discussed in many studies. For instance, the system in Literature (H. Chai et al., 2021) collects the car behaviors on the Internet of Vehicles (IoVs) through local stations and conducts the optimal parameter training via F.L., achieving the optimal conditions of IoVs. Moreover, the system increases the calculation speed and data volume through stratified training. Literature (P. Zhang et al., 2021) uses F.L. to update the equipment parameters in the Industrial Internet of Things (IIoT); nonetheless, due to the lower calculation capacity in IIoT, the F.L. technique is in charge of transmitting the optimal calculated parameters to the IIoT for updates, which improves IIoT's accuracy. The mechanism in Literature (S. Oh et al., 2020) delivers the identifying requests of A.I. models to user devices. When the user device collects specific error rates, the system calculates the parameters and sends the result back to the server for optimal computing, reducing network communication burdens and workload. Literature (Y Chen et al., 2020) collects IIoT data by adding a weighting mechanism; the system will select the data with higher weighting to execute parameter training for lowering communication costs and improving AI-model accuracy.

The user device in Literature (F. Sattler et al., 2020) sends the selected parameters to the server for calculating the optimal results after three layers of filters, cutting the transmission volume and costs. Apart from applying F.L. in IIoT, the technique can also operate in smart healthcare. The system in Literature (M. A. Rahman et al., 2021) delivers users' health information collected from a wearable device to F.L.; the F.L. technique will calculate the A.I. model's optimal parameters to maintain the best setting helping the device judge users' health conditions. The system in Literature (R. Kumar et al., 2021) scans patients' X-ray pictures to evaluate health conditions; to enhance accuracy, the researcher uses error data to develop the optimal parameters by F.L. and update the A.I. model for higher performance. Literature (H. Jin et al., 2021) attempts to run the optimal training on all A.I. models in the local network at a hospital, allowing the IIoT detection and intelligent recognition to reach higher accuracy. Finally, Literature (D. Połap, 2021) saves the medical equipment parameters in a database for F.L. to find the optimal setting, updating relevant data in the hospital and work stations through the database.

Furthermore, the design in Literature (Y. Zhan et al., 2020) encourages users to send their device parameters to the server for optimal setting training and enables nearby devices to share the data. Literature (L. Ahmed et al., 2020) trains and tests samples by active learning, and F.L., the optimal parameters generated by the combination can lower the recognition rate. To reduce the computing workload of the server, Literature (P. Tam et al., 2021) collects user data to train on the local side. Afterward, transmit the obtained parameters to the server for further calculation, and the process at the local side helps ease the communication loading of the server. In Literature (D. C. Attota et al., 2021), the study introduces the network security mechanism for the IoT; the F.L. technique prevents the server I.P. address from exposing, successfully avoiding DDoS attacks. Finally, Literature (I. Donevski et al., 2021) updates drone parameters by F.L., benefiting the drone equipment's recognition accuracy.

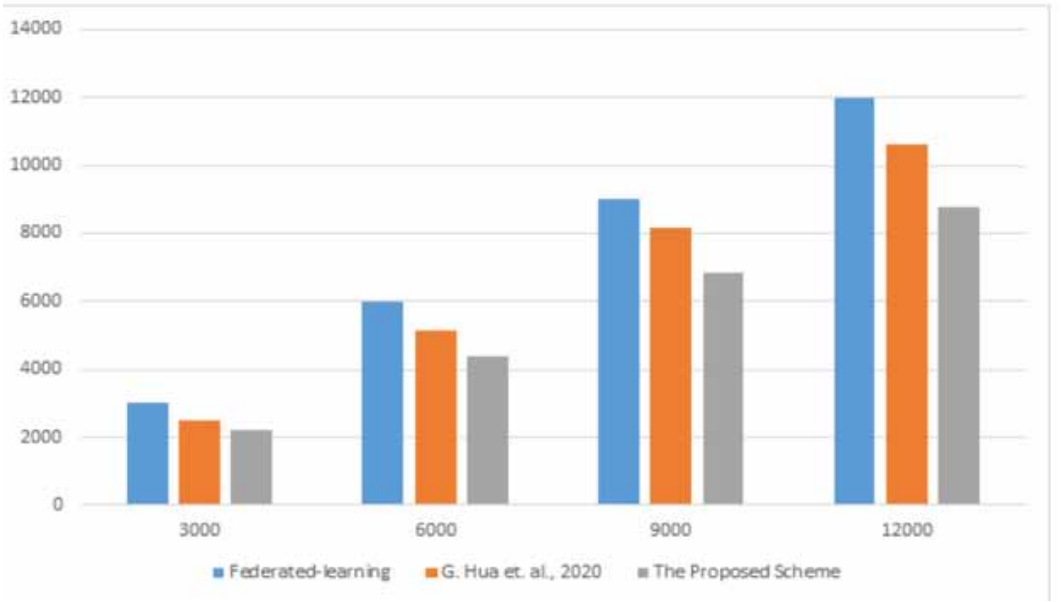
Literature (Y. Lu et al., 2019) stores user device parameters in a blockchain and F.L. can collect the parameters from the blockchain and find the optimal setting. Literature (W. Zhang et al., 2020) suggests an error detection method for IoT. Firstly, the system creates a Merkle Tree to record the result of each IoT recognition rate and then calculate the results accordingly; if the differences vary significantly, meaning the IoT device is abnormal. The system in Literature (G. Hua et al., 2020) updates the machine learning equipment synchronously by blockchains. Literature (H. Kim et al., 2019) utilizes blockchains to execute parameter sharing and updates. Literature (M. H. u. Rehman et al., 2021) employs blockchains to ensure data privacy and security in F.L., and the method allows many user devices to update the optimal parameters. Hence, this article presents a dendrogram clustering federated-learning system based on industry 4.0, and the system stores the optimal parameters in blockchains for user devices to update to the optimal setting from a blockchain node when needed. Dhiman, Gaurav, et al. (2017), Dhiman, Gaurav, et al. (2018), and Dhiman, G. et al. (2019) have proposed a novel bio-inspired heuristic algorithm. Satnam Kaur et al. (2022) also presented another heuristic algorithm based on the Tunicate Swarm Algorithm. On the other hand, Gupta V. K. et al. (2022) suggested a crime tracking system based on machine learning, and Kumar, R. et al. (2021) proposed the optimization of fuzzy algorithms. Finally, Chatterjee, I. (2021) analyzed the patentability possibility of A.I. models, and Vaishnav P. K. (2021) provided a rapid test analysis for COVID-19.

THE PROPOSED SCHEME

System Model

The system model is as Figure 1. Firstly, each machine will check whether there are new files or pictures added to the device by the hash function. The device will add new files when the system notices different product information; meanwhile, when the files are different, the hash value differs as well. Thus, the machine will send the calculation result to the base station to process the hash value of the device, eliminate those repetitive parameters, and transmit existing calculations to blockchain nodes

Figure 1. System Model



for further access, which can reduce the workload of repetitive tasks. Additionally, after calculating the optimal parameters, the server will save the data directly to blockchain nodes for each machine to access nearby locations and reduce the communication workload. The approach developed in this study focuses on smart contracts; each machine can auto-calculate through the smart contract to process the parameter calculation and update the data in blockchain nodes. Furthermore, the system verifies each blockchain parameter to ensure data completeness and privacy, preventing update errors from data tampering in nodes. Under the F.L. mechanism, this article collects the A.I. models' parameters from each device to optimize and update them via blockchain, enhancing the A.I. model's recognition rate. Originally, if we wanted to increase the recognition rate of each A.I. model, the system must transmit the collected data to the server for training; yet, this will cause privacy issues and produce massive communication costs. Consequently, we build privacy protection and lower the training data volume under the F.L. mechanism. Table 1 lists the symbols used in this article.

Smart Contract and Network Security Setting

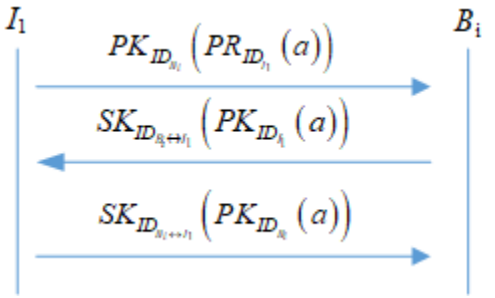
This study utilizes Public Key Infrastructure (PKI) to conduct smart contract verification and private communication. The below shows the system initialization process:

1. **The setting of the server is:** Public key ($\mathcal{PK}_{\mathbb{ID}_{TA}}$), private key ($\mathcal{PR}_{\mathbb{ID}_{TA}}$), and certificate ($\mathcal{CER}_{\mathbb{ID}_{TA}}$).
2. **The setting parameters in each base station are:** Public key ($\mathcal{PK}_{\mathbb{ID}_{B_i}}$), private key ($\mathcal{PR}_{\mathbb{ID}_{B_i}}$), and certificate ($\mathcal{CER}_{\mathbb{ID}_{B_i}}$).
3. **The setting parameters in each machine:** Public key ($\mathcal{PK}_{\mathbb{ID}_{I_i}}$), private key ($\mathcal{PR}_{\mathbb{ID}_{I_i}}$), and certificate ($\mathcal{CER}_{\mathbb{ID}_{I_i}}$).

Table 1. Notation table

Notation	Description	Notation	Description
$\mathcal{PK}_{\mathbb{ID}_{TA}}$	The public key of user $\mathbb{ID}_{TA}$.	$\Theta_{ID_{I_n}}$	The data set of user $\mathbb{ID}_n$.
$\mathcal{PR}_{\mathbb{ID}_{TA}}$	The private key of user $\mathbb{ID}_{TA}$.	$L_{ID_{B_1}}$	The optimal parameter of user B_1.
$\mathcal{CER}_{\mathbb{ID}_{TA}}$	The certificate of user $\mathbb{ID}_{TA}$.	$\frac{\partial L}{\partial \omega}$	The optimal value
$\mathcal{SK}_{TA \leftrightarrow ID_{B_u}}$	The common secret key.	H	The hash function.

This paragraph describes the verification calculation of the machine and the base station at the beginning. Firstly, the machine (I_1) will choose a random number ($a \in Z_q^*$); later, the base station will use the public key to encrypt the data. The base station will then decrypt the data by the private key when receiving the encrypted message; combining with the public key of the machine, the base station can obtain the random number a. Afterward, the machine and the base station will treat an as the secret key of symmetric encryption to initiate a private communication. The encryption communication process is as follows:

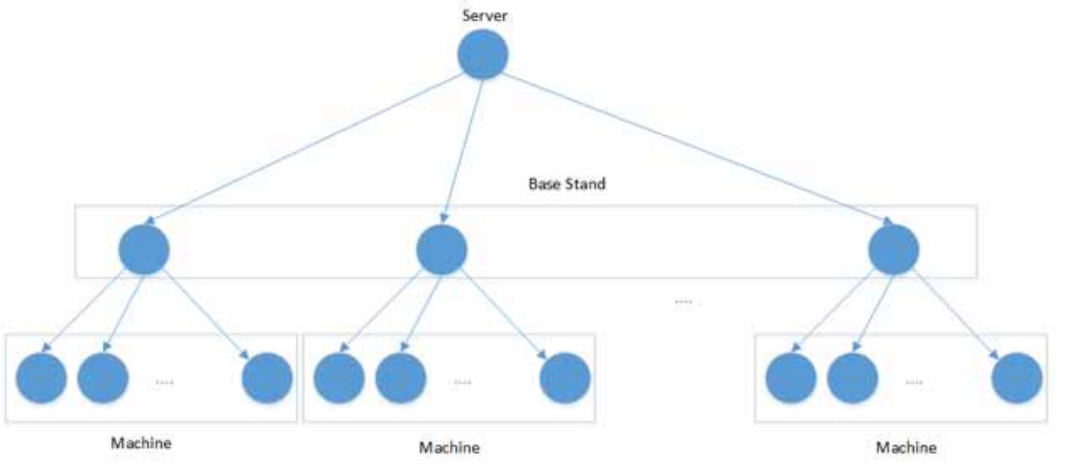


The smart contract enables the machine and the base station to verify identities, and the system automatically calculates the optimal parameters. Next, the smart contract will update the new parameters in the blockchain, which safeguards communication and information security.

The Dendrogram Clustering Federated-Learning System

The dendrogram clustering federated-learning system is demonstrated in Figure 2. Firstly, each machine will check whether there is new information regularly. Then, set H to represent the hash function, and $\mathcal{F}_{\mathbb{ID}_{I_i}}$ as the current machine information, the formula of the hash value is $H\left(\mathcal{F}_{\mathbb{ID}_{I_i}}\right)$. Further, the system will compare the new and existing hash values; if the values are different, the system will need to run an optimal calculation. If $H\left(\mathcal{F}_{\mathbb{ID}_{I_i}}\right) \neq H\left(\mathcal{F}_{\mathbb{ID}_{I_i}}\right)'$, the machine will process the A.I. model $\Theta_{ID_{I_1}}$ and transmit the encrypted data to the base station. Next, the base station will

Figure 2. Hierarchical Clustering Federated-learning System



run the hash function individually after decrypting the message. For example, if the data $\Theta_{ID_{I_n}}$ has existed in the data set, the station will eliminate the data, and the formula is:

$$\Theta_{ID_{I_n}} \in H\left(\Theta_{ID_{I_1}} \,||\, \Theta_{ID_{I_2}} \left\|...\right\| \Theta_{ID_{I_{n+1}}}\right)$$

Moreover, the station will calculate the optimal parameter $L_{ID_{B_1}}$, between $\Theta_{ID_{I_1}}$ and $\Theta_{ID_{I_{n+1}}}$, send the result to the server. Finally, the server will save the recently calculated parameters in a parameter cluster $L_{ID_{B_1}} \,||\, L_{ID_{B_i}} \left\|...\right\| L_{ID_{B_n}}$. Furthermore, the server will judge if the parameter is repetitive. If so, the server will provide the data to the blockchain node for base stations to download the optimal parameters; if not, the server will run the optimal parameter calculation by the below formula:

$$result = \begin{Bmatrix} True, L_{ID_{B_j}} \in L_{ID_{B_1}} \,||\, L_{ID_{B_i}} \left\|...\right\| L_{ID_{B_n}} \\ false, other \end{Bmatrix} \tag{1}$$

If the result is True, the parameter is a repetitive value; the server will notify the base station and allow devices to download from the blockchain node directly. If the result is False, the optimal calculation is as follows:

$$\frac{\partial L}{\partial \omega} = \frac{1}{n}\left[L_{ID_{B_1}} - L_{ID_{B_2}} - ... - \,||\, L_{ID_{B_n}}\right] \tag{2}$$

Afterward, the server will transmit the optimal value $\frac{\partial L}{\partial \omega}$ to the base station nodes for devices to update parameters from the node.

Blockchain Update and Verification

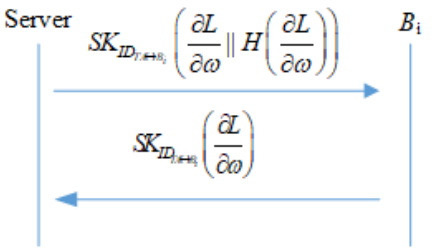
The server will execute private communication by the secret key with the base station after attaining the optimal parameter to transmit the data and utilize its private key to encrypt the data and calculate the hash value by a hash function before sending the information to a base station. The encryption formula is:

$$\mathcal{SK}_{TA\leftrightarrow ID_{B_u}}\left(\mathcal{PR}_{\mathbb{ID}_{TA}}\left(\frac{\partial L}{\partial \omega}\,||\,H\left(\frac{\partial L}{\partial \omega}\right)\right)\right)$$

When the base station receives the message, it decrypts the data with the secret key and the server's public key. Next, the base station will calculate the hash value $\left(H\left(\frac{\partial L}{\partial \omega}\right)'\right)$ of $\frac{\partial L}{\partial \omega}$. If $H\left(\frac{\partial L}{\partial \omega}\right)' = H\left(\frac{\partial L}{\partial \omega}\right)$, the data transmission is complete. Afterward, the base station employs the private key to encrypt the data and deliver the message to its blockchain node; the encrypted formula is:

$$\mathcal{PR}_{\mathbb{ID}_{B_u}}\left(\mathcal{PR}_{\mathbb{ID}_{TA}}\left(\frac{\partial L}{\partial \omega}\,||\,H\left(\frac{\partial L}{\partial \omega}\right)\right)\left\|L_{B_u}\right\|T\right)$$

Additionally, the encryption and decryption process between the server and the base station is:



After the base station updates the node, it will conduct a private communication with the device by the secret key, encrypt the data by:

$$\mathcal{PR}_{\mathbb{ID}_{B_u}}\left(\mathcal{PR}_{\mathbb{ID}_{TA}}\left(\frac{\partial L}{\partial \omega}\,||\,H\left(\frac{\partial L}{\partial \omega}\right)\right)\left\|L_{B_u}\right\|T\right)$$

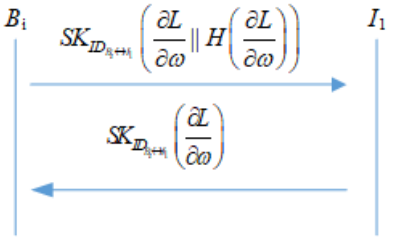
and transmit the encrypted message to the device. Moreover, the symmetric encryption formula is:

$$\mathcal{PR}_{\mathbb{ID}_{B_u}}\left(\mathcal{PR}_{\mathbb{ID}_{TA}}\left(\frac{\partial L}{\partial \omega}\,||\,H\left(\frac{\partial L}{\partial \omega}\right)\right)\left\|L_{B_u}\right\|T\right)$$

Next, the machine will decrypt the data by the secret key with the public key from the base station and the server. Finally, the machine calculates the hash value $H\left(\frac{\partial L}{\partial \omega}\right)^{'}$ of $\frac{\partial L}{\partial \omega}$. If $H\left(\frac{\partial L}{\partial \omega}\right)^{'} = H\left(\frac{\partial L}{\partial \omega}\right)$, the data transmission is complete, and the machine can update the parameter by formula:

$$\mathcal{PR}_{\mathbb{ID}_{I_u}}\left(\mathcal{PR}_{\mathbb{ID}_{TA}}\left(\frac{\partial L}{\partial \omega} \,||\, H\left(\frac{\partial L}{\partial \omega}\right)\right)\middle\|\Theta_{ID_{I_u}}\middle\| T\right)$$

and update the information to the blockchain node. The encryption and decryption transmission formula is as follows:



For anyone who wants to access the device or the blockchain node of the base station, the person needs to have the public key and hash function to verify the authenticity and integrity of the message, ensuring the data is provided by the server.

EXPERIMENT RESULTS

Network Security Analysis

This article employs PKI and symmetric encryption to encrypt messages. In cryptology, PKI and symmetric encryption are recognized as secured approaches, especially the encryption mechanism of PKI is developed with Elliptic Curve Cryptography generated by complex mathematical equations, which are difficult to tackle. As a result, the proposed network security mechanism can significantly ensure transmission safety. Table 2 illustrates the processing speed of various encryption methods. Table 3 compares the network security mechanism suggested in this study with others, showing that our approach verifies authenticity and integrity by encryption. On the other hand, bitcoins verify by the miner's hash algorithm, which is time-consuming and unacceptable for real-time systems. Our system verifies identities by PKI. After confirming the message's origin, the system will compare

Table 2. Execution time in milliseconds

Notation	Description	Execution time (ms)
H	HMAC	0.002
ASE	R.S.A. encryption	0.19
ASD	R.S.A. Decryption	4.65
S_e	A.E.S. Encryption	<0.19
S_d	A.E.S. Decryption	<4.65

Table 3. Effectiveness Analysis

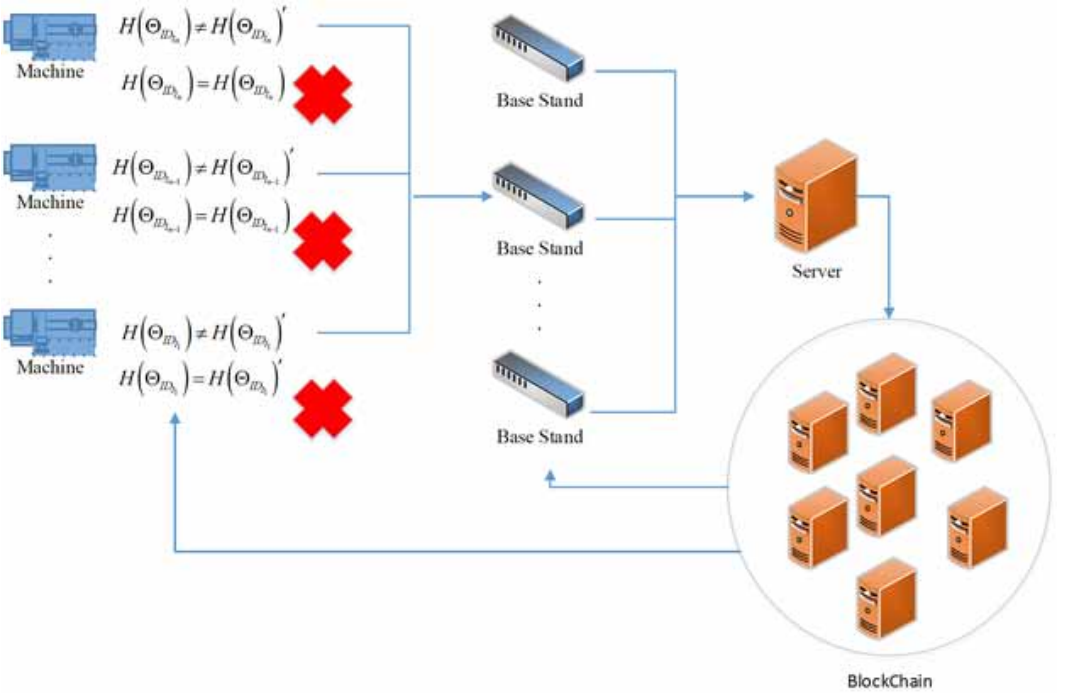
Property / Method	(H. Yang et al., 2021)	(Y. Wang et al., 2021)	(D. Liu and O. Simeone, 2020)	The Proposed Scheme
Encryption/Decryption	Encryption: 2**ASE* Decryption: 2**ASD* Spending time: 9.68 (ms)	Encryption: 2**ASE* Decryption: 2**ASD* Spending time: 9.68 (ms)	Encryption: 4**ASE* Decryption: 4**ASD* Spending time: 19.36 (ms)	Encryption: S_e Decryption: S_d Spending time: 4.65 (ms)

the plaintext with the hash value; if the results are correct, the information is complete. Furthermore, the system has a timestamp (T) to confirm the time source; the above explanation proves that the suggested method fulfills the requirements of developing blockchains.

Data Workload Verification

The dendrogram clustering federated-learning system primarily filters the data layer by layer to eliminate calculated or the same data to reduce the computing workload; the larger the data in federated learning, the heavier the server computing workload of gradient descent. Figure 3 demonstrates the differences between the suggested approach and the workload that does not eliminate redundant data; in other words, the massive amount of the same or calculated data will cause the communication workload in each device. Additionally, as the federated-learning system relies on the server to calculate data and update the devices, the server fails to provide the required services if it has encountered malicious attacks; the implementation of blockchain can significantly avoid such kinds of attacks and enable users to access the data easily. Therefore, the research has adopted dendrogram clustering

Figure 3. Experiment Result



to solve this issue; when a node has been attacked, other devices can access the data from the upper layer or nearby nodes, ensuring the entire system's operation.

The hardware and software equipment is shown in Table 4. We calculate the parameters using the TensorFlow Federated framework on mobile phones. In this article, we used 15 mobile phones in total, as shown in the Scenario Diagram in Figure 4. The system will share the parameters to the server for Hash calculation, eliminating the same values and only leaving the values with different Hash values.

The x-axis in Figure 3 illustrates that the mobile phones produced 3,000, 6,000, 9,000, and 12,000 pieces of parameters, and the y-axis shows the execution time (ms). The computational workload will significantly increase when sending a massive amount of parameters to the server for training. This article utilizes the blockchain platform developed by the N.A.R. Labs at the National Center for High-performance Computing, allowing the mobiles to download information directly from the nodes; moreover, the blockchain technique can prevent DDoS attacks. An effectiveness analysis comparing the different methods proposed by different studies is shown in Figure 3. According to the table, removing identical parameters enables the server to outperform other methods in terms of its effectiveness. Figure 5 is the cell phone screenshot of one of the machines at a certain time. This screen records the current parameters of the machine running and the content of the last updated parameters.

Table 4. Hardware and Software Equipment

Software	Hardware
TensorFlow Federated	Server
remix	Mobile Device
Python	

Figure 4. The Scenario Diagram for flow shop

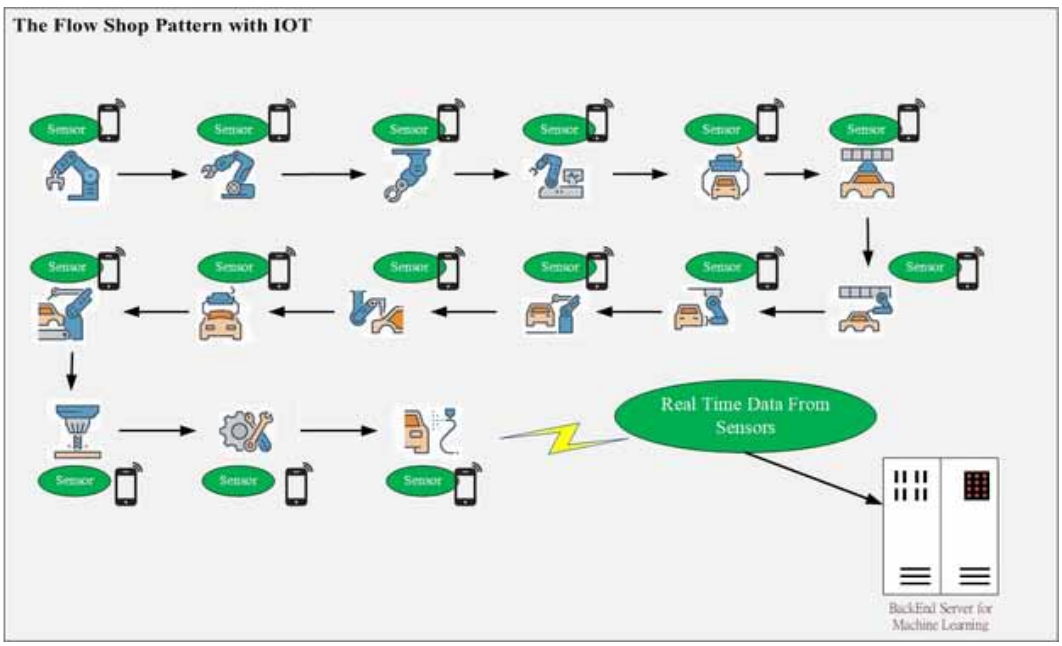
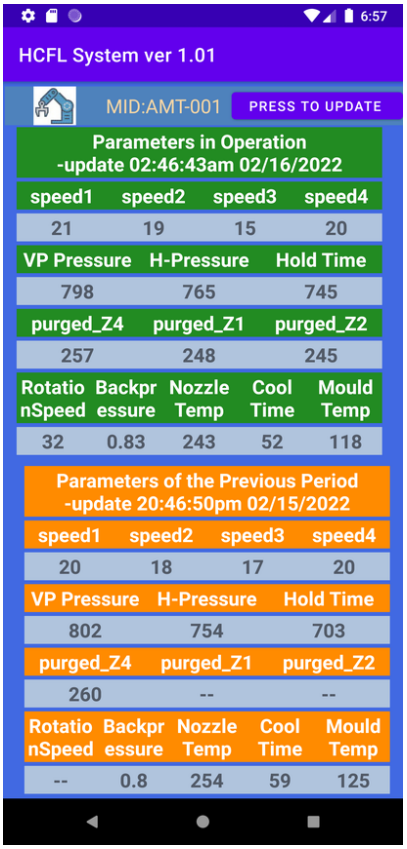


Figure 5. The cellphone Screenshot of one of the machines



CONCLUSION

This article proposes a dendrogram clustering federated-learning system based on industry 4.0. Because a federated-learning system needs to process massive parameters, it is a heavy workload for the server. Moreover, when encountering attacks, the system can allow other nodes to provide services due to the blockchain technique, which secures data safety, and any device can verify the data source on the node. The experimental result has proved the data security of the proposed method. Furthermore, the system can lower the server workload with the layer-by-layer filter to reduce data calculation. In conclusion, the suggested way can enhance the update of the Industry 4.0 models, helping to improve production yields and machine recognition rate.

Moreover, for the F.L. mechanism, this research collects the parameters of the A.I. models from each device to optimize and update the data by leveraging blockchain technology, enhancing the A.I. model's recognition rate. To increase the recognition rate, an A.I. model must send the collected data samples to the server for parameter training; however, such a step causes privacy issues and boosts communication costs due to the massive data transmission. Therefore, the approach proposed in this article entails constructing a protection mechanism and decreasing the parameter training volume under the F.L. technique. Overall, on the management side, this research contributes to making small-cost enterprises import Industry 4.0 without purchasing new equipment; on the technical side, The method proposed in this study has the following contributions:

1. Backend server performance optimization.

The system can group and distribute different A.I. model functions into corresponding servers for computation. This design reduces the backend server computational loading and optimizes the A.I. model parameter computation performance.

2. Decrease in original F.L. computational complexity.

The research uses a hash algorithm to lower the F.L. workload. Moreover, all training data must be sent to servers to train eigenvalues to enhance the recognition rates of A.I. models. However, sending large samples to servers increases server data workload. Accordingly, this study adopted F.L. to update the eigenvalues of A.I. models, thereby protecting user privacy and elevating A.I. model recognition rates.

3. Privacy protection mechanisms to prevent unauthorized alteration.

This paper used a blockchain mechanism based on R.S.A. asymmetric encryption and decryption for identity verification. Private keys for both ends were created to facilitate private communication. Hash functions are used to ensure the integrity of data. It can prevent data from being stolen or exposed to a third party by using private communication during transmission.

4. Combination with blockchain technologies to prevent DDoS attacks and protect backend servers in the continuous transmission of updated parameters.

Combining a blockchain mechanism with the server is necessary to avoid DDoS attacks, enabling client-side devices to identify optimal parameters from various nodes. This study performed classification calculations for the different A.I. models and used blockchain to update model parameters, allowing the A.I. models of each device to obtain optimal eigenvalues.

Finally, Some limitations should be noted. First, this paper only used a subnet under the same Internet communication protocol. The results may be less favorable if different subnets are involved. Secondly, the experiment of this study is to establish a simulated environment in the factory, which excludes the transmission behavior of other data in the network in the daily administrative work of the factory. However, In real-world enterprise environments, various data may be transmitted in the network backbones in addition to the manufacturing machines' parameters. Thus, in real-world operations, the bandwidths of network backbones and the loading of other data transmitted by the equipment must be considered.

CONFLICT OF INTEREST

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

FUNDING AGENCY

This work was supported in part by the Ministry of Science and Technology of Taiwan, R.O.C., under Contracts MOST 109-2622-E-197-012, MOST 110-2622-E-197-015, and MOST 111-2410-H-197 -002.

REFERENCES

Ahmed, Ahmad, Said, Qolomany, Qadir, & Al-Fuqaha. (2020). Active Learning Based Federated Learning for Waste and Natural Disaster Image Classification. *IEEE Access, 8,* 208518–208531. .10.1109/ACCESS.2020.3038676

Attota, M. Parizi, & Pouriyeh. (2021). An Ensemble Multi-View Federated Learning Intrusion Detection for IoT. *IEEE Access, 9*, 117734 – 117745. DOI:doi:10.1109/ACCESS.2021.3107337

Chai, H., Leng, S., Chen, Y., & Zhang, K. (2021). A Hierarchical Blockchain-Enabled Federated Learning Algorithm for Knowledge Sharing in Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *22*(7), 3975–3986. doi:10.1109/TITS.2020.3002712

Chatterjee, I. (2021). Artificial Intelligence and Patentability: Review and Discussions. *International Journal of Modern Research*, *1*, 15–21.

Chen, Y., Sun, X., & Jin, Y. (2020). Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(10), 4229–4238. doi:10.1109/TNNLS.2019.2953131 PMID:31899435

Dhiman, G., & Chahar, V. (2017). Spotted Hyena Optimizer: A Novel Bio-inspired based Metaheuristic Technique for Engineering Applications. *Advances in Engineering Software*, *114*, 48–70. doi:10.1016/j.advengsoft.2017.05.014

Dhiman, G., & Chahar, V. (2018). Emperor Penguin Optimizer: A Bio-inspired Algorithm for Engineering Problems. *Knowledge-Based Systems*, *159*, 20–50. Advance online publication. doi:10.1016/j.knosys.2018.06.001

Dhiman, G., & Kaur, A. (2019). STOA: A bio-inspired based optimization algorithm for industrial engineering problems. *Engineering Applications of Artificial Intelligence*, *82*, 148–174. doi:10.1016/j.engappai.2019.03.021

Donevski, Babu, Nielsen, Popovski, & Saad. (2021). Federated Learning With a Drone Orchestrator: Path Planning for Minimized Staleness. *IEEE Open Journal of the Communications Society, 2*, 1000 – 1014. DOI:.2021.307200310.1109/OJCOMS

Gupta, V. K., Shukla, S. K., & Rawat, R. S. (2022). Crime tracking system and people's safety in India using machine learning approaches. *International Journal of Modern Research*, *2*(1), 1–7.

Hua, Zhu, Wu, Shen, Zhou, & Lin. (2020). Blockchain-Based Federated Learning for Intelligent Control in Heavy Haul Railway. *IEEE Access, 8*, 176830 – 176839. 10.1109/ACCESS.2020.3021253

Hua, G., Zhu, L., Wu, J., Shen, C., Zhou, L., & Lin, Q. (2020). Blockchain-Based Federated Learning for Intelligent Control in Heavy Haul Railway. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 176830–176839. doi:10.1109/ACCESS.2020.3021253

Jin, H., Dai, X., Xiao, J., Li, B., Li, H., & Zhang, Y. (2021). Cross-Cluster Federated Learning and Blockchain for Internet of Medical Things. *IEEE Internet of Things Journal*, *8*(21), 15776–15784. doi:10.1109/JIOT.2021.3081578

Kaur, S., Awasthi, L. K., Sangal, A. L., & Dhiman, G. (2020). Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence*, *90*, 103541. Advance online publication. doi:10.1016/j.engappai.2020.103541

Kim, H., Park, J., Bennis, M., & Kim, S.-L. (2019). Blockchained On-Device Federated Learning. *IEEE Transactions on Industrial Informatics*, *24*(6), 1279–1283. doi:10.1109/LCOMM.2019.2921755

Kumar, R., & Dhiman, G. (2021). A Comparative Study of Fuzzy Optimization through Fuzzy Number. *International Journal of Modern Research*, *1*, 1–14.

Kumar, R., Khan, A. A., Kumar, J., Zakria, N. A. G., Zhang, S., Ting, Y., Zheng, C., & Wang, W. (2021). Blockchain-Federated-Learning and Deep Learning Models for COVID-19 Detection Using CT Imaging. *IEEE Transactions on Neural Networks and Learning Systems*, *21*(14), 16301–16314. doi:10.1109/JSEN.2021.3076767 PMID:35789224

Liu & Simeone. (2020). Privacy for Free: Wireless Federated Learning via Uncoded Transmission With Adaptive Power Control. *IEEE Journal on Selected Areas in Communications*, *39*(1), 170–185. doi:10.1109/JSAC.2020.3036948

Lu, Y., Huang, X., Dai, Y., Maharjan, S., & Zhang, Y. (2019). Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT. *IEEE Transactions on Industrial Informatics*, *16*(6), 4177–4186. doi:10.1109/TII.2019.2942190

Luo, Du, Huang, Chen, & Li. (2013). Hybrid flow shop scheduling considering machine electricity consumption cost. *International Journal of Production Economics*, *146*(2), 423-439.

Noor, , Khan, , Hussain, , & Khan, , Syed, Shah, & Babar. (2009). GA-Based Scheduling System for Flow Shop and Job Shop Scheduling Problems. *Journal of Engineering and Applied Sciences (Asian Research Publishing Network)*, *28*, 77–88.

Oh, S., Park, J., Jeong, E., Kim, H., Bennis, M., & Kim, S.-L. (2020). Mix2FLD: Downlink Federated Learning After Uplink Federated Distillation With Two-Way Mixup. *IEEE Communications Letters*, *24*(10), 2211–2215. doi:10.1109/LCOMM.2020.3003693

Po3ap, D. (2021). Fuzzy Consensus With Federated Learning Method in Medical Systems. *IEEE Access: Practical Innovations, Open Solutions*, *9*, 150383–150392. doi:10.1109/ACCESS.2021.3125799

Rahman, Hossain, Islam, Alrajeh, & Muhammad. (2021). Secure and Provenance Enhanced Internet of Health Things Framework: A Blockchain Managed Federated Learning Approach. *IEEE Access, 8*, 205071 –205087. 10.1109/ACCESS.2020.3037474

Ruiz, R., & Stutzle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, *177*(3), 2033–2049. doi:10.1016/j.ejor.2005.12.009

Sattler, F., Wiedemann, S., & Muller, K.-R. (2020). Robust and Communication-Efficient Federated Learning From Noni. i.d. Data. IEEE Transactions on Neural Networks and Learning Systems, 31(9), 3400 – 3413. doi:10.1109/TNNLS.2019.2944481

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, *64*(2), 278–285. doi:10.1016/0377-2217(93)90182-M

Tam, P., Sa Math, C. N., & Kim, S. (2021). Adaptive Resource Optimized Edge Federated Learning in Real-Time Image Sensing Classifications. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 10929–10940. doi:10.1109/JSTARS.2021.3120724

ur Rehman, Dirir, Salah, Damiani, & Svetinovic. (2021). TrustFed: A Framework for Fair and Trustworthy Cross-Device Federated Learning in IIoT. *IEEE Transactions on Industrial Informatics, 17*(12), 8485 – 8494. 10.1109/TII.2021.3075706

Vaishnav, P. K., Sharma, S., & Sharma, P. (2021). Analytical Review Analysis for Screening COVID-19. *International Journal of Modern Research*, *1*, 22–29.

Wang, Su, Zhang, & Benslimane. (2021). Learning in the Air: Secure Federated Learning for UAV-Assisted Crowdsensing. *IEEE Transactions on Network Science and Engineering, 8*(2), 1055 – 1069. 10.1109/TNSE.2020.3014385

Yang, H., Zhao, J., Xiong, Z., Lam, K.-Y., Sun, S., & Xiao, L. (2021). Privacy-Preserving Federated Learning for UAV-Enabled Networks: Learning-Based Joint Scheduling and Resource Management. *IEEE Journal on Selected Areas in Communications*, *39*(10), 3144–3159. doi:10.1109/JSAC.2021.3088655

Zhan, Y., Li, P., Qu, Z., Zeng, D., & Guo, S. (2020). A Learning-Based Incentive Mechanism for Federated Learning. *IEEE Internet of Things Journal*, *7*(7), 6360–6368. doi:10.1109/JIOT.2020.2967772

Zhang, P., Sun, H., Situ, J., Jiang, C., & Xie, D. (2021). Federated Transfer Learning for IIoT Devices With Low Computing Power Based on Blockchain and Edge Computing. *IEEE Access: Practical Innovations, Open Solutions*, *9*, 98630–98638. doi:10.1109/ACCESS.2021.3095078

Zhang, W., Lu, Q., Yu, Q., Li, Z., Liu, Y., Lo, S. K., Chen, S., Xu, X., & Zhu, L. (2020). Blockchain-Based Federated Learning for Device Failure Detection in Industrial IoT. *IEEE Transactions on Industrial Informatics*, *8*(6), 5926–5937. doi:10.1109/JIOT.2020.3032544

Zhao, F., He, X., & Wang, L. (2021). A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem. *IEEE Transactions on Cybernetics*, *51*(11), 5291–5303. doi:10.1109/TCYB.2020.3025662 PMID:33095728

Chun-Yi Lu is currently Assistant Professor of Information Management at the National Penghu University of Science and Technology of Taiwan. He earned his Bachelor of Economics at Tamkang University, Taiwan, in 1995, and his PhD in Engineering Science from the University of National Cheng Kung University in 2013. His research focuses on adapting data mining and AI Technology to diverse data types to gain insight into various practical scientific applications.

Hsin-Te Wu received the PhD Degree in Department of Computer Science and Engineering from National Sun Yat-Sen University, Taiwan, in 2013. Hsin-Te Wu is an Assistant Professor of Department of Computer Science and Information Engineering from National Ilan University, Taiwan. His research interests include computer networks, wireless network, speech compression, network security, blockchain, and internet of things.