Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Distributed Intelligent Model for Privacy and Secrecy in Preschool Education

Guoqiang He

To cite this article: Guoqiang He (2023) Distributed Intelligent Model for Privacy and Secrecy in Preschool Education, Applied Artificial Intelligence, 37:1, 2222494, DOI: 10.1080/08839514.2023.2222494

To link to this article: https://doi.org/10.1080/08839514.2023.2222494

Published online: 08 Jun 2023.

APPLIED ARTIFICIAL INTELLIGENCE
2023, VOL. 37, NO. 1, e2222494 (2050 pages)
https://doi.org/10.1080/08839514.2023.2222494

Taylor & Francis
Taylor & Francis Group

Distributed Intelligent Model for Privacy and Secrecy in Preschool Education

Guoqiang He

Zhengzhou Preschool Education College, Preschool Education Institute, Zhengzhou, China

ABSTRACT

Mobile devices, including phones, tablets, and smartwatches, have revolutionized the way we compute and have seamlessly integrated into education systems. These versatile devices store vast amounts of valuable personal data due to their rich user interactions and advanced sensor capabilities. By harnessing the potential of this data through model training, we can greatly enhance the functionality and effectiveness of smart applications, providing educators with invaluable insights for making informed decisions. However, it is crucial to acknowledge the significant risks and responsibilities associated with handling such sensitive information. One notable breakthrough in this field is distributed machine learning, which enables improved accuracy and scalability by employing a multi-node system. This approach is particularly advantageous for processing larger input data sizes, allowing for enhanced performance and reduced errors. Moreover, it facilitates assisting individuals in making well-informed choices and effectively analyzing extensive datasets. This work introduces an advanced distributed intelligent model that leverages fully distributed machine learning techniques. Through a consensus mechanism and the exchange of gradients, we ensure the utmost integrity of private data pertaining to sports activities, education, training, and the health of preschoolers. The robust privacy and security features of this model make it an ideal solution for preschool organizations and educational institutions seeking to harness the power of machine learning while upholding the strictest standards of data privacy and security.

ARTICLE HISTORY
Received 10 May 2023
Revised 2 June 2023
Accepted 4 June 2023

Introduction

Preschool education is an industry that is constantly evolving, and professionals are increasingly turning to technology to provide the best possible learning environment for young children (Ma et al. 2020). Wearable sensors have the potential to create significant opportunities for the development of early childhood education and training. By continuously monitoring physical signals, these devices can track various parameters and identify

CONTACT Guoqiang He heguoqiang@zzpec.edu.cn Zhengzhou Preschool Education College, Preschool Education Institute, Zhengzhou, 450000, China

biomarkers indicating potential health issues in children. They can also assist with fall detection, posture, and sleep analysis (Aira et al. 2019).

Traditionally, collecting data on a child's performance was a time-consuming process for educators. With the widespread commercialization of smart mobile devices, there has been an increase in the development of innovative applications that allow for real-time processing, high reliability, and connectivity even when no network connection is available (Fan et al. 2021).

The use of wearable sensors and mobile devices may raise concerns regarding the privacy and protection of sensitive data, such as health information. However, as with any technology, there are risks associated with collecting and processing personal data (Weinberg et al. 2015). The data collected may include information about a child's medical history, family background, medication usage, and other personal information, which can put them at risk of social stigma or other potential harm (Cai et al. 2020).

To address these concerns, it is essential to create a system that allows for the intelligent analysis and processing of preschoolers' data while ensuring their privacy. This work proposes an advanced distributed intelligent model that employs fully distributed machine learning through a consensus mechanism and exchange of gradients (Alzubaidi et al. 2021) ensures the integrity of private data related to preschool activities, education, training, and children's health (Mingxiao et al. (2017, October). A Review on Consensus Algorithm of Blockchain. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (Pp. 2567–2572). IEEE.).

While technology presents opportunities for significant improvements in preschool education and training, it is essential to balance these benefits with the need to protect personal data and ensure privacy. Through innovative technologies and advanced data management systems, we can create an environment that maximizes the potential benefits while minimizing the risks associated with the collection and processing of personal data.

The rapid proliferation of mobile devices, such as phones, tablets, and smartwatches, has transformed the way we interact with technology and has also made its way into the field of education. These devices possess vast storage capacities and collect a wealth of valuable data through user interactions and advanced sensors. This data presents an opportunity to train models and enhance the functionality of smart applications, ultimately benefiting educators in making informed decisions. However, the sensitive nature of this data also raises concerns regarding its use and disposal, necessitating the development of robust privacy and security measures.

The primary contribution of this work lies in the utilization of distributed machine learning to address the challenges associated with mobile device data. By employing a multi-node system, the approach significantly improves the accuracy and scalability of machine learning models when processing large

volumes of input data. This enhancement leads to more efficient performance, reduces errors, and enables the analysis of extensive datasets.

Furthermore, this work presents an advanced distributed intelligent model that incorporates fully distributed machine learning techniques. The model ensures the integrity and privacy of private data related to sports activities, education, training, and the health of preschoolers through the implementation of a consensus mechanism and the exchange of gradients. These robust privacy and security features make it an ideal solution for preschool organizations and educational institutions aiming to leverage the power of machine learning while safeguarding the confidentiality of their data.

In summary, this work acknowledges the potential benefits of utilizing mobile device data in education and addresses the associated risks through the application of distributed machine learning. The advanced distributed intelligent model presented herein not only enhances the usability and power of smart applications but also ensures the privacy and security of sensitive data, thereby making it valuable for educational institutions and preschool organizations.

Review Publications

The literature about privacy and secrecy concentrates on the Internet of Things (Karunarathne, Saxena, and Khurram Khan 2021). We can say that the preschool field lacks a scientific approach, especially in the holistic way the current study provides.

(Jain et al. 2021), investigated the present status of the Internet of Things and wearable technology, as well as the influence that wearable devices would have on the future of the IoT. They predicted that in the future, accessibility would play a significant role in promoting accessibility for additional preschoolers, including the deaf and the blind. Wearable e-textiles provide characteristics and functions such as thermal regulation, brightness, contact, and sensitivity. In contrast, on-body smart clothing offers real-time tracking and biometric record keeping evaluating people's effectiveness, including breathing and cardiovascular system rates, body heat, fluid intake, and tense muscles.

Weinberg et al. (2015) brought the Internet of Things to the management community and examined one of its core tensions: utility vs. privacy and secrecy. They emphasized the distinctions between IoT and Web 2.0 before highlighting potential problems and management advice. Moreover, they investigated the primary topic of confidentiality and anonymity. Due to considerable growth in user data quantities and their openness, as well as possible tradeoffs in IoT advantages and consumer impression personhood features, the management problem of privacy has reached an unprecedented degree.

Sharma and Park (2018) suggested a revolutionary hybrid network design for the intelligent city by utilizing the power of upcoming Software Defined

Connectivity and ledger technologies to deliver low delay, minimize bandwidth consumption, and enhance protection, confidentiality, and scalability. To improve efficiency and overcome existing constraints, their design was separated into two sections, the core structure, and the edge network. Their suggested method inherits the strengths of both controlled and dispersed network topologies via building a hybrid framework. Their solution used a memory-hardened Proof of Work technique to assure security and privacy and prevent information manipulation by attackers. To assess the viability and effectiveness of their approach, they simulated it and evaluated it using several performance measures. The outcome of the assessment demonstrated the efficiency of the model. There are still certain constraints in the model, such as the effective usage of edge nodes and the activation of caching approach at the edge nodes. Therefore, they will do further work in this area in the future.

Shen et al. (2022) described the environment and most recent advancements in associative training data confidentiality and security. They feel that transfer learning is an essential trend for developing decentralized and collaborative deep learning due to its capacity to offload calculations from the central server. They recognized the various privacy and security approaches, including differential privacy, safe multiparty computing, and strong aggregation. In addition, they examined the existing attack models, finding the zones of weakness and the tactics attackers use to compromise federated systems. The study ended with a review of the unresolved difficulties and possible future lines of work in this gaining popular learning style.

In fog computing, Zhou et al. (2020) suggested a federated learning approach that protects anonymity. As a member, each fog node can acquire data from Internet-of-Things devices and execute the assigned learning job. This strategy successfully improves the poor learning efficiency and model precision caused by the unequal data allocation and the high computing power disparity. They allowed IoT device data to comply with differential protection to withstand data assaults and used the mix of blind and Paillier elliptic curve cryptography versus model intrusions, which allowed the secure grouping of design variables. In addition, they explicitly confirmed that our approach not only ensures data protection and model security but also withstands collusion attempts made by several malevolent groups. Their trials demonstrate that their method is very effective at handling different data distributions. Future efforts may focus on increasing the scheme's efficacy and lowering its computing cost, making it more valuable and secure.

Ismagilova et al. (2022) evaluated a multitude of crucial topics in smart cities findings, including privacy and assurance of mobile devices and services, smart city transport systems, power technologies, healthcare, frameworks, algorithms, and procedures to enhance safety and privacy, operational dangers for smart cities, utilize and usage of smart services by citizens, blockchain use, and social media use. The results offered an

instructive study methodology and comparison point for researchers and practitioners but are somewhat restricted owing to the emphasis on security and privacy, which may have omitted a variety of human-centered aspects that may influence the implementation of smart cities in the future. Their analysis gave a helpful perspective on a few of the most important topics and provided essential guidance for future research.

The proposed approach introduces several innovative aspects in the field of mobile device data analysis and distributed machine learning. These include:

(1) Integration of Mobile Devices into Education: The paper recognizes the growing importance of mobile devices in education and highlights their potential as primary computing devices. By incorporating these devices into the educational context, the paper demonstrates an innovative approach to enhancing the learning experience and decision-making for educators.
(2) Utilization of Mobile Device Data: The paper explores the rich user interactions and robust sensor capabilities of mobile devices, emphasizing the vast amounts of valuable private data they can store. The innovative aspect lies in harnessing this data to train models and improve the usability and power of smart applications, ultimately benefiting educators in making efficient decisions.
(3) Distributed Machine Learning: The paper introduces the concept of distributed machine learning as a significant development in the field. This approach utilizes a multi-node system to increase accuracy and scale to larger input data sizes. The innovative aspect lies in the ability to improve performance, reduce errors, and analyze large amounts of data through the collaborative effort of multiple nodes.
(4) Consensus Mechanism and Gradient Exchange: The paper proposes a consensus mechanism and gradient exchange as means to ensure the integrity of private data related to sports activity, education, training, and the health of preschoolers. This innovative approach enables the protection of sensitive information while leveraging the benefits of distributed machine learning, providing a robust privacy and security framework.
(5) Privacy and Security Features: The paper emphasizes the importance of privacy and security when handling sensitive data. The innovative aspect lies in the development of a model with robust privacy and security features specifically tailored for preschool organizations and educational institutions. This ensures that the power of machine learning can be harnessed while maintaining the privacy and security of the data.

Overall, the paper's innovation lies in its recognition of the potential of mobile devices in education, the utilization of mobile device data, the application of distributed machine learning, and the development of privacy and security measures. These aspects collectively contribute to the advancement of the field and offer valuable insights for educational institutions and preschool organizations.

Distributed Intelligent Model

The entire distributed learning architecture can be seen as an evolutionary step beyond the parameter server architectures (Guo et al. 2022; Shen et al. 2022). For various reasons, the involvement of the parameter server is not desirable, and in other applications, it may not be feasible. The parameter server mechanism provides a relatively simple mechanism that ensures the consensus of all collaborators on a universal network-wide machine learning model. In the case of fully distributed learning, his participation is not possible. Nevertheless, mechanisms can still control the learning process and reach a consensus between the agents. In the conditions of fully distributed learning, the concept of the universal model does not exist (Guo et al. 2022).

Now the interest turns to algorithms that control the learning process in such a way that the local model of each collaborating agent converges toward the desired solution. The communication topology of a fully distributed learning network is quite different from the star created by the existence of the parameter server. The set of nodes $\mathcal{V}$ of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ describes the cooperating agents, and the group of edges $\mathcal{E}$ represents the communication channels between the agents. Now, cooperating agents rely only on communication with usually a small number of other cooperating agents (peer-to-peer communications) with which there is a communication channel. This leads primarily to communication topologies described by sparse graphs with a small maximum degree (Conti, Donadel, and Turrin 2021; Halabi, Bellaiche, and Abusitta 2018; Zhou et al. 2020)

Although the theoretical description below generalizes to the case of non-connected graphs, the focus will be on connected undirected graphs. In other words, it is implied that the communication between the agents for which a communication channel exists is two-way. Finally, recall that for any pair of nodes $(n_i, n_j) \in \mathcal{V}$ of a connected graph, there is a path between n_i and n_j.

Due to the nature of the fully distributed learning problem, it is helpful to define some additional concepts which are of particular importance for the description of the proposed algorithm. Initially, for each node $k \in \mathcal{V}$, the neighborhood of node k is defined as the subset $\mathcal{N}_k \subseteq \mathcal{V}$ of nodes with which node k is joined by edges (Goldreich and Goldreich 2011).

Figure 1 includes a graphical representation of a network of ten agents, with edges showing which agents can communicate.

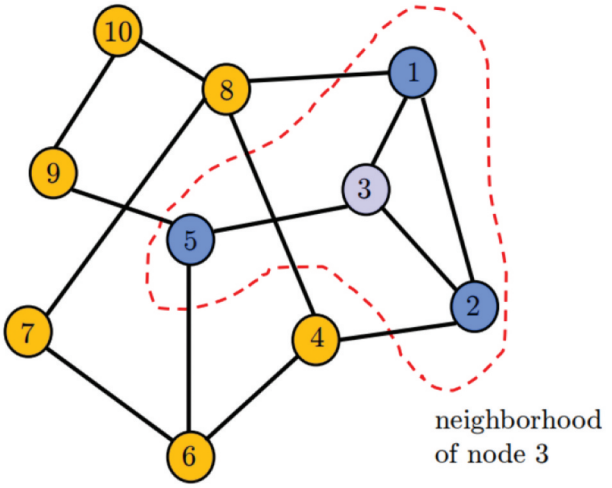

Figure 1. Graphical representation of a network with 10 nodes. Edges represent which nodes can communicate. The red area depicts the node's neighborhood $\mathcal{N}_3 \subseteq \mathcal{V}, \mathcal{N}_3 = \{1, 2, 3, 5\}$.

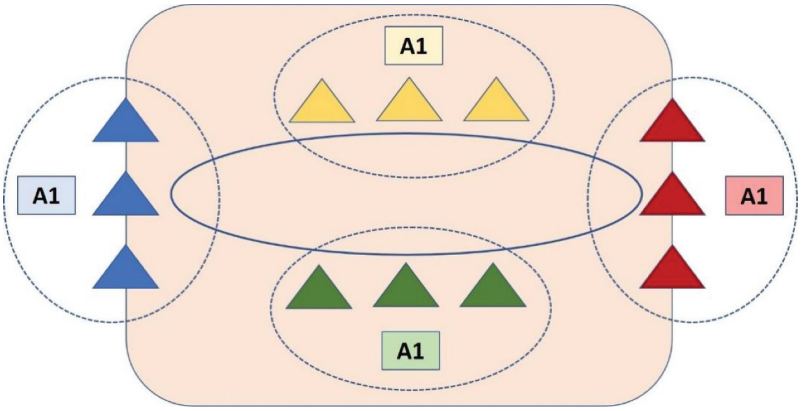

Figure 2. Graphical representation of application and peers on the distributed intelligent model.

The degree of each node k is represented by the plurality of its neighborhood $\deg(k) = |\mathcal{N}_k|$. Specifically, in the example of Figure 1 $\deg(3) = |\mathcal{N}_3| = 4$. Finally, a different class of algorithms (e.g., the family of Consensus algorithms) use a different version of the set $\mathcal{N}_k$ which will be denoted as $\mathcal{N}_{\bar{k}}$. The set $\mathcal{N}_{\bar{k}}$ includes all neighbors of node k, excluding node k. In the example of Figure 1 the group $\mathcal{N}_{\bar{k}} = \{1, 2, 5\}$.

As mentioned, the graph's edges refer to two-way communications between the nodes they join. However, this two-way relationship does not automatically imply that the flow of information is also symmetrical. For this reason, weights assigned by node k can be used to combine the information it receives from its neighborhood. This is how the coefficient α_{lk} is defined, which defines the weight given to the information of node l by node k. Consequently, the

indices l, k also show the flow of information. In this coefficient, the data flows from node k to node l. Accordingly, the coefficient α_{kk} symbolizes the flow of information from node k to node l. Therefore, the weights $\{\alpha_{lk}, a_{kl}\}$ can differ from each other or can be zero. A particular case is the coefficient α_{kk} which describes the weight given by the node k to the information it has (Shivaprasad and Shetty 2017).

In this work, an approach is proposed that aims to ensure the integrity of personal data while simultaneously speeding up the distributed learning process (Figure 2). To achieve this object, it is chosen to significantly increase the volume of exchanged messages in the network since, in addition to the exchange of parameters, it also requires the interaction of gradients during the learning process.

Initially, in the first step, the aggregation formula of the Consensus algorithm is used to calculate an intermediate model $\psi_{k,t}$ consisting of the local model of agent k and the models received from its neighbors (Aggarwal 2016; Goldreich and Goldreich 2011).

In networks of collaborating agents, the systems studied in this paper, the concept of consensus means that all collaborating agents will end up agreeing on a set of parameters of common interest. In learning problems, the standard set of parameters is learning model parameters (e.g., neural network weights) that cooperating agents learn through local data processing and sharing local parameters with their neighbors. This problem is more difficult now that no parameter server mechanism is available. Calculating a global model for the entire network is no longer feasible. The goal is for all local models to converge to the desired solution. The general form of the Consensus algorithm, specially adapted as an entire distributed learning algorithm, is described by the following equations, which respectively represent the local processing (adapt) and communication (combine) steps (Deka 2016; He et al. 2011):

$$\begin{aligned} \psi_{t,k} &= W_{t,k} + e_t \sum_{i \in \mathcal{N}_{\bar{k}}} a_{i,k}\left(W_{t,i} - W_{t,k}\right)(\text{combine}) \\ W_{t+1,k} &= \psi_{t,k} - \mu_t \hat{} F_{t,k}\left(\psi_{t,k}\right)(\text{adapt}) \end{aligned} \tag{1}$$

whereas $\hat{} F_{t,k}\left(\psi_{t,k}\right)$ denotes a stochastic approximation of the real gradient of the function $F_{t,k}\left(\psi_{t,k}\right), \quad F_{t,k}\left(\psi_{t,k}\right)$.

The provided equations describe the consensus algorithm, which is adapted as a distributed learning algorithm. The algorithm consists of two main steps: local processing (adapt) and communication (combine).

In the local processing step (adapt), each node or device in the distributed system performs local computation on its own data. This computation involves updating its local model parameters, based on the previous state and the stochastic approximation of the gradient. This step allows each node to adapt its model based on the available data and the current estimate of the gradient.

In the communication step (combine), the nodes exchange information with their neighbors. This exchange is done by combining their local model parameters with the weighted difference between their neighbors' parameters. The weights can be determined based on network topology or other factors. This step allows nodes to share information and learn from the collective knowledge of the network.

Together, these two steps, adapt and combine, form the consensus algorithm for distributed learning. The local processing step ensures that each node updates its model based on its own data, while the communication step enables nodes to exchange information and collectively improve the overall model. This iterative process continues until convergence, resulting in a distributed learning algorithm that leverages the collaborative effort of multiple nodes to train models and solve complex tasks while preserving data privacy and security.

The gradient estimation is a crucial step in the distributed learning algorithm as it allows each node to update its local model based on an approximation of the true gradient, even though the node may not have direct access to the entire dataset or the complete information required to compute the exact gradient.

The specific method used to estimate the gradient depends on the algorithm and the problem at hand. Common techniques include stochastic approximation methods, such as stochastic gradient descent, where the gradient is estimated based on a randomly selected subset of the available data.

In the context of distributed learning, the gradient estimation typically takes into account the data available at each node and may involve local computations or aggregations of partial gradients. The exact details of the gradient estimation process can vary depending on the specific algorithm and the distribution of data across the network.

The accuracy and efficiency of the gradient estimator are crucial factors in the performance of the distributed learning algorithm. The estimator should provide a reasonably accurate approximation of the true gradient to ensure effective learning. Additionally, the estimation process should be computationally efficient to allow for scalability in large-scale distributed systems.

Overall, the gradient estimator in the algorithm is responsible for approximating the true gradient of the function being optimized based on the available data and the local information at each node. Its accuracy and efficiency significantly impact the convergence and performance of the distributed learning algorithm.

Before starting the cooperative learning, it is assumed that the agents have prior knowledge of various parameters such as the number of epochs E and the number of communication cycles per epoch T. Initially, all cooperating agents in the network initialize the parameters of the local models $W_{k,0}$. They assumed that they have agreed on its characteristics (e.g., type of neural network, number of hidden layers, dimensions of weight vectors) at an earlier

time period. Then in each round of communication (which is denoted by the index t the agent k sends the parameters of its local model to its neighbors $(\mathcal{N}_{\bar{k}})$ and will receive from them their local models $\{W_{i,t}\}_{i\in\mathcal{N}_{\bar{k}}}$.

Using the parameter vectors obtained by each agent from its neighborhood, it performs a first intermediate update of its model using the combined equation. Then, the final update of agent k's local model is performed using its local dataset, specifically by performing one or more update steps via the Stochastic Gradient Descent (SGD) algorithm (Bottou 2010). More specifically, agent k shuffles its data set and divides it into mini-batches which are used at each step of SGD to compute a stochastic approximation of the gradient of the function $F_{t,k}\left(\psi_{k,t}\right)$. Therefore, it is assumed that several updates are performed through the SGD algorithm, using several mini-batches. Finally, the updated model W_(t,k) is sent to the neighbors. This event marks the end of a cycle of processing and communication and the beginning of a new one, at which point the steps are repeated (Wu and Khalil 2021).

The key feature of the algorithm for fully distributed learning through consensus and gradient exchange is that before using the $\psi_{k,t}$ model to update its local model, it introduces a gradient exchange step. More specifically, before using $\psi_{k,t}$ it is sent to the neighbors of node k who use $\psi_{k,t}$ to calculate the gradient $F_{i,t}\left(\psi_{t,k}\right), i \in \mathcal{N}_{\bar{k}}$. In other words, each neighbor calculates the above gradient using $\psi_{k,t}$ to minimize its local cost function. In the next step, the computed gradients are sent back to node k (node k receives the set $\left\{ F_{i,t}\left(\psi_{t,k}\right)\right\}_{i\in\mathcal{N}_{\bar{k}}}$ from its neighbors). This step allows agent k to exploit more gradients with information about its neighbors' data to then update its local model according to the following equation (Cheng, Wang, and Xin 2018):

$$\hat{\psi}_{k,t} = \psi_{k,t} - \mu_t \sum_{i\in\mathcal{N}_{\bar{k}}} c_{k,i} \quad F_{k,i}\left(\psi_{k,t}\right) \tag{2}$$

where the weights $C_{k,i}$ are used to blend the gradients obtained from the neighborhood. After the step of exchanging the gradients, the step of local renewal follows as follows:

$$W_{k,t+1} = \hat{\psi}_{k,t} - \mu_t \quad F_{k,t}\left(\hat{\psi}_{k,t}\right) \tag{3}$$

The Combine & Adapt equations of this particular method are as follows:

$$\begin{aligned} \psi_{k,t} &= \sum_{l\in\mathcal{N}_k} c_{lk} W_{k,t-1} \\ W_{k,t+1} &= \psi_{k,t} - \mu \sum_{l\in\mathcal{N}_k} c_{lk} \hat{} F_l\left(\psi_{k,t}\right) \end{aligned} \tag{4}$$

To lighten the communication load of the gradient exchange step, with the ultimate goal that each agent does not have to wait for the gradients from its neighborhood to perform the local update, the implementation proposed in work modifies the gradient exchange step in such a way, so that agent*k* does not need to get back the estimates of the gradients from its neighbors to proceed to the step of locally updating its model parameters. More specifically, it implements and uses an estimate for them, instead of agent k waiting to receive the neighboring gradients $F_{t,k}\left(\psi_{t,i}\right)$, it implements and use an estimate for them, $\bar{F}_{k,t}\left(\psi_{t-1,i}\right)$. The estimator used is inspired by techniques that use momentum information and uses past models received from its neighbors to estimate the gradient it will receive in the current round according to the following formula (Hou et al. 2021; Lv et al. 2019):

$$\bar{F}_{k,t}\left(\psi_{t-1,i}\right) = \rho \quad F_{k,t}\left(\psi_{t-1,k}\right) + (1-\rho)\,\bar{F}_{k,t-1} \tag{5}$$

where the parameter $\rho \in (0, 1]$ controls whether the above terms in the sum are considered to calculate the estimate for the gradient $\bar{\mathrm{F}}_{\mathrm{k,t}}\left(\psi_{\mathrm{t-1,i}}\right)$.

According to the above relation, the solution of the neighboring gradients is replaced by an estimate based on the model $\psi_{t-1,i}$, i.e., of this previous round (the most recent one available). Based on the above modification, the steps of the algorithm are transformed as follows:

$$\psi_{t,k} = W_{t,k} + e_t \sum_{i \in \mathcal{N}_{\bar{k}}} a_{k,i}\left(\psi_{t,i} - W_{t,k}\right) \tag{6}$$

While the refresh step using the neighboring gradients transforms into the form below:

$$\hat{\psi}_{t,k} = \psi_{t,k} - \mu_t \sum_{i \in \mathcal{N}_{\bar{k}}} c_{k,i}\,\bar{F}_{i,t}\left(\psi_{t-1,k}\right) \tag{7}$$

Finally, the last remaining step is the local update step described by the above relationship (Axelsson and Nylander 2018).

The measurement of the communication complexity of the specific method will be done as a function of the messages exchanged during learning. More specifically, it is assumed that the machine learning model stored locally in each agent consists of a total of M parameters (e.g., the weights of a Neural Network). The specific method requires each agent to exchange the local parameters $\psi_{t,k}$ and the gradient $\bar{F}_{k,t}\left(\psi_{t-1,i}\right) \quad i \in \mathcal{N}_{\bar{k}}$. Consequently, the communication load of each agent now depends on the size of its neighborhood and is of the order of $\mathcal{O}(M|\mathcal{N}_{\bar{k}}|)$.

Next, we present the proposed Proximal Stochastic Gradient Descent (PSGD) method (T. Papastergiou and V. Megalooikonomou, "A Distributed

Proximal Gradient Descent Method for Tensor Completion," 2017 IEEE International Conference on Big Data (Big Data), 2017, Pp. 2056–2065, Doi: 10.1109/BigData.2017.8258152.) to solve the problem of computing the elliptic values of a tensor through a decomposition. The PSGD algorithm requires the proximity operator to be applied to the current point of the optimization, producing the next point, starting from a random point (e.g., from a random initialization of the decomposition factors U_0,V_0,W_0).

So, the proximity operator, in the case of a deconstruction, takes the following form (Ganzfried 2021; Sreelakshmi et al. 2019):

$$\begin{aligned}&\mathrm{prox}_{\mathrm{gL}}\left(\widetilde{\mathrm{U}},\widetilde{\ },\widetilde{\mathrm{W}}\right)\\&\quad=\underset{U,V,W}{\mathrm{argmin}}\left(\mathrm{L}(\mathrm{U},\mathrm{V},\mathrm{W})+\frac{1}{2\mathrm{g}}\parallel \mathrm{U}-\widetilde{\mathrm{U}}\parallel_{\mathrm{F}}^{2}\right.\\&\quad\left.+\parallel \mathrm{V}-\widetilde{\mathrm{V}}\parallel_{\mathrm{F}}^{2}+\parallel \mathrm{W}-\widetilde{\mathrm{W}}\parallel_{\mathrm{F}}^{2}\right)\end{aligned}\tag{8}$$

Therefore, an optimization problem should be solved at each step of the algorithm. The objective function that should be optimized at each step is (Dai et al. 2016):

$$\begin{aligned}\mathcal{L}(\mathrm{U},\mathrm{V},\mathrm{W})=&\parallel W\odot\left(X-\sum_{\mathrm{r}=1}^{\mathrm{R}}\mathrm{U}_{\cdot,\mathrm{r}}\mathrm{V}_{*,\mathrm{r}}\mathrm{W}_{+,\mathrm{r}}\right)\parallel_{\mathrm{F}}^{2}\\&+\frac{1}{2\mathrm{g}}\left(\parallel \mathrm{U}-\widetilde{\mathrm{U}}\parallel_{\mathrm{F}}^{2}+\parallel \mathrm{V}-\widetilde{\mathrm{V}}\parallel_{\mathrm{F}}^{2}+\parallel \mathrm{W}-\widetilde{\mathrm{W}}\parallel_{\mathrm{F}}^{2}\right)\end{aligned}\tag{9}$$

where **U,V,W** are the degradation factors, which have been calculated in the previous step of the algorithm. In our approach, we will use the SGD method to solve these optimization problems, and for this purpose, we should express the objective function as the sum of the local errors of the observed values of the **X** tensor. To arrive at such an analytical form, we should write the second part of the Equation as a sum of the observed values of **X**. We will prove the formula for the first term of the sum $\parallel \mathrm{U}-\tilde{D}\parallel_{F'}^{2}$ as the other terms are generated similarly (Bellare and Oded 2011):

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\mathcal{J}[(i,j,k)\in\Omega]\frac{\parallel \mathrm{U}_{\mathrm{i},*}-U_{\mathrm{i},\parallel_{\mathrm{F}}}^{2}}{N_{i,*}}=\sum_{(i,j,k)\in X}\frac{\parallel \mathrm{U}_{\mathrm{i},*}-U_{\mathrm{i},*}\parallel_{\mathrm{F}}^{2}}{N_{i,,*}}\tag{10}$$

where $\mathrm{N}_{\mathrm{i},*,*}$ is the number of observed values in the *i* front part of the tensor **X** and **J**[(*i,j,k*) ∈ Ω] is the indicator function of the set of indicators of the observed values of **X** (**J**(*i,j,k*) = 1, ∀(*i,j,k*) ∈ Ω and zero everywhere else). Combining the equations, we can arrive at the following objective function, expressed as the sum of the observed values of the tensor (Berger 2013):

$$L(U,V,W) = \sum_{(i,j,k)\in\Omega} \left[\begin{array}{l} \left(\mathrm{x}_{\mathrm{ijk}} - \sum_{\mathrm{r}=1}^{\mathrm{R}} U^{\circ}_{*,r} V_{*,r} W_{*,r} \right)^2 \\ + \frac{1}{2\mathrm{g}} \left(\frac{U_{i,*} - \tilde{U}_{i,*}{}^2_{\mathrm{F}}}{\mathcal{N}_{i,*,*}} + \frac{V_{j,*} - \tilde{V}_{i,*}{}^2_{\mathrm{F}}}{\mathcal{N}_{*,j,*}} + \frac{W_{k,*} - \tilde{W}_{i,*}{}^2_{\mathrm{F}})}{\mathcal{N}_{*,*,k}} \right) \end{array} \right] \tag{11}$$
$$= \sum_{(i,j,k)\in\Omega} \mathcal{L}_{\mathrm{x}_{ijk}}(U,V,W)$$

Optimizing the objective function at each step is essential in the context of machine learning and optimization algorithms for several reasons:

(1) Convergence: The objective function represents the goal or task that the machine learning algorithm aims to optimize. By continuously optimizing the objective function at each step, the algorithm iteratively approaches the optimal solution or the best possible outcome. Optimization ensures that the algorithm converges toward a point where the objective function is maximized or minimized, depending on the specific problem.
(2) Model Improvement: The objective function typically quantifies the performance or quality of the model being learned. By optimizing the objective function, the algorithm seeks to improve the model's performance, accuracy, or generalization capabilities. This iterative optimization process helps refine the model and achieve better predictive or decision-making abilities.
(3) Learning from Data: The objective function is constructed based on the available data and the desired learning task. By optimizing the objective function, the algorithm leverages the information in the data to update the model's parameters and make it more representative of the underlying patterns and relationships present in the data. This enables the model to make better predictions or decisions based on new, unseen data.
(4) Adaptation to Changing Environments: In some cases, the objective function may need to be optimized continuously to adapt to changing environments or evolving data distributions. By updating the objective function at each step, the algorithm can adapt the model to new patterns or dynamics in the data, ensuring that it remains relevant and performs well in real-world scenarios.
(5) Decision-Making and Utility: In certain applications, the objective function represents a utility function or a measure of the value or benefit derived from the decisions made by the model. By optimizing the objective function, the algorithm aims to maximize the utility or the desired outcome of the decisions made by the model. This is particularly relevant in reinforcement learning or decision-making tasks.

In summary, optimizing the objective function at each step is crucial for convergence, model improvement, learning from data, adaptation to changing environments, and achieving the desired utility or outcome. It allows the algorithm to refine the model, make better predictions or decisions, and align its behavior with the underlying task or problem being solved.

The partial derivatives of $\mathcal{L}_{\mathrm{x}_{\mathrm{ijk}}}(\vartheta)$, for the objective function, for the parameters $u_{\rho l}, v_{\mathrm{el}}, w_{\mathrm{el}}$are given in the following equations (Demertzis, Iliadis, and Anezakis 2017; Demertzis, Iliadis, and Bougoudis 2020):

$$\frac{\partial \mathcal{L}_{\mathrm{x}_{ijk}}(\mathrm{U},\mathrm{V},\mathrm{W})}{\partial \mathrm{u}_{el}} = \begin{cases} -2\left(\mathrm{x}_{ijk}\sum_{r-1}^{R} u_{ir}v_{jr}w_{kr}\right)v_{jl}w_{kl} + \frac{2}{N_{i**}}(u_{il}-\tilde{u}), Q=i \\ 0, Q\neq i \end{cases}$$

$$\frac{\partial \mathcal{L}_{\mathrm{x}_{ijk}}(\mathrm{U},\mathrm{V},\mathrm{W})}{\partial \mathrm{u}_{el}} = \begin{cases} -2\left(\mathrm{x}_{ijk}\sum_{r-1}^{R} u_{ir}v_{jr}w_{kr}\right)i_{il}w_{kl} + \frac{2}{N_{\cdot i*}}\left(v_{jl}-\tilde{v}_{jl}\right), Q=j \\ 0, Q\neq j \end{cases}$$

$$\frac{\partial \mathcal{L}_{\mathrm{x}_{ijk}}(\mathrm{U},\mathrm{V},\mathrm{W})}{\partial \mathrm{u}_{el}} = \begin{cases} -2\left(\mathrm{x}_{ijk}\sum_{r-1}^{R} u_{ir}v_{jr}w_{kr}\right)u_{il}w_{kl} + \frac{2}{N_{*+k}}(w_{kl}-\tilde{w}_{kl}), Q=k \\ 0, Q\neq k \end{cases} \tag{12}$$

To also ensure the confidentiality of the process described above, RSA encryption is used. This scheme owes its widespread use, not so much to its efficiency, as to the fact that it is nothing more than an implementation of the RSA cryptosystem with the role of the keys reversed (public-private). More specifically, if we denote by $\mathrm{Encrypt}_K(m)$ the RSA encryption function for a plaintext m and a key K and by $\mathrm{Decrypt}_K(c)$ the corresponding decryption function for the ciphertext c. Obviously (by the definition of cryptosystem) it does (Alshalali, M'Bale, and Josyula 2018):

$$\mathrm{Decrypt}_K(\mathrm{Encrypt}_K(m)) = m \tag{13}$$

Suppose A wants to send B the weights of a Neural Network m digitally signed with RSA. The key K is a fifth:

$$K = ((n,e),(p,q,d)) : n = pq, p, q : \pi\rho\dot{\sigma}\tau\mathrm{O}, ed \equiv 1(mod\phi(n)) \tag{14}$$

The values *n, e* is the public key, while *p, q, and d* are the private key. To create a signature, A calculates the:

$$s = \left(\mathrm{sig}_K(m) =\right)\mathrm{Decrypt}_K(m) = m^d(modn) \tag{15}$$

s is his digital signature which he sends to B. For signature verification B uses A's public key to verify signature s while retrieving the original message:

$$m_1 = \mathrm{Encrypt}_K(s) = s^e(modn) \tag{16}$$

It is true that $\mathrm{ver}_K(m_1, s) = 1$ as $s^e = m^{de} = m$, i.e., m_1 is the original message, i.e., m.

The additional complexity gained through the operations involved in the distributed learning algorithm, such as the local processing (adapt) and communication (combine) steps, can vary depending on several factors, including the size of the network, the complexity of the model, the amount of data, and the specific implementation details. However, it is important to note that distributed learning algorithms often introduce some level of additional complexity compared to traditional centralized learning approaches.

Here are a few factors that contribute to the additional complexity in distributed learning:

(1) Communication Overhead: In the communication (combine) step, nodes exchange information, which introduces communication overhead. The amount of communication required depends on the network topology, the number of nodes, and the frequency of communication. As the size of the network grows, the amount of communication and the associated overhead increase, adding complexity to the algorithm.
(2) Synchronization and Coordination: Distributed learning algorithms often require synchronization and coordination among nodes to ensure that they exchange information and update their models correctly. Managing synchronization and coordination in a distributed setting can be challenging and adds complexity to the algorithm.
(3) Data Partitioning and Distribution: Data is typically distributed across multiple nodes in distributed learning. Partitioning and distributing the data among nodes can introduce additional complexity, particularly when dealing with imbalanced data or when the data needs to be shared or combined in a meaningful way during the communication step.
(4) Fault Tolerance and Robustness: Distributed learning algorithms often need to handle failures or nodes leaving the network. Building fault tolerance and robustness into the algorithm increases its complexity as it requires mechanisms to handle node failures, data inconsistencies, and ensure the algorithm can continue functioning even in the presence of failures.
(5) Scalability Considerations: Distributed learning algorithms aim to scale to large datasets and networks. Achieving scalability requires careful consideration of algorithmic design and implementation choices, introducing additional complexity in terms of managing computational resources, load balancing, and efficient utilization of the distributed system.

It is important to note that while distributed learning introduces additional complexity, the benefits it offers, such as improved performance, scalability,

and privacy preservation, often outweigh the added complexity. Continuously work to address these complexities by developing efficient algorithms, optimizing communication patterns, and leveraging parallel and distributed computing techniques to mitigate the additional complexity introduced by distributed learning.

Applications and Peers

We'll now demonstrate how client apps communicate with peers, especially the suggested architecture that runs on peers to access the Distributed Intelligent Model. Queries are basic conversations between an application and a peer, while updates (writes) need more steps.

A client application connects to the Distributed Intelligent Model on a peer to access data. An API allows applications to submit transaction proposals (invoke data), obtain endorsements, receive events, and route approved transactions to the ordering service through a gateway.

Applications may query or change the point of contact using a peer connection on the gateway. A query transaction's result is returned with simple processing, but an update (write) transaction includes a more sophisticated workflow amongst applications, peers, and orderers. Let's go at this update procedure in depth.

In collaboration with orderers, Peers guarantees that the procedure is consistent and up to date on every peer in a communication channel.

The above illustration depicts companies and their peers in the Distributed Intelligent Model. Four companies each contribute 12 peers to establish a network. Each channel links the person to their peers in each organization. These organizations' peers have not joined other organizations' communication channels, although they usually join at least one other channel. An organization's applications link to peers in the same organization and others via a track using the Distributed Intelligent Model.

The following three-phase sequence depicts interactions between a client application, a peer's gateway service, orderer nodes, and further peer updates.

Phase 1: Transaction Proposal and Approval

The first (write) phase of an update consists of transaction proposal submission, execution, and endorsement:

(1) Transaction proposal – By connecting to the gateway service on P1, the client application presents a signed transaction proposal. A1 must either

outsource the choice of endorsing organizations to the gateway service or expressly list the organizations that must be supported.
(2) Transaction execution – The gateway service chooses P1 or another peer within its organization to carry out the transaction. The assigned peer maintains the proposal's code and creates a proposal response (containing the read-write set). The chosen peer signs and returns the proposal answer to the gateway.
(3) Transaction endorsement – The gateway repeats transaction execution for each organization that the code endorsement rules demand. The gateway service gathers the signed proposal answers and generates a transaction envelope, which it then provides to the client for signature.

Phase 2: Submission and Ordering of Transactions

The second phase of an update consists of submitting a transaction and arranging it into the Distributed Intelligent Model:

(1) Transaction submission – The signed transaction envelope is sent to the gateway service by the client. The gateway passes the envelope to an ordering node and sends the client a success message.
(2) Transaction ordering – After verifying the signature, the ordering node organizes the transaction and bundles it with other ordered transactions into blocks. Following that, the ordering service distributes the block to all peers in the channel for validation and commitment to the ledger.

Phase 3: Transaction Validation and Commitment

The third phase of an update includes transaction validation, ledger commitment, and a commit event:

(1) Transaction validation – Each peer verifies that the client signature on the transaction envelope corresponds to the signature on the original transaction proposal. Each peer verifies that all read-write sets and status answers are comparable (i.e., all peers' endorsements match) and that the blessings comply with the endorsement rules. Then, for commitment to the Distributed Intelligent Model, each peer stamps each transaction as legitimate or invalid.
(2) Transaction commitment – Each peer commits to the communication channel the ordered block of transactions. The commit is a permanent (write) communication channel. The communication channel's total of all legitimate transactions is only updated with the outcomes of valid transactions.

Commit event – When a peer commits to the Distributed Intelligent Model, the client receives a commit status event with evidence of the modification.

To ensure the effectiveness of the proposed solution, there are several measures and considerations that can be taken into account:

(1) Robust Privacy Framework: Implement a robust privacy framework that ensures the protection of sensitive data. This can include techniques such as data anonymization, encryption, access controls, and strict data handling policies. Adhere to relevant privacy regulations and guidelines to ensure compliance and safeguard the privacy of user data.
(2) Secure Communication: Employ secure communication protocols to protect the transmission of data and gradients between nodes. Use encryption and authentication mechanisms to ensure confidentiality and integrity during data exchange. This helps prevent unauthorized access or tampering with the exchanged information.
(3) Data Minimization and Purpose Limitation: Minimize the collection and storage of personal data to only what is necessary for the learning task. Adopt a purpose limitation approach where data is used solely for the intended educational or research purposes and not shared or utilized for other unrelated activities.
(4) Consent and Transparency: Obtain informed consent from individuals whose data is being used in the learning process. Clearly communicate the purpose of data collection, how it will be used, and any potential risks involved. Provide individuals with transparency and control over their data, allowing them to opt-out or manage their privacy preferences.
(5) Regular Risk Assessments and Audits: Conduct regular risk assessments to identify potential vulnerabilities in the system and evaluate the effectiveness of privacy and security measures. Perform audits to ensure compliance with privacy regulations and assess the overall security posture of the distributed learning infrastructure.
(6) User Empowerment and Education: Educate users, including educators, administrators, and students, about data privacy and security practices. Promote awareness of the potential risks and benefits of data sharing and distributed learning. Provide resources and guidelines on best practices for protecting privacy and securing sensitive data.
(7) Ethical Considerations: Ensure that the proposed solution adheres to ethical guidelines and principles. Avoid biases, discrimination, or unfair treatment in the learning process. Consider the potential implications and consequences of the decisions made by the intelligent model and promote fairness, transparency, and accountability.

(8) Continuous Monitoring and Incident Response: Implement a robust monitoring system to detect any security breaches or privacy incidents. Establish an incident response plan to quickly respond to and mitigate any potential privacy or security breaches. Regularly review and update security protocols and procedures to address emerging threats and vulnerabilities.

By incorporating these measures into the proposed solution, you can enhance the effectiveness, privacy, and security of the distributed learning system while ensuring the protection of sensitive data and maintaining the trust of stakeholders involved.

Conclusion

This study presents an advanced distributed intelligent model that employs entirely distributed machine learning while ensuring the security of private data linked to preschool education activity, sports education, training, and the health of preschoolers through a consensus process and exchange of gradients.

The proposed model enables the intelligent analysis and processing of athletes' data while ensuring their privacy. This work presents an advanced distributed intelligent model that uses fully distributed machine learning and, through a consensus mechanism and exchange of gradients, ensures the integrity of private data related to sports activity, education, training, and athletes' health. It is decided to considerably increase the number of messages exchanged in the network to accomplish this goal since the learning process needs the interaction of gradients and the transmission of parameters.

The suggested approach of protecting personal data privacy while enhancing the distributed learning procedure by increasing the number of exchanged messages and incorporating gradient communication opens up several potential applications, including:

(1) Healthcare and Medical Research: The method can be applied in healthcare settings to enable collaborative analysis of sensitive patient data, such as electronic health records or genomic data. By preserving data privacy while exchanging gradients, medical researchers and institutions can collectively train models on a distributed network, leading to improved diagnostics, personalized treatment recommendations, and advancements in medical research.
(2) Financial Services: In the financial sector, where data privacy is of utmost importance, the suggested approach can be utilized to securely analyze financial transactions, customer profiles, and fraud detection. By leveraging distributed machine learning with enhanced privacy

measures, financial institutions can collaboratively detect patterns and anomalies, thereby improving risk assessment and fraud prevention while safeguarding customer information.

(3) Educational Institutions: The method can be applied within educational institutions to protect student privacy while enabling collaborative analysis and personalized learning. By securely exchanging gradients and preserving the integrity of private data, educational institutions can leverage distributed machine learning to enhance student assessment, adaptive learning systems, and educational analytics while ensuring compliance with data protection regulations.

(4) Smart City and Internet of Things (IoT) Applications: The suggested approach can be applied in the context of smart cities and IoT deployments. By securely exchanging gradients and protecting personal data, distributed machine learning can be utilized to analyze data from various sources, such as sensors, traffic cameras, and environmental sensors. This can enable intelligent decision-making for urban planning, traffic optimization, energy management, and environmental monitoring, while respecting individual privacy.

(5) Collaborative Research and Development: The method can be employed in collaborative research and development projects where multiple organizations or teams need to analyze sensitive data. By securely exchanging gradients and protecting data privacy, distributed machine learning can enable collaborative research efforts in fields such as drug discovery, climate modeling, or material science, fostering innovation while maintaining confidentiality.

These applications demonstrate the potential of the suggested approach to address privacy concerns while enabling distributed machine learning in various domains, leading to improved insights, decision-making, and innovation while safeguarding sensitive data.

As a continuation of this work, we will focus on developing a model with quality elements that will highlight to the effective usage of edge nodes and the activation of caching approach at the edge nodes.

Disclosure statement

No potential conflict of interest was reported by the author.

Funding

This study was supported by the 2022 Urgent and Special Projects of Zhengzhou Local Colleges and Universities for Early education.

References

Aggarwal, C. C. 2016. *An introduction to recommender systems*. vol 1. Springer International Publishing. doi:10.1007/978-3-319-29659-3_1.

Aira, T., K. Salin, T. Vasankari, R. Korpelainen, O. H. Jari Parkkari, K. Savonen, L. Alanko, L. Kannas, H. Selänne, and H. Selänne. 2019. Training volume and intensity of physical activity among young athletes: The health promoting sports club (HPSC) study. *Advances in Physical Education* 9 (4):270–87. doi:10.4236/ape.2019.94019.

Alshalali, T., K. M'Bale, and D. Josyula. 2018. Security and privacy of electronic health records sharing using hyperledger fabric. Paper read at 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA.

Alzubaidi, L., J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan. 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data* 8 (1):1–74. doi:10.1186/s40537-021-00444-8.

Axelsson, J., and S. Nylander. 2018. An analysis of systems-of-systems opportunities and challenges related to mobility in smart cities. Paper read at 2018 13th Annual Conference on System of Systems Engineering (SoSE), Paris, France.

Bellare, M., and O Goldreich. 2011. On probabilistic versus deterministic provers in the definition of proofs of knowledge. *Studies in Complexity and Cryptography. Miscellanea on the Interplay Between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman* 6650: 114–23.

Berger, J. O. 2013. *Statistical decision theory and Bayesian analysis*. Berlin / Heidelberg, Germany: Springer Science & Business Media.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, Large-scale machine learning with stochastic gradient descent, Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France (pp. 177–86). Physica-Verlag HD.

Cai, H., Z. Liu, L. Ruiyang, and J. Chen. 2020. Leakage protection device based on smart home. Paper read at 2020 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Zhangjiajie, China.

Cheng, Z., T. Wang, and Y. Xin. 2018. High-order distributed consensus in multi-agent networks. Paper read at 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), Enshi, China.

Conti, M., D. Donadel, and F. Turrin. 2021. A survey on industrial control system testbeds and datasets for security research. *IEEE Communications Surveys & Tutorials* 23 (4):2248–94. doi:10.1109/COMST.2021.3094360.

Dai, J., Y. Li, H. Kaiming, and J. Sun. 2016. R-fcn: Object detection via region-based fully convolutional networks. *Advances in Neural Information Processing Systems* 29: 367.

Deka, B. K. 2016. Transformations of graph database model from multidimensional data model. Paper read at 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India.

Demertzis, K., L. Iliadis, and V.-D. Anezakis. 2017. Commentary: Aedes albopictus and Aedes japonicus—two invasive mosquito species with different temperature niches in Europe. *Frontiers in Environmental Science* 85 (5). doi:10.3389/fenvs.2017.00085.

Demertzis, K., L. Iliadis, and I. Bougoudis. 2020. Gryphon: A semi-supervised anomaly detection system based on one-class evolving spiking neural network. *Neural Computing and Applications* 32 (9):4303–14. doi:10.1007/s00521-019-04363-x.

Fan, C., M. Chen, X. Wang, J. Wang, and B. Huang. 2021. A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. *Frontiers in Energy Research* 9:652801. doi:10.3389/fenrg.2021.652801.

Ganzfried, S. 2021. Computing Nash equilibria in multiplayer DAG-structured stochastic games with persistent imperfect information. Paper read at Decision and Game Theory for Security: 12th International Conference, GameSec 2021, Virtual Event, October 25–27, 2021, Proceedings 12, Prague, Czech Republic.

Goldreich, O., and O. Goldreich. 2011. *Studies in complexity and cryptography. Miscellanea on the interplay between randomness and computation*. vol 6650. Springer Berlin Heidelberg. doi:10.1007/978-3-642-22670-0.

Guo, Y., R. Zhao, S. Lai, L. Fan, X. Lei, and G. K. Karagiannidis. 2022. Distributed machine learning for multiuser mobile edge computing systems. *IEEE Journal of Selected Topics in Signal Processing* 16 (3):460–73. doi:10.1109/JSTSP.2022.3140660.

Halabi, T., M. Bellaiche, and A. Abusitta. 2018. A cooperative game for online cloud federation formation based on security risk assessment. Paper read at 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Shanghai, China.

He, B., Y. Cui, J. Chen, and P. Xie. 2011. A spatial data mining method for mineral resources potential assessment. Paper read at Proceedings 2011 IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services, Fuzhou, China.

Hou, R., F. Tang, S. Liang, G. Ling, and Y. Zhu. 2021. Multi-party verifiable privacy-preserving federated k-means clustering in outsourced environment. *Security and Communication Networks* 2021 (2021):1–11. doi:10.1155/2021/3630312.

Ismagilova, E., L. Hughes, N. P. Rana, and Y. K. Dwivedi. 2022. Security, privacy and risks within smart cities: Literature review and development of a smart city interaction framework. *Information Systems Frontiers* 24 (2):393–414. doi:10.1007/s10796-020-10044-1.

Jain, N., A. Chaudhary, N. Sindhwani, and A. Rana. 2021. Applications of Wearable devices in IoT. Paper read at 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), Noida, India.

Karunarathne, S. M., N. Saxena, and M. Khurram Khan. 2021. Security and privacy in IoT smart healthcare. *IEEE Internet Computing* 25 (4):37–48. doi:10.1109/MIC.2021.3051675.

Lv, Z., W. Deng, Z. Zhang, N. Guo, and G. Yan. 2019. A data fusion and data cleaning system for smart grids big data. Paper read at 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Xiamen, China.

Ma, B., S. Nie, J. Minghui, J. Song, and W. Wang. 2020. Research and analysis of sports training real-time monitoring system based on mobile artificial intelligence terminal. *Wireless Communications and Mobile Computing* 2020:1–10. doi:10.1155/2020/8879616.

Mingxiao, D., M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun (2017, October). A review on consensus algorithm of blockchain. In 2017 IEEE international conference on systems, man, and cybernetics (SMC) (pp. 2567–72). IEEE.

Sharma, P., and J. H. Park. 2018. Blockchain based hybrid network architecture for the smart city. *Future Gener Comput Systs* 86:650–55. doi:10.1016/j.future.2018.04.060.

Shen, S., T. Zhu, D. Wu, W. Wang, and W. Zhou. 2022. From distributed machine learning to federated learning: In the view of data privacy and security. *Concurrency & Computation: Practice & Experience* 34 (16):e6002. doi:10.1002/cpe.6002.

Shivaprasad, T. K., and J. Shetty. 2017. Sentiment analysis of product reviews: A review. Paper read at 2017 International conference on inventive communication and computational technologies (ICICCT), Coimbatore.

Sreelakshmi, K., T. Tulasi Sasidhar, N. Mohan, and K. P. Soman. 2019. A methodology for spikes and transients detection and removal in power signals using Chebyshev approximation. Paper read at 2019 9th International Conference on Advances in Computing and Communication (ICACC), Kochi, India.

Weinberg, B. D., G. R. Milne, Y. G. Andonova, and F. M. Hajjat. 2015. Internet of Things: Convenience vs. privacy and secrecy. *Business Horizons* 58 (6):615–24. doi:10.1016/j.bushor.2015.06.005.

Wu, W., and A. M. Khalil. 2021. The discrete Gaussian expectation maximization (gradient) algorithm for differential privacy. *Computational Intelligence and Neuroscience* 2021:1–13. doi:10.1155/2021/7962489.

Zhou, C., A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang. 2020. Privacy-preserving federated learning in fog computing. *IEEE Internet of Things Journal* 7 (11):10782–93. doi:10.1109/JIOT.2020.2987958.