An International ICT R&D Journal Sponsored by ZTE Corporation

ISSN 1673-5188
CN 34-1294/TN

ZTE COMMUNICATIONS

中兴通讯技术(英文版)

September 2022, Vol. 20 No. 3

Special Topic:
Federated Learning for IoT and Edge Computing

Cover Paper: Toward Low-Cost Flexible Intelligent OAM in Optical Fiber Communication Networks

ISSN 1673-5188

The 9th Editorial Board of ZTE Communications

CONTENTS

ZTE COMMUNICATIONS
September 2022 Vol. 20 No. 3 (Issue 80)

Special Topic ▶ **Federated Learning for IoT and Edge Computing**

CONTENTS

ZTE COMMUNICATIONS
September 2022 Vol. 20 No. 3 (Issue 80)

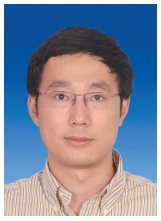
Serial parameters: CN 34–1294/TN*2003*q*16*98*en*P*¥ 30.00*2200*12*2022-09

PAN Yi, CUI Laizhong, CAI Zhipeng, LI Wei

Editorial: Special Topic on Federated Learning for IoT and Edge Computing

Guest Editors >>>

PAN Yi is currently a Chair Professor and the Dean of Faculty of Computer Science and Control Engineering at Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China and a Regents' Professor Emeritus at Georgia State University, USA. He served as Chair of Computer Science Department at Georgia State University from 2005 to 2020. Dr. PAN received his BE and ME degrees in computer engineering from Tsinghua University, China, in 1982 and 1984, respectively, and his PhD degree in computer science from the University of Pittsburgh, USA in 1991. His profile has been featured as a distinguished alumnus in both Tsinghua and University of Pittsburgh CS Alumni Newsletters. His current research interests mainly include bioinformatics and health informatics using big data analytics, cloud computing, and machine learning technologies. Dr. PAN has published more than 450 papers including over 250 journal papers published in IEEE/ACM Transactions/Journals. In addition, he has edited/authored 43 books. His work has been cited more than 19 000 times based on Google Scholar and his current h-index is 88. He has served as an Editor-in-Chief or editorial board member for 20 journals including seven IEEE Transactions. He is the recipient of many awards including one IEEE Transactions Best Paper Award, five IEEE and other international conference or journal Best Paper Awards, four IBM Faculty Awards, two JSPS Senior Invitation Fellowships, IEEE BIBE Outstanding Achievement Award, IEEE Outstanding Leadership Award, NSF Research Opportunity Award, etc. He has organized numerous international conferences and delivered keynote speeches at over 70 international conferences around the world. Dr. PAN is a member of the Academy of the United Nations Sciences and Technology Organization, foreign member of Ukrainian Academy of Engineering Sciences, Fellow of American Institute for Medical and Biological Engineering, Fellow of Institute of Engineering Technology, and Fellow of Royal Society for Public Health.



CUI Laizhong is currently a professor in the College of Computer Science and Software Engineering at Shenzhen University, China. He received his BS degree from Jilin University, China in 2007 and PhD degree in computer science and technology from Tsinghua University, China in 2012. His research interests include future Internet architecture and protocols, edge computing, multimedia systems and applications, blockchain, Internet of Things, cloud and big data computing, computational intelligence, and machine learning. He led more than 10 scientific research projects, including National Key Research and Development Plan of China, National Natural Science Foundation of China, etc. He has published more than 100 papers in prestigious journals, including *IEEE JSAC*, *IEEE TC*, *IEE TKDE*, *IEEE TMM*, *IEEE IoT Journal*, *IEEE TII*, *IEEE TVT*, *IEEE TNSM*, *ACM TOIT*, *IEEE TCBB*, etc. He serves as an associate editor or a member of editorial boards for several international journals, including *IEEE IoT Journal*, *IEEE TNSM*, and *International Journal of Machine Learning and Cybernetics*. He is a senior member of the IEEE and a senior member of the CCF.

CAI Zhipeng received his PhD and MS degrees from the Department of Computing Science at University of Alberta, Canada and BS degree from the Department of Computer Science and Engineering at Beijing Institute of Technology, China. Dr. CAI is currently a professor in the Department of Computer Science and the College of Business at Georgia State University, USA. His research areas focus on machine learning, Internet of Things, privacy, and big data. Dr. CAI is the recipient of an NSF CAREER Award. His research has been supported by the National Science Foundation, the U. S. Department of State, and other academic and industrial sponsors. Dr. CAI has published more than 100 papers in top journals and conferences including more than 70 IEEE/ACM transaction papers. His publications have been cited for more than 11 000 times. He is the Editor-in-Chief for *Wireless Communications and Mobile Computing* and Associate Editor-in-Chief for Elsevier *High-Confidence Computing Journal*. He serves as an editor for several prestigious journals including *IEEE TKDE*, *IEEE TVT*, *IEEE TWC*, *IEEE TCSS*, *IEEE Internet of Things Journal*, etc. Dr. CAI is a Steering Committee Co-Chair for the international conferences of WASA, IPCCC and COCOON. He has served as a General/Program Chair for many international conferences.

LI Wei received his PhD from The University of Sydney, Australia. He is currently an ARC DECRA Fellow with the Center for Distributed and High Performance Computing, School of Computer Science, The University of Sydney. He is the technical lead of the Australia-China Joint Research Center on Energy Informatics

DOI: 10.12142/ZTECOM.202203001
Citation: Y. Pan, L. Z. Cui, Z. P. Cai, and W. Li, "Editorial: special topic on federated learning for IoT and edge computing," *ZTE Communications*, vol. 20, no. 3, pp. 1 - 2, Sept. 2022. doi: 10.12142/ZTECOM.202203001.

and Demand Response Technologies. His research interests include edge computing, sustainable computing, task scheduling, decision making, Internet of Things, energy informatics, and medical informatics. He is the recipient of Australian Research Council Discovery Early Career Researcher Award in 2020, the IEEE TCSC Award for Excellence in Scalable Computing for Early Career Researchers in 2018, and the IEEE Outstanding Leadership Award in 2018. He serves as the information director for ACM Computing Surveys and the editors and PC members of tens of journals and conferences.

Recent years have witnessed the proliferation of Internet of Things (IoT), in which billions of devices are connected to the Internet, generating an overwhelming amount of data. It is challenging and infeasible to transfer and process trillions and zillions of bytes using the current cloud-device architecture, due to bandwidth constraints of networks and potentially uncontrollable latency of cloud services. Edge computing, an emerging computing paradigm, has received a tremendous amount of interest to boost IoT. By pushing data storage, computing and controls closer to the network edge, edge computing has been widely recognized as a promising solution to meeting the current-day requirements of low latency, high scalability, and energy efficiency, as well as to mitigating the network traffic burdens.

However, with the emergence of diverse IoT applications (e. g., the smart city, industrial automation, and connected cars), it becomes challenging for edge computing to deal with these heterogeneous IoT environments and gather the data feasibly for training in a centralized manner. Furthermore, data privacy has become fast-growing concerns when data are being accessed and obtained from IoT devices.

The aforementioned issues necessitate Federated Learning (FL), which enables edge devices to collaboratively train a locally-standard model using mobile data generated in real time. Federated learning is well suited for IoT and edge computing applications and can leverage the computation power of edge servers and the data collected on widely dispersed edge devices. Instead of collecting data to some centralized servers, FL allows machine learning models to be deployed and trained on the user end to alleviate some potential privacy leakage. Building such FL systems into edge architecture poses many technical challenges that need to be addressed. The goal of this special issue is to stimulate discussions around open problems of FL for IoT and edge computing. It focuses on sharing of the most recent and groundbreaking work on the study and application of FL in IoT and network edge.

This special issue receives both theoretical and application-based contributions in FL for IoT and edge computing. The following five papers are accepted after rigorous reviews by several external reviewers and guest editors.

The paper "A Collaborative Medical Diagnosis System Without Sharing Patient Data" by NAN, et al. proposes and builds a secured and explainable machine learning framework to address the issue of current machine learning not being able to fully exploit its potentials because the data usually sit in data silos and privacy and security regulations restrict their access and use. Their approach can share valuable information among different medical institutions to improve the learning results without sharing the patients' data. It also reveals how the machine makes a decision through eigenvalues to offer a more insightful answer to medical professionals.

HAN et al. give an overview of numerous state-of-the-art methods in the literature related to FL on non-IID data in the paper "A Survey of Federated Learning on Non-IID Data". This paper also proposes a motivation-based taxonomy, which classifies these methods into two categories, including heterogeneity reducing strategies and adaptability enhancing strategies. Moreover, the core ideas and main challenges of these methods are analyzed and several promising research directions are outlined.

The third paper "Federated Learning Based on Extremely Sparse Series Clinic Monitoring Data" by LU et al. designs a medical data resampling and balancing scheme for federated learning to eliminate model biases caused by sample imbalance and provide accurate disease risk prediction on multi-center medical data. Experimental results on a real-world clinical database demonstrate the improvement of accuracy and tolerance for missing data.

The fourth paper "MSRA-Fed: A Communication-Efficient Federated Learning Method Based on Model Split and Representation Aggregate" by LIU et al. verifies that the outputs of the last hidden layer can record the features of training data. Accordingly, they propose a communication-efficient strategy based on model split and representation aggregate. Specifically, the authors make the client upload the outputs of the last hidden layer instead of all model parameters when participating in the aggregation, and the server distributes gradients according to the global information to revise local models. Experimental results indicate that their new method can upload less than one-tenth of model parameters, while preserve the usability of the model.

The last paper "Neursafe-FL: A Reliable, Efficient, Easy-to-Use Federated Learning Framework" by TANG et al. introduces a reliable, efficient and easy-to-use federated learning framework named Neursafe-FL. Based on the unified application program interface (API), the framework is not only compatible with mainstream machine learning frameworks such as Tensorflow and Pytorch, but also supports further extensions, which can preserve the programming style of the original framework to lower the threshold of FL. At the same time, the design of componentization, modularization, and standardized interface makes the framework highly extensible. Their source code is also publicly available.

We would like to thank the authors and reviewers for their hard work and contributions. This special issue would not be possible without their help and collaboration.

A Collaborative Medical Diagnosis System Without Sharing Patient Data

NAN Yucen[1], FANG Minghao[2], ZOU Xiaojing[2], DOU Yutao[3], Albert Y. ZOMAYA[3]

(1. College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410003, China;
2. Tongji Hospital, Tongji Medical College, Huazhong University of Science and Technology, Wuhan 430074, China;
3. Center for Distributed and High Performance Computing, University of Sydney, Sydney 2008, Australia)

DOI: 10.12142/ZTECOM.202203002

https://kns.cnki.net/kcms/detail/34.1294.TN.20220901.1320.002.html, published online September 1, 2022

Manuscript received: 2022-06-10

Abstract: As more medical data become digitalized, machine learning is regarded as a promising tool for constructing medical decision support systems. Even with vast medical data volumes, machine learning is still not fully exploiting its potential because the data usually sits in data silos, and privacy and security regulations restrict their access and use. To address these issues, we built a secured and explainable machine learning framework, called explainable federated XGBoost (EXPERTS), which can share valuable information among different medical institutions to improve the learning results without sharing the patients' data. It also reveals how the machine makes a decision through eigenvalues to offer a more insightful answer to medical professionals. To study the performance, we evaluate our approach by real-world datasets, and our approach outperforms the benchmark algorithms under both federated learning and non-federated learning frameworks.

Keywords: explainable machine learning; federated learning; secured data analysis; medical applications

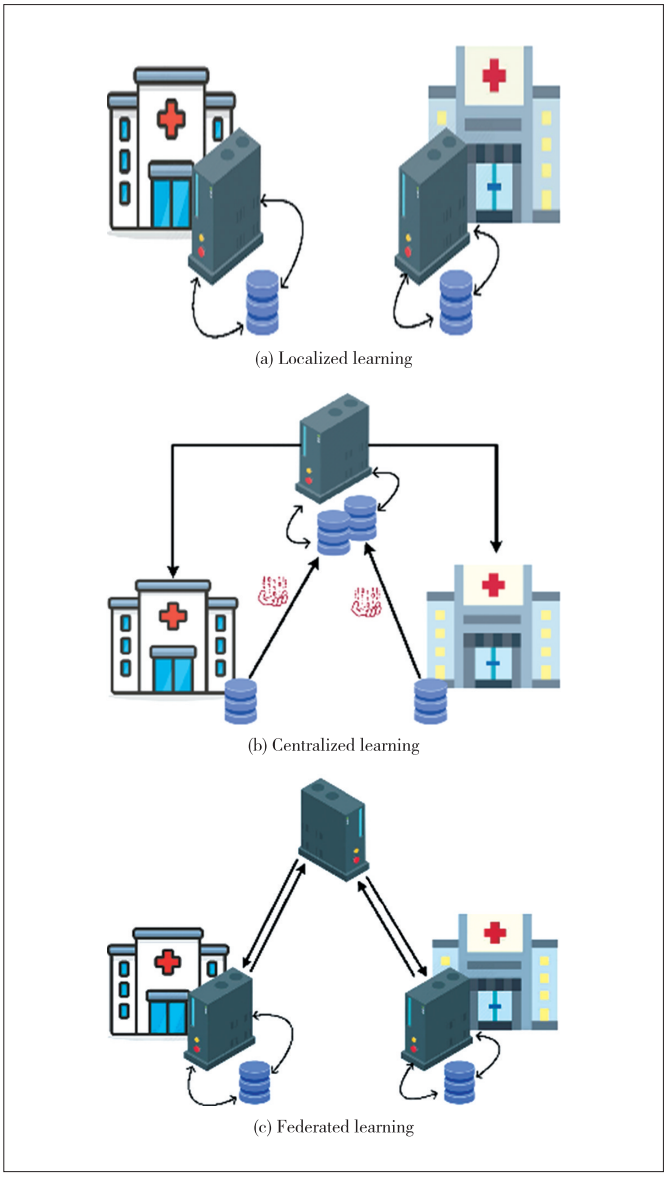
Citation (IEEE Format): Y. C. Nan, M. H. Fang, X. J. Zou, et al., "A collaborative medical diagnosis system without sharing patient data," *ZTE Communications*, vol. 20, no. 3, pp. 3 - 16, Sept. 2022. doi: 10.12142/ZTECOM.202203002.

1 Introduction

Machine learning (ML) has played an important role in the healthcare industry, serving as a decision support system for medical diagnosis, and actively promotes smart medicine development[1-2]. It can be used to complete some laborious and often time-consuming routine tasks for better resource utilization. More importantly, ML can offer meaningful support for clinical decision-making by comprehensively analyzing electronic healthcare records (EHR)[3-5]. More than 70% of medical institutions worldwide have implemented EHR systems, but just 3% of them can exchange data over the network[6]. Without a secured framework for managing the use of EHR[7], patients' information is at a high risk of cyber threats. Meanwhile, the performance of ML could be severely degraded by the limited data available locally.

The data security concerns lead to EHRs which are often used locally for analysis and learning, as depicted in Fig. 1(a). On the other hand, the medical data available in a single place are often not enough to fully exploit the advancement of ML. The lack of data can be solved by using a centralized system as shown in Fig. 1(b) to store the data from multiple sources. However, the security threats to the centralized system come from multiple aspects, such as data transmission over the network, data leaking from the centralized server, and manipulation and misconduct in handling patient data. All these threats pose unique technical and ethical challenges for this solution. Federated learning (FL) is a burgeoning distributed ML paradigm to collectively train a model as a whole without explicitly exchanging data samples[8]. It enables a party (such as medical institutions and organizations) to transparently and securely share knowledge with other parties[9]. FL can be roughly divided into three groups, namely, horizontal FL, vertical FL, and transfer FL[10]. The horizontal FL means that the datasets used for training have the same feature space across all parties. The vertical FL uses different datasets of different feature spaces to jointly train a global model. The transfer FL refers to using transfer learning to utilize a pre-trained model that is trained on a similar dataset for solving a different problem. In this study, the federated learning applied to different medical institutions is horizontal FL[11-12]. Horizontal FL can help solve the problem of lack of data for some hospitals, and only the model parameters are exchanged among parties while developing a global diagnostic model, as shown in Fig. 1(c).

Furthermore, clinical heterogeneity, lack of specific moni-



▲ **Figure 1. Different types of learning**

toring markers, and interpretive uncertainty may lead to misdiagnosis in computer-aided diagnosis. Therefore, in addition to data privacy, a secured medical decision-support system also involves generating reliable and trustful results by providing instrumental clues to medical professionals on why the decision is made, which is also known as explainable/interpretable machine learning[13-15]. Some of the explainable ML techniques are model-dependent, especially for linear models and decision trees, while the others are model agnostic and can be applied to any supervised ML model. The model interpretability is often available for those trained locally, but it is still an open question for FL.

In this study, we develop an explainable XGBoost model (a tree-based extreme gradient boost model) under a horizontal federated learning framework, EXPERTS, to construct a secured medical decision-support system. In the system, we can achieve the desired learning performance without sharing patient data and provide consistent global interpretability among parties. To fully demonstrate and understand the system, we extended the edge analytics framework[16] to collect the same set of features (demographic characteristics, clinical features, vital signals, and laboratory tests) of COVID-19 patients from different places and store the collected data locally. Then EXPERTS is applied to predict the status of the hospitalized COVID-19 patients during their stay. The main contributions of this paper are summarized as follows:

• We propose an explainable XGBoost under the horizontal FL framework, called EXPERTS, to construct a secured medical decision-support system. In this system, only model parameters are shared among parties to build a global model without sharing any patient's data, thereby protecting patients' privacy without losing performance.

• We implement the Shapley value to provide the horizontal FL model interpretability by revealing the detailed feature importance at each party. Within the system, the feature importance is consistent between parties, which means we can provide global model interpretability for all parties.

• We demonstrate the practicality of EXPERTS by a real-world COVID-19 dataset and an open medical dataset named Cerebral Vasoregulation in the elderly with stroke. Our results confirm that EXPERTS can achieve the same performance level as the centralized learning approach.

The remainder of this paper is organized as follows. Section 2 gives the motivation for our work. Section 3 shows the design of our study. Section 4 reveals the experimental results of our design. Section 5 concludes this work and sketches the future work.

2 Related Work

Data-driven ML has emerged as a promising option for developing accurate and efficient diagnostic tools from large volumes of medical data. In Ref. [17], the authors argued that an AI-based tumor detector requires massive and a wide range of data, including possible anatomies, pathologies and many others, to make valuable clinical suggestions, and to be practical and generalizing well to new patients. However, it is impractical to include all of them among medical institutions as the data are highly sensitive and the usage is strictly regulated. Even when the patients are de-identified by removing their personal information, their privacy could still be exposed by reconstructing faces from computed tomography (CT) or magnetic resonance imaging (MRI) data[18].

Federated learning[8] is one of the emerging approaches to address security challenges by introducing the idea of sharing the characteristics of the ML model rather than the data itself. More specifically, it keeps the patient data locally for each participant and only transmits the intermediate results of the model at local servers to the centralized server for model iteration and

update, thereby reducing communication intensity and improving data privacy. Since proposed by Google first in 2017[8], FL has attracted more and more attention among researchers and has been widely utilized in various privacy-sensitive domains. It has great potential for medical and healthcare applications[19 - 21].

Nevertheless, there are still several issues that remain in FL. For example, to improve performance, researchers focus heavily on neural networks but ignore other machine learning models, such as decision trees. Not only that, by emphasizing performance using neural networks, researchers also ignore the interpretability of the model, which is crucial for medical professionals to understand what drives the ML to make the decision. The study on the interpretability of FL is very limited. The authors in Ref. [22] studied the model interpretability under the framework of vertical FL. In this work, a party contributes to the vertical FL model by sharing its features with others. The contribution of the party can thus be represented by the combined contributions of its shared features. In other words, the interpretability of the federated model is provided by the group Shapley values instead of the individual ones. To our knowledge, this work is the first interpretability analysis based on original features under the horizontal FL framework, and will not affect the global interpretability for using different data samples stored in medical institutions for model training.

3 Method

The technical detail of EXPERTS is given in this section. To make the entire system clearer, we illustrate the detailed processing flow in Fig. 2. As can be seen, the local data in each hospital will never leave the local physical area. In the local database, data pre-processing is first performed through the patients/variables filter and abnormal removal. Then, we rely on statistical transferring and one-hot sampling to further refine the pre-screened data. After getting the available data, we use the tree-based SHapley Additive exPlanations (SHAP) to rank all the features by correlation, and select the top-20 features for subsequent local learning. Consequently, we perform local fast learning through the initialized model, and upload the local model's parameters to the central processor in the federated node. By applying certain mathematical methods to weight or average the various parameter sets from different hospitals, which generalizes local model parameters to global parameters, we send them back to each local node for model update and learning. In this section, we briefly describe these steps encompassing the learning strategy, like data pre-processing, followed by horizontal federated-XGBoost and model interpretability.

3.1 Data Pre-Processing

Before the data analysis model is performed, the raw data are obtained, organized, and pre-processed locally. The data pre-processing includes variable extraction, unification, artifact removal, feature generation and others. In this regard, we first need to obtain patient data from the local hospital database. The data can be retrieved in different forms and need to be turned into a table-like structure. The steps of data unification involve timestamp unification (unifying time count), unit unification (unifying measurement unit for each variable), categorical variables form unification (converting categorical variables to numeric variables), and representation unification (unifying the name of the feature). In the artifact removal step, multiple procedures are performed to ensure the validity and quality of the data. For example, the timestamp artifact removal procedure is used to remove unrelated medical records. The out-of-range artifact removal procedure is used to remove the values of the feature that greatly exceed its physiological range. The data normalization is still valuable to reduce the adverse effects associated with the use of physiological data. More specifically, Z-score is used to normalize all included features to obtain a normalized version of variables, which is computed by $\text{Normalized}(x) = \frac{x - \bar{x}}{\text{std}(x)}$, where $\bar{x}$ and $\text{std}(x)$ represent the mean and standard deviation of x, respectively. The time-series variables are converted into static features via discretization.

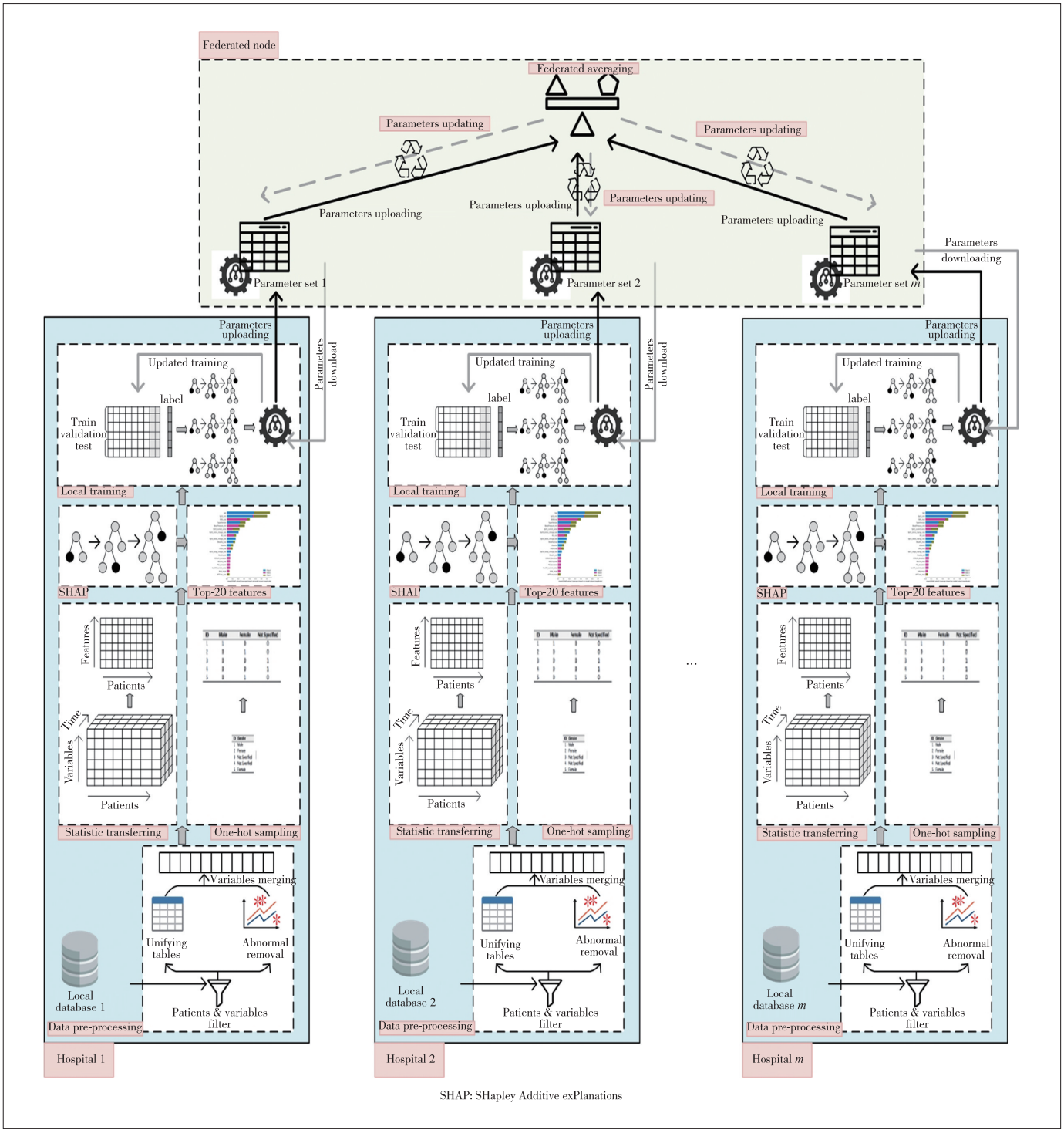
3.2 Horizontal Federated-XGBoost

Although neural networks are currently the most popular ML models, the lack of clear interpretability makes them hard to justify their decisions, which is a prerequisite for the widespread adoption of machine learning approaches by healthcare communities. Instead, decision trees (DT) are regarded as reliable alternatives for balancing accuracy and interpretability. DT is a tree-like ML model that consists of nodes and edges, where the internal nodes present the test instances, the edges present the results, and the leaf nodes present the prediction results. In short, the path from the root to the leaf represents the prediction rule. Although the gradient boosting decision tree (GBDT) has not yet received enough attention under the FL framework, the representative XGBoost is a promising candidate to achieve the desired ML performance.

We first give a recap of the the XGBoost algorithm. For a given set of n independently identically distributed and labeled examples $\{(x_i, y_i),\ i = 0,\cdots,n\}$, where $\boldsymbol{X} \in \mathcal{R}^{n \times d}$ and d represents the feature dimension. The goal of XGBoost is to train a learning model with a set of parameters to minimize the objective loss function for the K iterations, which can be represented as follows:

$$\text{Objective} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{1}$$

where $\sum_{i=1}^{n} l(y_i, \hat{y}_i)$ is the total training loss after K iterations to measure how well the model fits. More specifically, y_i is the real label, and $\hat{y}_i$ presents the predicted output for the i-th



▲ **Figure 2. Complete workflow of EXPERTS**

data sample after K iterations through using K CARTs, which can be calculated as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \ . \tag{2}$$

$\sum_{k=1}^{K} \Omega(f_k)$ in Eq. (1) is the regularization term to measure the complexity of the model, and $\Omega(f_k)$ can be depicted as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda ||\omega||^2, \tag{3}$$

with the component ω_j of ω being the score/weight of the j-th leaf node of the tree. T is the number of leaf nodes, and $\frac{1}{2}\lambda||\omega||^2$ is the L2 regularization term of the leaf node score. The score of each leaf node is increased by L2 smoothing to prevent overfitting. In short, by minimizing the objective function of Eq. (1), both the accuracy and stability of the model can be considered, and it is the balance between the deviation and the variance.

Moreover, XGBoost is an additive model and the newly generated tree needs to fit the last predicted residual, which means the objective is no longer to directly optimize the entire objective function, but to optimize the objective function step by step from the first tree to the K-th tree. Then, $\hat{y}_i$ can be rewritten as $\hat{y}_i^k = \hat{y}_i^{(k-1)} + f_k(x)$ for the k-th iteration.

After that, we need to find the best split of samples of the tree from root to leaf. By using the greedy algorithm to search for the best split which aims to maximize the learning gain at each iteration, the gain can be calculated as follows:

$$\text{Gain} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} + \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right], \tag{4}$$

where I_L and I_R represent the left and right sets of data sample indices. When searching for the best split point, instances g_i and h_i in the left and right space will be calculated for getting the value of Gain. When a CART structure is fixed, the weight ω_j of a leaf node j is calculated by:

$$\omega_j^{\dagger} = -\frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}. \tag{5}$$

Considering the generality, we apply a particular logistic loss function $l\left(y_i, \hat{y}_i^{(t-1)}\right) = y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i)\ln(1 + e^{\hat{y}_i})$ as our picked loss function. Then, the first and second order gradient of the loss function can be derived as:

$$g_i = \frac{1}{1 + e^{-\hat{y}^{(k-1)}}} - y_i, \tag{6}$$

and

$$h_i = \frac{1}{1 + e^{-\hat{y}^{(k-1)}}} \times \left(1 - \frac{1}{1 + e^{-\hat{y}^{(k-1)}}}\right), \tag{7}$$

separately.

In this work, we study the horizontally partitioned data for different nodes, which means the nodes have the same feature dimension and each node holds the entire features of an instance. For better understanding, we modeled this method as the following. Assuming there are L distributed parties $P_0,\cdots,P_L$ that hold sample sets $X_0,\cdots,X_L$, where each $X_l = \begin{bmatrix} X_{l0} \\ \cdots \\ X_{lm} \end{bmatrix}$ involves m samples in the l-th party, entities accompanied with the label involved in the l-th party can be shown as $\left[(x_{l0}^0,\cdots,x_{l0}^d, y_{l0}),\cdots,(x_{lm}^0,\cdots,x_{lm}^d, y_{lm})\right]$. To implement the XGBoost under the federated-learning framework, the key idea is to calculate the parameters g_i and h_i at each local party discussed in Eqs. (6) and (7), and then pass them to the central aggregator to determine an optimal split through iterative model averaging to further update the model. In short, XGBoost under the FL framework is summarized as follows:

- Each party downloads the latest XGBoost model from the central aggregate server.
- Each party uses local data to train the downloaded XGBoost model and uploads the gradient to the central aggregate server, and the server aggregates the gradient of each user to update the model parameters.
- The central aggregate server distributes the updated model to each party.
- Each party updates the local model accordingly.

3.3 Model Interpretability

When the final model updates, we will conduct a feature importance analysis on local nodes and compare the explanation from each local node to check the robustness of our model.

In the past, people used Gain[23] or Split[24] to explain the model as they could summarize a complicated ensemble model and provide insight into what features drive the model's prediction. However, it cannot be ignored that in some cases, the rankings of Gain and Split are often inconsistent even for the same features. To solve the problem of inconsistency in the feature attribution method, we choose Shapley value as an explanatory tool for our model. As defined in Ref. [25], the Shapley value for the j-th feature is a solution concept in the cooperative game theory, which can be obtained by:

$$\phi_j(\text{val}) = \sum_{\mathcal{S}\subseteq\{\boldsymbol{X}_1,\cdots,\boldsymbol{X}_d\}\setminus\{\boldsymbol{X}_j\}} \frac{|\mathcal{S}|!\left(d - |\mathcal{S}| - 1\right)!}{d!}\left[\text{val}\left(\mathcal{S}\cup\left\{\boldsymbol{X}_j\right\}\right) - \text{val}(\mathcal{S})\right], \tag{8}$$

where $\mathcal{S}$ is the sub-set of features used in the model, $\boldsymbol{X}$ is the vector of features of the instance to be explained, and d is the number of features defined above. $\text{val}_X(\mathcal{S})$ is the prediction of the eigenvalues in the set S, and the features excluded in the set S are marginalized as:

$$\mathrm{val}_X(\mathcal{S}) = \int \hat{f}(x_1,\cdots,x_d)\mathrm{d}P_{X\notin\mathcal{S}} - E_X(\hat{f}(X)), \tag{9}$$

which performs multiple integrations for each excluded feature. The Shapley value obtained in this way can satisfy efficiency, symmetry, dummy, and additivity[26] at the same time, which can be regarded as the definition of fair expenditures.

However, the exact Shapley value must be estimated using the j-th feature and all possible subsets that exclude the j-th feature. As more features are involved, the computational complexity of the accurate solution to this problem increases exponentially. To reduce the complexity, we adopt SHapley Additive exPlanations as an alternative. SHAP is the Shapley value estimate based on the game theory, and it has two variants, namely KernelSHAP and TreeSHAP. The computation cost of KernelSHAP is very high as it aims for serving all ML models, so it can only approximate the actual Shapley value. TreeSHAP is fast; it can calculate the accurate Shapley value, and even correctly estimate the Shapley value when the features are correlated.

The TreeSHAP value is defined below:

$$f(x) = g(x') = \phi_0 + \sum_{j=1}^{M}\phi_j z_j', \tag{10}$$

where $f(x)$ represents the predicted value of the sample in the decision tree, $z_j' \in \{0,1\}^M$ represents how many features of all d features are included in the decision path where the sample is located. For example, if the feature k is not in its decision path, the SHAP value of the corresponding feature is 0, that is, $\phi_k = 0$, which means that the feature k will not contribute to the final predicted value. Moreover, ϕ_i is represented as below:

$$\phi_j = \sum_{\mathcal{S}\subseteq N\{j\}} \frac{|\mathcal{S}|!(M-|S|-1)!}{M!}\left[f_x(\mathcal{S}\cup\{j\}) - f_x(\mathcal{S})\right], \tag{11}$$

where N is the collection of all the features in the training set, and its dimension is M; $\mathcal{S}$ is a subset extracted from N and its dimension is $|\mathcal{S}|$.

The pseudo code of our proposed algorithm EXPERTS is provided in Algorithm 1, which is formed by the above process.

Algorithm 1: EXPERTS

Input: each party P_l inputs m samples, and each sample has all d features and the corresponding label y_{lm}
Output: K decision trees with global feature interpretability
1: Perform pre-processing steps discussed in Section 3.1 in each local party for every sample
2: **Aggregate server**
3: **for** each round $t = 1, 2, \cdots$ **do**
4: set L local parties with hyper-parameters
5: for the maximal score, aggregate server sends gain to other local Ps
6: **end for**
7: **Local client:**
8: **for** $l = 1 \rightarrow L$ **do**
9: split samples in P_l into Ω batches
10: receive default hyper-parameter from aggregate server
11: **for** each local epoch from 1 to E **do**
12: **for** batch $\omega \in \Omega$ **do**
13: P_l initializes $\{\hat{y}\}_{ml}$ with hyper-parameters
14: **end for**
15: **end for**
16: **end for**
17: **for** $k = 1 \rightarrow K$ **do**
18: **for** $l = 1 \rightarrow L$ **do**
19: P_l computes g_i and h_i described in Eqs. (6) and (7)
20: **end for**
21: **for** each node in the current tree **do**
22: **for** $j = 1 \rightarrow d$ **do**
23: P_l run Eq. (4) for split
24: **end for**
25: for the maximal score, P_l sends gain to other Ps
26: **end for**
27: update $y_0, \cdots, y_d$ based on the weights in Eq. (5)
28: calculate the approximate Shapley value through Eq. (10)
29: **end for**

4 Performance Evaluation

This section first presents our experiments' setup, which involves both a testbed study and a numerical study. The testbed is a real-world prototype for COVID-19 diagnosis. To study the flexibility of the proposed framework EXPERTS, we also used a publicly available dataset for stroke in our experiments. For each medical application, we treated the data collected from two different hospitals but the approach can be extended to multiple parties. The framework's performance was comprehensively evaluated using multiple metrics, including accuracy, precision, recall, F1 score, receiver operating characteristic (ROC) curve, and Precision-Recall (PR) curve. We also implemented numerous benchmark algorithms involving the federated learning framework and its counterparts, namely, federated-multilayer perceptron (MLP), XGBoost, MLP, and Random forest. All algorithms are performed after the missing values imputed with the mean, except for EXPERTS and XGBoost.

4.1 Experiment Setting

To evaluate the performance of our proposed algorithm and to conduct a fair comparison, all data analytics were carried out on the same setting servers, which was a laptop with a 2.3 GHz Intel Core i5 CPU and 8 GB memory. Additionally, all computational steps involved in this study, such as preprocessing and learning, and the proposed algorithm, along with the selected benchmark algorithms, were all implemented in Python 3.8 with PyTorch and TensorFlow.

4.1.1 Dataset

• Real-world dataset: Our study was performed at two designated hospitals for treating COVID-19 patients during the outbreak. We retrospectively analyzed 1 012 and 1 642 hospitalized patients separately, involving patients with the mild symptom, severe symptom, and critical symptom diagnosed according to WHO interim guidance[27]. Laboratory confirmation of SARS-CoV-2 infection was performed by the local health authority[1]. In total, 24 items within CBC (shown in Table 1), two demographic variables (gender and age), five types of comorbidities (including hypertension, coronary heart disease, diabetes, stroke, and cancer), and five time-series vital signals (breath, blood pressure, SpO2, pulse, and temperature) were used to represent the physical condition of patients in this study.

▼ **Table 1. Feature abbreviation checklist within complete blood count (CBC) test**

Abbreviation	Full Name
EON (#)	Eosinophils (#)
EON (%)	Eosinophils (%)
EOP (#)	Basophils (#)
EOP (%)	Basophils (%)
HCT	Hematocrit
HGB	Hemoglobin
LYM (#)	Lymphocyte (#)
LYM (%)	Lymphocyte (%)
MCH	Mean corpuscular hemoglobin
MCHC	Mean corpuscular hemoglobin concentration
MCV	Mean corpuscular volume
MONON (#)	Monocyte (#)
MONON (%)	Monocyte (%)
MPV	Mean platelet volume
NEU (#)	Neutrophils (#)
NEU (%)	Neutrophils (%)
PCT	Procalcitonin
PDW	Platelet distribution width
P-LCR	Platelet-large cell ratio
PLT	Platelet
RBC	Red blood cell
RDW-CV	Red blood cell distribution width CV
RDW-SD	Red blood cell distribution width SD
WBC	White blood cell

• Open dataset: We also used a public non-image based real-world dataset, known as Cerebral Vasoregulation in the Elderly with Stroke[2][28] in our experiment. Cerebral Vasoregulation in the Elderly with Stroke with numerous feature values can be identified as a binary category (stroke or non-stroke). This dataset involves 164 patient instances and contains a large number of missing values, since it was produced from the data collected in a real medical care environment after a long period of time. To simulate the framework of federated learning, we first split this data into 70% training set, 20% validation set, and 10% test set. Then we randomly split the training set to simulate data from two different institutions. In this study, we divided the training set into 65% and 35% for performance evaluation.

4.1.2 Benchmark Algorithm

To quantitatively evaluate the performance of our EXPERTS algorithm, we implemented multiple algorithms as our performance benchmarks, including:

• Federated-MLP (with mean value imputation): a multilayer perceptron model under the federated learning framework. Additionally, MLP cannot handle the missing values, so we used the mean value for the imputation.

• XGBoost (without data imputation): an extreme tree-based model under a non-federated learning framework.

• MLP (with mean value imputation): MLP model under non-federated learning framework for data processing and the mean value is used for the imputation.

• Random forest (with mean value imputation): a tree-based model under the non-federated learning framework. Like MLP, the random forest cannot handle the missing values as well, so we used the mean value for imputation.

4.2 Results Analysis

4.2.1 Performance Evaluation of Federated-XGBoost

In these tests, we evaluated the performance of EXPERTS on the COVID-19 dataset. We used 70% of the samples as the training set, 20% of the samples as the validation set, and the rest as the testing set. Our approach can achieve 93% accuracy in predicting the patients' clinical courses. However, accuracy is not always enough to evaluate the clinical performance of the algorithm. We also employed the averaged Precision, Recall, and F1-score as performance metrics in our experiments. Precision and recall are both used to evaluate the quality of classification to show the accuracy of the model. More specifically, precision indicates the percentage of the relevant results retrieved, and recall refers to the percentage of the total relevant results correctly classified. The F1-Score is the harmonic mean of Precision and Recall. The results of predicting COVID-19 patients' clinical course are shown in Table 2.

Moreover, the results of the tests are plotted in Fig. 3, and Fig. 3(a) shows the areas under the receiver operator curves (AUROCs) for different COVID-19 patients. The accuracy for the mild, severe and critical patients reaches 0.994, 0.981 and

1.The studies involving human participants were reviewed and approved by the ethical committee of Tongji Hospital, Tongji Medical College, Huazhong University of Science and Technology, China. Informed patient consent was waived by the Ethics Commission due to the retrospective and observational nature of this study.
2.The dataset is available on the website: https://physionet.org/content/cves/1.0.0/.

▼Table 2. Classification results for EXPERTS

	Precision	Recall	F1-Score
Mild	0.96	0.91	0.93
Severe	0.94	0.96	0.95
Critical	0.89	0.87	0.88
Macro average	0.93	0.91	0.92
Weighted average	0.93	0.93	0.93

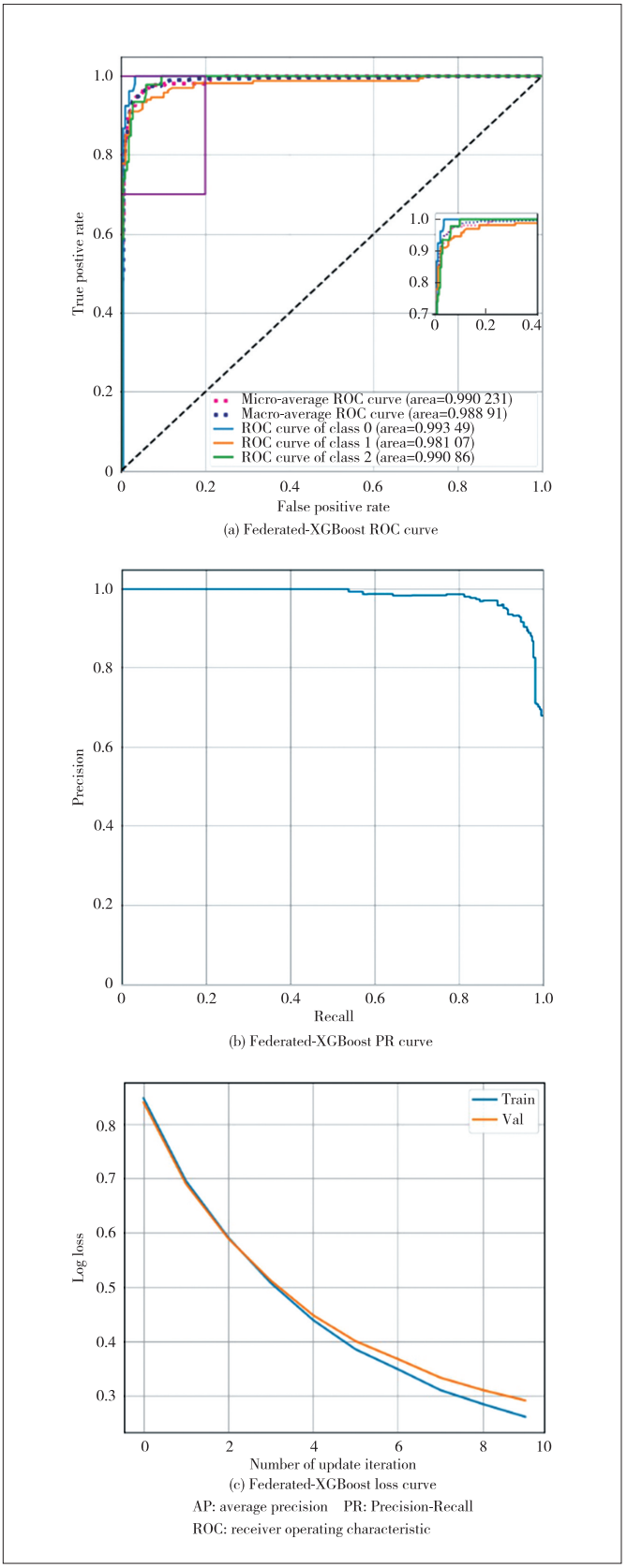
0.991, respectively. The embedded figure enlarges the details of the top left corner of Fig. 3(a). The PR curve is a non-decreasing function of the true positive rate (TPR) with respect to the false positive rate (FPR). The PR curve is shown in Fig. 3(b) with Precision as the *Y*-axis and Recall as the *X*-axis. It can be seen that when Recall is less than 0.5, Precision is always 1; while Recall is greater than 0.5, Precision gradually decreases from 1 to around 0.7.

The loss value of our iteration-like method is shown in Fig. 3(c). The *x*-axis of Fig. 3 (c) is the number of update iterations of the training, and the *y*-axis represents the loss function of our model. It can be seen from the figure that the loss value drops greatly in the initial iteration of the training stage, indicating that the learning rate is appropriate and the gradient descent process is carried out. After the six-th iteration, it can be seen obviously that the loss curve tends to be stable, and the change in the loss was not as obvious as in the beginning.

4.2.2 Feature Importance

In this experiment, we studied the model interpretability of EXPERTS by identifying the important features. We performed the averaging on the aggregated server to update the parameters gathered from local parties and returned the updated parameters to each party for the next iteration. In our tests, we found that the feature importance of EXPERTS derived from each party was the same no matter how the data varied. This suggests that EXPERTS can address the unevenly distributed datasets and avoid using the local optima for a global explanation.

Fig. 4 shows the top 20 important features and their individual contribution to the final diagnosis results. Figs. 4(a), 4 (b), and 4(c) represent that the summary plot of COVID-19 patients is in mild, severe, and critical status. In these figures, the *y*-axis lists the features in the reverse order of their importance from top to bottom, and the *x*-axis represents the SHAP value. Besides, the features that drive the prediction toward positive are in red, and those pushing the prediction negative are in blue. By reviewing the influence of the selected features in the model, it is obvious that age plays a crucial role for mild-symptom patients shown in Fig. 4(a) and severe-symptom patients shown in Fig. 4(b). The elder the age, the less likelihood for those patients to be less affected by COVID-19, and they will develop into a severe or worse situation. For the comorbidities, cancer could be a useful bio-marker to identify the risk of COVID-19 patients being mild shown in Fig. 4(a) or severe shown in Fig. 4(b). Cancer will increase the chance of poor



▲Figure 3. Performance metrics for federated XGBoost on COVID-19 cases

prognosis, but comorbidities are not the main factors for critically ill patients. The vital signal SpO2 also plays a key role in our experiments. The greater the minimum value of SpO2 in an observation period, the more likely for the patients to stay in a mild condition shown in Fig. 4(a). Otherwise, the possibility of turning severe condition is higher shown in Fig. 4(b). For those critically ill patients in Fig. 4(c), the SpO2 current reading becomes more important. Our findings are consistent with the earlier medical studies[29 - 30]. In summary, Fig. 4 shows the top 20 important features among all features in descending order of their mean absolute SHAP values, and plots their distribution across all predictions accordingly.

4.2.3 Model Generalization

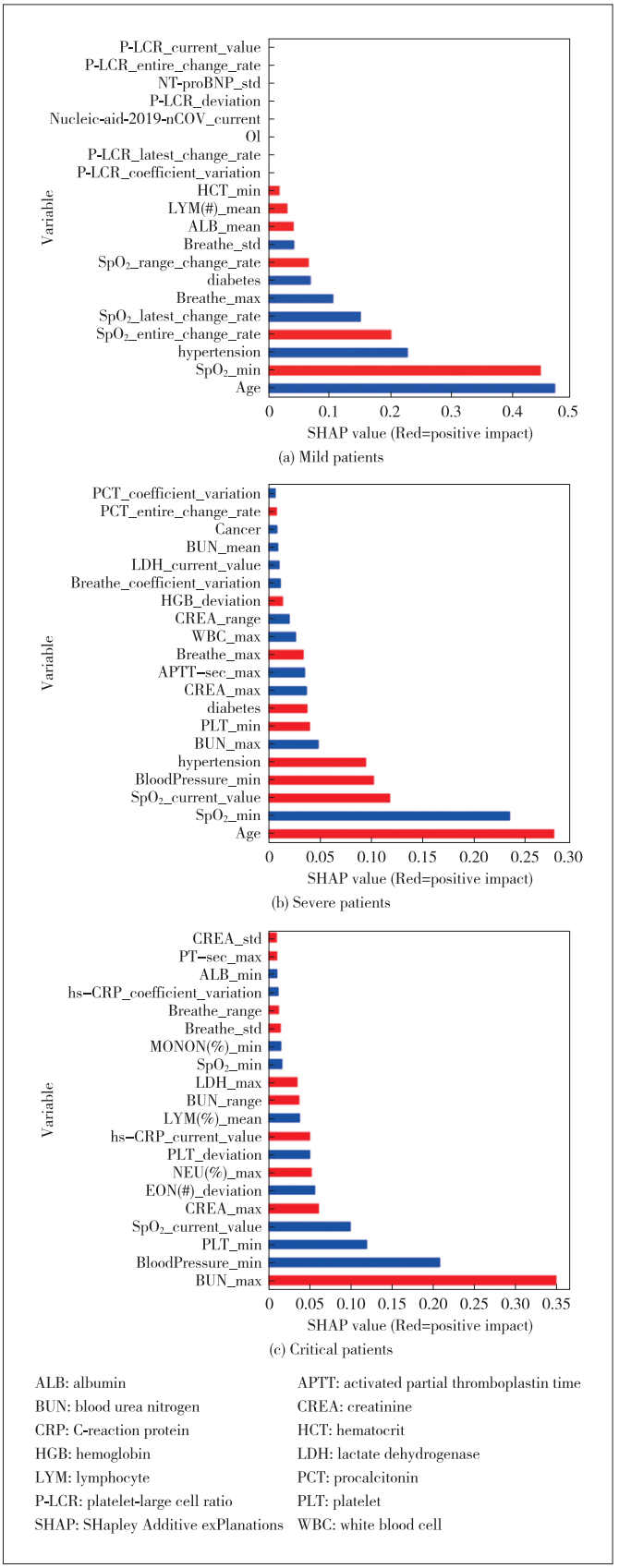
To prove the generalization of EXPERTS, we also tested it on a public dataset, called Cerebral Vasoregulation in the Elderly with Stroke. In Fig. 5(a), we can see that the area under the curve (AUC) achieves 70.8% after ten times the model update, and the PR curve can be seen in Fig. 5(b). Furthermore, in Fig. 5(c), it is obvious that within the process of the ten times iteration, the loss curve shows a non-increasing trend. EXPERTS cannot achieve the same performance level of the COVID-19 case as ML is restricted by the relatively small sample size. Meanwhile, we also showed the top 20 important features in Fig. 5 (d).

4.2.4 Performance Comparison with Benchmark Algorithms

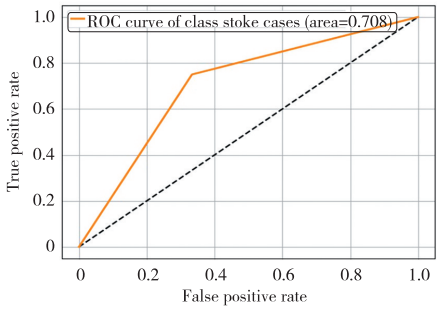
In this part, we compared EXPERTS with the selected benchmark algorithms, and all the experiments were performed on the COVID-19 dataset.

In our tests, the best accuracy of Federated-MLP can only reach 78% on the COVID-19 dataset, which lags far behind EXPERTS. A possible reason for this limited performance is that MLP cannot properly handle a large number of missing data. As we used the mean imputation method, it might change the distribution of the original data and affect the final performance. Another possible cause is that MLP is a relatively simple neural network, so its capacity is not as strong as the complex deep neural networks. Fig. 6 shows the learning results based on a fully connected neural network MLP which is typically simple under the framework of federated learning. As can be seen from Fig. 6(a), for different COVID-19 patients, their AUC can only reach 89%, 87%, and 92% respectively. Fig. 6(b) shows its PR curve, which is a non-increasing curve. As can be seen, within the interval of Recall from 0 to 1, the value of Precision drops from 1 to 0.3. Fig. 6(c) shows the loss curve of Federated-MLP within the process of iteration and model update, and the loss gradually decreases from 0.78 to 0.56 in these continuous update iterations.

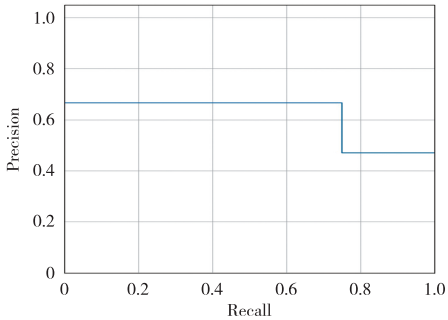
We also implemented three non-federated benchmark algorithms in our experiments, including two tree-based methods and one neural network method. We evaluated their perfor-



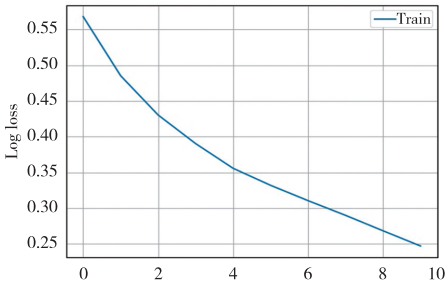
▲ **Figure 4. Feature importance among different patients types**



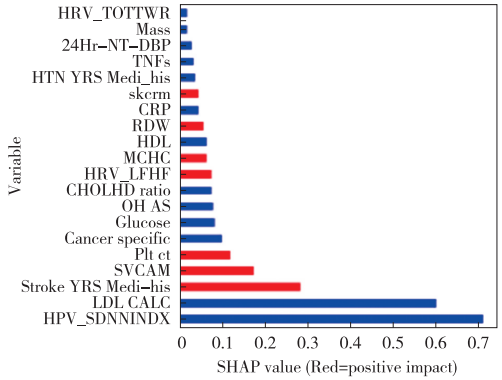
(a) Federated-XGBoost ROC curve



(b) Federated-XGBoost PR curve



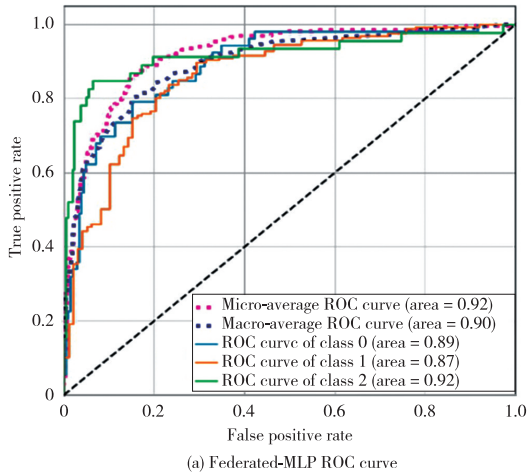
(c) Federated-XGBoost loss curve



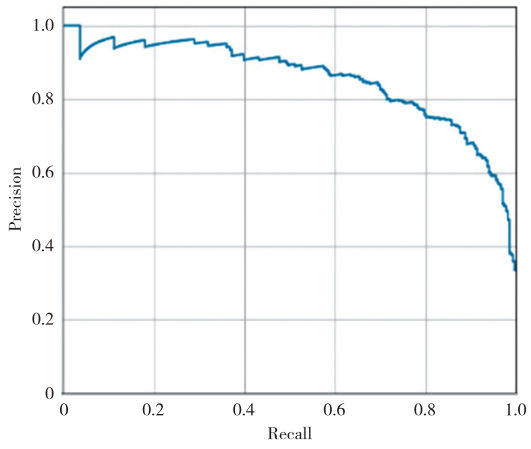
(d) Federated-XGBoost feature importance

CHOL: total cholesterol
CRP: C-reaction protein
HDL: high density lipoprotein
ROC: HRV: heart rate variability
LDL: low density lipoprotein
MCHC: mean corpuscular hemoglobin concentration
PLT: platelet
PR: Precision-Recall
ROC: receiver operating characteristic

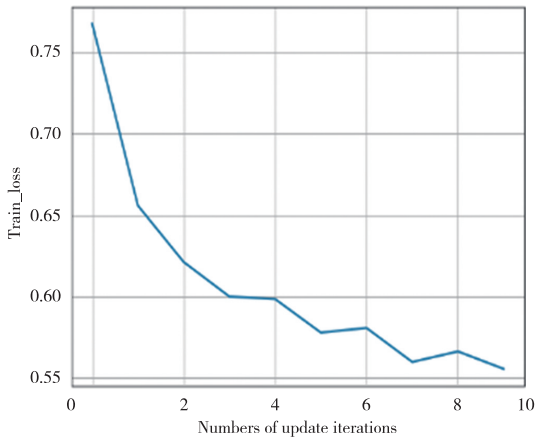
▲ **Figure 5. Performance evaluation of EXPERS on the stroke cases**



(a) Federated-MLP ROC curve



(b) Federated-MLP PR curve

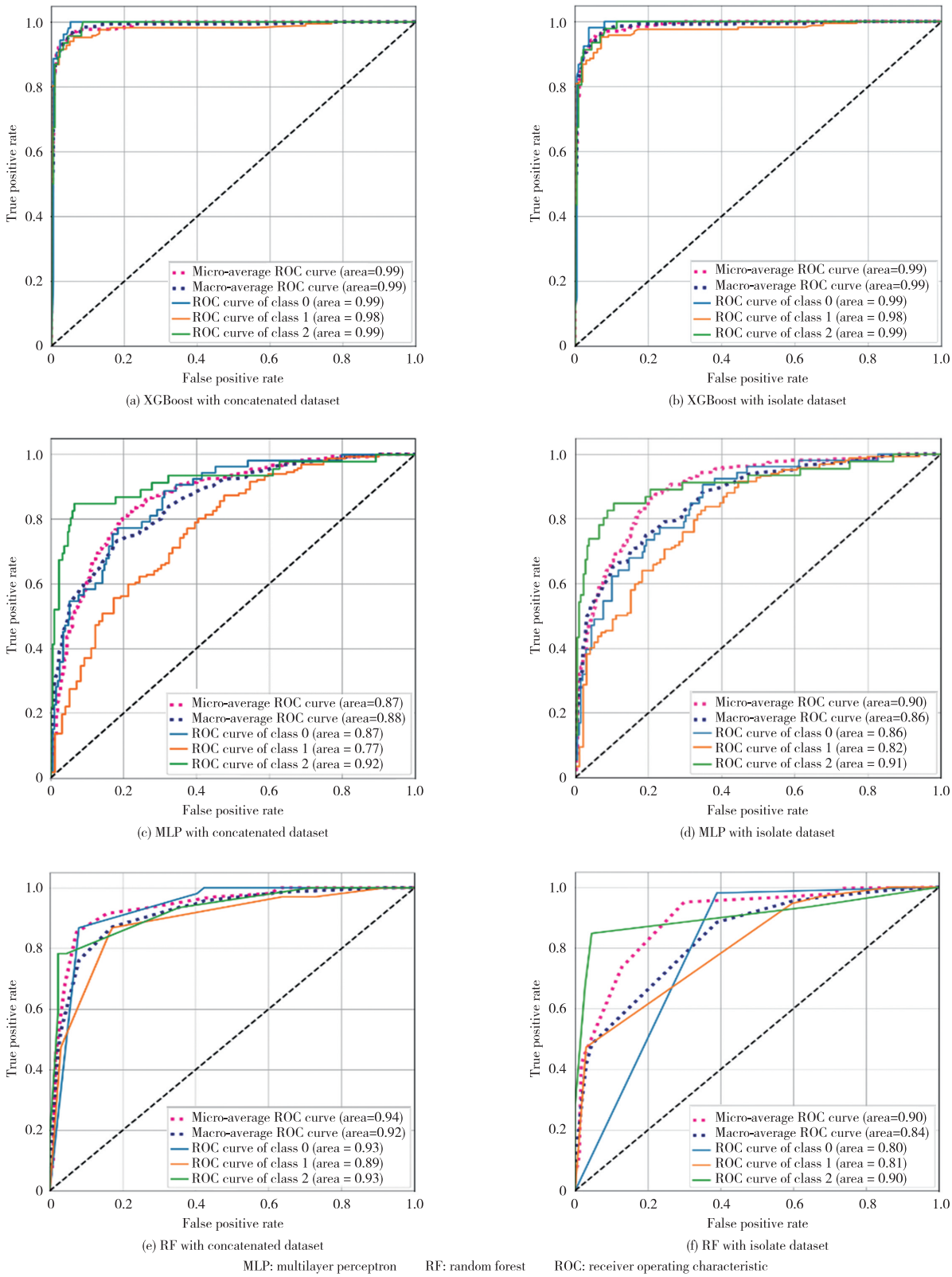


(c) Federated-MLP loss function

AP: average precision
MLP: multilayer perceptron
PR: Precision-Recall
ROC: receiver operating characteristic

▲ **Figure 6. Performance metrics for federated MLP**

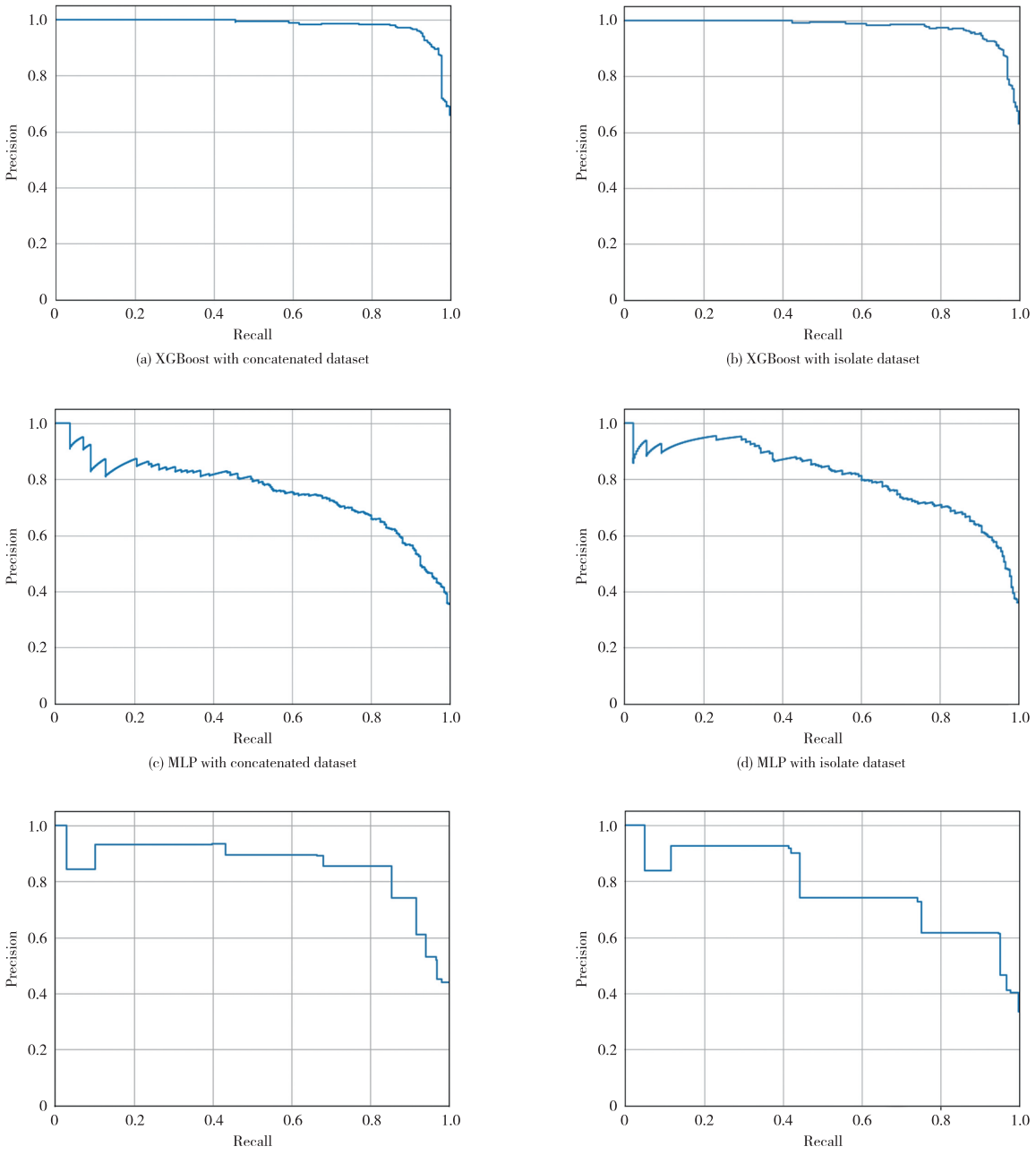
mance on the centralized dataset (bigger size) and the distributed local datasets (smaller size). Fig. 7 presents all ROC



▲**Figure 7. Comparison among ROCs**

curves for the benchmarks. Fig. 7(a) shows the result of the XGBoost algorithm on the centralized dataset. Correspondingly, Fig. 7(b) shows the result of the XGBoost algorithm on the distributed local dataset. It is easy to observe that XGBoost works well in both cases. Figs. 7(c) and 7(d) respectively show the performance of MLP on the centralized data set and the distributed local datasets. We can see that the learning effect of the distributed data sets is slightly worse than the centralized data set. Finally, in Figs. 7(e) and 7(f), we studied the performance difference of the random forest on different dataset cases. It shows that the performance of the random forest significantly drops when the size of the data reduces.

Fig. 8 shows the PR curves of all the benchmark algorithms without running under the federated learning framework, and



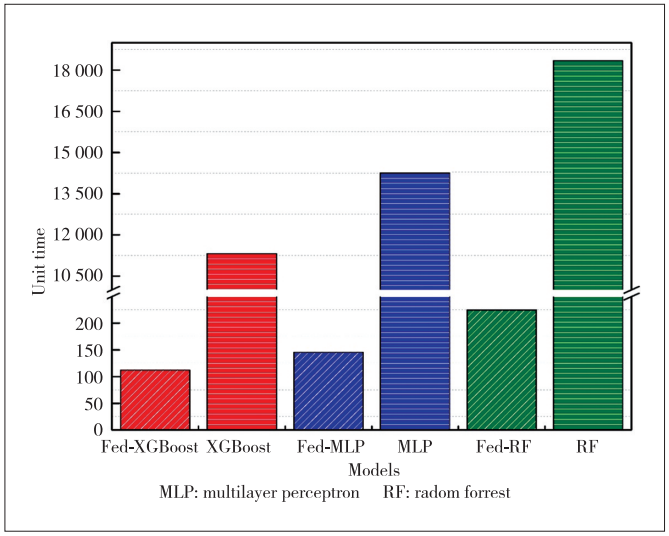
(a) XGBoost with concatenated dataset
(b) XGBoost with isolate dataset
(c) MLP with concatenated dataset
(d) MLP with isolate dataset
(e) RF with concatenated dataset
(f) RF with isolate dataset

AP: average precision MLP: multilayer perceptron RF: random forest

▲ **Figure 8. Comparison among Precision-Recall (PR) curves**

it can be seen that they are all non-increasing curves. Within the range of Recall from 0 to 1, Precision all reduces from 1 to less than 0.5 except for the two XGBoost cases. In addition, comparing them with the ROC curves shown in Figs. 3(a) and Fig 6(a) under the federated learning framework, we find that EXPERTS inherits the legacy of the XGBoost and can achieve the same performance of the XGBoost under the centralized data setting. This would suggest that EXPERTS can achieve the desired performance without sharing patients' data.

In addition to the excellent performance of learning, another advantage of federated learning is that it consumes fewer transmission resources. As depicted, the difference between federated learning and traditional centralized learning is that the original data transmission is replaced by the transmission of model parameters only, which greatly reduces the overload of data transmission and effectively improves the overall performance. From the perspective of time consumption, the results are shown in Fig. 9. It can be seen that no matter what model is used, the federated model is far superior to the one without the federated architecture.



▲**Figure 9. Overhead comparison among different models on COVID-19 cases**

5 Conclusions

The effective application of federated learning in the medical field is essential to address the security threats of personal medical data and the resource imbalance at all levels of hospitals. We show that EXPERTS could build a global model for COVID-19 patients' diagnoses without sharing their data. It also gives the leading factors of the COVID-19 patients in different statuses. To test the flexibility of EXPERTS that handles different medical applications, our system has also been verified with an open dataset for stroke. EXPERTS can adapt to the new application with traceable decision support, making it more suitable than static scoring that requires manual processing.

There are several limitations to this study. Our study is now designed as retrospective ones, and we will extend our framework to prospective studies. EXPERTS is tested as a horizontal federated learning model, and vertical federated learning should also be considered to further prove the reliability of the model since the features collected in different hospitals are usually not the same.

Acknowledgement

We are grateful to AMD Product (China) Co., Ltd. and the Sugon Information Industry Co., Ltd. for its X785-g30 series GPU server.

References

[1] DA SILVA D B, SCHMIDT D, DA COSTA C A, et al. DeepSigns: a predictive model based on deep learning for the early detection of patient health deterioration [J]. Expert systems with applications, 2021, 165: 113905. DOI: 10.1016/j.eswa.2020.113905
[2] YE B, YUAN X X, CAI Z C, et al. Severity assessment of COVID-19 based on feature extraction and V-descriptors [J]. IEEE transactions on industrial informatics, 2021, 17(11): 7456 – 7467. DOI: 10.1109/TII.2021.3056386
[3] NAPI N M, ZAIDAN A A, ZAIDAN B B, et al. Medical emergency triage and patient prioritisation in a telemedicine environment: a systematic review [J]. Health and technology, 2019, 9(5): 679 – 700. DOI: 10.1007/s12553-019-00357-w
[4] CORDERO A, GARCÍA-ACUÑA J M, RODRÍGUEZ-MAÑERO M, et al. Prevalence, long-term prognosis and medical alternatives for patients admitted for acute coronary syndromes and prasugrel contraindication [J]. International journal of cardiology, 2018, 270: 36 – 41. DOI: 10.1016/j.ijcard.2018.06.057
[5] FERRONI P, ZANZOTTO F M, RIONDINO S, et al. Breast cancer prognosis using a machine learning approach [J]. Cancers, 2019, 11(3): 328. DOI: 10.3390/cancers11030328
[6] CHOUDHURY O, GKOULALAS-DIVANIS A, SALONIDIS T, et al. Differential Privacy-enabled Federated Learning for Sensitive Health Data [EB/OL]. (2019-10-07)[2022-02-28]. https://arxiv.org/abs/1910.02578v2
[7] LIANG W, LI K C, LONG J, et al. An industrial network intrusion detection algorithm based on multifeature data clustering optimization model [J]. IEEE transactions on industrial informatics, 2020, 16(3): 2063 – 2071. DOI: 10.1109/TII.2019.2946791
[8] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C]//Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. AISTATS, 2017
[9] LI L, FAN Y X, TSE M, et al. A review of applications in federated learning [J]. Computers & industrial Engineering, 2020, 149: 106854
[10] LI T, SAHU A K, TALWALKAR A, et al. Federated learning: challenges, methods, and future directions [J]. IEEE signal processing magazine, 2020, 37 (3): 50 – 60. DOI: 10.1109/MSP.2020.2975749
[11] BRISIMI T S, CHEN R D, MELA T, et al. Federated learning of predictive models from federated Electronic Health Records [J]. International journal of medical informatics, 2018, 112: 59 – 67. DOI: 10.1016/j.ijmedinf.2018.01.007
[12] KAISSIS G A, MAKOWSKI M R, RÜCKERT D, et al. Secure, privacy-preserving and federated machine learning in medical imaging [J]. Nature machine intelligence, 2020, 2(6): 305 – 311. DOI: 10.1038/s42256-020-0186-1
[13] DOSILOVIC F K, BRCIC M, HLUPIC N. Explainable artificial intelligence: a survey [C]//Proceedings of 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).

IEEE, 2018. DOI: 10.23919/mipro.2018.8400040
[14] LUNDBERG S, LEE S I. A unified approach to interpreting model predictions [EB/OL]. [2022-02-28]. https://arxiv.org/abs/1705.07874
[15] STOJIĆ A, STANIĆ N, VUKOVIĆ G, et al. Explainable extreme gradient boosting tree-based prediction of toluene, ethylbenzene and xylene wet deposition [J]. Science of the total environment, 2019, 653: 140 – 147. DOI: 10.1016/j.scitotenv.2018.10.368
[16] NAN Y C, LI W, LU F, et al. Developing practical multi-view learning for clinical analytics in P4 medicine [J]. IEEE transactions on emerging topics in computing, 2021: 1. DOI: 10.1109/tetc.2021.3054761
[17] RIEKE N, HANCOX J, LI W Q, et al. The future of digital health with federated learning [J]. NPJ digital medicine, 2020, 3(1): 119. DOI: 10.1038/s41746-020-00323-1
[18] SCHWARZ C G, KREMERS W K, THERNEAU T M, et al. Identification of anonymous MRI research participants with face-recognition software [J]. The New England journal of medicine, 2019, 381(17): 1684 – 1686. DOI: 10.1056/nejmc1908881
[19] SHELLER M J, EDWARDS B, REINA G A, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data [J]. Scientific reports, 2020, 10(1): 12598. DOI: 10.1038/s41598-020-69250-1
[20] SILVA S, GUTMAN B A, ROMERO E, et al. Federated learning in distributed medical databases: meta-analysis of large-scale subcortical brain data [C]// IEEE 16th International Symposium on Biomedical Imaging. IEEE, 2019: 270 – 274. DOI: 10.1109/ISBI.2019.8759317
[21] XU J, GLICKSBERG B S, SU C, et al. Federated learning for healthcare informatics [J]. Journal of healthcare informatics research, 2021, 5(1): 1 – 19. DOI: 10.1007/s41666-020-00082-4
[22] WANG G. Interpret federated learning with shapley values [EB/OL]. [2022-02-28]. https://arxiv.org/abs/1905.04519
[23] RYZIN J V, BREIMAN L, FRIEDMAN J H, et al. Classification and regression trees [J]. Journal of the American statistical association, 1986, 81(393): 253. DOI: 10.2307/2288003
[24] CHEN T Q, GUESTRIN C. XGBoost: A scalable tree boosting system [C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016: 785 – 794. DOI: 10.1145/2939672.2939785
[25] SHAPLEY L S. A value for n-person games [EB/OL]. [2022-02-28]. https://www.rand.org/pubs/papers/P295.html
[26] LUNDBERG S M, ERION G G, LEE S I. Consistent individualized feature attribution for tree ensembles [EB/OL]. [2022-02-28]. https://arxiv. org/abs/1802.03888
[27] WHO. Coronavirus disease 2019 (COVID-19): situation report [R], Geneva, Switzerland: WHO, 2020
[28] NOVAK V, HU K, DESROCHERS L, et al. Cerebral flow velocities during daily activities depend on blood pressure in patients with chronic ischemic infarctions [J]. Stroke, 2010, 41(1): 61 – 66. DOI: 10.1161/STROKEAHA.109.565556
[29] FERRARI D, MOTTA A, STROLLO M, et al. Routine blood tests as a potential diagnostic tool for COVID-19 [J]. Clinical chemistry and laboratory medicine (CCLM), 2020, 58(7): 1095 – 1099. DOI: 10.1515/cclm-2020-0398
[30] AN X S, LI X Y, SHANG F T, et al. Clinical characteristics and blood test results in COVID-19 patients [J]. Annals of clinical and laboratory science, 2020, 50(3): 299 – 307

Biographies

NAN Yucen (yucen.nan@sydney.edu.au) received her PhD and MPhil degrees from University of Sydney, Australia in 2022 and 2017. She is currently a lecturer in the College of Intelligence Science and Technology, National University of Defense Technology, China. Her current research interests are in the area of edge computing and the Internet of Things.

FANG Minghao received his MD degree from Huazhong University of Science and Technology, China in 2006. He is currently an associate professor with Tongji Hospital, Tongji Medical College, Huazhong University of Science and Technology. He has worked in emergency and critical care medicine for 20 years. His research interests include the diagnosis and treatment of critical respiratory and cardiovascular disease.

ZOU Xiaojing received her MD degree from Tongji Medical College, Huazhong University of Science and Technology, China in 2011. She is an associate chief physician in Emergency Department and Intensive Care Unit of Tongji Hospital, Tongji Medical College, Huazhong University of Science and Technology. Her research interests are in sepsis and the application of artificial intelligence in critical diseases.

DOU Yutao received his BE degree in software engineering from University of Canberra, Australia in 2020. He is currently working toward the master of philosophy degree with the University of Sydney, Australia. His research interests mainly include distributed computing, bioinformatics, and artificial intelligence.

Albert Y. ZOMAYA is the chair professor of high performance computing & networking in the School of Computer Science and Director of the Center for Distributed and High Performance Computing at the University of Sydney, Australia. He has published more than 600 scientific papers and is the (co-)author/editor of more than 30 books. As a sought-after speaker, he has delivered more than 190 keynote addresses, invited seminars, and media briefings. His research interests span several areas in parallel and distributed computing and complex systems. He is currently the Editor in Chief of the ACM Computing Surveys and served in the past as Editor in Chief of the *IEEE Transactions on Computers* (2010—2014) and the *IEEE Transactions on Sustainable Computing* (2016—2020).

A Survey of Federated Learning on Non-IID Data

HAN Xuming[1], GAO Minghan[2], WANG Limin[3], HE Zaobo[1], WANG Yanze[1]
(1. Jinan University, Guangzhou 510632, China;
2. Changchun University of Technology, Changchun 130012, China;
3. Guangdong University of Finance & Economics, Guangzhou 510320, China)

DOI: 10.12142/ZTECOM.202203003
https://kns.cnki.net/kcms/detail/34.1294.TN.20220818.0945.002.html, published online August 18, 2022
Manuscript received: 2022-06-10

Abstract: Federated learning (FL) is a machine learning paradigm for data silos and privacy protection, which aims to organize multiple clients for training global machine learning models without exposing data to all parties. However, when dealing with non-independently identically distributed (non-IID) client data, FL cannot obtain more satisfactory results than centrally trained machine learning and even fails to match the accuracy of the local model obtained by client training alone. To analyze and address the above issues, we survey the state-of-the-art methods in the literature related to FL on non-IID data. On this basis, a motivation-based taxonomy, which classifies these methods into two categories, including heterogeneity reducing strategies and adaptability enhancing strategies, is proposed. Moreover, the core ideas and main challenges of these methods are analyzed. Finally, we envision several promising research directions that have not been thoroughly studied, in hope of promoting research in related fields to a certain extent.

Keywords: data heterogeneity; federated learning; non-IID data

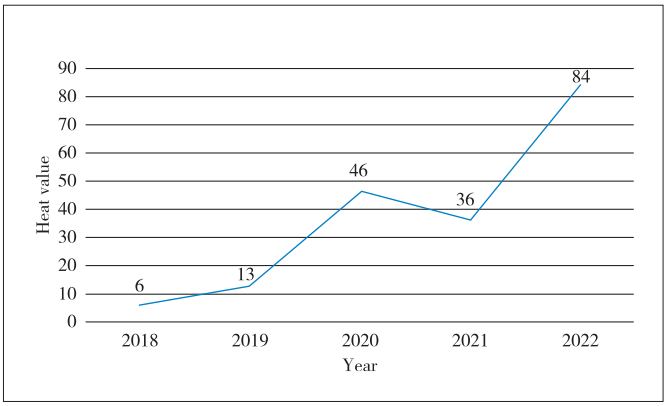
Citation (IEEE Format): X. M. Han, M. H. Gao, L. M. Wang, et al., "A survey of federated learning on non-IID data," *ZTE Communications*, vol. 20, no. 3, pp. 17 - 26, Sept. 2022. doi: 10.12142/ZTECOM.202203003.

1 Introduction

The potent ability of machine learning methods[1] comes from learning the representation and internal laws of a large number of sample data. The extensive use of edge devices makes data collection less expensive. However, the sample data collected by edge devices are often scattered and small-scale in the real world. Hence, it is not easy to train a practical machine learning model solely on edge devices' data. Centralizing edge data for training also becomes more challenging with the gradual improvement of laws and regulations related to data security, and therefore federated learning (FL) comes into being.

In FL, a central server can unite different clients (such as edge devices and an entire organization) to cooperatively train a global model that performs well on most clients while preserving privacy. FL has attracted much attention from researchers in recent years due to its excellent characteristics and is widely used in various fields, including mobile edge devices[2], the Internet of Things (IoT)[3], and medical collaboration[4]. Fig. 1 shows the heat value of FL in Google search in the past five years[5]. The higher the heat value, the more interested people are in federated learning in the current year.

Although FL solves the problem of cooperative learning with small data under privacy constraints, it still faces some challenges. Due to the geographic distribution and usage patterns of edge devices, the data in the edge devices tend to be skewed to varying degrees (including label skew, feature skew, volume skew, time skew, and hybrid skew), also known as data heterogeneity. The data heterogeneity challenges the data's independent and identically distributed (IID) assumption. The existence of data heterogeneity challenges the assumption of IID data, adding complexity to problem modeling,



▲ **Figure 1. Heat value of federated learning in Google search[5]**

theoretical analysis, and empirical evaluation of solutions. As a result, the global model becomes difficult to adapt to individual clients. On the non-independent and identically distributed (non-IID) data, the FL of a single global model, FedAvg[6], proved ineffective by experiments. Hence, a survey of improved methods is necessary for researchers to further analyze and solve FL problems on non-IID data.

Existing surveys on FL on non-IID data[7-8] focus on the action position (such as data, models, and architecture) of processing non-IID methods and cannot show the purpose and motivation of the improved methods. To remedy this regret, this survey investigates many improved methods that mitigate the impact of non-IID data on FL and provides a new perspective on FL methods for analyzing non-IID data. These methods are categorized into heterogeneity reducing strategies and adaptability enhancing strategies from the perspective of core motivations. On this basis, a detailed classification is carried out respectively. The main contributions of this survey are summarized as follows:

1) This survey provides a brief overview of FL concepts, methods, and challenges posed by the non-IID data setting.

2) This survey proposes a unique perspective based on core motivations. According to the core motivations of a method, existing state-of-the-art methods are classified into heterogeneity reducing strategies and adaptability enhancing strategies. On this basis, many FL methods for non-IID data are reviewed, and their basic ideas and main challenges are analyzed.

3) We look forward to future research trajectories in some related fields on non-IID data.

The rest of the article is organized as follows. Section 2 provides an overview of FL and its non-IID data setting. Section 3 presents our induction of a unique taxonomy based on core motivations. Section 4 analyzes the ideas and main challenges of heterogeneity reducing strategies. Section 5 analyzes the ideas and main challenges of adaptability enhancing strategies. In Section 6, we look forward to future research directions in FL on non-IID data. Finally, we summarize the work of this survey.

2 Preliminary Knowledge

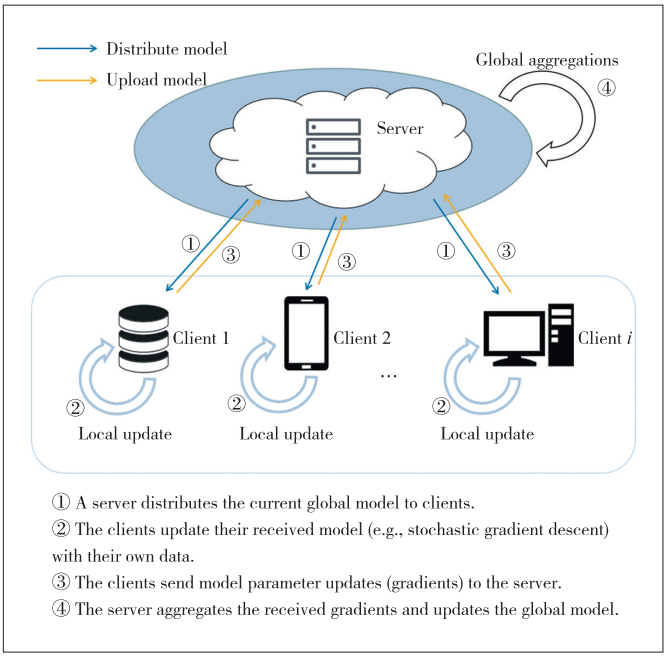
In this section, we provide an overview of FL and non-IID data settings of FL for understanding the problem of FL on non-IID data.

2.1 Federated Learning

FL[6] aims to get a single globally optimal model from data across thousands of clients to minimize the training loss for each client. The data and training processes of all clients must remain local to meet participants' needs for privacy. Fig. 2 shows the architecture of FL. The clients participating in the training are different types of devices with different hardware and software characteristics, and each client maintains a local model. In each training, the server distributes the initial model to the clients in the training. The clients update their model parameters by utilizing their local data, and upload their model parameters to the server for aggregation, thereby completing the updating of the current round of the global model. Finally, the server uses the updated global model as a new initial model to participate in the next training. The global model training objective function can be formulated as:

$$\boldsymbol{\Theta}^{*} = \arg\min_{\Theta} \frac{1}{n} \sum_{i=1}^{n} F_i(\theta_i, x_{ij}, y_{ij}), \tag{1}$$

where $\boldsymbol{\Theta}^{*}$ is the global model parameter, θ_i is the local model parameter of the i-th client, n is the number of all clients, x is the data feature, y is the data label, and F_i is the empirical risk of the i-th client data.



▲ **Figure 2. Architecture of federated learning (FL) in each training**

2.2 Non-IID Data Setting

FL requires that the client data involved in the training satisfy the IID assumptions. However, the collection of data in the real world often depends on the usage of a specific device. There is a certain degree of heterogeneity in the distribution and quantity of data between clients, also known as skew [8-9]. Depending on the skew situation, we categorizes it as follows in this survey.

•Label skew. The label distribution of data between clients is different. Since each client may rate the same feature differently, records with the same characteristics may have different labels. For example, many people love furry pets, but people allergic to animal hair may not think so.

•Feature skew. The features of data between clients do not overlap or partially overlap. Due to differences in viewing

angle and modality, records with the same label may have very different characteristics or even be completely different. For example, when two cameras at different positions capture the same object, the description (front view and left view) of the object's features may be quite different.

•Quantity skew. The quantity of client data varies. Due to differences in computing and storage capabilities of client devices, the numbers of data that devices can use for federated training may be different. For example, there will be hundreds of fold differences in the frequency of temperature measurements and the number of temperature data stored between home and industrial electronic thermometers, which may lead to a preference for clients with larger data sets.

•Time skew. The distribution of client data is time-dependent. The data collected by the device may vary by day, night, or season. For example, the usage and driving characteristics of shared bicycles may be significantly different in the morning and the evening, and the transmission characteristics of COVID-19 may also be significantly different in summer and winter.

•Hybrid skew. Client data have two or more skews of the above.

Client data take on the characteristics of non-IID by the different types of skew mentioned above. Due to the difference in the distribution of clients, the convergence direction of a small number of clients may deviate from most of the other clients when FL is trained on non-IID data. This is known as client drift[10], which is an essential factor that impairs the effect of FL.

3 Federated Learning Strategies on Non-IID Data

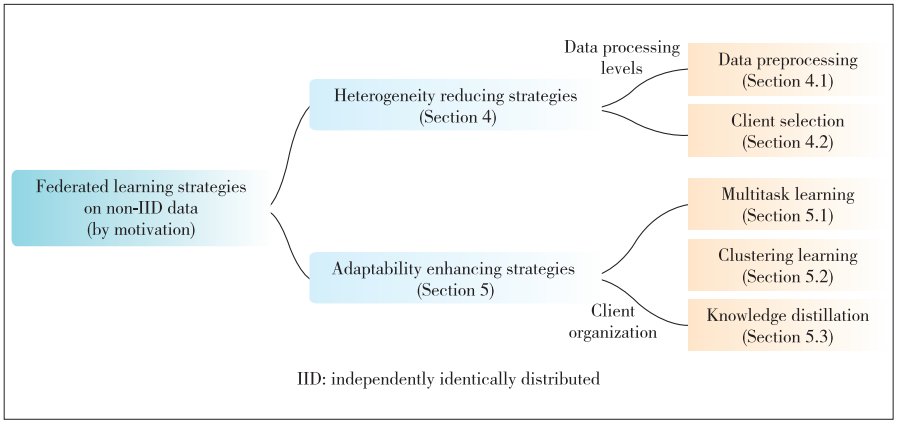
This survey provides a comprehensive examination of the FL on non-IID data in recent years. On this basis, it classifies existing FL strategies on non-IID data from the perspective of motivation, mainly including heterogeneity reducing strategies (Section 4) and adaptability enhancing strategies (Section 5). Then, the specific methods of the two strategies are further subdivided according to the data processing level and client organization. Our proposed taxonomy is shown in Fig. 3, which is the basis for a comprehensive review and systematic analysis of existing methods. Fig. 4 shows the setup of two basic strategies, namely heterogeneity reducing strategies, which perform preprocessing before the client participates in federated training so that the data participating in federated training is close to the IID data, and adaptability enhancing strategies, which use various means to obtain a personalized model to enhance the model's adaptability to non-IID data when the client performs (or completes) federated training. This section will describe both strategies in detail.

3.1 Heterogeneity Reducing Strategies

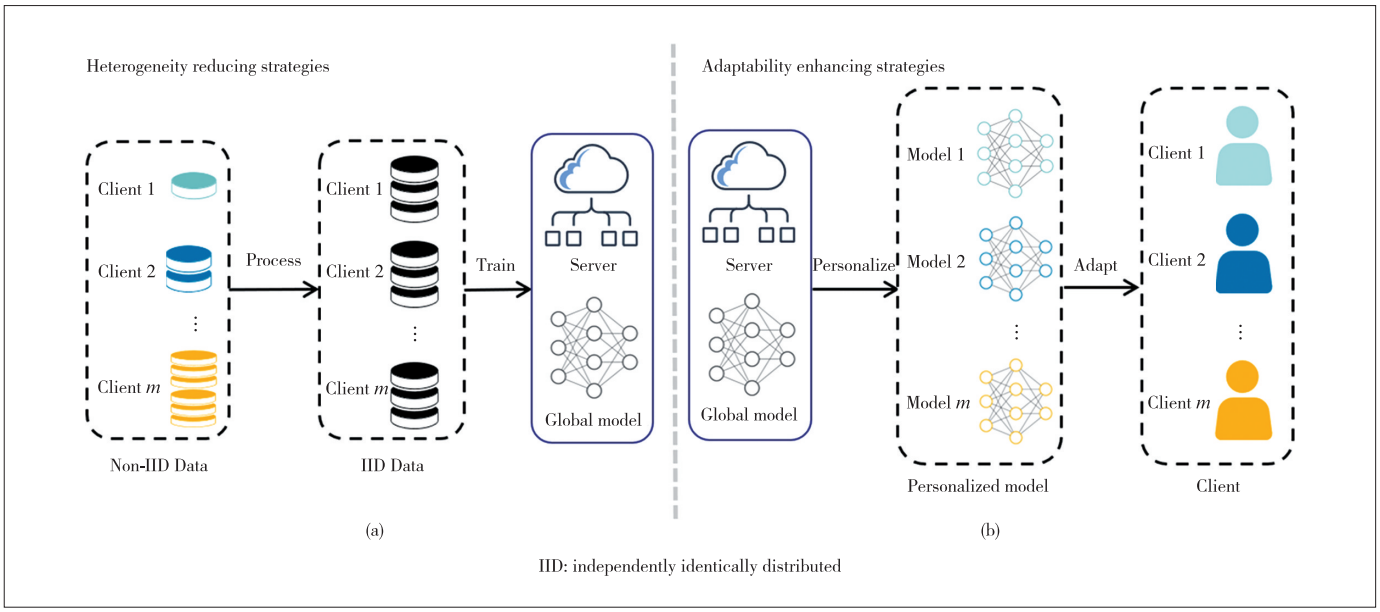
The heterogeneity reducing strategies aim to keep the data involved in the federated training close to the IID data, as shown in Fig. 4(a). Since the data distribution between clients may differ, the server may experience client drift while organizing clients to train a single global model collaboratively. The client drift makes it difficult for the global model on the server to achieve the desired training effect. A natural idea for this challenge is that the server can perform federated training on data approaching the IID by reducing the heterogeneity between client data. Reduction in heterogeneity simplifies the difficulty of server model aggregation and reduces the risk that the model cannot converge. The primary means of heterogeneity reducing strategies summarized in this survey include data preprocessing and client selection. The methods of data preprocessing aim to directly or indirectly convert the non-IID data of the client participating in the training into IID data before the client uses the respective data to participate in the federated training. The methods of client selection aim to select the subset of clients with the slightest degree of data skew to participate in federated training.

3.2 Adaptability Enhancing Strategies

The adaptability enhancing strategies aim to learn personalized models for clients with different distributions, as shown in Fig. 4(b). The heterogeneity reducing strategies can effectively prevent the adverse effects of non-IID data in FL. However, even a high-quality global model may lose some of the client's private information. The loss of client information causes the model to degrade on specific clients. Therefore, some scholars have proposed an FL method with more robust client adaptability. Unlike the heterogeneity reducing strategies summarized in this survey, the server no longer trains a single global model in the adaptability enhancing strategies. Multiple refined models are adapted to clients with different data distri-



▲ **Figure 3. Taxonomy of federated learning (FL) strategies on non-IID data proposed in this survey**



▲ **Figure 4. Heterogeneity reducing strategies and adaptability enhancing strategies: (a) Heterogeneity reducing strategies and (b) adaptability enhancing strategies**

butions by personalizing the global model aggregated by the server. The primary means of adaptability enhancing strategies summarized in this survey include federated multitask learning, federated clustering learning, and federated knowledge distillation. Federated multitask learning aims to find related subtasks in FL and use domain-specific knowledge to train similar models for them. Federated clustering learning aims to cluster clients with similar distributions into a class on client data with inherent partitions and train a cluster model to adapt to its inherent partitions. Federated knowledge distillation aims to transfer knowledge between the server and client models (or only between client models), improving its performance on unknown heterogeneous data.

4 Heterogeneity Reducing Strategies

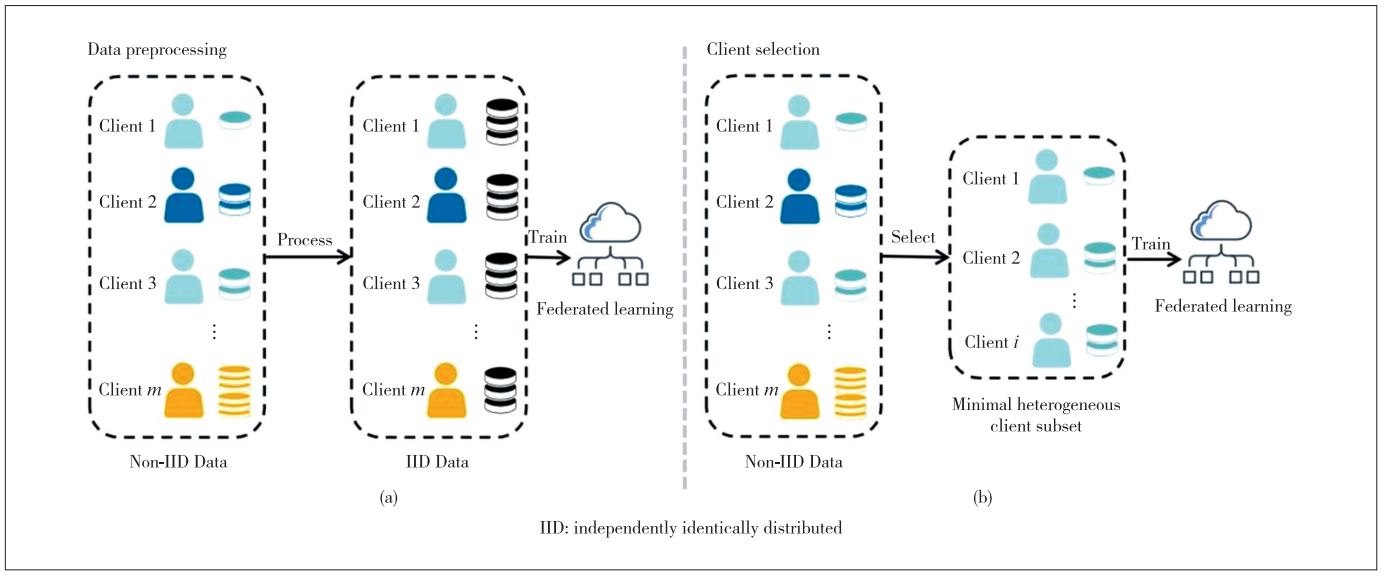
This section will introduce FL methods on non-IID motivated by reducing heterogeneity. The primary setting of these methods is shown in Fig. 5. This survey classifies them into data preprocessing methods and client selection methods according to the different levels of data processed by the server, where the data preprocessing method preprocesses the client's data before joining the training and converts the non-IID data into IID data, and client selection selects the client subset with the most negligible heterogeneity to join the training. Table 1 shows the advantages and disadvantages of these methods.

4.1 Data Preprocessing

Classical data preprocessing methods include oversampling[11] and undersampling[12]. Before performing machine learning training, increasing or decreasing the number of training times for specific types of samples in the dataset can effectively reduce data heterogeneity. This survey classifies the data preprocessing methods in FL into direct preprocessing and indirect preprocessing.

In the direct preprocessing methods, the server directly alters the data involved in training so that training is done on data close to the IID. TUOR et al.[13] proposed federated learning based on data correlation. They set up a small task-specific benchmark data set on the server and trained a benchmark model against it. The benchmark model was used to determine the relevance of local data on the clients and filter out data samples irrelevant to the learning task. Each client used only its selected subset of relevant data during FL. However, since the distribution of participating training clients in the federated environment is unknown, the setting of the benchmark dataset faces incredible difficulties. The difference between the benchmark dataset distribution and the real data set will be an essential factor in determining the model's performance. YOSHIDA et al.[14] proposed a hybrid FL. Clients allowed their data to be uploaded to the server to build an approximate IID dataset. The server then updated the global model with IID data and aggregated it with other locally updated models from clients. Such methods are relatively easy to implement and do not require a preset benchmark dataset. However, building IID datasets directly from the server raises data privacy concerns when a trusted central server is not guaranteed. YOON et al.[15] proposed mean augmented federated learning (MAFL), an average enhanced FL framework, which exchanged model parameters and additional data generated by the mix. Based on MAFL, FedMix is proposed to approximate the loss function of global mixing by Taylor expansion. This approximation involved only the average data from other clients, with some privacy to the server.

In the indirect preprocessing methods, the server designs the



▲Figure 5. Method settings for heterogeneity reducing strategies: (a) data preprocessing and (b) client selection

▼Table 1. Summary of specific methods based on heterogeneity reducing strategies

Methods	Ways	Advantages	Disadvantages
Data preprocessing	Direct	• Easy to implement	• May reveal privacy • Proxy dataset required
	Indirect	• Strong privacy	• Contextual information may be required • More complex to implement
Client selection	Context-based	• Faster model converges	• May reveal privacy
	Deep-learning-based	• No context required • Better effect	• Higher time and space costs

encoding method to obtain the encrypted data distribution indirectly and thus balance the data distribution. DUAN et al.[16] proposed a self-balancing FL framework named Astraea. Before training, the server classified the majority and minority classes based on the Z-score outlier detection algorithm and then performed data preprocessing to adapt to the classes. In training, the mediator of asynchronously receiving and applying client updates was proposed to average local imbalance. The mediator made the distribution of data collection close to unity by rearranging the clients to participate in the training. However, this algorithm requires the server to have more understanding of the context information of the client. WU et al. designed a generative convolutional autoencoder (GCAE) in Ref. [17]. By synthesizing minority class samples through GCAE, a class-balanced dataset was generated to retrain the client's local model. The process alleviated the non-IID of clients' data and achieved better-personalized prediction. Furthermore, because GCAE contains only a small number of model parameters, it can significantly reduce the communication overhead during model transfer.

4.2 Client Selection

Such work designs client-level data distribution balancing methods. Because the server does not need to preprocess the data directly, the methods avoid the privacy problems caused by directly processing the client data. This survey classifies them into context-based methods and deep-learning-based methods according to the difference in server selection of clients.

The context-based methods focus on utilizing available environmental information in FL. ZHAO et al.[18] proposed an enhanced FL method, Newt, which selects participating clients in heterogeneous FL. On the one hand, under the joint consideration of the client dataset and weight update size, the server selected the available clients in a specific FL task by setting selectors to explore the trade-off between accuracy performance and system progress for each round. On the other hand, the frequency of client selection was taken as an additional dimension to optimize the client selection algorithm. This allows the server to maintain fundamental fairness in its biased selection of clients. SHU et al.[19] proposed a computation and communication efficient federated learning via adaptive sampling of data and clients, called FLAS. The server captured data distribution among different clients and set adaptive thresholds during the learning process to improve local computing efficiency and accelerate client convergence. In addition, the server selected clients with the same convergence phase to reduce the communication cost between the client and the server. The context-based methods require the server to have a priori knowledge of the client data distribution, limiting the application of FL in environments with strict information constraints.

The deep-learning-based methods bring practical experience from deep learning to FL and use online learning to perform client selection. ZHANG et al.[20] designed an experience-

driven FL method based on deep reinforcement learning. The server mitigated the negative impact of non-IID data by selecting a subset of participants and adaptively adjusting their batch size. This method can adaptively determine system parameters without knowing any prior information to control local model training and global aggregation and maximize the model accuracy of each round of communication. WANG et al.[21] proposed an experience-driven FL framework, FAVOR. An agent with dual deep Q-learning network (DDQN) training was designed to perform active client selection to obtain the optimal client terminal set. The agent offset bias was introduced by non-IID data and speeded up the FL process. One of the advantages of Q-learning is comparing the expected utility of available actions without prior environmental information. Therefore, this method can train and reuse data more effectively than the context-based methods in federated environments with strict information constraints. However, the deep-learning-based methods tend to have high costs in time and space, which imposes higher requirements on the performance of federated networks.

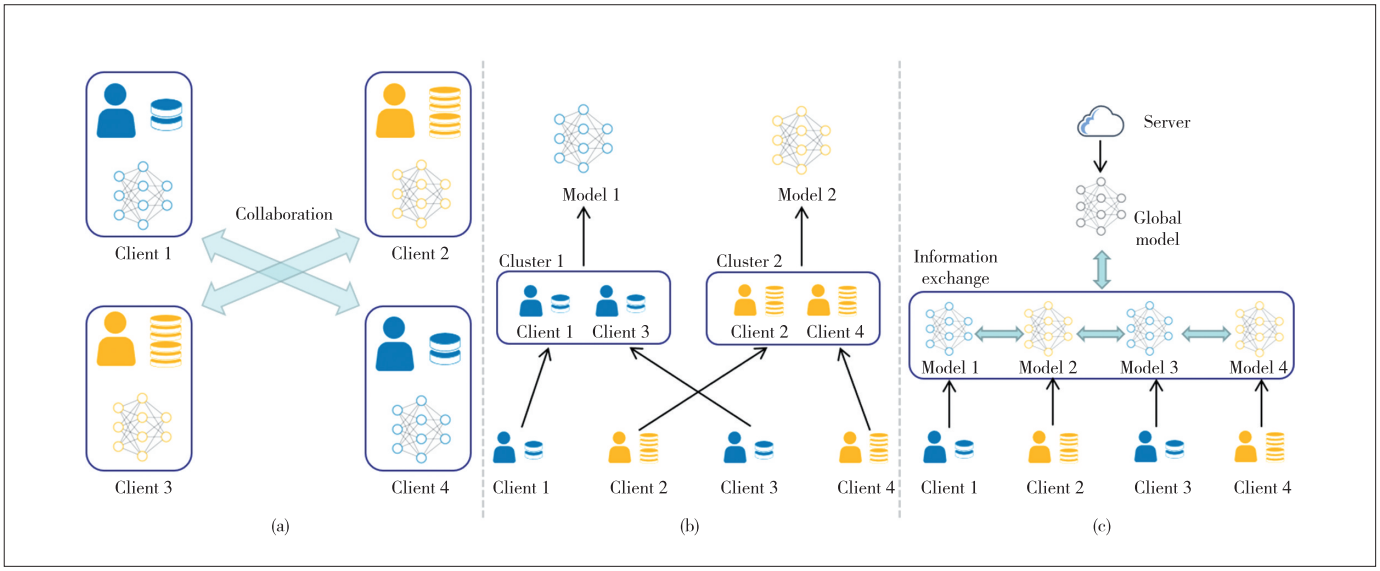
5 Adaptability Enhancing Strategies

This section will investigate FL methods on non-IID motivated by enhancing adaptability. The settings of the different methods are shown in Fig. 6. This survey classifies FL tasks into federated multitasking learning, federated clustering learning, and federated knowledge distillation based on how they are organized between the server and the client. Federated multitask learning finds relevant task clients for knowledge exchange and collaborative training in the training process. Federated clustering learning classifies clients with similar data distribution as clusters and learns the cluster model according to clusters. Federated knowledge distillation exchanges the knowledge in their models between the server and the client (or between the clients), which makes the model output close to each other. Table 2 shows the advantages and disadvantages of these methods.

5.1 Federated Multitask Learning

Multitask learning exploits the similarity between tasks while solving multiple potentially related tasks[22]. These tasks are somewhat related but not identical. By introducing multitask learning, each client learns knowledge from all relevant tasks, which facilitates the training of more adaptive client models. This survey divides federated multitask learning on non-IID into client-based methods, and subtask-division-based methods based on how multitasks are set up.

The client-based methods regard clients with different data distribution as different tasks. The server constructs an association matrix between the clients to organize the relevant client to participate in the collaboration. SMITH et al. [23] introduced multitask learning into FL and proposed a novel systems-aware optimization method, MOCHA. This algorithm extended primal-dual optimization into a federated multitasking setting and defined a data-local subproblem to separate computation across clients. MOCHA learned a personalized model for each client during joint optimization of multiple subproblems. However, this algorithm can only be applied to convex target problems, and all clients must be guaranteed to participate in each round of training. HUANG et al.[24] introduced the attention message passing mechanism into FL and proposed FedAMP. This algorithm implemented an attention mechanism by calculating the similarity between client model weights, iteratively encouraging more cooperation between similar clients. Based on this, a personalized cloud model was maintained for each client using a messaging mechanism. Cli-



▲ **Figure 6. Method settings for adaptability enhancing strategies: (a) federated multitask learning, (b) federated clustering learning, and (c) federated knowledge distillation**

▼ **Table 2. Summary of specific methods based on adaptability enhancing strategies**

Methods	Ways	Advantages	Disadvantages
Federated multitask learning	Client-based	• Easy to implement	• Possibility to isolate heterogeneous clients
	Subtask-division-based	• Part-time joins allowed	• Data quality sensitive
Federated clustering learning	Model-loss-based	• Easy to implement • Predictable effect	• Need to preset the number of clusters • Communication overhead is high
	Client-similarity-based	• No need to preset the number of clusters	• Lack of theoretical analysis
Federated knowledge distillation	One-way distillation	• Strong privacy	• Poor to heterogeneity model • Contextual information may be required
	Mutual distillation	• Robust to heterogeneous models • Suitable for a large number of clients	• Negative transfer possible • Lack of theoretical analysis

ents with similar models could achieve closer cooperation and improved cooperation efficiency through this positive feedback mechanism. However, such an algorithm may actively segregate clients with different distributions when the data are highly non-IID. These segregated clients will be more likely to converge to the local optimum. JAMALI-RAD et al.[25] proposed an FL algorithm with Taskonomy (FLT). Unlike FedAMP, this algorithm initialized an encoder with an on-server standard dataset and sent it to each client. The server received the latent representation obtained by the client compressed by the encoder and generated a task association matrix accordingly. On this basis, the personalization model was trained using the associations between clients. Since this algorithm only needs to pass the encoder to the client at one time, higher operating efficiency is obtained.

The subtask-division-based methods divide the FL task into multiple sub-tasks and perform multitask learning in a federated setting by organizing the subtasks. LI et al.[26] proposed Ditto, a federated multitask learning framework. Ditto added a regularization term to the original objective function of the client model. This algorithm used an objective function with added regularization terms for local training, while the original objective function was used for global training. This algorithm achieved a trade-off between robustness and fairness by adjusting a hyperparameter λ under the condition of personalization. Ditto achieves promising results on both convex and non-convex targets. MARFOQ et al.[27] designed a federated multitask model FedEM based on mixed data distribution. This algorithm assumed that the data distribution for each client was a mixture of M underlying distributions, with different distributions as subtasks. Clients used only the points sampled from the mixture distribution to construct an unbiased estimate of the true risk on each subtask and jointly learned shared component models and personalized hybrid weights through EM-like algorithms. Even if the two clients have completely different data distributions, both can benefit from knowing the same distribution drawn from all other clients' datasets. Notably, this method allows clients to join training at any time.

5.2 Federated Clustering Learning

Clustering is an unsupervised machine learning method that aims to generate multiple clusters of data with similar characteristics[28]. In FL, cluster models can be obtained by clustering clients and intra-cluster aggregation between global and local models. The cluster model has more robust adaptability to the clients within the clusters. This survey classifies the clustering into model-loss-based methods and client-similarity-based methods by their clustering bases.

The model-loss-based methods select a representative model as the cluster center, and clients can join the cluster of the model with the most negligible loss. GHOSH et al.[29] proposed the iterative federated clustering algorithm, IFCA. The server trained K models simultaneously and broadcasted the K models to all clients simultaneously. Each client joined a unique cluster by finding the model with the smallest loss. Cluster-based FL model aggregation was then performed on the server. However, since the server needs to broadcast K cluster models to all clients, its communication overhead is K times that of FedAvg. Based on Ref. [29], LI et al.[30] absorbed the idea of soft clustering and considered that different clusters have gradients and blurred boundaries. The authors divided clients into N-associated clusters and performed model fusion and local updates based on multiple cluster models. This method could utilize the information of boundary clients more effectively and realized information fusion between different clusters to a certain extent. Its communication cost is the same as in Ref. [28]. The loss-based method can ensure a specified number of cluster partitions. However, since several representative models need to be selected as cluster models (cluster centers), the clustering effect may be sensitive to the selected models and the number of them. Furthermore, the clients need to perform a cluster center model for each cluster to find the cluster with the smallest loss. This leads to extra computation for model loss-based methods.

The client-similarity-based methods take the similarity between model parameters or model parameter updates to represent client similarity. SATTLER et al.[31] proposed a recursive cluster FL method. After training the global model, this algorithm accorded with the cosine similarity between the last gradient updates of the client. Hierarchical clustering was used to iteratively bisect the client until the lower bound of the cosine distance within the cluster or the upper bound of the cosine distance between the clusters was satisfied. With the recursive method, the user does not need to pre-set the number of clusters. Even on non-convex optimization problems, solid mathematical guarantees for clustering quality can be pro-

vided. However, the model gradient-based methods have certain limitations. Because gradient descent methods may get stuck in local optima, those models' gradients pointing to the local optima cannot represent the similarity of these clients. Furthermore, in the gradient descent process, the model takes a mini-batch (a small subset of the data set) from the full data set each time to calculate the gradient, and then adjusts the parameters. The gradient directions given by these small mini-batches will vary. FRABONI et al.[32] proposed two aggregation sampling methods based on sample size and similarity. This algorithm pre-sets M different distributions and then puts clients into different distributions based on the number of samples or similarities. Experiments show that it can converge to a smaller value on non-IID data. ZHANG et al.[33] considered a measure of the same similarity between the client's computing power and its network conditions with the server and the skewed data distribution. Therefore, the client similarity was defined as the gradient direction and model update delay while solving the problems of data skew and system heterogeneity.

It is worth noting that clustering-based methods can achieve excellent results when applied to data distributions with transparent partitions. However, in the real world, such scenarios are minimal. More importantly, there is no good theoretical analysis to prove the validity of clustering basis, including methods based on model loss and model gradients.

5.3 Federated Knowledge Distillation

Knowledge distillation hopes to transfer the knowledge learned by machine learning models from specific tasks to related tasks[34]. The fundamental difference from traditional machine learning is that knowledge distillation relaxes the assumptions of IID data and allows for direct knowledge transfer between models. Therefore, in FL, clients can benefit from this process even if they have different data distributions. By reducing the importance of the data in the training process, a more adaptive client model can be obtained using federated knowledge extraction in a non-IID data setting. This survey classifies federated knowledge distillation into one-way distillation methods and mutual distillation methods based on the direction of knowledge transfer.

The one-way distillation methods can quickly transfer the knowledge contained in the dominant teacher model to the student model. In FL, both the server and the client can be set as teacher models. LIN et al. [35] proposed an ensemble distillation FedDF for federated model fusion. This algorithm built P groups of heterogeneous client models (which may vary in structure and numerical precision), evaluated on small batches of unlabeled data pre-stored by the server. The classification ensembles distill their logit output to train the student model on the server. This method improves the efficiency of client model training and has good robustness to data skew. LI et al.[36] proposed a federated learning method via model distillation (FedMD) by combining transfer learning and knowledge distillation. Each client used its own model prediction server to share the dataset to obtain class scores, and the server averaged the class scores as a global consensus. Each client learned this consensus through model distillation to obtain better client models. In this way, other clients' knowledge could be leveraged without the need to share its private data or model architecture explicitly. However, both the FedDF and FedMD have to pre-store a representative dataset on the client, which is a significant challenge. ZHU et al.[37] proposed a data-free knowledge distillation (FEDGEN) based on generative learning. The server used the client label prediction module (instead of the data) to learn a global generator that generated a feature representation matching the client-side labels. Each client model implemented knowledge distillation from the server to the client by sampling the generated feature representation. However, the one-way distillation may be challenging to achieve good results in the face of model heterogeneity.

The mutual distillation methods can be applied to diverse network architectures and are robust to heterogeneous models of different sizes. Better accuracy can also be achieved when training with a large number of clients. BISTRITZ et al. [38] proposed a distributed distillation algorithm that established a new topological relationship between clients, and each client could only connect and communicate with a few nearby devices. In each round of iterations, the clients accepted the soft network decisions of their neighbors in a chain, updated their soft network decisions through the consensus algorithm, and sent them to other neighbors. A more adaptive client model was obtained by limiting the loss of self-model features through knowledge distillation between adjacent clients. LI et al.[39] proposed FedH2L, which took the federated network as a collection of students, and all clients taught each other. To manage the global and client gradient conflict, they designed projected gradients to update the model to maximize intra-domain and cross-domain performance, performing well on non-IID data. Bidirectional distillation avoids the dependence on the powerful teacher model, and the student model can improve the learning efficiency and generalization ability of the network through online mutual learning. However, the methods of mutual distillation still lack theoretical analysis, and sometimes unavoidable negative knowledge transfer occurs. Participants may get caught up in groupthink, where the blind leads the blind.

6 Future Directions

Many methods have been proposed for the FL on non-IID data, but some problems are still not well solved. This section will discuss some of these challenges and examine their future research trajectories.

•Heterogeneity Analysis: A client heterogeneity analysis method is still missing, though the FL on non-IID data has re-

ceived extensive attention and research. Specifically, existing methods for heterogeneity analysis based on model loss[29] or model parameter update (gradient)[31] have limitations such as being sensitive to manual settings and lack of theoretical proofs (details in Section 5.2) and cannot achieve the desired goal well. So how to design a client heterogeneity analysis method with good generalization is still an open problem.

•Hyperparameter: Existing FL methods for non-IID data have achieved good performance. However, the vast number of hyperparameters presented in these methods adversely affects the debugging and use of FL networks. Furthermore, due to the significant differences in the number and usage of hyperparameters for different methods, it is not easy to evaluate the actual effectiveness of those proposed innovative methods fairly.

•Security Assurance: The FL applications tend to have high privacy and high-risk characteristics, such as in the business and medical field, because of FL's better privacy. Some recent studies have shown that the privacy guarantees of FL methods, such as FedAvg, can be easily broken by attackers using methods such as inversion[40] and inference[41]. So it is necessary to conduct more in-depth research on possible attacks and corresponding preventions and design FL methods with more security assurance.

•Interpretability: The interpretability of deep learning has always been the focus and difficulty of research. Since the federated setting has the characteristics of distributed training and data heterogeneity, how interpreting its training and decision-making process will be more complicated. There is little discussion on the interpretability of FL today, and more reliable explanations can enhance users' confidence in FL.

•Dedicated Datasets: In FL, researchers need to design their data partitioning algorithms for CIFAR100, Fashion MINIST, and other existing datasets. Since the algorithm's performance may be diverse under different data distributions, the algorithm's performance cannot be well proved using the self-divided data set. Dedicated homogeneous and heterogeneous datasets and data partition algorithms must be designed to align with real-world environments.

7 Conclusions

This survey provides an overview of FL on non-IID data. First, the background and settings of both FL and non-IID data are introduced. Then, according to the motivation of existing methods, a new taxonomy is proposed. Specifically, the existing methods are classified into two categories: heterogeneity reducing strategies and adaptability enhancing strategies. In addition, the core ideas, key technologies, and main challenges of the methods are emphasized. Finally, the future research trajectories for some existing challenges in this field are conceived. We hope this work will help researchers to further overcome the challenges of FL on non-IID data.

References

[1] LECUN Y, BENGIO Y, HINTON G. Deep learning [J]. Nature, 2015, 521 (7553): 436 – 444. DOI: 10.1038/nature14539
[2] LIM W Y B, LUONG N C, HOANG D T, et al. Federated learning in mobile edge networks: a comprehensive survey [J]. IEEE communications surveys & tutorials, 2020, 22(3): 2031 – 2063. DOI: 10.1109/COMST.2020.2986024
[3] NGUYEN D C, DING M, PATHIRANA P N, et al. Federated learning for Internet of Things: a comprehensive survey [J]. IEEE communications surveys & tutorials, 2021, 23(3): 1622 – 1658. DOI: 10.1109/COMST.2021.3075439
[4] PFITZNER B, STECKHAN N, ARNRICH B. Federated learning in a medical context: a systematic literature review [J]. ACM transactions on Internet technology, 2021, 21(2): 1 – 31. DOI: 10.1145/3412357
[5] Google. Google trends [EB/OL]. [2022-06-01]. https://trends.google.com/trends/explore?q=federated%20learning&geo=US
[6] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C]//Artificial Intelligence and Statistics. PMLR, 2017: 1273 – 1282. DOI: 10.48550/arXiv.1602.05629
[7] TAN A Z, YU H, CUI L, et al. Towards personalized federated learning [J]. IEEE transactions on neural networks and learning systems, 2022. DOI: 10.1109/TNNLS.2022.3160699
[8] ZHU H Y, XU J J, LIU S Q, et al. Federated learning on non-IID data: a survey [J]. Neurocomputing, 2021, 465: 371 – 390. DOI: 10.1016/j.neucom.2021.07.098
[9] HSIEH K, PHANISHAYEE A, MUTLU O, et al. The non-IID data quagmire of decentralized machine learning [C]//International Conference on Machine Learning. PMLR, 2020: 4387 – 4398. DOI: 10.48550/arXiv.1910.00189
[10] KARIMIREDDY S P, KALE S, MOHRI M, et al. Scaffold: stochastic controlled averaging for federated learning [C]//International Conference on Machine Learning. PMLR, 2020: 5132 – 5143. DOI: 10.48550/arXiv.1910.06378
[11] KOVÁCS G. An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets [J]. Applied soft computing, 2019, 83: 105662. DOI: 10.1016/j.asoc.2019.105662
[12] GUZMÁN-PONCE A, SÁNCHEZ J S, VALDOVINOS R M, et al. DBIG-US: a two-stage under-sampling algorithm to face the class imbalance problem [J]. Expert systems with applications, 2021, 168: 114301. DOI: 10.1016/j.eswa.2020.114301
[13] TUOR T, WANG S Q, KO B J, et al. Overcoming noisy and irrelevant data in federated learning [C]//The 25th International Conference on Pattern Recognition (ICPR). IEEE, 2020: 5020 – 5027. DOI: 10.1109/ICPR48806.2021.9412599
[14] YOSHIDA N, NISHIO T, MORIKURA M, et al. Hybrid-FL for wireless networks: cooperative learning mechanism using non-IID data [C]//2020 IEEE International Conference on Communications. IEEE, 2020: 1 – 7. DOI: 10.1109/ICC40277.2020.9149323
[15] YOON T, SHIN S, HWANG S J, et al. FedMix: approximation of mixup under mean augmented federated learning [EB/OL]. [2021-06-01]. https://arxiv.org/abs/2107.00233v1
[16] DUAN M M, LIU D, CHEN X Z, et al. Self-balancing federated learning with global imbalanced data in mobile systems [J]. IEEE transactions on parallel and distributed systems, 2021, 32(1): 59 – 71. DOI: 10.1109/TPDS.2020.3009406
[17] WU Q, CHEN X, ZHOU Z, et al. FedHome: cloud-edge based personalized federated learning for in-home health monitoring [J]. IEEE transactions on mobile computing, 2022, 21(8): 2818 – 2832. DOI: 10.1109/TMC.2020.3045266
[18] ZHAO J X, CHANG X Y, FENG Y H, et al. Participant selection for federated learning with heterogeneous data in intelligent transport system [J]. IEEE transactions on intelligent transportation systems, 2022, 99: 1 – 10. DOI: 10.1109/TITS.2022.3149753
[19] SHU J G, ZHANG W Z, ZHOU Y, et al. FLAS: computation and communication efficient federated learning via adaptive sampling [J]. IEEE transactions on network science and engineering, 2022, 9(4): 2003 – 2014. DOI: 10.1109/TNSE.2021.3056655
[20] ZHANG J, GUO S, QU Z H, et al. Adaptive federated learning on non-IID data with resource constraint [J]. IEEE transactions on computers, 2022, 71(7): 1655 – 1667. DOI: 10.1109/TC.2021.3099723
[21] WANG H, KAPLAN Z, NIU D, et al. Optimizing federated learning on non-

IID data with reinforcement learning [C]//IEEE Conference on Computer Communications. IEEE, 2020: 1698 – 1707. DOI: 10.1109/INFOCOM41043.2020.9155494
[22] STANDLEY T, ZAMIR A, CHEN D, et al. Which tasks should be learned together in multitask learning? [C]//International Conference on Machine Learning. PMLR, 2020: 9120 – 9132. DOI: 10.48550/arXiv.1905.07553
[23] SMITH V, CHIANG C K, SANJABI M, et al. Federated multitask learning [J]. Advances in neural information processing systems, 2017, 30. DOI:10.48550/arXiv.1705.10467
[24] HUANG Y T, CHU L Y, ZHOU Z R, et al. Personalized cross-silo federated learning on non-IID data [C]//Proceedings of the AAAI Conference on Artificial Intelligence. AAAI, 2021: 7865 – 7873. DOI: 10.48550/arXiv.2007.03797
[25] JAMALI-RAD H, ABDIZADEH M, SINGH A. Federated learning with taskonomy for non-IID data [J]. IEEE transactions on neural networks and learning systems, 2022. DOI: 10.1109/TNNLS.2022.3152581
[26] LI T, HU S, BEIRAMI A, et al. Ditto: fair and robust federated learning through personalization [C]//International Conference on Machine Learning. PMLR, 2021: 6357 – 6368. DOI: 10.48550/arXiv.2012.04221
[27] MARFOQ O, NEGLIA G, BELLET A, et al. Federated multi-task learning under a mixture of distributions [EB/OL]. [2022-06-01]. https://arxiv. org/abs/2108.10252
[28] LI Y, HU P, LIU Z, et al. Contrastive clustering [C]//2021 AAAI Conference on Artificial Intelligence. AAAI, 2021. DOI: 10.48550/arXiv.2009.09687
[29] GHOSH A, CHUNG J, YIN D, et al. An efficient framework for clustered federated learning [J]. Advances in neural information processing systems, 2020, 33: 19586 – 19597. DOI: 10.48550/arXiv.2006.04088
[30] LI C X, LI G, VARSHNEY P K. Federated learning with soft clustering [J]. IEEE Internet of Things journal, 2022, 9(10): 7773 – 7782. DOI: 10.1109/JIOT.2021.3113927
[31] SATTLER F, MÜLLER K R, SAMEK W. Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints [J]. IEEE transactions on neural networks and learning systems, 2021, 32(8): 3710 – 3722. DOI: 10.1109/TNNLS.2020.3015958
[32] FRABONI Y, VIDAL R, KAMENI L, et al. Clustered sampling: low-variance and improved representativity for clients selection in federated learning [C]//International Conference on Machine Learning. PMLR, 2021: 3407 – 3416. DOI: 10.48550/arXiv.2105.05883
[33] ZHANG Y, DUAN M, LIU D, et al. CSAFL: a clustered semi-asynchronous federated learning framework [C]//Proceedings of 2021 International Joint Conference on Neural Networks (IJCNN). IEEE, 2021: 1 – 10. DOI: 10.1109/IJCNN52387.2021.9533794
[34] PARK D Y, CHA M H, KIM D, et al. Learning student-friendly teacher networks for knowledge distillation [J]. Advances in neural information processing systems, 2021, 34. DOI: 10.48550/arXiv.2102.07650
[35] LIN T, KONG L, STICH S U, et al. Ensemble distillation for robust model fusion in federated learning [J]. advances in neural information processing systems, 2020, 33: 2351 – 2363. DOI: 10.48550/arXiv.2006.07242
[36] LI D L, WANG J P. FedMD: heterogenous federated learning via model distillation [EB/OL]. (2021-01-27) [2022-06-01]. https://arxiv.org/abs/2101.11296
[37] ZHU Z D, HONG J Y, ZHOU J Y. Data-free knowledge distillation for heterogeneous federated learning [J]. Proceedings of machine learning research, 2021, 139: 12878 – 12889. DOI:10.48550/arXiv.2105.10056
[38] BISTRITZ I, MANN A, BAMBOS N. Distributed distillation for on-device learning [J]. Advances in neural information processing systems, 2020, 33: 22593 – 22604
[39] LI Y Y, ZHOU W, WANG H M, et al. FedH2L: federated learning with model and statistical heterogeneity [EB/OL]. (2021-01-27) [2022-06-01]. https://ui.adsabs.harvard.edu/abs/2021arXiv210111296L/abstract
[40] JIN X, CHEN P Y, HSU C Y, et al. CAFE: catastrophic data leakage in vertical federated learning [EB/OL]. [2022-06-01]. https://arxiv. org/abs/2110.15122
[41] ZHANG J W, ZHANG J L, CHEN J J, et al. GAN enhanced membership inference: a passive local attack in federated learning [C]//2020 IEEE International Conference on Communications. IEEE, 2020, 1 – 6. DOI: 10.1109/ICC40277.2020.9148790

Biographies

HAN Xuming received his PhD degree from Jilin University, China. Now he is a professor and PhD supervisor at Jinan University, China. He is in charge of about 10 important scientific research projects and 80 journal papers and conference papers, and has publish four academic monographs. His research interests include artificial intelligence, federated Learning, and machine learning.

GAO Minghan is currently a graduate student in Changchun University of Technology, China. His research interests include federated learning, multitasking optimization, and clustering.

WANG Limin (20211016@gdufe.edu.cn) received her master's and PhD degrees in computer science and technology from Jilin University, China in 2004 and 2007, respectively. Now she is a professor with the Guangdong University of Finance & Economics, China. Her current research interests include big data analysis, evolutionary algorithm, and intelligent decision optimization. She is a member of China Computer Federation. She has published more than 90 research papers in international and domestic journals or international conferences.

HE Zaobo received his PhD degree from Georgia State University, USA, MS degree from Shaanxi Normal University, China, and BS degree from Yan'an University, China, all in the Department of Computer Science. Dr. HE is currently a professor in the Department of Computer Science at Jinan University, China. His research areas focus on data privacy and Internet of Things.

WANG Yanze is currently a graduate student in Jinan University, China. His research interests include federated learning, pattern recognition, and computer vision.

Federated Learning Based on Extremely Sparse Series Clinic Monitoring Data

LU Feng[1], GU Lin[1], TIAN Xuehua[1], SONG Cheng[1], ZHOU Lun[2]

(1. School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China;
2. Tongji Hospital, Tongji Medical College, Huazhong University of Science and Technology, Wuhan 430074, China)

DOI: 10.12142/ZTECOM.202203004

https://kns.cnki.net/kcms/detail/34.1294.TN.20220824.1508.002.html, published online August 24, 2022

Manuscript received: 2022-08-18

Abstract: Decentralized machine learning frameworks, e.g., federated learning, are emerging to facilitate learning with medical data under privacy protection. It is widely agreed that the establishment of an accurate and robust medical learning model requires a large number of continuous synchronous monitoring data of patients from various types of monitoring facilities. However, the clinic monitoring data are usually sparse and imbalanced with errors and time irregularity, leading to inaccurate risk prediction results. To address this issue, this paper designs a medical data resampling and balancing scheme for federated learning to eliminate model biases caused by sample imbalance and provide accurate disease risk prediction on multi-center medical data. Experimental results on a real-world clinical database MIMIC-IV demonstrate that the proposed method can improve AUC (the area under the receiver operating characteristic) from 50.1% to 62.8%, with a significant performance improvement of accuracy from 76.8% to 82.2%, compared to a vanilla federated learning artificial neural network (ANN). Moreover, we increase the model's tolerance for missing data from 20% to 50% compared with a stand-alone baseline model.

Keywords: federate learning; time-series electronic health records (EHRs); feature engineering; imbalance sample

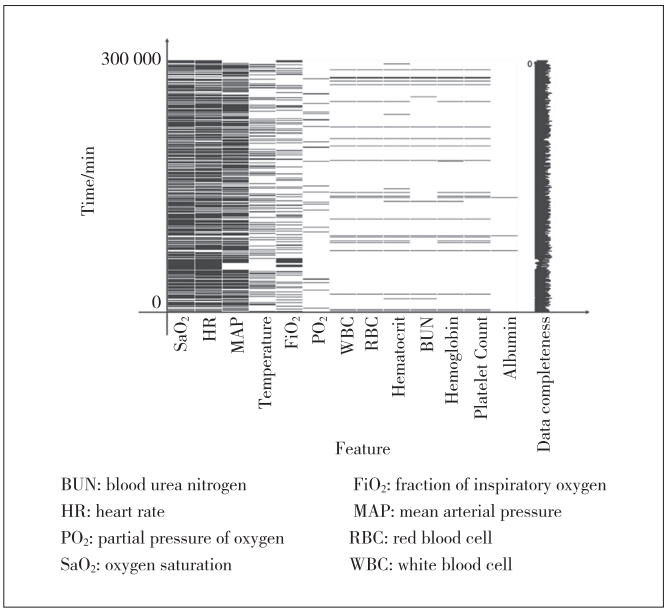
Citation (IEEE Format): F. Lu, L. Gu, X. H. Tian, et al., "Federated learning based on extremely sparse series clinic monitoring data," *ZTE Communications*, vol. 20, no. 3, pp. 27 - 34, Sept. 2022. doi: 10.12142/ZTECOM.202203004.

1 Introduction

With the increasing availability of electronic health records, artificial intelligence such as neural networks has been widely applied and explored to provide medical risk prediction[1-5]. Neural networks can predict the morbidity risk of patients in advance and find abnormalities in time. Application systems with prediction and early warning functions can be developed based on the neural networks. Predictive and early warning systems have been shown to improve patient outcomes by alerting surgeons to action in advance[6-9]. To train a general and accurate neural network, more medical data from multiple hospitals or medical institutes are desired. However, due to mandatory privacy practices and ethical constraints, hospitals or medical institutes cannot freely share patients' electronic medical records with each other. To this end, decentralized machine learning frameworks, e.g., federated learning[10], are proposed and designed to enable distributed learning with medical data privacy protection.

This work is supported by Hubei Provincial Development and Reform Commission Program "Hubei Big Data Analysis Platform and Intelligent Service Project for Medical and Health".

To establish an accurate and robust medical prediction model, many indicators obtained from different monitoring facilities with temporal integrity must be collected as training data. However, the electronic health record (EHR) system continuously enrolls sparse and error data in the clinic[11-13]. For example, there is a relatively large amount of missing data (Fig. 1) when we extract vital signs, laboratory measurements and other assessment data and combine them statistically according to the time in the Medical Information Mart for Intensive Care IV (MIMIC-IV)[14] dataset. This is partially due to the missing of data acquisition in the current equipment and the hand-filled omission of the nurse. Therefore, it is almost impossible to align data with the timeline to ensure the uniform frequency of patients' different indicators[15-16] (Fig. 1). The clinic monitoring data are incredibly sparse and imbalanced with error and time irregularity, so they cannot be used directly to train risk or disease prediction models[17]. Moreover, since the medical risk, such as septic shock, is critical, training local models using mass imbalance samples with clinic monitoring indicators is most challenging. Therefore, it is essential to propose a risk prediction model with a decentralized machine learning framework that can use these valuable but seriously sparse clinical data.



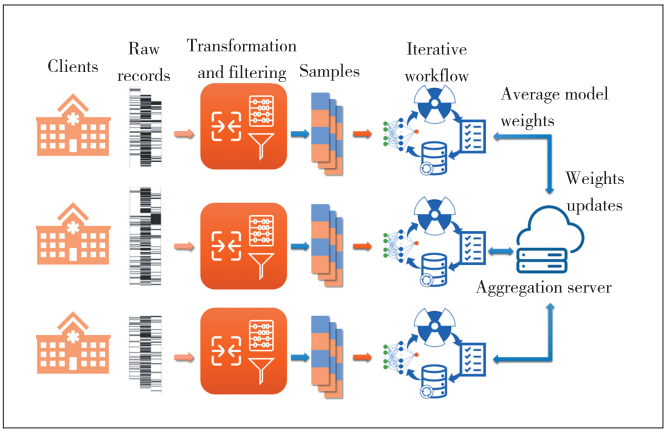
▲**Figure 1. Sparse data with time irregularity in the Medical Information Mart for Intensive Care IV (MIMIC-IV) dataset**

Unfortunately, most existing federated learning methods ignore the data sparsity and errors[18], hence the accuracy and reliability of the prediction models cannot be guaranteed. A multi-center dataset was used in the multi-center federated learning mortality prediction study, but it had different missing values in 11 indicators[19]. Although there are already machine learning frameworks dedicated to the problem of sparse data, this scheme is suitable for more traditional SVMs rather than neural networks with large-scale complex parameters. Solutions in a distributed environment lack validation[20]. Some researchers focus on compressing sparse data to improve the speed of the training phase[21]. For example, Google engineers can use federated learning to predict which emoji a user will likely choose based on sparse data and poorly balanced classes[22]. They tend to include as many participants as possible, but such a global model may not enable the prediction with good performance and reliable robustness[23]. That is to say, low-quality and untrusted participants need to be excluded from federated learning to improve the reliability of the prediction model.

In this paper, we propose a data transformation framework to transform sparse, error-prone and temporally irregular raw data from clients into more accurate patient records for federated learning. We first explore and analyze the MIMIC-IV dataset and obtain two findings about sample imbalance of medical data. One is the difference between the number of negative and positive samples. The other is the missing ratio of features, which results in the model tending to highlight them excessively. Based on these two findings we further improve the quality and reliability of the client and the final model through iteratively training the local models and comprehensively resampling and balancing feature missing rates. Then, we build a horizontal federated learning model of an artificial neural network (ANN) and apply the iterative feature balancing method based on SHapley Additive exPlanations (SHAP) to reduce model bias caused by different proportions of missing features. Finally, the performance of the transforming framework and final model is evaluated based on a real-world clinical database, MIMIC-IV.

2 Methodology

Fig. 2 briefly introduces our EHR transformation and iterative learning workflow based on federated learning architecture. Each client consists of transformation and iteration learning, interpretation and resampling. We transform raw clinic records into sample features in the first module, and then input them to a machine learning model such as an ANN and explain the model with SHAP[24].



▲**Figure 2. Overview of the proposed medical data resampling and balancing scheme**

Records with similar medical semantics on the clients are first merged, which are categorized into two types: text and numerical data. To better abstract the features of text indicators, we adopt one-hot encoding to transform the text into a Boolean value. Meanwhile, we remove some abnormal numerical data and encode some numerical data also into a Boolean value.

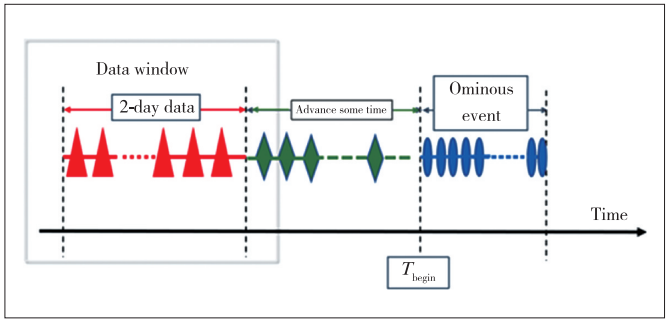
At this time, the encoded indicators are still sparse and have variable data acquisition frequency. It is again a challenge for our clients to train local models. We further transform the indicators using the statistic method, that is, calculating the statistic value of each indicator. The statistic transformation can help to solve the sparse and frequency alignment difficulties.

Finally, after all the patient records are transformed, we will get samples with statistical features on clients to start the iteration of training their local models and explaining the model with SHAP. Combined with the physician's experience and SHAP's results, we can further optimize the models through iteratively resampling the data.

2.1 Transformation and Data Filtering

As the analysis above, raw medical monitoring records in certain a time slot are sparse with a high missing rate. This tends to result in lengthy training procedures and unsatisfactory prediction performance. The EHR system records multiple indicators representing similar medical contexts by exploiting the raw time-series data. For example, the difference between left-hand systolic blood pressure and right-hand systolic blood pressure is less than 10 mmHg for most patients. This difference can always be ignored in the clinical judgment for shock disease. Therefore, in septic shock prediction, both left-hand and right-hand blood systolic pressures can represent patients' blood systolic pressure. Motivated by such facts, we first merge raw indicators with related or similar medical concepts into one. In this way, the missing rate of each indicator can be significantly reduced and the quality of data increases.

Here, we design a time window to collect a sequence of item data within a fixed period, as shown in Fig. 3. Because medical data are highly sparse, the acquisition frequencies of different items are uneven and there are NULL or duplicated values. Thus, it is challenging to design a model to predict a disease based on data with a high missing rate.



▲ **Figure 3. Data window**

We calculate the timely-sequential data series of each item in the window from maximum, minimum, mean, and other statistical views. By recording the eight statistical views for each medical indicator, we get one training sample for this time window of a patient, and the sample's features are the derived views. In this way, the features' NULL value is reduced but not cleaned up. We calculate the eigenvalue miss rate of each sample. If the NULL values in a sample exceeds a certain threshold, e.g., 50%, we choose to eliminate the whole sample. Otherwise, we keep the sample as a record of the training dataset.

After the data transformation on distributed clients, these samples are used to train local prediction machine learning models for clients of a federated learning system. Here, we take an ANN as an example. The network consists of an input layer of 209 neurons and one hidden layer to fit the data on each client. Each iteration generates a set of SHAP values, an updated model, and a report of feature missing rates. The updates of the local model weights are uploaded to the server of our federated learning system and aggregated by averaging weighted by the dataset size of each client. Then every client gets the global model's weights as their new weights. To validate and improve the correctness of our final model, we take the medical basis, doctor experience and adjustment of the positive and negative sample distribution into consideration.

2.2 Iterative Learning and Imbalanced Data Resampling

First, for a client to train the local model, we have dataset *samples*, including the negative sample set N and positive sample set P. We use f to denote the element's initial feature set of *samples*, including medical indicators' statistics, such as feature $HR_{\max}$ (the maximum of the indicator item Heart Rate). We also use key-value pairs (f, v) to represent features f and the eigenvalue of f. As shown in Algorithm 1, we first input (f, v) into the neural network to make the local train.

After inputting (f, v) into the neural network, we get a model with weights fitted. We can use an AUC evaluation index to measure the advantages and disadvantages of the model. We also take SHAP (Kernel Model), a game-theoretic approach, to explain the output of the model and obtain the classical Shapley values f_{SHAP} with the highest ranking from SHAP (Kernel Model). SHAP values for each feature are first evaluated on clients. The summary of SHAP values is expected to be similar to the medical basics and physician's experience. For each feature f_x in f_{SHAP}, we use $m(\ )$, the miss rate calculation method deployed as a part of each client, to calculate the miss rate $m\left(P_{f_x}\right)$ and $m\left(N_{f_x}\right)$ based on f_x. When the difference between the positive and negative samples is larger than ε, that is, $\left|m\left(P_{f_x}\right)-m\left(N_{f_x}\right)\right|>\varepsilon$, the sample allocation is imbalanced. Then, we need to resample the imbalanced data based on the feature f_x.

If the missing rate of positive or negative samples is 100%, that is, $m\left(P_{f_x}\right)==100\%$ or $m\left(P_{f_x}\right)==100\%$, the values of the positive or negative samples of the feature f_x are completely missing. Then, we use $delete\left(f_x\right)$ to delete the feature f_x because it has no positive contribution to the prediction model. Otherwise, we use $Cmp\left(m\left(N_{f_x}\right), m\left(P_{f_x}\right)\right)$ to get the samples set $smp_{\max}$ with the largest miss rate and the samples set $smp_{\min}$ with the smallest miss rate based on the feature f_x, i. e., $\left(smp_{\max}, smp_{\min}\right)=Cmp\left(m\left(N_{f_x}\right), m\left(P_{f_x}\right)\right)$. After calculating the miss rate $m(s)$ of each single sample $s\left(s \in smp_{\max}\right)$ about all feature values, we use $sort\left(smp_{\max}\right)$ to sort $smp_{\max}$ according to $m(s)$. Next, we delete s if its miss rate $m(s)$ is very high until $\left|m\left(smp_{\max}\right)-m\left(smp_{\max}\right)\right| \leqslant \varepsilon$. Now, the miss rates of the negative and positive samples based on the feature f_x are balanced.

Then we get a new sample set $samples' = smp_{\max} \cup smp_{\max}$ with fewer features in set f. Based on the clinical experience,

we obtained a new feature set f'. If $samples' \neq samples$ or $f' \neq f$, we must rebuild the model and balance the sample repeatedly. Otherwise, the model *NeuralNetworkModel*, the feature set f' and AUC are returned and ready to be uploaded to the server.

Algorithm 1. Imbalanced Data Resampling Algorithm

Require: *samples* is the dataset which includes negative samples as N and positive samples as P, as $samples = N \bigcup P$; f is an element of *samples* and a set of the initial features for training model.
Ensure: $AUC, f', NeuralNetworkModel$;
1: initialization: $samples' = [\ ], f' = [\ ]$
2: **while** $samples' \neq samples \| f' \neq f$ **do**
3: $(AUC, NeuralNetworkModel) = FitModel(f, v)$
4: $f_{SHAP} = SHAP(NeuralNetworkModel)$
5: **for each** $f_x \in f_{SHAP}$ **do**
6: **while** $\left| m(P_{f_x}) - m(N_{f_x}) \right| > \varepsilon$ **do**
7: **if** $m(P_{f_x}) == 100\% \| m(N_{f_x}) == 100\%$ **then**
8: delete(f_x)
9: **else**
10: $(smp_{\max}, smp_{\min}) = Cmp(m(P_{f_x}), m(N_{f_x}))$
11: sort$(smp_{\max})$ according to $m(s|s \in smp_{\max})$
12: delete (s) until $\left| m(smp_{\max}) - m(smp_{\min}) \right| \leqslant \varepsilon$
13: **end if**
14: **end while**
15: **end for**
16: $sample = sample', f = f'$
17: $sample' = smp_{\max} \bigcup smp_{\max}$
18: $f' = doctorSelect(f)$
19: **end while**
20: Return: $AUC, f', NeuralNetworkModel$

3 Experiment and Evaluation

Our experiment is based on the closely related clinical indicators and a public real dataset in the intensive care unit, the MIMIC-IV clinical database. We compared the prediction by the final neural network of our federated learning system with other traditional standalone models for sepsis analysis after integrating data with the proposed method. The performance was comprehensively evaluated through multiple indicators. We explained our final model with the help of SHAP, adjusted the balance of the sample features, combined with the physician's experience, selected the eigenvalue, and tried to set it in line with the physician's clinical cognition. All our experiments were carried out on the same equipment. The computer was equipped with Intel Core i5 8400 CPU, 24 GB memory and GPU acceleration disabled. The training of our selected benchmark algorithm and model is implemented in Python 3.9 for Windows.

3.1 Task Background

Sepsis is defined as life-threatening organ dysfunction caused by a dysregulated host response to infection. Septic shock is a subset of sepsis with circulatory and cellular/metabolic dysfunction associated with a higher risk of mortality[25-26]. Sepsis and septic shock are major healthcare problems, affecting millions of people around the world each year and killing as many as one in four (and often more) people worldwide. It sometimes takes only about 24 hours to develop from sepsis to septic shock[27]. Some studies show that the delayed diagnosis and treatment of severe sepsis and septic shock within the first six hours of entering the ICU are closely related to the increased mortality and increased utilization of hospital resources[28-29]. It is a complex problem for clinicians to improve the ability of early clinical recognition, accurately evaluate the condition, and implement reasonable treatment strategies as soon as possible to improve the treatment effect and reduce mortality.

3.2 Dataset and Preprocessing

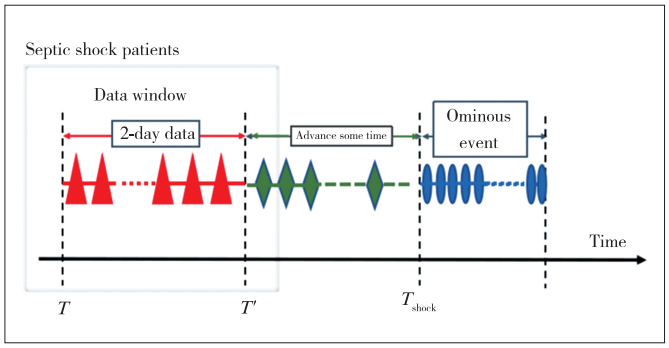
ICU patients are most prone to septic shock. Many patients are difficult to obtain long-term close monitoring in the early stage of shock. And most nurses record data manually. Therefore, the data are too sparse to predict the risk with the existing time series method. To validate our system, we carried out septic shock prediction experiments on MIMIC-IV that includes the data recorded by two different systems in the ICU of Beth Israel Deaconess Medical Center from 2012 to 2019.

More than 30 clinical detection indicators, such as demographics, vital signs, and laboratory values of 121 214 patients were collected, including 6 036 patients diagnosed with septic shock (positive samples) and 115 178 patients who were not diagnosed with any shock (negative samples). These required data include more than 30 monitor features generated from the patients in ICU, which were collected at different times. The time series data is required for subsequent data preprocessing.

All patient characteristics were aligned to the data at the same time, which simplifies the development and testing of the model. The data of each patient were saved to a CSV file. It contains the data of more than 30 index variables such as demographics, vital signs and laboratory results of the patients in ICU. We divided the data sample into four datasets, three of which were for clients in our federated learning system and one was for evaluating our final model.

We sorted the data according to the time of data collection, to facilitate the selection of the later data window and the marking of septic shock events.

As shown in Fig. 4, when a patient's mean arterial pressure (MAP) stays lower than 65 mmhg for at least five minutes, the starting time point of the five minutes is marked as the event of septic shock, which is recorded as T_{shock}. The first 30 minutes of a septic shock event are marked as t' and the window start time before t' is recorded as T. The size of the time window is $t' - t =$ 24 h. The data in the time window is statistically used as mean



▲Figure 4. Definition of positive sample

values, max values and other forms, that is, statistical transformation. A piece of data generated after conversion is a sample data.

For the control experiment, we selected the patient records without any shock and the time series data were generated in the way we selected septic shock records. For these time series data, we selected the data window with the least two days to generate the data, and the sample generated by calculating the statistics is taken as the negative sample. We chose the two days with the least data because the patients of lower risk usually had fewer records in a hospital.

3.3 Performance Metrics

A typical objective function in multi-view classification is usually the internal measurement of a classification algorithm. However, the effectiveness and efficiency of this kind of objective function are poor. Instead, we employed multiple external validation metrics to evaluate whether the classification matches the specified label. Selected external indicators used in the following experiments were AUC (the area under the receiver operating characteristic), ACC (accuracy), Pre. (precision), Sens. (sensitivity), Spe. (specificity) and F1 score. AUC and ACC indicate the correctness of the classification; the higher the correctness is, the better. Usually they are the primary indicators. Pre. focuses only on the proportion of true positives among all results classified as positive. Sens. reflects the proportion of positive results detected in all true positive samples. Spe. represents the proportion of all true negative samples classified as negative samples. F1 comprehensively evaluates the recognition rate of positive samples and the overall accuracy.

3.4 Result Analysis

We comprehensively compared various performance indicators to determine the effectiveness and accuracy of the prediction. Additionally, we compared our experimental results to the baseline method and ablation studies.

1) Comparison to the baseline method

When the frequency of data collection is high and there is no missing data, the time series model can have higher accuracy, but for sparse data, its accuracy (ACC) decreases. Our design is characterized by data sparseness. Tables 1 and 2 show the comparison of the results of the time series model and our model on the dataset. Compared to the time series model, our final model has a 11% improvement in accuracy. The comparison of AUC, ACC, Rec., Pre., Sens., Spe. and F1 is following.

As can be seen from Table 1, the 5-minute, 10-minute and 15-minute values of AUC in advanced prediction are about 0.51, indicating that the authenticity of prediction is very low and there is no reference value. Most of the ICU clinical data are collected with low frequency and relatively sparse data, so it is not feasible to use the time series model for prediction.

▼Table 1. LSTM prediction results

Time/min	AUC	ACC	Pre.	Sens.	Spe.	F1
15	0.502	0.990	0.044	0.013	0.998	0.020
10	0.512	0.991	0.085	0.026	0.998	0.040
5	0.506	0.993	0.229	0.013	0.999	0.025

ACC: accuracy
AUC: the area under the receiver operating characteristic
LSTM: Long Short-Term Memory
Pre.: precision
Sens.: sensitivity
Spe.: specificity

As shown in Table 2, we use the proposed method to perform transformation and statistical view of data and fit the data with our neural network on each client. The final model has an AUC of 62.8%, while the recall is 26.7% and precision is 87.8%. The AUC is 11% higher than that of Long Short-Term Memory (LSTM) which is not optimized for sparse data. It shows that our transformation and resampling for feature balance can help to improve the model predicting performance.

2) Ablation studies

Unlike other systems, we take SHAP into consideration to pick out the corresponding features and rebalance the proportions of their samples. Table 3 shows the comparison between the initial results and the adjusted results.

▼Table 2. Prediction results of the proposed model

AUC	ACC	Rec.	Pre.	Sens.	Spe.	F1
0.628	0.822	0.267	0.878	0.989	0.267	0.409

ACC: accuracy
AUC: the area under the receiver operating characteristic
Pre.: precision
Rec.: recall
Sens.: sensitivity
Spe.: specificity

▼Table 3. Comparison of ablation experiments

	AUC	ACC	Rec.	Pre.	Sens.	Spe.	F1
Before Rebalancing	0.501	0.768	0.003	0.333	0.998	0.003	0.005
After Rebalancing	0.628	0.822	0.267	0.878	0.989	0.267	0.409

ACC: accuracy
AUC: the area under the receiver operating characteristic
Pre.: precision
Rec.: recall
Sens.: sensitivity
Spe.: specificity

We evaluated the model with SHAP to check the factors behind these numbers. Aiming at the top 20 features of SHAP values that had a significant impact on the model, we checked the balance degree for these features in the sample. Because the imbalanced sample missing rate of certain features may

bring a high impact, we balanced the positive and negative sample missing rates. After rounds of iterative experiments, with the adjusted sample balance degree, AUC, ACC and Precision are improved by 12.7%, 5.4% and 57.5%, respectively.

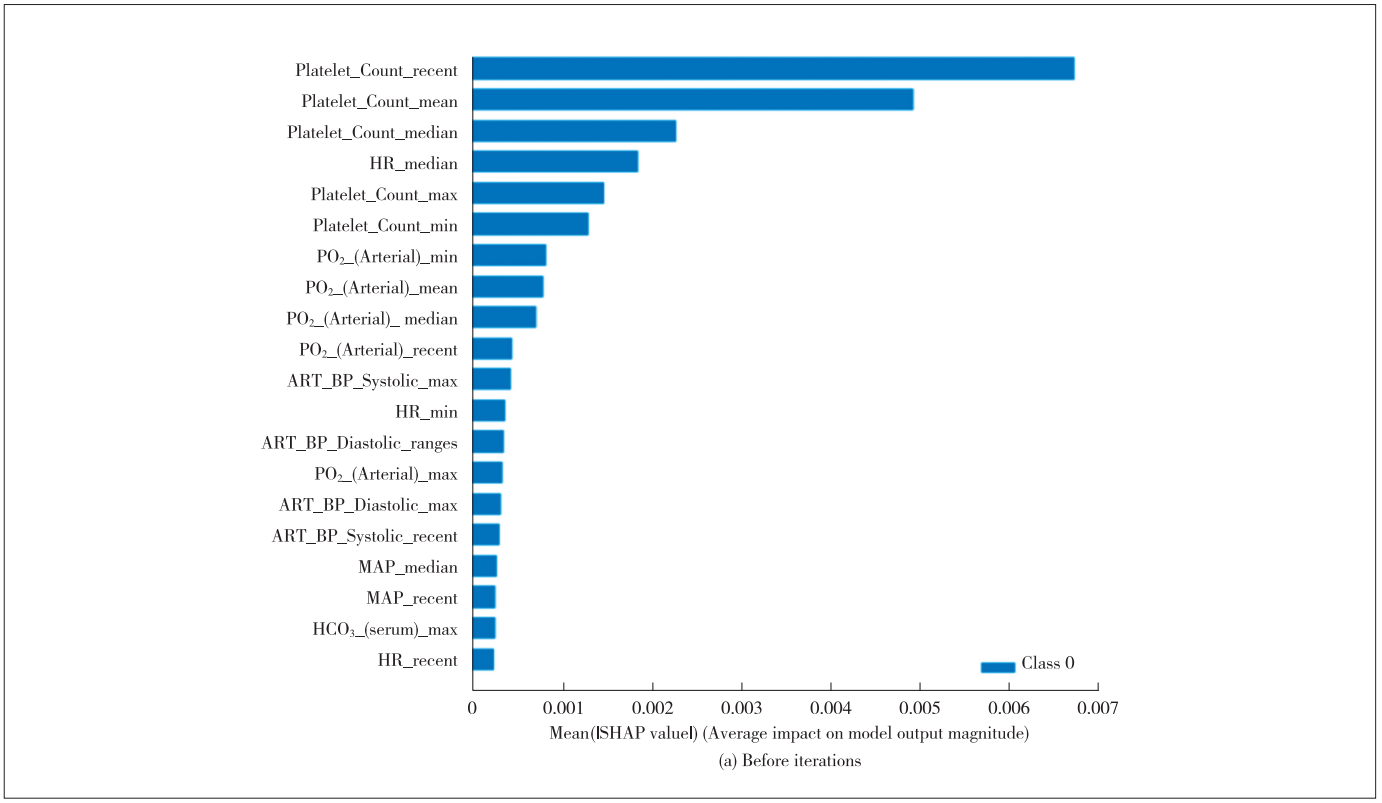
After several rounds of iterative adjustments, features were selected by balancing positive and negative samples, combining physician's experience and SHAP evaluation, thus our model has the accuracy of prediction, and the results are consistent with doctors' cognition. Fig. 5 shows SHAP evaluation results before and after iterations. Balanced features derive a more reasonable final model, which is not overly affected by just Platelet related features.

4 Conclusions

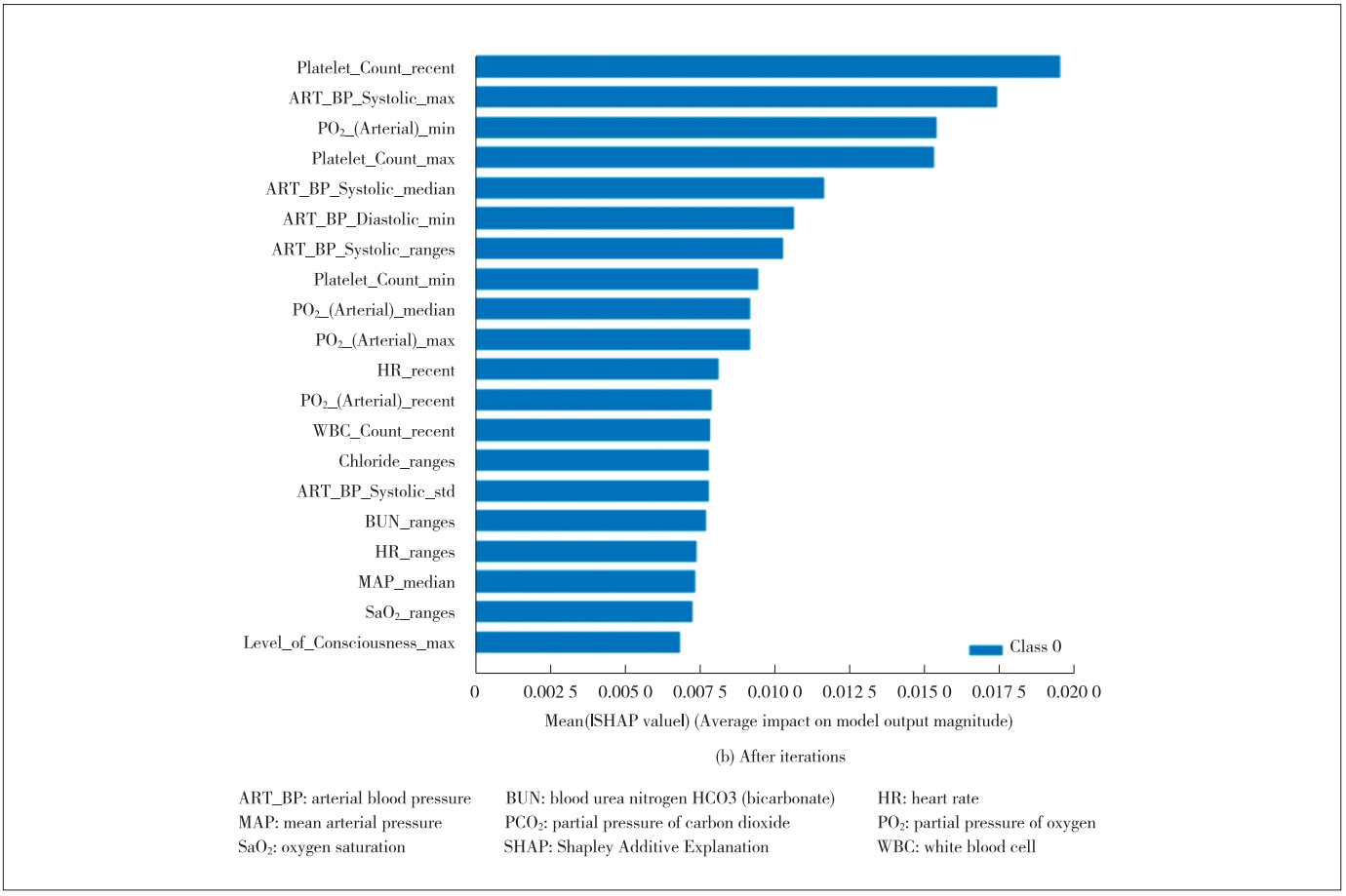
We propose a federated learning system optimized for sparse time series data with transformation and resampling. In this design, we merge multiple types of sparse time series data for each client and clean them sufficiently so that they are interpretable, design a view for each type of data, and perform statistical data processing, that is, filling in data and selection of data for local models. In this way, we get a batch of valid serialized data ready for clients to fit local artificial neural networks. Next, the network is fitted, the SHAP is used as the interpretation toolkit, and the SHAP ranking fed back is judged by the sample data balanced and physician selected. Due to the sparseness of this kind of data from hospitals and the implementation of the above methods, our federated learning clients depend on less the alignment of time series data on the timeline in the case of extremely poor data quality, which is an effective system to break the limit of high medical time series data missing ratio. The experiments were only conducted in a simple simulated experimental environment. For further verification and exploration, experiments in a real distributed environment and large-scale experiments are required. Due to the research goals, we did not additionally consider the scalability of the system in terms of transmission and induction, which may require more experimental and theoretical analysis.

References

[1] DENNY J C, COLLINS F S. Precision medicine in 2030—seven ways to transform healthcare [J]. Cell, 2021, 184(6): 1415 – 1419. DOI: 10.1016/j.cell.2021.01.015
[2] RAGHUNATH S, ULLOA CERNA A E, JING L Y, et al. Prediction of mortality from 12-lead electrocardiogram voltage data using a deep neural network [J]. Nature medicine, 2020, 26(6): 886 – 891. DOI: 10.1038/s41591-020-0870-z
[3] GAO Y, CAI G Y, FANG W, et al. Machine learning based early warning system enables accurate mortality risk prediction for COVID-19 [J]. Nature communications, 2020, 11: 5033. DOI: 10.1038/s41467-020-18684-2
[4] TOMAŠEV N, GLOROT X, RAE J W, et al. A clinically applicable approach to continuous prediction of future acute kidney injury [J]. Nature, 2019, 572(7767): 116 – 119. DOI: 10.1038/s41586-019-1390-1
[5] LIANG H Y, TSUI B Y, NI H, et al. Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence [J]. Nature medicine, 2019, 25(3): 433 – 438. DOI: 10.1038/s41591-018-0335-9
[6] KOCH M. Artificial intelligence is becoming natural [J]. Cell, 2018, 173(3): 531 – 533. DOI: 10.1016/j.cell.2018.04.007
[7] BACCHI S, TAN Y, OAKDEN-RAYNER L, et al. Machine learning in the prediction of medical inpatient length of stay [J]. Internal medicine journal, 2022, 52(2): 176 – 185



(a) Before iterations



▲**Figure 5. Comparison of SHAP values**

[8] VAROQUAUX G, CHEPLYGINA V. Machine learning for medical imaging: methodological failures and recommendations for the future [J]. NPJ digital medicine, 2022, 5(1): 1 – 8
[9] COUDRAY N, OCAMPO P S, SAKELLAROPOULOS T, et al. Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning [J]. Nature medicine, 2018, 24(10): 1559 – 1567. DOI: 10.1038/s41591-018-0177-5
[10] YANG Q, LIU Y, CHENG Y, et al. Synthesis lectures on artificial intelligence and machine learning: federated learning [M]. Berlin Heidelberg, Germany: Springer, 2019: 1 – 207. DOI: 10.2200/s00960ed2v01y201910aim043
[11] KELLY C J, KARTHIKESALINGAM A, SULEYMAN M, et al. Key challenges for delivering clinical impact with artificial intelligence [J]. BMC medicine, 2019, 17(1): 195. DOI: 10.1186/s12916-019-1426-2
[12] LI R W, CHEN Y, RITCHIE M D, et al. Electronic health records and polygenic risk scores for predicting disease risk [J]. Nature reviews genetics, 2020, 21(8): 493 – 502. DOI: 10.1038/s41576-020-0224-1
[13] SHILO S, ROSSMAN H, SEGAL E. Axes of a revolution: challenges and promises of big data in healthcare [J]. Nature medicine, 2020, 26(1): 29 – 38. DOI: 10.1038/s41591-019-0727-5
[14] JOHNSON A, BULGARELLI L, POLLARD T, et al. MIMIC-IV-ED [DB]. PhysioNet, 2021. DOI: 10.13026/as7t-c445
[15] LI J, YAN X S, CHAUDHARY D, et al. Imputation of missing values for electronic health record laboratory data [J]. NPJ digital medicine, 2021, 4(1): 147. DOI: 10.1038/s41746-021-00518-0
[16] WU J R, VODOVOTZ Y, ABDELHAMID S, et al. Multi-omic analysis in injured humans: patterns align with outcomes and treatment responses [J]. Cell reports medicine, 2021, 2(12): 100478. DOI: 10.1016/j.xcrm.2021.100478
[17] WEERAKODY P B, WONG K W, WANG G J, et al. A review of irregular time series data handling with gated recurrent neural networks [J]. Neurocomputing, 2021, 441: 161 – 178. DOI: 10.1016/j.neucom.2021.02.046
[18] KUMAR Y, SINGLA R. Federated learning systems for healthcare: perspective and recent progress [J]. Federated learning systems, 2021: 141 – 156
[19] VAID A, JALADANKI S K, XU J, et al. Federated learning of electronic health records improves mortality prediction in patients hospitalized with covid-19 [EB/OL]. (2020-08-11)[2021-11-21]. https://www.medrxiv.org/content/10.1101/2020.08.11.20172809v1
[20] SATTLER F, WIEDEMANN S, MULLER K R, et al. Robust and communication-efficient federated learning from non-i.i.d. data [J]. IEEE transactions on neural networks and learning systems, 2020, 31(9): 3400 – 3413. DOI: 10.1109/TNNLS.2019.2944481
[21] RAMASWAMY S, MATHEWS R, RAO K, et al. Federated learning for emoji prediction in a mobile keyboard [EB/OL]. (2019-06-11)[2021-11-21]. https://arxiv.org/abs/1906.04329
[22] KANG J W, XIONG Z H, NIYATO D, et al. Reliable federated learning for mobile networks [J]. IEEE wireless communications, 2020, 27(2): 72 – 80. DOI: 10.1109/MWC.001.1900119
[23] WEI R M, WANG J Y, SU M M, et al. Missing value imputation approach for mass spectrometry-based metabolomics data [J]. Scientific reports, 2018, 8(1): 663. DOI: 10.1038/s41598-017-19120-0
[24] LUNDBERG S, LEE S I. A unified approach to interpreting model predictions [EB/OL]. (2017-11-25)[2021-12-05]. https://arxiv.org/abs/1705.07874
[25] ANNANE D, BELLISSANT E, CAVAILLON J M. Septic shock [J]. The lancet, 2005, 365(9453): 63 – 78. DOI: 10.1016/S0140-6736(04)17667-8
[26] ASTIZ M E, RACKOW E C. Septic shock [J]. The lancet, 1998, 351(9114): 1501 – 1505. DOI: 10.1016/S0140-6736(98)01134-9
[27] RIVERS E P, MCINTYRE L, MORRO D C, et al. Early and innovative interventions for severe sepsis and septic shock: taking advantage of a window of opportunity [J]. CMAJ, 2005, 173(9): 1054 – 1065. DOI: 10.1503/cmaj.050632

Biographies

LU Feng received her MS and PhD degrees in computer science from Huazhong University of Science and Technology, China in 1997 and 2006. She is currently an associate professor in School of Computer Science and Technology, Huazhong University of Science and Technology. Her current research interests include big data, artificial intelligence and distributed computing. She has authored two books and over 20 papers in refereed journals and conferences in these areas. She is a member of CCF and a senior member of the first Session of Chinese Hospital Association, Health Data Application and Management Committee. She was the recipient of three teaching achievement and curriculum development awards.

GU Lin (lingu@hust.edu.cn) received her MS and PhD degrees in computer science from University of Aizu, Fukushima, Japan in 2011 and 2015. She is currently an associate professor in School of Computer Science and Technology, Huazhong University of Science and Technology, China. Her current research interests include serverless computing, network function virtualization, cloud computing, software-defined networking, and data center networking. She has authored two books and over 40 papers in refereed journals and conferences in these areas. She is a member of IEEE and a senior number of CCF.

TIAN Xuehua received her MS degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2022. She works on medical data mining and machine learning. She mainly focuses on working with sparse time series data.

SONG Cheng received his bachelor's degree from the School of Computer Science and Artificial Intelligence, Wuhan University of Technology, China in 2021. He majored in software engineering. He is working on data mining and machine learning for an MS degree at Huazhong University of Science and Technology, China.

ZHOU Lun received his MS Degree in Tongji Medical College, Huazhong University of Science and Technology, China in 2003. He is an associate professor of geriatrics at Tongji Hospital. His research interests include the pathogenesis of congenital heart disease and early warning of severe diseases in the elderly. He has published more than 20 papers in refereed journals such as PNAS and JMCC.

MSRA-Fed: A Communication-Efficient Federated Learning Method Based on Model Split and Representation Aggregate

LIU Qinbo[1,2], JIN Zhihao[1], WANG Jiabo[1], LIU Yang[1,3], LUO Wenjian[1,3]

(1. School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China;
2. Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Shenzhen 518055, China;
3. Peng Cheng Laboratory, Shenzhen 518055, China)

DOI: 10.12142/ZTECOM.202203005

https://kns.cnki.net/kcms/detail/34.1294.TN.20220804.1854.002.html, published online August 5, 2022

Manuscript received: 2022-06-20

Abstract: Recent years have witnessed a spurt of progress in federated learning, which can coordinate multi-participation model training while protecting the data privacy of participants. However, low communication efficiency is a bottleneck when deploying federated learning to edge computing and IoT devices due to the need to transmit a huge number of parameters during co-training. In this paper, we verify that the outputs of the last hidden layer can record the characteristics of training data. Accordingly, we propose a communication-efficient strategy based on model split and representation aggregate. Specifically, we make the client upload the outputs of the last hidden layer instead of all model parameters when participating in the aggregation, and the server distributes gradients according to the global information to revise local models. Empirical evidence from experiments verifies that our method can complete training by uploading less than one-tenth of model parameters, while preserving the usability of the model.

Keywords: federated learning; communication load; efficient communication; privacy protection

Citation (IEEE Format): Q. B. Liu, Z. H. Jin, J. B. Wang, et al., "MSRA-Fed: a communication-efficient federated learning method based on model split and representation aggregate," *ZTE Communications*, vol. 20, no. 3, pp. 35–42, Sept. 2022. doi: 10.12142/ZTECOM.202203005.

1 Introduction

As the Internet has found its way into our everyday life, the deep integration of the artificial intelligence technology represented by deep learning and IoT technology has become a new trend, and then the concept of Artificial Intelligence of Things (AIoT)[1] came into being. AIoT not only requires each device to be intelligent, but also that intelligent terminals can cooperate and integrate with each other, so as to give full play to the great value of IoT. AIoT already has a wide range of applications. For example, the intelligent camera can do object tracking and detect abnormal sound according to real-time voice; the smart bracelet can analyze users' health status according to the monitored data; intelligent systems in autonomous vehicles collect data from monitoring equipment to make driving decisions. However, traditional AI techniques usually require centralized data collection and centralized training of models, which needs to upload the data on IoT devices to application providers. Such paradigms obviously raise privacy concerns. For example, there is a high probability that users will not allow smart security cameras to upload their bedroom surveillance video to the server for model training[2]. Therefore, the practical deployment of AIoT faces the challenge of privacy risks.

To alleviate privacy concerns, an effective approach is to employ federated learning[3]. Federated learning stipulates that each client saves data locally, and only uses parameters instead of raw data to communicate between clients and the server. Therefore, it can be used as a privacy-preserving computing paradigm and has received extensive attention in academia and industry. Especially in the field of IoT, federated learning has been proven to power IoT in data sharing, attack detection, mobile group perception, privacy and security[4]. With federated learning, each IoT device is no longer limited to acquiring knowledge from its own dataset, but can benefit

This work is supported by Shenzhen Basic Research (General Project) under Grant No. JCYJ20190806142601687, Shenzhen Stable Supporting Program (General Project) under Grant No. GXWD20201230155427003-20200821160539001, Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies under Grant No. 2022B1212010005, and Shenzhen Basic Research (Key Project) under Grant No. JCYJ20200109113405927.

The corresponding author is LIU Yang. LIU Qinbo, JIN Zhihao and WANG Jiabo contribute equally in this work.

from the data and information of other devices while protecting their own privacy. Taking the widely used FedAvg algorithm[5] as an example, in each round of training, IoT clients randomly selected by the central server upload model parameters to the server. The central server only receives and aggregates these parameters to obtain an enhanced global model for distributed learning. Then, the central server distributes the aggregated model to guide local training. The data of each client are always stored locally, and knowledge sharing among IoT devices is achieved through the aggregation of model parameters.

However, there are still many challenges for federated learning to be deployed on IoT devices, one of which is the urgent need to improve communication efficiency. Although the iterative development of communication technologies such as 4G and 5G in recent years has made the bandwidth quite impressive, compared with the computing performance of the server and clients, what hinders training efficiency of federated learning the most is the communication between the server and clients[6]. Especially in edge computing and IoT, there are some applications that require extremely high communication efficiency. In some practical settings where federated learning is deployed, in order to improve the overall operational efficiency, the system will ignore some devices with limited network bandwidth or limited access, that is, they will not participate in the training round. However, such simple processing will lead to some local models not being optimized, which will seriously affect user experience[7].

In order to improve the communication efficiency of federated learning, an effective way is to reduce the amount of communication data between clients and the server. FedProto[8] proposed to replace the gradient-based aggregation with prototype-based aggregation. The prototype size is much smaller than the size of gradient, so the prototype-based method can effectively improve communication efficiency. However, FedProto simply uploads the prototype to the server for weighted average to obtain global prototype, which cannot effectively fuse the information of clients. SplitFed[9] uses the idea of split learning to split the neural network into a client-side network and a server-side network, thereby reducing the parameters required for communication. However, multi-client split learning is done asynchronously, so it is inefficient and causes clients to be idle. Subsequently, SplitFed introduces a fed server to execute FedAvg on the client side, which can synchronously train the split learning and greatly accelerate convergence. However, this method requires clients to upload smash data and the server to send gradient back to the clients in each round of model update. The amount of data transferred between the server and clients is still very large.

To cope with the problem of high communication load in federated learning, we propose a novel method for efficient communication based on model split and representation aggregate—MSRA-Fed. MSRA-Fed considers the advantages of federated learning and split learning, and significantly reduces the amount of data communicated between clients and the server. It also provides stronger privacy protection than the traditional gradient-based communication. At the same time, the proposed method can ensure the deep fusion of information of each client during the aggregation process. The paradigm in this paper is suitable for federated learning scenarios that do not require particularly high accuracy, but require low communication costs and strict security.

Our contributions are as follows:

1) Through empirical evidence from experiments, we verify that the outputs of the last hidden layer of a neural network can carry the characteristics of the training set. Therefore, these outputs can be used to replace the parameters of the model to cooperate with each client for model training.

2) In response to the above observations, we propose a communication optimization strategy based on model split and representation aggregate. This approach can significantly reduce the parameters the client needs to upload, reduce the communication load of federated learning, and ensure the accuracy of the model.

The rest of this paper is organized as follows. Section 2 presents the background and related work. Section 3 introduces the scheme of MSRA-Fed in detail. Section 4 verifies the effectiveness of MSRA-Fed through experiments and shows the experimental results. Finally, a summary and discussion of future work are presented in Section 5.

2 Background and Related Work

Federated learning is a machine learning framework, whose purpose is to use distributed data to collectively train a common model[10]. In this way, the storage and computing power of each participant can be fully utilized. During the training process, decentralized clients will only have parameter communication with the central server, and no raw data will be exchanged between any clients[11]. Compared with the way where each participant trains independently, the participants in federated learning can obtain other client-side knowledge from the global model issued by the server, so as to make local models more effective. These characteristics of federated learning not only allow us to combine multi-party data for mining and analysis, but also avoid direct interaction of raw data and protect data privacy. Since federated learning was proposed, a large number of related papers and achievements have emerged. We note that there is extensive research on how to improve efficiency and effectiveness of federated learning.

MCMAHAN et al.[5] proposed FedAvg based on a centralized training architecture, which is robust to non-IID (independent and identically distributed) data distributions and can reduce the communication rounds required to train the model.

FedAvg adopts a synchronous update scheme during each round of training. For a fixed number of K clients, the server will randomly select a portion of the clients to participate in each round of training according to a fraction C. Ref. [5] shows the reward brought by increasing the number of clients will gradually decrease if the number of participating clients exceeds a certain threshold. In each round of local training, the client k calculates the gradient g_k of the local data under the current model to update its local model w_k. Typically, the client k can update w_k through multiple local epochs of training, and then the central server aggregates. Compared with the general distributed stochastic gradient descent, FedAvg reduces the number of iterations of global training by increasing the amount of computation on the client side, thereby reducing the communication rounds.

In the training process of FedAvg, each participant needs to frequently communicate parameters with the central server to update the local model. The amount of data communicated is generally large, resulting in a high communication cost[4]. WANG et al.[12] designed a Communication-Mitigated Federated Learning (CMFL) framework by identifying irrelevant updates on clients and excluding them in advance to prevent invalid updates, which aims to reduce communication costs by avoiding uploading irrelevant parameters. Although this method improves communication efficiency, it increases a lot of computational cost. SATTLER et al.[13] proposed an Sparse Ternary Compression (STC) compression framework for the shortcomings of several compression-based solutions that only compress upstream communication and are only effective for ideal conditions (such as IID data distribution). Experiments show that STC outperforms the FedAvg algorithm under certain conditions. Refs. [14] and [15] proposed two ways to reduce the cost of upstream communication: the structured update and summary update. The former uses fewer samples and fewer updates; the latter uses lossy compression to update parameters. However, both approaches lack robustness when dealing with poor quality data. CALDAS et al. [16] proposed a federated mechanism that makes it possible to efficiently train a smaller subset of the global model and address the downlink communication pressure with server-to-client compression. This work broke the situation that downlink communication has not been studied. The above-mentioned research reduces the amount of transmitted data by compressing the original model or obtaining more compact updates, although the accuracy decreases to some extent.

In order to communicate efficiently, there are also some studies that reduce the number of communication parameters by splitting the model into different devices. Our idea is primarily inspired by the scheme of model split. DEAN et al.[17] proposed DistBelief, which uses the paradigm of model parallelism by model split. DistBelief can manage the data transmission between participants in the process of bottom communication, synchronization, training and inference. However, since many clients will be idle when the system is running, DistBelief is not efficient. GPipe[18] proposed by Google introduces pipelines on the basis of model parallelism, which improves the utilization of devices in the parallelism. However, as a special setting of distributed learning, federated learning assumes that the central server is not allowed to manage and schedule the clients participating in the local training. Therefore, the conventional model split and pipeline parallelization are not suitable for federated learning. Recently, some researchers have also tried to introduce model split into federated learning. Ref. [19] applied split learning to Long Short-Term Memory (LSTM) networks and proposed LSTMSPLIT to classify time-series data with multiple clients. Ref. [20] proposed an asynchronous learning strategy, which divided the neural network into deep and shallow layers. The method of updating the shallow layers more frequently than the deep layers can reduce the communication cost. THAPA et al.[9] proposed SplitFed by combining federated learning and split learning[21], which solves the problem that each client cannot be updated synchronously under the model split strategy. However, the amount of data to be transmitted in these methods is still large and the communication frequency is too high for efficient training.

3 MSRA-Fed Method

3.1 Outputs of Last Hidden Layer Carry Characteristics

The trained model can reflect the characteristics of a training set and even remember training samples when overfitting[22]. That is, the model parameters carry the characteristics of the training samples. Therefore, the method adopted by federated learning when updating a global model is to upload the parameters of local models to the server for aggregation, in order to obtain a better global model without violating the privacy of training sets. However, the parameters of a neural network model are usually of high dimension, which makes clients suffer from high communication load in neural network based federated learning tasks. Therefore, we choose to use other data carrying the characteristics of training samples instead of all model parameters for server aggregation, to trade off the communication load and model accuracy of federated learning. From the structure of neural network models, it can be found that the data directly involved in determining the prediction results are the outputs of the last hidden layer. Due to this fact, we propose an assumption that the outputs of the last hidden layer of neural networks should be similar for samples with similar prediction results.

We take a classification task on the public Modified National Institute of Standards and Technology (MNIST) database as an example to verify the hypothesis. To facilitate presentation in the space-constrained paper[11], we set up a Multilayer Perceptron (MLP) with two hidden layers, each with 12

neurons. The learning rate is set to 0.01. Our experimental results are shown in Table 1. The first three rows of Table 1 are the results obtained during ten rounds of training, which record the dissimilarity of outputs of the last hidden layer under each predicted label. We denote the number of samples with label l by N_l. The dissimilarity of outputs of the k-th neuron can be formulated as:

$$\sigma_{l,k}^2 = \frac{1}{N_l}\sum_{n=1}^{N_l}(X_{l,k}^n - \bar{X}_{l,k})^2, \quad (1)$$

where $X_{l,k}^n$ refers to the output of the n-th sample with label l on the k-th neuron of the last hidden layer, and $\bar{X}_{l,k} = \sum_n X_{l,k}^n / N_l$. We take the samples with Labels 3, 6, and 7 in the dataset as examples, and similar results can be obtained under other labels. The last row of Table 1 is the dissimilarity of outputs of the last hidden layer blending all predicted labels. The 12 components represent the variance of the corresponding outputs of 12 neurons in the last hidden layer.

From the above experimental results, it can be found that the outputs of training samples with the same label in the last hidden layer are similar, because their variances are smaller than those in the scenario where we do not distinguish the labels. A similar phenomenon on the public dataset CIFAR-10 is also found. Accordingly, it can be considered that for neural network models on clients, the outputs of the last hidden layer can carry the characteristics of training samples to a certain extent.

3.2 Communication-Efficient Strategy Based on Model Split and Representation Aggregate

Based on the assumption in Subsection 3.1, we can conclude that for neural network models on different clients, an effect close to aggregating all parameters can be obtained by aggregating the outputs of the last hidden layer. The method of splitting neural networks has made great progress in split learning[23]. However, split learning requires too much communication between clients and the server due to communicating high-dimensional parameters. In terms of communication consumption per round, the communication load of split learning is not significantly reduced compared to federated learning. To adopt the paradigm of aggregating hidden layers in federated learning, this paper sets an output layer on the server side for complete training and retains a local model on each client side. As shown in Fig. 1, the server calculates the gradient of the loss function to the aggregated last hidden layer after local forward propagation, and distributes the gradient to clients. Each client uses this gradient to assist the update of its local model. Such learning process revises local models indirectly through back-propagation of gradients.

It is worth noting that we reform the neural network based federated learning on the server side and the client side respectively. Each client trains a local neural network, which together with the server constitutes a federated neural network framework as a whole. Unlike split learning, we keep the complete local model on each client and follow the setting of federated learning to support local training. In addition, the server does not collect local models or gradients uploaded by clients, which can reduce the risk of model increments or gradients compromising data privacy. Even malicious participants cannot launch inference attacks[24] or model inversion attacks[25] against honest clients by eavesdropping on local models. This means that our method could have stricter privacy and security, and in particular, has high communication efficiency. Each client participates in the aggregation process of federated learning by uploading the outputs of the last hidden layer, while the server coordinates a collaborative training by distributing the back-propagated gradient. Clients use this gradient to modify their local models after receiving the back-propagated gradient. More details are shown in Algorithm 1. We implement the federation of model training on the client and server through a local model training stage, a communication stage, and a back-propagation stage.

Algorithm 1. MSRA-Fed based on model split and representation aggregate

Require: The number of clients: K; learning rate: η; the number of local epochs: E; the number of global iterations: T; local datasets: $\{D_i\}_{i=1}^K$

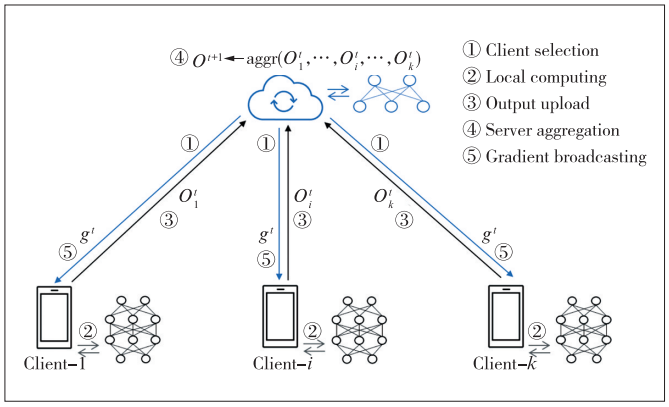
Ensure: Local models on clients: $\{w_i\}_{i=1}^K$

//Server executes:

1: Initialize: w_i //Randomly initialize local models
2: **for** each round $t = 1$ to T **do**
3: $m = \max(C \times K, 1)$
4: Z_t = random set of m clients
5: **for** each client $i \in Z_t$ in parallel **do**

▼**Table 1. Variance of the outputs of each neuron in the last hidden layer**

Label	1st Neuron	2nd Neuron	3rd Neuron	4th Neuron	5th Neuron	6th Neuron	7th Neuron	8th Neuron	9th Neuron	10th Neuron	11th Neuron	12th Neuron
Label-3	1.00	2.94	1.98	11.82	1.80	2.16	0.01	9.60	0.28	4.29	3.84	11.03
Label-6	4.82	0.00	3.98	1.88	0.00	3.65	1.17	0.03	36.54	8.05	3.65	0.72
Label-7	0.00	0.01	1.37	13.00	4.92	5.20	3.33	0.10	17.91	12.54	3.82	0.08
Blended	52.11	34.38	18.51	24.65	16.18	11.99	3.99	35.46	17.24	18.57	27.22	23.68



▲ **Figure 1. Overview of MSRA-Fed**

6: $\left(\boldsymbol{X}_{i,l}, l\right) = \text{ClientTrain}(i)$
7: **end for**
8: **for** each label l **do**
9: $\boldsymbol{X}_l = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{X}_{i,l}$
10: Forward propagation with $\left(\boldsymbol{X}_l, l\right)$, calculate $\boldsymbol{g}$
11: **end for**
12: **for** each client $i \in Z_t$ in parallel **do**
13: ClientUpdate(i, $\boldsymbol{g}$)
14: **end for**
15: **end for**
//Clients execute:
16: **function** ClientTrain(i):
17: **for** each local epoch $e = 1$ to E **do**
18: **for** each sample $\left(s_n, l_n\right) \in D_i$ **do**
19: **if** $e < E$ **do**
20: $w_i \leftarrow w_i - \eta \cdot \nabla loss(w_i; s_n, l_n)$
21: **else do**
22: Forward propagation with $\left(s_n, l_n\right)$, calculate $\boldsymbol{X}_l^n$
23: **end if**
24: **end for**
25: **end for**
26: **for** each label l **do**
27: $\boldsymbol{X}_{i,l} = \frac{1}{N_l}\sum_{n=1}^{N_l} \boldsymbol{X}_l^n$
28: **end for**
29: **return** a set of $\left(\boldsymbol{X}_{i,l}, l\right)$
30: **function** ClientUpdate(i,$\boldsymbol{g}$):
31: Back propagation with $\boldsymbol{g}$, update w_i

1) Local model training stage. As shown in Lines 17 – 28 of Algorithm 1, in each round of federated learning, each client trains a local model for several epochs and then uploads aggregated representation to the server for aggregation. The client preserves a full model locally instead of a split model. Unlike conventional federated learning, the client uploads the aggregated outputs of the last hidden layer instead of all parameters. In order to reduce communication consumption, the client adopts a class-wise average aggregation of the local hidden layer output set. The data that finally participate in global federated aggregation is the averaged hidden layer outputs under each label.

2) Communication stage. At Line 29 of Algorithm 1, the client communicates with the server. When the client uploads local data, the server collects the outputs of all clients' hidden layers and averages these local representations separately according to class (or label). Taking the classification task on MNIST as an example, the server will finally obtain the outputs of the last hidden layer corresponding to ten different labels after one communication.

3) Back-propagation stage. In Lines 8 – 14 of Algorithm 1, after averaging the hidden layers, the aggregated output is used as input to the neural network on the server to continue training and compute a gradient for back-propagation. After receiving the back-propagated gradient from the server, clients call it for back-propagation. That is, the back-propagated gradient of the federated averaged hidden layer is used to revise local models. For the client, the local training of the last epoch in each round is back-propagated using the gradient issued by the server side.

3.3 Summary

For the federated learning framework, we modify the collaborative training process and propose a communication optimization strategy based on model split and representation aggregation. Our method aggregates local representations on the server through the outputs of the last hidden layer, and uses gradients delivered by the server to coordinate training on each client. On the basis of the local training model, each client additionally introduces the server's gradient containing global information to correct its local model, which significantly reduces the communication load and also makes the model update more stable. In addition, since clients do not need to upload local models or gradients to any third party, the security risk of local data and models could be low. Although we did not focus on improving the accuracy of the model but on efficient communication, this will be improved in future work to sacrifice less accuracy under conditions of efficient communication and strict protection of privacy and security.

4 Evaluation

4.1 Dataset and Experimental Setup

We use the public dataset MNIST as the experimental dataset, which consists of ten classes of images. MNIST includes ten categories of handwritten digits, and each image is a gray-scale image of size 28×28. This dataset contains 60 000 training samples and 10 000 test samples. Since the method designed in this paper is suitable for various neural network

models, the commonly used model MLP is used here to verify the effectiveness of our method. Without loss of generality, we set up a learning environment with one central server and ten clients. The proportion of clients participating in each global training is $C = 1$, and the number of local epochs of the clients is $E = 5$. For the MNIST dataset, we employ an MLP with one hidden layer with 256 neurons. The learning rate η of MLP is set to 0.01. In the case of independent and identically distributed data, the samples are randomly shuffled. For each dataset used for training, all training samples are randomly and uniformly distributed to clients and each client randomly draws samples from it.

This experiment is carried out under Python 3.6, and the computer hardware is configured as 16 G memory, Intel i5-10400F CPU and GTX1650 GPU.

4.2 Evaluation Metrics

The communication efficiency and performance of the model is measured from the communication load required for model convergence and the final accuracy of the model. The definition of accuracy used here is shown in Eq. (2).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{2}$$

where TP is the number of true positives, which means that the positive samples are also predicted as positive by the model; TN is the number of true negatives, which means that the negative samples are also predicted to be negative; FP is the number of false positives, which means that the negative samples are predicted to be positive; FN is the number of false negatives, representing positive samples that are incorrectly predicted. The positive and negative here represent the true class and other classes of the sample, respectively.

4.3 Experimental Results and Analysis

Every time the federated learning progresses to the communication stage between the client and the server, we use an element in the tensor as a unit to count the amount of data transmitted in the communication. We define the communication load as the total amount of transmitted data. By comparing the communication load and the improvement percentage of model accuracy, or the communication load required to improve the model accuracy, the gap between the conventional federated neural network model and the improved communication-efficient MSRA-Fed can be clearly displayed.

Table 2 is a comparison of communication efficiency after five rounds of training on MNIST using MSRA-Fed and FedAvg. The value of communication load per 1% accuracy improvement indicates the average communication load consumed when the final trained model improves the accuracy by 1% compared to the initial model accuracy. The results show that although the initial accuracy of MSRA-Fed is lower than FedAvg by about 12%, the accuracy of two methods has reached more than 80% after five rounds of model training. Meanwhile, MSRA-Fed consumes less than 2% of the communication load of FedAvg.

The results after ten rounds of training on the MNIST dataset using MSRA-Fed and FedAvg are shown in Table 3. Both algorithms have almost converged after ten rounds of training. The results demonstrate that the communication load consumed by conventional FedAvg is 56.19 times that of MSRA-Fed when training for ten rounds. Moreover, the communication load of MSRA-Fed training for ten rounds is much lower than that of FedAvg training for five rounds, which shows that MSRA-Fed can be efficiently trained for more rounds in the same time and the accuracy improvement per communication load is high.

▼ Table 2. Communication efficiency comparison after five rounds of training on MNIST dataset

Method	Initial Accuracy/%	Accuracy After Five Rounds of Training/%	Communication Load/B	Communication Load per 1% Accuracy Improvement
FedAvg	71.62	86.10	6 352 000	54 834.25
MSRA-Fed	59.97	80.11	123 280	765.14

MNIST: Modified National Institute of Standards and Technology

▼ Table 3. Communication efficiency comparison after ten rounds of training on MNIST dataset

Method	Initial Accuracy/%	Accuracy After Ten Rounds of Training/%	Communication Load/B	Communication Load per 1% Accuracy Improvement
FedAvg	71.62	86.15	12 704 000	109 291.12
MSRA-Fed	59.97	80.48	226 080	1 337.86

MNIST: Modified National Institute of Standards and Technology

When we focus on "communication load per 1% accuracy improvement" in Tables 2 and 3, it can be seen that the strategy based on model split and representation aggregation requires far less communication load than FedAvg with the same accuracy improvement. This shows that our proposed method has much higher communication efficiency while sacrificing a little model accuracy. It is worth mentioning that the accuracy of our method is lower than that of FedAvg when the algorithm converges. This is because we only use the outputs of the last hidden layer instead of all parameters to aggregate global information in order to significantly improve the communication efficiency. Each client maintains a complete model locally, while introducing global information sent by the server during the training process to correct the local model. Compared to FedAvg, the accuracy of our method drops slightly, but remains within acceptable limits. Most importantly, we reduce the amount of communication significantly.

5 Conclusions and Future Work

In this paper, we modify the federated learning model for the problem of high communication load. Specifically, a method for efficient communication is designed based on model split and representation aggregation. By enabling the client to upload the outputs of the last hidden layer instead of all parameters and using the global information issued by the server to guide and correct the updates of each local model, the communication consumption of federated learning can be significantly reduced while ensuring the accuracy of the model. Furthermore, we experimentally validate the method proposed in this paper. The experimental results show that the model spit and representation aggregation mechanism can significantly reduce the required communication consumption, and the traditional training method FedAvg consumes 56.19 times the communication load than our method. While improving the communication efficiency, our method also guarantees the stability and accuracy of the model.

Backbone models other than neural networks are not explored in this paper, which will be our future work. This mechanism we design does not improve the accuracy of the model much when it converges, and may even have a slight negative impact. Future work will focus on addressing the above problems and exploring the possibility of large-scale application of our scheme in real environments.

References

[1] TALUKDER A, HAAS R. AIoT: AI meets IoT and web in smart healthcare [C]// 13th ACM Web Science Conference 2021. ACM, 2021: 92 – 98. DOI: 10.1145/3462741.3466650
[2] ALKHATIB S, WAYCOTT J, BUCHANAN G, et al. Privacy and the Internet of Things (IoT) monitoring solutions for older adults: a review [J]. Studies in health technology and informatics, 2018, 252: 8 – 14
[3] LI T, SAHU A K, TALWALKAR A, et al. Federated learning: challenges, methods, and future directions [J]. IEEE signal processing magazine, 2020, 37 (3): 50 – 60. DOI: 10.1007/978-3-030-85559-8_13
[4] NGUYEN D C, DING M, PATHIRANA P N, et al. Federated learning for internet of things: a comprehensive survey [J]. IEEE communications surveys & tutorials, 2021, 23(3): 1622-1658. DOI: 10.1109/COMST.2021.3075439
[5] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C]//20th International Conference on Artificial Intelligence and Statistics (AISTATS). PMLR, 2017: 1273 – 1282. DOI: 10.48550/arXiv.1602.05629
[6] SHAHID O, POURIYEH S, PARIZI R M, et al. Communication efficiency in federated learning: Achievements and challenges [EB/OL]. (2021-07-23)[2022-05-01]. https://arxiv.org/abs/2107.10996v1
[7] CHAI Z, ALI A, ZAWAD S, et al. TiFL: a tier-based federated learning system [C]//29th International Symposium on High-Performance Parallel and Distributed Computing. ACM, 2020: 125 – 136. DOI: 10.1145/3369583.3392686
[8] TAN Y, LONG G, LIU L, et al. Fedproto: federated prototype learning across heterogeneous clients [C]//AAAI Conference on Artificial Intelligence. AAAI, 2022: 8432 – 8440. DOI: 10.1609/aaai.v36i8.20819
[9] THAPA C, CHAMIKARA M A P, CAMTEPE S, et al. Splitfed: when federated learning meets split learning [EB/OL]. (2022-02-16)[2022-05-01]. https://arxiv.org/abs/2004.12088
[10] KONEČNÝ J, MCMAHAN H B, RAMAGE D, et al. Federated optimization: distributed machine learning for on-device intelligence [EB/OL]. (2016-10-08) [2022-05-01]. https://arxiv.org/abs/1610.02527
[11] KHAN L U, SAAD W, HAN Z, et al. Federated learning for internet of things: recent advances, taxonomy, and open challenges [J]. IEEE communications surveys & tutorials, 2021, 23(3): 1759 – 1799. DOI: 10.1109/COMST.2021.3090430
[12] WANG L, WANG W, LI B. CMFL: mitigating communication overhead for federated learning [C]//IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2019: 954 – 964. DOI: 10.1109/ICDCS.2019.00099
[13] SATTLER F, WIEDEMANN S, MÜLLER K R, et al. Robust and communication-efficient federated learning from non-IID data [J]. IEEE transactions on neural networks and learning systems, 2019, 31(9): 3400 – 3413. DOI: 10.1109/TNNLS.2019.2944481
[14] KONEČNÝ J, MCMAHAN H B, YU F X, et al. Federated learning: strategies for improving communication efficiency [EB/OL]. (2017-10-30)[2022-05-01]. https://arxiv.org/abs/1610.05492
[15] SURESH A T, FELIX X Y, KUMAR S, et al. Distributed mean estimation with limited communication [C]//International conference on machine learning. PMLR, 2017: 3329 – 3337. DOI: 10.48550/arXiv.1611.00429
[16] CALDAS S, KONEČNY J, MCMAHAN H B, et al. Expanding the reach of federated learning by reducing client resource requirements [EB/OL]. (2019-01-08)[2022-05-01]. https://arxiv.org/abs/1812.07210
[17] DEAN J, CORRADO G, MONGA R, et al. Large scale distributed deep networks [C]//25th International Conference on Neural Information Processing Systems, NIPS. 2012: 1223 – 1231
[18] HUANG Y, CHENG Y, BAPNA A, et al. GPipe: efficient training of giant neural networks using pipeline parallelism [C]//33rd International Conference on Neural Information Processing Systems. NIPS, 2019: 103 – 112
[19] JIANG L, WANG Y, ZHENG W, et al. LSTMSPLIT: Effective SPLIT Learning based LSTM on Sequential Time-Series Data [EB/OL]. (2022-03-08)[2022-05-01]. https://arxiv.org/abs/2203.04305
[20] CHEN Y, SUN X, JIN Y. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation [J]. IEEE transactions on neural networks and learning systems, 2019, 31 (10): 4229 – 4238. DOI: 10.1109/TNNLS.2019.2953131
[21] THAPA C, CHAMIKARA M A P, CAMTEPE S A. Advancements of federated learning towards privacy preservation: from federated learning to split learning [M]//Federated Learning Systems. Springer, Cham, 2021: 79 – 109
[22] WANG W, FENG F, HE X, et al. Denoising implicit feedback for recommendation [C]//14th ACM International Conference on Web Search and Data Mining. ACM, 2021: 373 – 381. DOI: 10.1145/3437963.3441800
[23] SINGH A, VEPAKOMMA P, GUPTA O, et al. Detailed comparison of communication efficiency of split learning and federated learning [EB/OL]. (2019-01-08)[2022-05-01]. https://arxiv.org/abs/1909.09145
[24] WAINAKH A, VENTOLA F, MÜßIG T, et al. User-level label leakage from gradients in federated learning [J]. Proceedings on privacy enhancing technologies, 2022(2): 227 – 244. DOI: 10.2478/popets-2022-0043
[25] KHOSRAVY M, NAKAMURA K, HIROSE Y, et al. Model inversion attack: Analysis under gray-box scenario on deep learning based face recognition system [J]. KSII transactions on internet and information systems (TIIS), 2021, 15 (3): 1100 – 1118. DOI: 10.3837/tiis.2021.03.015

Biographies

LIU Qinbo received his BS degree in mathematics and physics basic science from the School of Mathematical Sciences, University of Electronic Science and Technology of China in 2021. He is currently pursuing his ME degree with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include federated learning and GNNs.

JIN Zhihao received his BE degree in computer science and technology from the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China in 2022. His research interests include federated learning.

WANG Jiabo received his BE degree in software engineering from the School of Information Science and Technology, Dalian Maritime University, China in 2021. He is currently pursuing his ME degree with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include federated learning.

LIU Yang (liu.yang@hit.edu.cn) received his BE degree in computer science from the Ocean University of China, China in 2010, MSc degree in software engineering from Peking University, China in 2013, and DPhil (PhD) degree in computer science from the University of Oxford, UK in July 2018, advised by Prof. Andrew SIMPSON. He is currently an assistant professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He is interested in security and privacy problems and, in particular, the privacy issues related to mobile and IoT devices.

LUO Wenjian received his BS and PhD degrees from the Department of Computer Science and Technology, University of Science and Technology of China, in 1998 and 2003, respectively. He is currently a professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His current research interests include computational intelligence and applications, network security and data privacy, machine learning, and data mining. Dr. LUO is also a senior member of the Association for Computing Machinery (ACM) and the China Computer Federation (CCF). He has been a member of the organizational team of more than ten academic conferences, in various functions, such as the program chair, the symposium chair and the publicity chair. He also serves as the chair of the IEEE CIS ECTC Task Force on Artificial Immune Systems. He also serves as an associate editor or an editorial board member for several journals, including *Information Sciences*, *Swarm and Evolutionary Computation*, *Journal of Information Security and Applications*, *Applied Soft Computing*, and *Complex & Intelligent Systems*.

Neursafe-FL: A Reliable, Efficient, Easy-to-Use Federated Learning Framework

TANG Bo[1], ZHANG Chengming[1], WANG Kewen[1], GAO Zhengguang[2], HAN Bingtao[2]
(1. ZTE Corporation, Shenzhen 518057, China;
2. The State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China)

DOI: 10.12142/ZTECOM.202203006
https://kns.cnki.net/kcms/detail/34.1294.TN.20220826.1322.001.html, published online August 26, 2022
Manuscript received: 2022-06-18

Abstract: Federated learning (FL) has developed rapidly in recent years as a privacy-preserving machine learning method, and it has been gradually applied to key areas involving privacy and security such as finance, medical care, and government affairs. However, the current solutions to FL rarely consider the problem of migration from centralized learning to federated learning, resulting in a high practical threshold for federated learning and low usability. Therefore, we introduce a reliable, efficient, and easy-to-use federated learning framework named Neursafe-FL. Based on the unified application program interface (API), the framework is not only compatible with mainstream machine learning frameworks, such as Tensorflow and Pytorch, but also supports further extensions, which can preserve the programming style of the original framework to lower the threshold of FL. At the same time, the design of componentization, modularization, and standardized interface makes the framework highly extensible, which meets the needs of customized requirements and FL evolution in the future. Neursafe-FL is already on Github as an open-source project[1].

Keywords: federated learning; privacy-preserving; Neursafe-FL

Citation (IEEE Format): B. Tang, C. M. Zhang, K. W. Wang, et al., "Neursafe-FL: a reliable, efficient, easy-to-use federated learning framework," *ZTE Communications*, vol. 20, no. 3, pp. 43 - 53, Sept. 2022. doi: 10.12142/ZTECOM.202203006.

1 Introduction

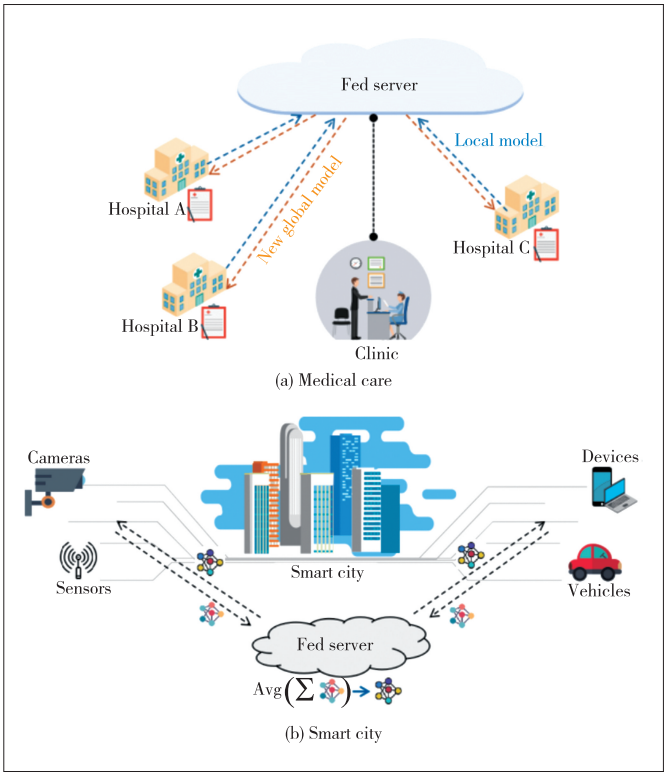
Federated learning (FL) is a distributed machine learning method that uses decentralized data residing on the client side to complete model training with the coordination of a central server[1-5]. It is a general method of "bringing the code to the data, instead of the data to the code"[3], and focuses on the security and privacy of data. Since the data are limited in the client domain during the training process and the intermediate data are encapsulated by the privacy-preserving algorithm, it could avoid privacy leakage in the training and inference process of machine learning. Especially with data protection laws and regulations, like the European General Data Protection Regulation (GDPR)[6], federated learning has been regarded as a hotspot in AI research.

Federated learning has many practical cases in the fields of finance[7], medical care[8], and smart city[9]. Fig. 1(a) shows the scene in medical care. It is expected to integrate the data of multiple hospitals to train a model, but the patient's information is very sensitive and private, which cannot be shared among hospitals. Fig. 1(b) shows that more and more intelligent edge devices with computing power, such as mobile phones, sensors, and cameras, join the smart city ecosystem. A large number of valuable data are generated on the devices. Since the data are private and impractical to collect, federated learning is very suitable for solving the problems in the above mentioned scenarios.

Federated learning breaks the data silos caused by privacy and security issues, which broadens the prospects of AI applications in many fields. However, FL introduces some unique challenges due to its distributed characteristics[10] shown as follows. 1) The problem of privacy leakage is caused by attack technologies such as membership inferring and feature inferring through intermediate data including model weights and gradients[11-13]; 2) The low efficiency of model convergence is caused by non-independent identically distribution (IID) data[14-15]; 3) The robustness of a model can be decreased by data poisoning[16], model poisoning[17] and other attack methods. There also exist efficiency problems which are caused by insufficient client-side computing power, data transmission bandwidth, etc[18].

We have developed a reliable, efficient, and easy-to-use open source framework to address the challenges mentioned

1. Neursafe-FL can be seen on the website of Github: https://github.com/neursafe/federated-learning.



▲ **Figure 1. Federated learning scenarios**

above in federated learning. Compared with the existing open source implementation for federated learning, more consideration is given to the migration of existing machine learning models from centralization to federation. The proposed framework cooperates well with mainstream machine learning frameworks and supports further upgradation, and it also retains the programming features of the original framework to simplify the implementations for various federated learning algorithms. Finally, the framework has advantages in future upgrades and evolution due to the componentization, modularization, and standardized interface.

The rest of this paper is organized as follows. Section 2 introduces the current research work on federated learning, including the technical challenges faced by federated learning, the comparison, and shortcomings of mainstream federated learning frameworks. In Section 3, we propose efficient and easy-to-use design solutions and principles. The design architecture, working principles, and the process of the system are introduced in Section 4. Section 5 presents experimental verification of multiple scenarios based on Neursafe FL. Finally, we summarize the contributions of this paper and point out directions for future work.

2 Related Work

With the increasing popularity of federated learning, a large amount of research work has been published to overcome the technical challenges of federated learning shown as follows. Differential privacy[19 – 23], secure multi-party computation[24 – 28], homomorphic encryption[29 – 33] and other privacy computing techniques were proposed to protect intermediate data such as the model weights and gradients, which avoids possible privacy leaks. Optimization and aggregation algorithms of FL such as FedAvg[1], FedNova[34], FedProx[35], SCAFFOLD[36], FedNas[37] were introduced to solve the problems of convergence efficiency caused by non-IID data. In Ref. [38 – 43], the authors proposed robust federated algorithms, such as Krum, FLRA, and Sageflow, to address the challenge of model robustness in the face of model attacks. The techniques of client-side incentives, quantization, models, and data compression were proposed to address the communication and computational efficiency problems in federated learning[44 – 53]. The incentive mechanism was adopted to solve the fairness problem in federated learning[54 – 56].

With the development of theoretical research on FL, a number of open source frameworks or libraries have emerged including Tensorflow Federated (TFF)[57], FATE[58], PySyft[59], FedML[60], and PaddleFL[61]. Among them, TFF, PySyft, and FedML are presented as the libraries for the research, while FATE and PaddleFL are frameworks or platforms for the production applications. However, these open source implementations have their own limitations as follows.

1) Most of the above work is developed based on a single underlying machine learning framework. For example, TFF is developed based on Tensorflow, FedML is implemented based on Pytorch, and PaddleFL is based on PaddlePaddle. Poorly substantial framework support leads to unnecessary costs for migrating FL applications.

2) The existing work supports limited application scenarios. For example, TFF only supports single-machine distributed simulation for research purposes; FATE and PaddleFL mainly solve cross-silo scenarios across data silos, while they are not suitable for cross-device scenarios. Although FedML supports a variety of application scenarios, the deployment process is very complicated. For example, the premise of FedML for various scenarios is that users must perform topology management, which makes user implementation more complex because network changes require the implementation change.

3) Most APIs of current frameworks are too complicated. Developers must learn proprietary API interfaces and programming specifications to implement federated learning, resulting in high costs for the migration of existing AI models.

4) Trade-offs between flexibility and usability are unsophisticated. On the one hand, some FL frameworks have a relatively high level of API encapsulation, which loses a certain degree of flexibility. On the other hand, the library represented by FedML has high flexibility, but the design makes development complicated, which leads to a high threshold for users.

Table 1 presents a comparison of open source projects. To solve the main challenges of federated learning mentioned

▼Table 1. Comparison of open source frameworks

Concerns	Features	TFF	PySyft	FedML	FATE	PaddleFL	Neursafe-FL
Supported running mode	Standalone	√	√	√	√	√	√
	Cross-device	×	×	√	×	×	√
	Cross-silo	×	×	√	√	√	√
Aggregation algorithms	IID (FedAvg, etc.)	√	√	√	√	√	√
	Non-IID (FedProx, etc.)	×	×	√	√	-	√
Supported underlying framework	Tensorflow	√	√	×	√	×	√
	Pytorch	×	√	√	√	×	√
Privacy protection methods	DP	√	√	√	×	√	√
	MPC	×	√	×	√	√	√
	HE	×	√	×	√	×	×
Flexibility	Device management	×	×	×	×	×	√
	Customization	×	×	√	×	×	√

DP: differential privacy FL: federated learning HE: homomorphic encryption IID: independent identically distribution MPC: multi-party computation TFF: tensorflow federated

above, Neursafe-FL is proposed as an efficient, simple, and easy-to-use federated learning framework without losing flexibility.

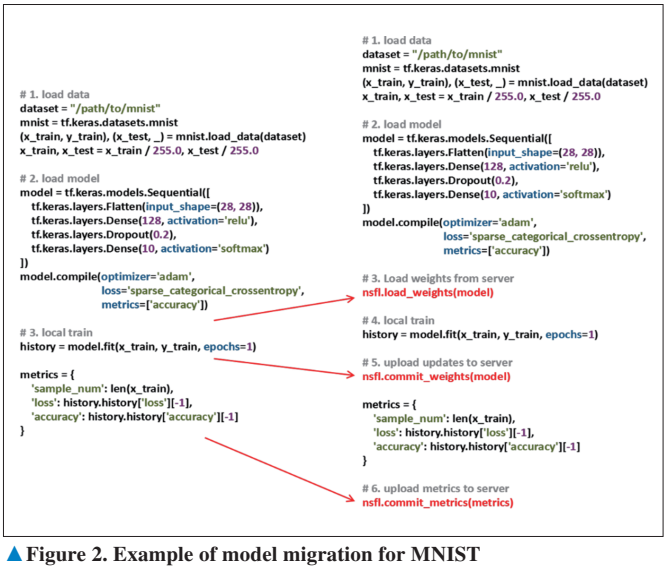
3 Design of Neursafe-FL

Neursafe-FL adopts the principles of componentization and modularization in the design. According to different functions and characteristics, we divide the system into several components and modules such as job scheduling, client management and selection, privacy protection, and optimization aggregation. The components are decoupled to reduce system complexity and provide feature-level scalability. Through componentized design, Neursafe-FL enables reliable services based on microservice management, the high availability (HA) mechanism, and job-level fault tolerance processing. In order to improve the usability and meet the requirements of long-term evolution for federated learning, we have made more efforts in the following areas:

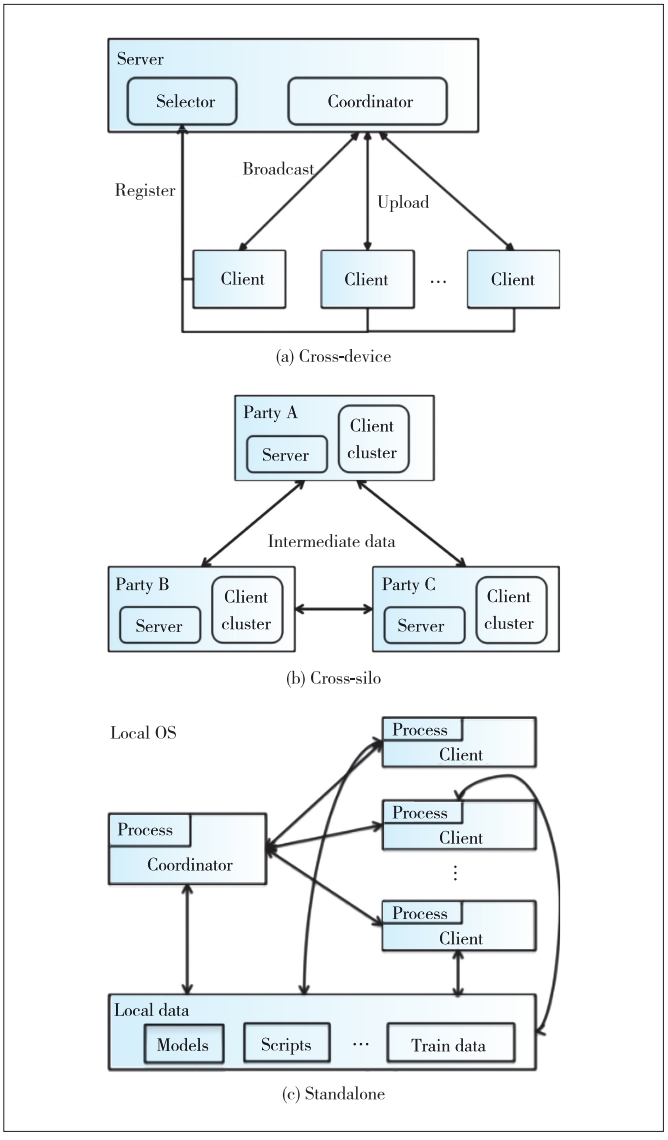
•Portability: There are a large number of existing center-based learning programs to be migrated to federated learning programs. Therefore, it is valuable to simplify the FL migration and even complete the migration with zero coding. To achieve this goal, Neursafe-FL has the following designs: 1) A minimalist unified API design is adopted; 2) On the basis of the unified API, it supports mainstream machine learning frameworks that support Tensorflow and Pytorch currently, and it also supports new frameworks by implementing the corresponding interfaces. In this way, it retains the programming style of the original framework, which significantly simplifies the program development of FL. Fig. 2 is an example of the FL migration of training on MNIST.

•Multi-running mode: Neursafe-FL supports standalone modes for research purposes. In this scenario, Neursafe-FL only deploys the server-side coordinator and one or more client processes on a single node for distributed simulation. For cross-silo and cross-device, we comprehensively consider the compatibility of two running modes in client management and scheduling design. In the cross-device mode, it enables clients to join in and log out of the system by registering and quitting, and provides a set of client selection algorithms with extension capabilities, which meets the requirements of training efficiency and model robustness[3]. There are fewer participants in the cross-silo mode, where clients can join the system by configuration, and can be selected by configuration or label matching. At the same time, each participant can deploy the server and client simultaneously, and the client also supports multi-task parallelism. Figs. 3(a), 3(b), and 3(c) are examples of the above running mode respectively.

•Extensibility: Federated learning is a fast-growing field, with new requirements and more advanced algorithms emerging. It requires the system with a high degree of flexibility, which may rise complexity in use. To trade off the flexibility and usability, we provide a user-friendly API for normal users, and an advanced interface for researchers to make further ex-



▲Figure 2. Example of model migration for MNIST



▲ **Figure 3. Multi-running mode**

tensions. In the second case, Neursafe-FL provides two approaches for the extension. 1) It extends the standardized algorithm interface and integrates it into the system as a part of the framework. 2) It extends through the callback interface for more customized requirements, but only one callback is validated at the same time. A detailed description of the extensibility is shown as follows.

• Extension of security algorithms: Users can implement new security and privacy algorithms through the form of libraries. To facilitate this form of extension, we provide the standardized interfaces such as "encrypt" and "decrypt", and decouple the security algorithms from federated learning processes. Users can integrate the security algorithm into the federated training process by extending the above standardized interface and referencing it in the configuration file. Based on the standard interface, the framework provides security algorithms such as differential privacy and secret sharing aggregation, and users can extend other security algorithms, homomorphic encryption, and secure multi-party computation as well.

• Extension of client selection algorithms: In a scenario with a large number of devices, selecting unreasonable training devices leads to resource mismatch, unfairness, and stragglers. Therefore, two expansion interfaces are provided for pre-selection and on-selection during the client selection process. Users can add filtering rules through the pre-selection interface, and prioritize qualified devices through the on-selection interface. For example, users can add trusted device matching rules through the pre-selected interface to filter out untrusted devices in the case of malicious parties. They can also use priority scores through an on-selection interface to select a more stable and reliable device based on the computing resources of the device, network status, and other information. We have provided rules for tag matching, resource matching, performance priority, and data priority. Users can add custom rules according to the above extension interface in two ways: by file and by webhook.

• Extension of aggregation algorithms: Aggregation algorithms have different effects in different scenarios. For example, the traditional FedAvg algorithm cannot face the challenges such as data heterogeneity and imbalance. Therefore, we provide two ways to extend the aggregation algorithm. 1) We inherit the base class of aggregator and integrate it directly into the framework; 2) Through the callback interface, we abstract the training process of federated learning into three steps: server-side broadcast, client-side reporting, and server-side aggregation, corresponding to the callback interfaces to broadcast custom data, aggregate updates on the server side, and process the final result. Users can implement the callback functions and validate them in the coordinator's configuration file. For example, we implement the SCAFFOLD aggregation algorithm with the second method to solve the problem in the non-IID scenario.

• Extension of machine learning framework: In order to be compatible with different machine learning frameworks such as Tensorflow, Pytorch, and Caffe, we encapsulate the framework and provide a unified standard interface for the upper layer. To support a new machine learning framework, users just need to complete the following adaptations through the standard interface: 1) adaptation of model operations such as loading, saving, and evaluation; 2) adaptation of weight operations, such as weight acquisition, weight assignment, and weight operations; 3) preprocessing operations on datasets for model evaluation and verification; 4) implementation of some security and privacy interfaces. An adapted ML framework can be integrated into the federated learning framework and run by the configuration parameters. The framework currently supports Tensorflow and Pytorch, and users can also extend the support of other machine learning frameworks according to the above steps.

4 Architecture of Neursafe-FL

The architecture of Neursafe-FL is shown in Fig. 4, and the cooperation between the components is shown in Fig. 5. Core components of Neursafe-FL are presented as follows.

•Infrastructure layer: Neursafe-FL has completed the adaptation at the infrastructure layer. On the server side, we deploy it on the Container as a Service (CaaS), which is Kubernetes by default. High reliability of FL core component services is guaranteed through the HA mechanism of CaaS. In the cross-device scenario, the client side supports two process modes: native OS process or containerized process. Containerized deployment improves system portability. In the cross-silo scenario, CaaS deployment is still used on the client side, which enables better parallel scheduling of tasks.

•Job scheduler: It is the core component of job management and scheduling. We schedule jobs according to job queuing and the current system resource status. The scheduling algorithm needs to consider the efficiency and fairness of the system resource when satisfying job requirements.

•Coordinator: It is dynamically created by the job scheduler for each job. It is responsible for the organization and coordination of federated training, including client selection, task dispatching, and model aggregation.

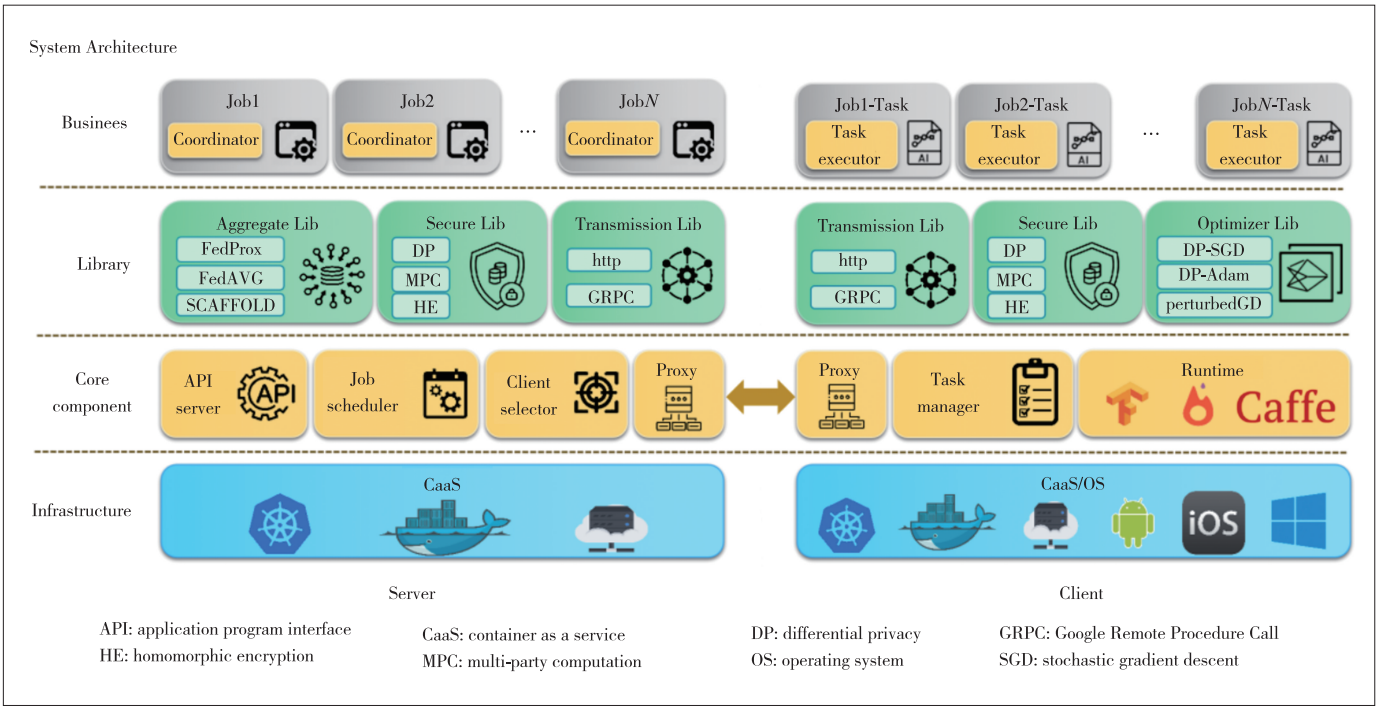
•Client selector: It is responsible for managing clients and responding to client selection requests. A client selector supports clients to join the system by registration or configuration. The client selection algorithm includes filtering and prioritization. The basic filters include whether there is a required dataset, whether it supports the specified machine learning framework (Runtime) or operating systems, etc., and whether it supports the extension of the filtering algorithm. The basic priorities include the number of data, computing power, parallelism, bandwidth, etc. The scalability of the client selection algorithm is expected to meet the needs of federated learning for client selection in terms of efficiency, robustness, and fairness[42 – 45].

• Task manager: It is the daemon component of the client. The main functions include the client's resources and status reporting, responding to the task issued by the server, and completing the task scheduling. In the cross-silo scenario, the client needs to execute tasks concurrently, and the task manager needs to schedule tasks according to the task queue and its local resource status.

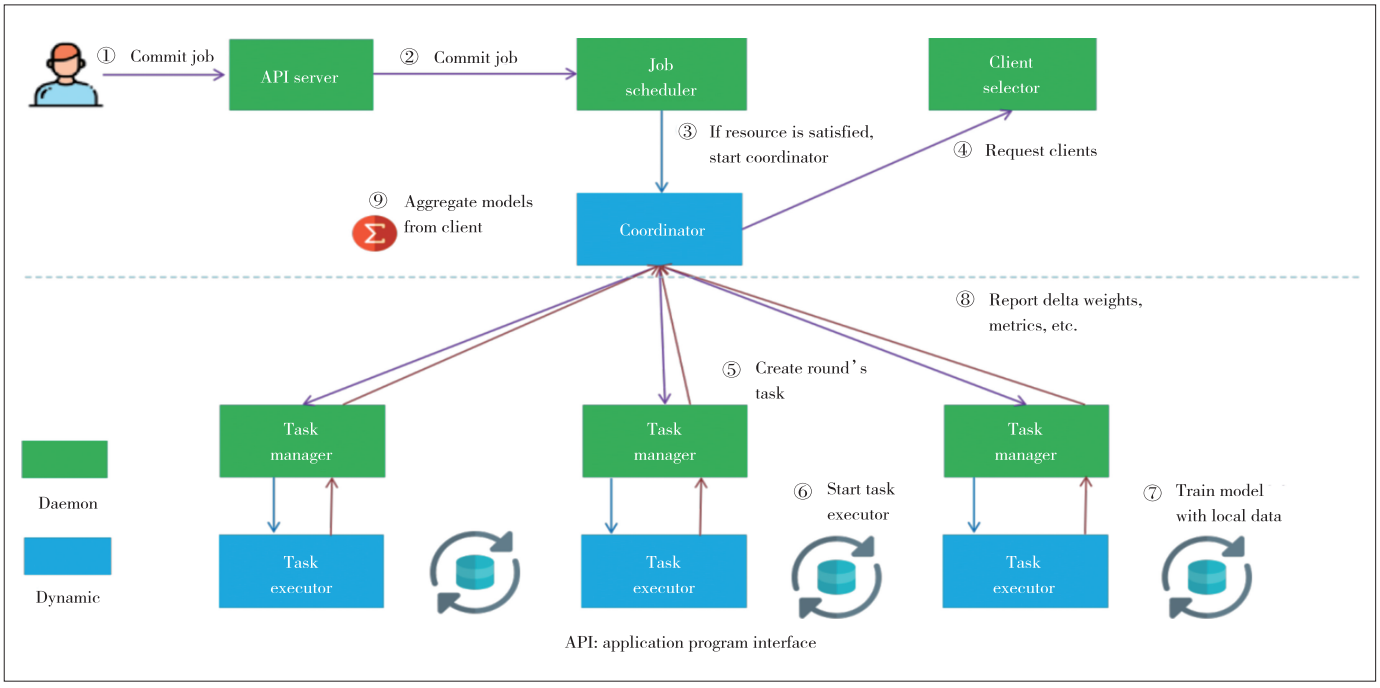
• Task executor: The client-side local executor of the federated task, which is dynamically created by the task manager when it receives tasks issued by the server. One task executor only manages one task to ensure the decoupling between tasks.

• Algorithms and basic libraries: Neursafe-FL encapsulates privacy security algorithms, federated optimization and aggregation algorithms, federated robustness algorithms, and low-level communications to simplify frameworks and application development. Users can extend the algorithm through the standardized algorithm interface, and benefit from the loose coupling between the algorithm and the framework.

Fig. 5 illustrates the interaction between Neursafe-FL core components in the job scheduling procedure as an example. 1) The user submits a job request to the job scheduler via the API server. 2) The job scheduler starts the coordinator when the system resources are satisfied. 3) The coordinator starts



▲**Figure 4. Architecture of Neursafe-FL**



▲**Figure 5. Interaction between Neursafe-FL core components**

the job execution process and requires clients to participate in federated training from the client selector. 4) The coordinator sends federated training tasks to clients. 5) After receiving the federated task, the client dynamically starts the local task executor. 6) The task executor uses local data to train the model. 7) Clients submit the model weight delta and the measurement generated by local training to the job coordinator. 8) The coordinator aggregates the client models to obtain a new global model. The coordinator decides to finish the federated training according to various criteria, such as model convergence and the max number of rounds reached.

5 Results of Neursafe-FL

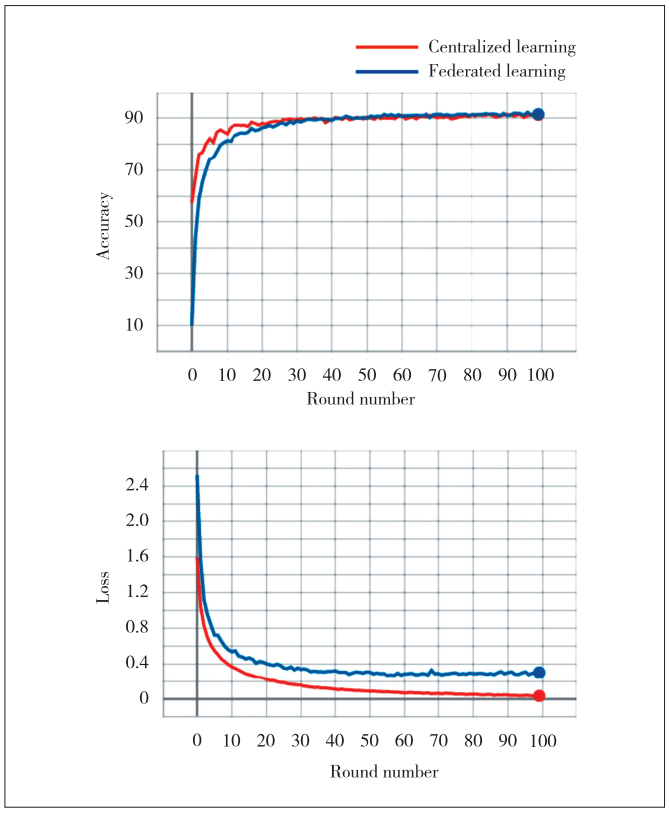
In order to evaluate the performance of Neursafe-FL, we design the following experiments: the comparison of convergence efficiency and accuracy of centralized learning and federated learning, the convergence comparison of federated training with different client numbers, the comparison of convergence efficiency of different federated aggregation algorithms under non-IID data, and the impact of privacy security algorithms on model accuracy and training efficiency. All experiments adopt a CNN model on two datasets: MNIST and CIFAR10.

The experiment in Fig. 6 is based on the CIFAR10 dataset. Three clients participate in federated training and the sample data for federated training are split according to the IID method. In order to compare the convergence efficiency, we set the client to perform only one epoch iteration per round. The experimental results in Fig. 6 show that the convergence efficiency and model accuracy of federated training and centralized training are almost the same under the IID data.

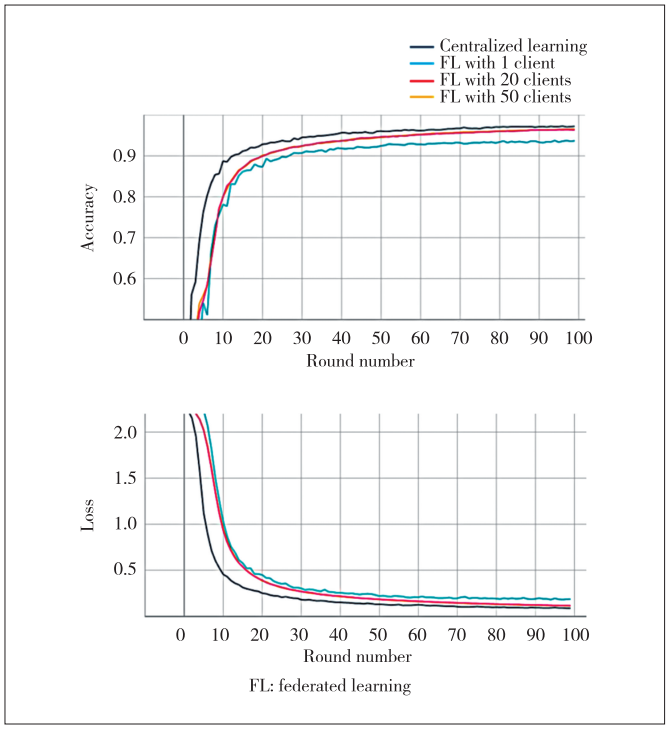
In Fig. 7, we compare the convergence performance of federated learning and centralized learning in terms of loss and accuracy. Data are evenly distributed to different clients by IID sampling. We test the scenario of federated learning with 1 client, 20 clients, and 50 clients. From the convergence curve of loss and accuracy in Fig. 7, the convergence of centralized learning is better than that of federated learning. On the other hand, the convergence performance of federated learning with multiple clients is significantly better than that of a single client, and it is close to the centralized learning. It can be seen that the convergence rate of federated learning and centralized learning are consistent. In addition, federated learning can ensure the privacy and security of clients.

Five federated clients participate in federated training, and the data are split in an extremely unbalanced manner, in which there are no samples with the same label among clients. Three federated aggregation algorithms, FedAvg, FedProx and SCAFFOLD, are used in the experiment. The results in Fig. 8 show that FedAvg is difficult to converge under this extreme data distribution, while FedProx and SCAFFOLD can converge with similar convergence efficiency. The results prove that it's necessary to select appropriate federated optimization and aggregation algorithms according to different data distributions to ensure the convergence efficiency of the model.

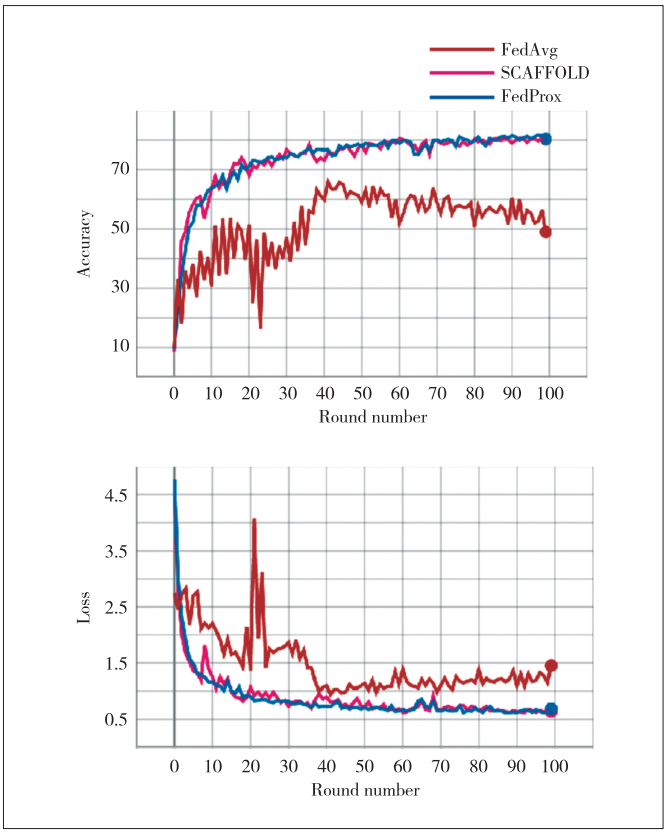
Two security algorithms, differential privacy and secret share aggregation (SSA) are tested in terms of overhead, model accuracy, and convergence efficiency. Both algorithm experiments are based on the MNIST and CIFAR10 datasets. The impact of differential privacy on the accuracy of a model is evaluated in the experiment. The results are shown in Fig. 9.



▲Figure 6. Comparison of federated training and centralized training



▲Figure 7. Convergence comparison of federated training with different client numbers



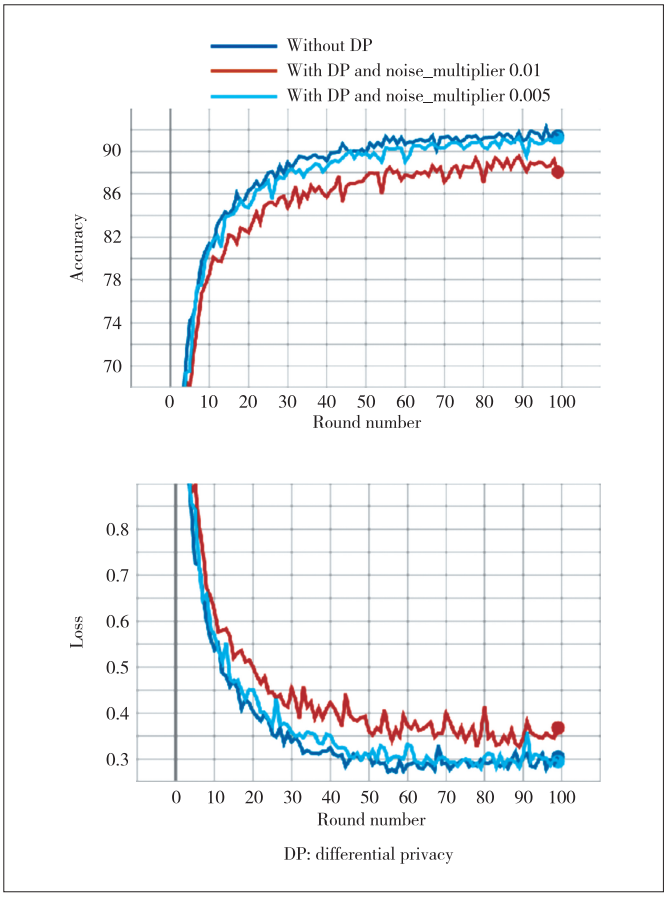
▲Figure 8. Results of different aggregation algorithms under non-IID data

After 100 rounds of training, the model accuracy with differential privacy is 0.912, while the accuracy of the model without differential privacy is 0.914. The results show that differential privacy has a limited impact on the model's accuracy. The overhead of differential privacy is shown in Table 2. Compared with the overall overhead of federated training, the overhead of differential privacy accounts for a small proportion. Neursafe-FL reduces the times of adding noise by adding corresponding noise to the updated weights of the client completing one round of model training. The budget of Neursafe-FL is much smaller than that of the method of adding noise during gradient update.

A comparative test of introducing SSA and not introducing SSA is conducted, and the results in Fig. 10 show that the curves of convergence and accuracy are completely consistent. The SSA algorithm based on the principle of cryptography is lossless. Therefore, the SSA-based algorithm has no effect on the convergence and accuracy of federated training.

The SSA algorithm is tested in the scenario where the client is disconnected. The results in Fig. 11 show that the convergence curve is almost the same as that in the normal scenario, and the SSA algorithm remains lossless. The SSA algorithm randomly selects disconnected clients every round, which only causes slight differences in the disconnected case.

The impact of the number of federated training clients on the SSA algorithm is studied in the experiment, and we analyze the



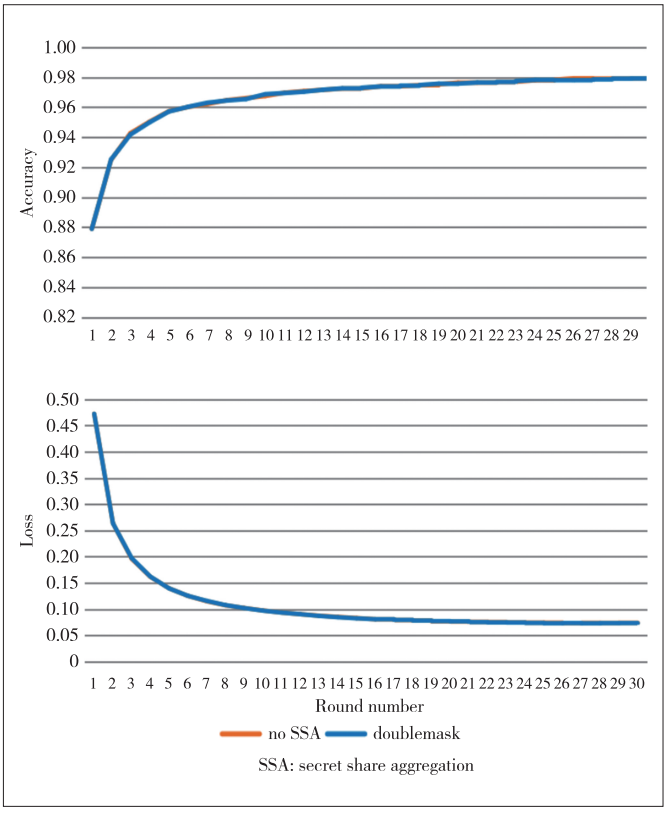
▲Figure 9. A comparative test of introducing DP and not introducing differential privacy

▼Table 2. Results of different parameters for differential privacy

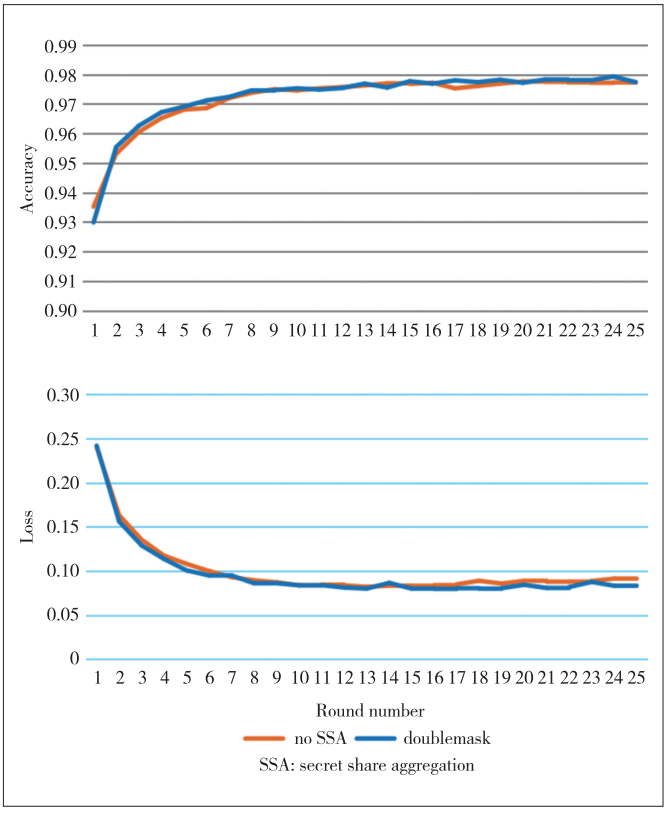
Type	Round Number	Noise_multiplier	Time Cost/s	Accuracy
Without DP	100	-	2 873.22	0.914 3
	50	-	1 396.56	0.898 4
With DP	100	0.01	2 879.36	0.912 2
	100	0.005	2 882.86	0.891 0
	50	0.01	1 390.42	0.876 3

DP: differential privacy

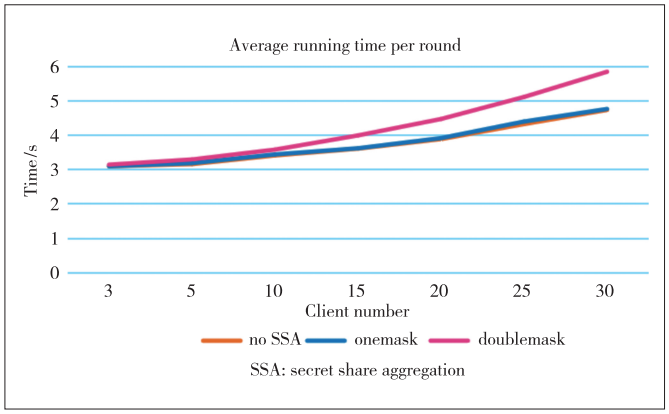
effect of clients' number on the overhead of federated training in the models without using the SSA algorithm, using the SSA onemask, and using the SSA doublemask respectively. The results are shown in Fig. 12. In the one-mask mode, the performance of SSA is basically the same as the performance when SSA is not used. The number of clients nearly has no impact on it, because each client only needs to send its own Diffie-Hellman (DH) public key to other clients under the one-mask model. This process is performed concurrently with model training, and the performance is not affected. In the double-mask mode, the performance of SSA increases with the number of clients. When the results are aggregated, additional communication is required to recover the mask. Therefore, the perfor-



▲Figure 10. A comparative test of federated training with SSA and without SSA



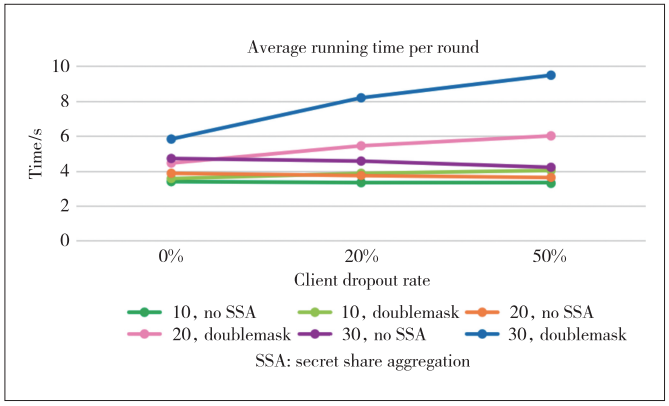
▲Figure 11. Impacts of SSA in the scenario where the clients are disconnected



▲**Figure 12. Impacts of client's number on the performance of SSA**

mance overhead will also increase with the increase in the number of clients.

The SSA algorithm is tested in the case of different drop rates of clients participating in the federated training, and we analyze the effect of different drop rates on the performance of federated training without using the SSA algorithm, using the SSA one-mask, and using the SSA double-mask respectively in Fig. 13. In the figure, "10, no SSA" denotes the training with 10 clients participating without using SSA algorithm, and the other abbreviations can be translated similarly. As shown in the figure, the overhead of using SSA increases substantially as the drop rate increases, and as the number of dropout clients increases, the overhead also increases. The overhead increases, because it requires the SSA algorithm to process the mask related to the disconnected clients, which leads to additional communication.



▲**Figure 13. Impacts of different dropout rates on the performance of federated training**

It can be seen that the two privacy security algorithms have their own advantages. Differential privacy introduces noise, which has an impact on the convergence of model training, but the impact is relatively small for the overhead. On the other hand, SSA can guarantee secure aggregation as a cryptographic algorithm without accessing the original data. This is a lossless algorithm, which ensures the accuracy and convergence of the model compared with the differential privacy, but it introduces a large overhead in calculation and transmission. Therefore, users can choose different security algorithms according to the requirements of the accuracy, calculation, and transmission for the applications.

6 Conclusions and Future Work

This paper introduces a new federated learning framework: Neursafe-FL, which focuses on the main challenges of federated learning, such as privacy, security, efficiency, and robustness, and on the usability to reduce the cost of model migration. In the design, we simplify the API, make it compatible with multiple mainstream machine learning frameworks, and enable framework extensions. All of these designs lower the threshold for federated learning. Through componentization, modulization, and standardization design, the scalability of the system is improved to meet the diverse needs of users. Experiments show that this framework has a good performance in terms of model convergence and accuracy under various settings.

In the future, we will focus on the following aspects: 1) Integrating more algorithms into Neursafe-FL to improve the efficiency, security, and robustness of federated learning. 2) Supporting vertical and transfer federation to meet the needs of different data distribution scenarios. 3) Enriching system features such as enabling more infrastructure and OS, and expanding support for more machine learning frameworks, federated inference, etc., to facilitate more FL applications.

References

[1] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C]//International Conference on Artificial Intelligence and Statistics. PMLR, 2017: 1273 – 1282
[2] YANG Q, LIU Y, CHEN T J, et al. Federated machine learning: concept and applications [EB/OL]. [2022-04-01]. https://arxiv.org/abs/1902.04885
[3] BONAWITZ K, EICHNER H, GRIESKAMP W, et al. Towards federated learning at scale: system design [EB/OL]. [2022-04-01]. https://arxiv. org/abs/1902.01046
[4] KAIROUZ E B P, MCMAHAN H B. Advances and open problems in federated learning [J]. Foundations and trends in machine learning, 2021, 14(1): 1 – 21. DOI: 10.1561/2200000083
[5] LI T, SAHU A K, TALWALKAR A, et al. Federated learning: challenges, methods, and future directions [J]. IEEE signal processing magazine, 2020, 37(3): 50 – 60. DOI: 10.1109/MSP.2020.2975749
[6] VOIGT P, VON DEM BUSSCHE A. The EU general data protection regulation (GDPR): a practical guide [M]. Cham: Springer International Publishing, 2017
[7] LONG G D, TAN Y, JIANG J, et al. Federated learning for open banking federated learning [J]. Federated Learning 2020: 240 – 254. DOI: 10.1007/978-3-030-63076-8_17
[8] XU J, GLICKSBERG B S, SU C, et al. Federated learning for healthcare informatics [J]. Journal of healthcare informatics research, 2021, 5(1): 1 – 19. DOI: 10.1007/s41666-020-00082-4

[9] JIANG J C, KANTARCI B, OKTUG S, et al. Federated learning in smart city sensing: challenges and opportunities [J]. Sensors, 2020, 20(21): 6230. DOI: 10.3390/s20216230
[10] LYU L, YU H, YANG Q. Threats to federated learning: a survey [EB/OL]. [2022-04-01]. https://deepai.org/publication/threats-to-federated-learning-a-survey
[11] BAGDASARYAN E, VEIT A, HUA Y, et al. How to backdoor federated learning [C]//International Conference on Artificial Intelligence and Statistics. PMLR, 2020: 2938 – 2948
[12] WANG Z B, SONG M K, ZHANG Z F, et al. Beyond inferring class representatives: user-level privacy leakage from federated learning [C]//IEEE Conference on Computer Communications. IEEE, 2019: 2512 – 2520. DOI: 10.1109/INFOCOM.2019.8737416
[13] NASR M, SHOKRI R, HOUMANSADR A. Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning [C]//IEEE Symposium on Security and Privacy. IEEE, 2019: 739 – 753. DOI: 10.1109/SP.2019.00065
[14] ZHAO Y, LI M, LAI L, et al. Federated learning with non-IID [EB/OL]. (2018-06-02) [2022-04-01]. https://arxiv.org/abs/1806.00582
[15] LI X, HUANG K, YANG W, et al. On the convergence of FedAvg on non-IID data [EB/OL]. (2020-06-25) [2022-04-01]. https://arxiv.org/abs/1907.02189
[16] ZHANG J L, CHEN J J, WU D, et al. Poisoning attack in federated learning using generative adversarial nets [C]//18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2019: 374 – 380. DOI: 10.1109/TrustCom/BigDataSE.2019.00057
[17] FANG M, CAO X, JIA J, et al. Local model poisoning attacks to byzantine-robust federated learning [C]//29th USENIX Security Symposium. USENIX, 2020: 1605 – 1622
[18] REISIZADEH A, TZIOTIS I, HASSANI H, et al. Straggler-resilient federated learning: leveraging the interplay between statistical accuracy and system heterogeneity [EB/OL]. [2022-04-01]. https://arxiv.org/abs/2012.14453
[19] DWORK C. Differential privacy: a survey of results [C]//International Conference on Theory and Applications of Models of Computation. TAMC, 2008: 1 – 19
[20] MCMAHAN H B, ANDREW G, ERLINGSSON U, et al. A general approach to adding differential privacy to iterative training procedures [EB/OL]. [2022-04-01]. https://arxiv.org/abs/1812.06210
[21] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy [C]//ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 308 – 318
[22] DWORK C, ROTH A. The algorithmic foundations of differential privacy [J]. Foundations and trends in theoretical computer science, 2014, 9(3 – 4): 211 – 407
[23] BU Z, DONG J, LONG Q, et al. Deep learning with gaussian differential privacy [J]. Harvard data science review, 2020, 23. DOI: 10.1162/99608f92.cfc5dd25
[24] GOLDREICH O. Secure multi-party computation [EB/OL]. [2022-04-01]. https://www.wisdom.weizmann.ac.il/~oded/pp.html
[25] DU W, ATALLAH M J. Secure multi-party computation problems and their applications: a review and open problems [C]//Proceedings of the 2001 Workshop on New Security Paradigms. ACM, 2001: 13 – 22
[26] MOHASSEL P, RINDAL P. ABY3: a mixed protocol framework for machine learning [C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM: 2018: 35 – 52. DOI: 10.1145/3243734.3243760
[27] CHEN V, PASTRO V, RAYKOVA M. Secure computation for machine learning with SPDZ [EB/OL]. [2022-04-01]. https://arxiv.org/abs/1901.00329
[28] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning [C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017. DOI: 10.1145/3133956.3133982
[29] GENTRY C. Fully homomorphic encryption using ideal lattices [C]//Proceedings of the 41st Annual ACM Symposium on Theory of Computing. ACM, 2009: 169 – 178. DOI: 10.1145/1536414.1536440
[30] GENTRY C. A fully homomorphic encryption scheme [M]. Stanford, USA: Stanford University, 2009
[31] ACAR A, AKSU H, ULUAGAC A S, et al. A survey on homomorphic encryption schemes [J]. ACM computing surveys, 2019, 51(4): 1 – 35. DOI: 10.1145/3214303
[32] ZHANG C, LI S, XIA J, et al. BatchCrypt: efficient homomorphic encryption for cross-silo federated learning [C]//2020 USENIX Annual Technical Conference. USENIX, 2020: 493 – 506
[33] FANG H K, QIAN Q. Privacy preserving machine learning with homomorphic encryption and federated learning [J]. Future Internet, 2021, 13(4): 94. DOI: 10.3390/fi13040094
[34] WANG J, LIU Q, LIANG H, et al. Tackling the objective inconsistency problem in heterogeneous federated optimization [J]. Advances in neural information processing systems, 2020, 33: 7611 – 7623
[35] LI T, SAHU A K, ZAHEER M, et al. Federated optimization in heterogeneous networks [EB/OL]. [2022-04-01]. https://arxiv.org/abs/1812.06127
[36] KARIMIREDDY S P, KALE S, MOHRI M, et al. SCAFFOLD: stochastic controlled averaging for on-device federated learning [EB/OL]. [2022-04-01]. https://arxiv.org/abs/1910.06378
[37] HE C Y, ANNAVARAM M, AVESTIMEHR S. Towards non-IID and invisible data with FedNAS: federated deep learning via neural architecture search [EB/OL]. [2022-04-01]. https://arxiv.org/abs/2004.08546
[38] BLANCHARD P, EL MHAMDI E M, GUERRAOUI R, et al. Machine learning with adversaries: Byzantine tolerant gradient descent [C]//The 31st International Conference on Neural Information Processing Systems. ACM, 2017
[39] REISIZADEH A, FARNIA F, PEDARSANI R, et al. Robust federated learning: the case of affine distribution shifts [J]. Advances in neural information processing systems, 2020, 33: 21554 – 21565
[40] PARK J, HAN D J, CHOI M, et al. Sageflow: robust federated learning against both stragglers and adversaries [J]. Advances in neural information processing systems, 2021, 34: 840 – 851
[41] PILLUTLA K, KAKADE S M, HARCHAOUI Z. Robust aggregation for federated learning [J]. IEEE transactions on signal processing, 2022, 70: 1142 – 1154. DOI: 10.1109/TSP.2022.3153135
[42] LI S, CHENG Y, WANG W, et al. Learning to detect malicious clients for robust federated learning [EB/OL]. [2022-04-01]. https://arxiv.org/abs/2002.00211
[43] SATTLER F, WIEDEMANN S, MULLER K R, et al. Robust and communication-efficient federated learning from non-IID data [J]. IEEE transactions on neural networks and learning systems, 2020, 31(9): 3400 – 3413. DOI: 10.1109/TNNLS.2019.2944481
[44] NISHIO T, YONETANI R. Client selection for federated learning with heterogeneous resources in mobile edge [C]//2019 IEEE International Conference on Communications. IEEE, 2019 : 1 – 7. DOI: 10.1109/ICC.2019.8761315
[45] GEYER R C, KLEIN T, NABI M. Differentially private federated learning: A client level perspective [EB/OL]. [2022-04-01]. https://arxiv. org/abs/1712.07557
[46] KANG J W, XIONG Z H, NIYATO D, et al. Incentive design for efficient federated learning in mobile networks: a contract theory approach [C]//2019 IEEE VTS Asia Pacific Wireless Communications Symposium. IEEE, 2019: 1 – 5. DOI: 10.1109/VTS-APWCS.2019.8851649
[47] CHEN W, HORVATH S, RICHTARIK P. Optimal client sampling for federated learning [EB/OL]. [2022-04-01]. https://arxiv.org/abs/2010.13723
[48] KONEČNÝ J, MCMAHAN H B, YU F X, et al. Federated learning: strategies for improving communication efficiency [EB/OL]. [2022-04-01]. https://arxiv.org/abs/1610.05492
[49] TRAN N H, BAO W, ZOMAYA A, et al. Federated learning over wireless networks: optimization model design and analysis [C]//IEEE Conference on Computer Communications. IEEE, 2019: 1387 – 1395. DOI: 10.1109/INFOCOM.2019.8737464
[50] GUHA N, TALWALKAR A, SMITH V. One-shot federated learning [EB/OL]. [2022-04-01]. https://arxiv.org/abs/1902.11175
[51] CHOI B, SOHN J Y, HAN D J, et al. Communication-computation efficient secure aggregation for federated learning [EB/OL]. [2022-04-01]. https://arxiv.org/abs/2012.05433
[52] DU W, ZENG X, YAN M, et al. Efficient federated learning via variational dropout [EB/OL]. [2022-04-01]. https://openreview.net/forum?id=BkeAf2CqY7
[53] REN J J, WANG H C, HOU T T, et al. Federated learning-based computation offloading optimization in edge computing-supported Internet of Things [J]. IEEE access, 7: 69194 – 69201. DOI: 10.1109/ACCESS.2019.2919736
[54] YU H, LIU Z L, LIU Y, et al. A fairness-aware incentive scheme for federated

learning [C]//Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. ACM, 2020: 393 – 399. DOI: 10.1145/3375627.3375840
[55] LI T, SANJABI M, SMITH V. Fair resource allocation in federated learning [EB/OL]. [2022-04-01]. https://arxiv.org/abs/1905.10497
[56] LYU L, XU X, WANG Q, et al. Collaborative fairness in federated learning [M]//Federated Learning. Cham: Springer, 2020: 189 – 204. DOI: 10.1007/978-3-030-63076-8_14
[57] INGERMAN A, OSTROWSKI K. TensorFlow federated [EB/OL]. [2022-04-01]. https://www.tensorflow.org/federated 2019
[58] LIU Y, FAN T, CHEN T, et al. FATE: an industrial grade platform for collaborative learning with data protection [J]. Journal of machine learning research, 2021, 22(226): 1 – 6
[59] ZILLER A, TRASK A, LOPARDO A, et al. Pysyft: a library for easy federated learning [M]//Federated learning systems. Cham: Springer, 2021: 111 – 139
[60] HE C Y, LI S Z, SO J, et al. FedML: a research library and benchmark for federated machine learning [EB/OL]. [2022-04-01]. https://arxiv.org/abs/2007.13518
[61] MA Y, YU D, WU T, et al. PaddlePaddle: an open-source deep learning platform from industrial practice [J]. Frontiers of data and computing, 2019, 1(1): 105 – 115

Biographies

TANG Bo received his master's degree from Northwestern Polytechnical University, China in 2005. Currently, as the system architect of ZTE Corporation, he is mainly responsible for federated learning and AI security solutions. His current research interests include container and container cloud, heterogeneous resource scheduling, and AI security and trustworthy AI. He holds several patents in the above research areas.

ZHANG Chengming received his BS degree from Yangzhou University, China in 2011, and ME degree from Nanjing University of Posts and Telecommunications, China in 2015. He is a senior R&D engineer of ZTE Corporation. His current research interests include AI platform, container cloud, resource scheduling, federated learning, and AI security.

WANG Kewen received his BS degree from China University of Mining and Technology, China in 2015, and the ME degree from Beijing Institute of Technology, China in 2017. He is an AI senior algorithm engineer of ZTE Corporation. His current research interests include AI Systems and AI security, such as federated learning, adversarial attack, and defense in deep learning.

GAO Zhengguang received his PhD degree from Beijing University of Posts and Telecommunications, China in 2020. He was a visiting PhD student in High Performance Networks Group, University of Bristol, UK from 2018 to 2019. After graduation, he was selected for "LAN JIAN" program of ZTE Corporation as an algorithm researcher. His current research interests include 5G/6G communication technologies, mobile networks, and machine learning for future communications.

HAN Bingtao (han.bingtao@zte.com.cn) received his BS degree from Tianjin University, China in 2001, and MS degree from Nankai University, China in 2004. He is the deputy director of the State Key Laboratory of Mobile Network and Mobile Multimedia Technology, and the leader for "Adlik" project of the LF AI & Data Foundation. Currently, he is the AI system architect of Central R&D Institute, ZTE Corporation. His current research interests include deep learning algorithms, AI systems, and network intelligence. He is the author and co-author for numbers of patents and related monographs.

Toward Low-Cost Flexible Intelligent OAM in Optical Fiber Communication Networks

YAN Baoluo, WU Qiong, SHI Hu, ZHAO Yan, JIA Yinqiu, FENG Zhenhua, CHEN Weizhang, ZHU Mo, ZHAO Zhiyong, FANG Yu, CHEN Yong
(WDM System Department of Wireline Product R&D Institute, ZTE Corporation, Beijing 100029, China)

DOI: 10.12142/ZTECOM.202203007
https://kns.cnki.net/kcms/detail/34.1294.TN.20220728.1550.004.html, published online July 28, 2022
Manuscript received: 2022-01-10

Abstract: Low-cost, flexible and intelligent optical performance monitoring and management is a key enabling technology for network quality guarantee, especially in the era of explosive growth of communication capacity and network scale. However, to the best of our knowledge, it is extremely challenging to implement real-time performance monitoring and operations, administration and maintenance (OAM) in a highly complex dynamic network. In this paper, we propose an innovative optical identification (OID) scheme that can realize both performance monitoring and some advanced OAM sub-functions. The basic concepts, applications, challenges and evolution directions of this OID tool are also discussed.

Keywords: fiber optics communication; optical performance monitoring; pilot tone; wavelength-division multiplexing (WDM)

Citation (IEEE Format): B. L. Yan, Q. Wu, H. Shi, et al., "Toward low-cost flexible intelligent OAM in optical fiber communication networks," *ZTE Communications*, vol. 20, no. 3, pp. 54 - 60, Sept. 2022. doi: 10.12142/ZTECOM.202203007.

1 Introduction

According to Omdia's forecast[1], the revenue of the global optical network market will exceed $17.4 billion by 2025. Driven by the vigorous development of information services, such as 5G, 4K and virtual reality (VR), the optical network served as a data bearer network is evolving from a rigid and homogeneous one to a flexible and heterogeneous one. Predictably, the five aspects of demands for optical transmission networks will be ultra-large capacity, ultra-low transmission delay, flexible service access, intelligent operations, administration and maintenance (OAM), and high reliability. Among them, it is of great importance to achieve refined network management and control so that network performance can be guaranteed, especially for diverse service requirements in vertical industries.

However, the current optical transport network (OTN) lacks mature OAM techniques for the optical layer just like the electrical layer. With continuous system upgrade and network expansion, the backbone network has increased its demand for intelligent OAM in the optical layer, especially for optical channel layer (OCh) performance monitoring, fault location and service scheduling. Moreover, current networks excessively rely on cumbersome manual data collection and complex analysis methods, which is difficult to adapt to the development trend of intelligence.

In 1993, HILL et al. firstly proposed the basic concepts of adding kHz-level pilot tones (PTs) to the high-speed optical signal for signal identification, power optimization and fault management[2]. In 1996, Bell Labs and Lucent Technologies used frequency-shift keying-modulated kHz-level frequency PTs with a date rate of 100 bit/s to achieve end-to-end signal tracking and real-time performance monitoring, fault location, rerouting, and wavelength conversion[3]. Furthermore, Tropic networks Inc. used the fast Fourier transform (FFT) technique to identify PTs with different carrier frequencies, laying the foundation for multi-carrier PT techniques in wavelength-division multiplexing (WDM) networks in 2006[4]. Additionally, the PT technique also shows unique advantages in optical network operations. For example, Lucent realized optical path tracing[5] and topology discovery (TD)[6] using PT signals around 2010. The above is the prototype of the early OCh OAM related to PT techniques, and its common feature is that the PT frequency is less than 1 MHz.

In recent years, PT techniques have been extensively investigated in the large-scale dynamic WDM network for perfor-

This work was supported in part by the National Key R&D Program of China under Grant No. 2019YFB2205302.

mance monitoring, such as optical power, chromatic dispersion (CD)[7], polarization-mode dispersion (PMD)[8], optical signal-to-noise ratio (OSNR)[9], and nonlinear noise[10]. However, with the expansion of network scale and the increase of transmission distance and capacity, the influence of CD accumulation and nonlinear effect on the system becomes much more obvious. In the WDM system loaded with multiple PTs, the crosstalk induced by stimulated Raman scattering (SRS) effect after long fiber transmission becomes more serious. Fortunately, higher frequency PT signals possess the smoothing properties to the CD walk-off effect on the SRS, thus it can significantly suppress the SRS interference[11]. Therefore, the carrier frequency of PTs has gradually transitioned from a low frequency (<1 MHz) to a higher frequency larger than 10 MHz for optical network monitoring and OAM application in recent years[12]. Overall, there still remain some challenges that require further investigation in the deployment of real-time and intelligent OAM in the optical layer.

In this paper, we develop and demonstrate an OCh monitoring tool based on the PT technique, named optical identification (OID), which can achieve real-time channel-level performance monitoring and signaling issuance over an entire optical network. The modulation frequency and amplitude of the PT signals we used are much lower than those of service optical signals, avoiding performance degradation of the service signal. The OID can be applied into two scenarios. One is real-time optical performance monitoring and management, such as channel power, OSNR and channel loss of signal (LOS) alarms. The other is channel-associated overhead management; for example, the OID can be used to carry service labels and path information, and can even carry a small amount of channel signaling overhead, which is the key to realize intelligent OAM in the optical layer.

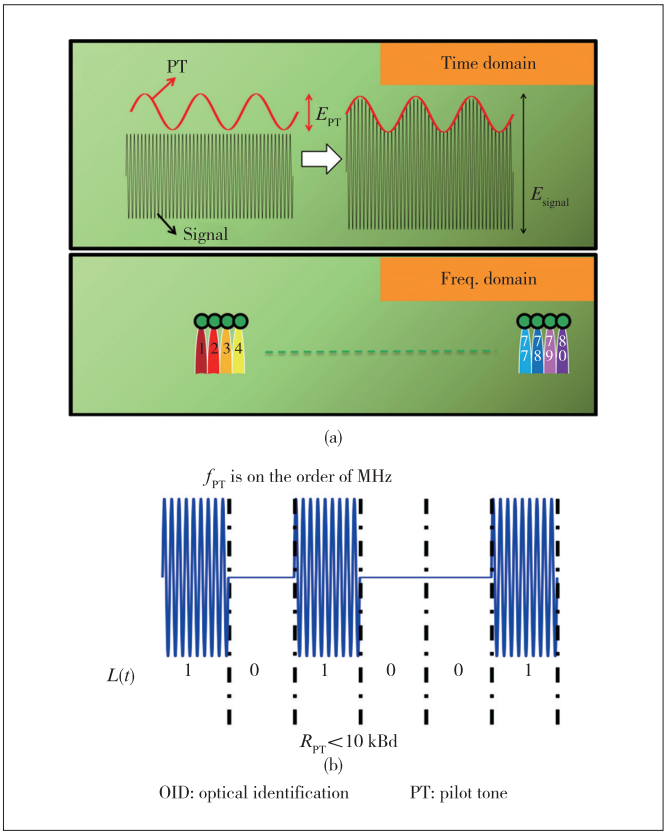
2 Basic Concept of Optical Identification Technique

2.1 Operating Principle

As shown in Fig. 1(a), for a single-channel OID signal, the electric field $E_{\mathrm{PT}}(t)$ expression of the low-speed PT modulated signal can be written as:

$$E_{\mathrm{PT}}(t) = E_{\mathrm{signal}}(t)\sqrt{1 + 2L(t)m_d\cos(2\pi f_{\mathrm{PT}}t + \varphi)}, \quad (1)$$

where $E_{\mathrm{signal}}(t)$ is the electric field of a high-speed optical signal, $L(t)$=0, 1,⋯ is the data stream of the OID in the form of on-off keying (OOK) modulation, f_{PT} is the PT frequency that is generally selected to be greater than 10 MHz. Owing to the degrading effect of subcarrier-signal beating interference (SSBI) on the signal-to-noise ratio (SNR), the OID date rate R_{PT} should not be high, which is recommended to be designed as lower than 10 kBd; m_d is the modulation index (MI) of the OID, which is defined as the ratio of the RF power of PT signals to the DC power of service optical signals. Actually, the PT signal can be regarded as a kind of low frequency noise for the service optical signal, so its amplitude or MI should be limited to control the OSNR penalty to be less than 0.5 dB@FEC limit. As a conservative design, its MI is recommended to be controlled below 10%. In the case of multi-channel monitoring, the electric field of OIDs can be derived from Eq. (1):

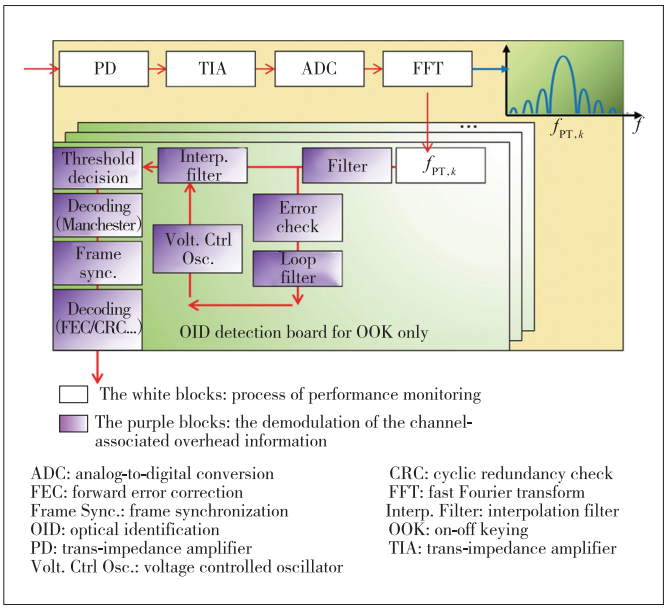


▲ **Figure 1. Schematic diagram of (a) OID loading and (b) channel-associated overhead data**

$$E_{\mathrm{PT}}(t) = \sum_k E_k(t)\sqrt{1 + 2L_k(t)m_{d,k}\cos(2\pi f_{\mathrm{PT},k}t + \varphi_k)}$$
$$(k = 1,2,3\cdots), \quad (2)$$

where k is the WDM channel number. As shown in Fig. 1(b), the channel-associated overhead data use the OOK modulation format with a specific frame structure. It is worth mentioning that Manchester coding possesses direct-current balance property, so that OIDs may carry the channel-associated overhead data while ensuring its average power is stable. Moreover, the error correction coding, such as forward error correction (FEC) and cyclic redundancy check (CRC), can improve the reliability of the channel-associated overhead data carried by OIDs. The above coding techniques are all adopted in the proposed OID tool.

As shown in Fig. 2, the basic process of OID detection for performance monitoring includes photodetection (PD), amplification by trans-impedance amplifier (TIA), analog-to-digital



▲ **Figure 2. OID detection process, taking OOK modulation as an example**

conversion (ADC) sampling, and FFT operation for extracting the voltage amplitude corresponding to $f_{PT,k}$. Through the derivation of the above process, the single-channel optical power P_k (dBm) and the electrical power of OID $P_{PT,k}$ (dBm) will follow a linear rule, which is the basic mathematical form of using OID for channel monitoring. Furthermore, to demodulate the channel-associated data overhead carried by OID, the following additional process steps are essential: smoothing filtering, clock recovery based on phase-locked loop using the Gardner algorithm, threshold decision, Manchester decoding, frame synchronization, and error correction coding. The purple module is implemented through field programmable gate array (FPGA) in the verification stage.

2.2 Potential Problems

As described in Table 1, there are some factors that limit the measurement accuracy of OID tools, according to our theoretical model and experimental verification. The practical applications are the permutation and combination of the basic scenarios shown in Table 1. Although the complexity has increased, the error analysis method is universal. In addition, the parameters that mainly contribute to the error sources are also listed. It is worth noting that the SRS effect decreases as the OID frequency f_{PT} increases, while the CD fading behaves on the contrary, so there is a tradeoff in the design of the carrier frequency f_{PT}. The error polarities of OCh power monitoring induced by SRS in different bands are inconsistent, which come from the direction of power transfer caused by SRS.

In the case of channel-associated overhead, we are more concerned about signal crosstalk caused by SRS. The G.655 fiber has small dispersion and effective area A_{eff}, so the Raman gain (nonlinear effect) is larger, which leads to larger SRS and crosstalk over different channels.

▼ **Table 1. Influencing factors on optical performance monitoring using OIDs**

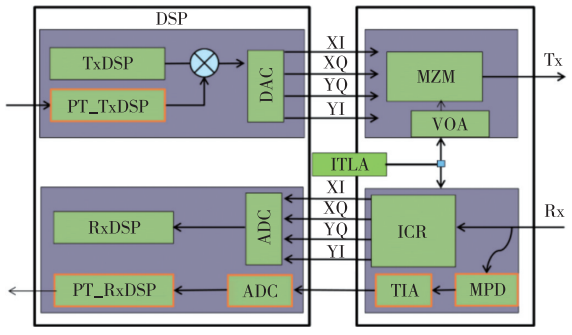
Error Source	Related Application	Error Polarity	Main Contribution
SSBI	Large capacity, multi-wave	+	m_d: Modulation index N: Channel number B_L: Bandwidth of PT signal B_S: Bandwidth of optical signal
ASE	Long haul, multi-span	−	OSNR
CD fading	Long haul, high baud rate, fiber with large D	−	f_{PT}: PT frequency $\Delta\lambda_{eff}$: Effective spectral width of optical signal D: Dispersion coefficient L: Transmission distance
SRS	Long haul, fiber with small D	Longwave: + Mid-wave: relatively small value Shortwave: −	f_{PT}: PT frequency g_{12}: Raman gain coefficient A_{eff}: Fiber effective area L: Transmission distance

Notes: + indicates the measured value of OID is greater than the actual value, while − means the opposite results.
ASE: amplifier spontaneousemission noise
CD: chromatic dispersion
OID: optical identification
OSNR: optical signal-to-noise ratio
PT: pilot tone
SRS: stimulated Raman scattering
SSBI: subcarrier-signal beating interference

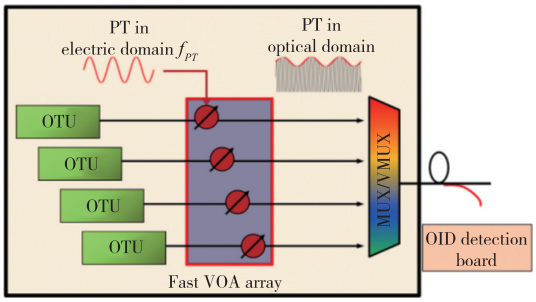
3 OID Loading Scheme and Its Features

The current mainstream OID loading tools can be divided into built-in digital signal process (DSP) chips[11] and photonic chips. Fig. 3(a) exhibits the built-in DSP scheme, that is, the digital modulation signal of OID is directly multiplied by the framed XI, XQ, YI, and YQ high-speed digital signals to realize OID loading. Fig. 3(b) shows the photonic chip loading scheme. The variable optical attenuator (VOA) array is driven by the OID analog signal to load the low-frequency PT on the service optical signal. Actually, the latter scheme can also be realized through built-in VOA with silicon photonics (SiP)-integrated transceiver chips or through a semiconductor optical amplifier (SOA) with indium phosphide (InP)-integrated ones. In order to make OID compatible with various future optical modules, the verification of OID loading ability on InP-integrated and SiP-integrated photonic chips is essential. Based on our research, both VOA and SOA can be driven by the OID carrier frequency up to 50 MHz for SiP- and InP-based coherent optical modules, respectively, so the frequency resources are abundant. Additionally, we have noticed other reports about tuning laser current driving amplitude to achieve PT loading. However, the potential impact of this method on service signals[13-14] remains the main concern before its practical application.

The performance difference of the above two mainstream schemes is as follows. The DSP-based method saves the cost of additional devices thanks to its higher integration as OID loading and detection are realized by the built-in DSP. Moreover, its advantages on integration will be more prominent in future optical modules with limited space and power consumption. Additionally, with the help of FFT tools, an entire spectrum can be divided into multiple bands in the digital domain of DSP scheme, and PTs



(a)



(b)

ADC: analog-to-digital converter
DAC: digital-to-analog converter
DSP: digital signal process
ICR: integrated coherent receivers
ITLA: integrable tunable laser assembly
MPD: monitor photodiode
MUX: multiplexer
MZM: Mach-Zehnder interferometer electro-optic modulator
OID: optical identification
OTU: optical transform unit
PT: pilot tone
Rx: receive
TIA: trans-impedance amplifier
Tx: transport
VMUX: variable optical attenuator plus multiplexer
VOA: variable optical attenuator

▲**Figure 3. (a) Built-in DSP chips and (b) an example of photonic chip loading scheme by using external VOA array**

are loaded for each frequency band, where each PT has a different phase. It is reported that the CD fading can be suppressed under the above configuration[11]. However, it also has its shortcomings. The first one is that the OID signal and service signal are coupled in the DSP solution, so the sampling rates mismatch between the two and the quantization noise needs to be resolved, which may cause signal distortion. The second is that not all vendors are willing to develop an OID function built-in DSP application specific integrated circuit (ASIC), as it requires more design and verification effort and increases risk of chip failure. On the contrary, the photonic chip loading scheme is decoupled from the service signal, which provides better flexibility. It can also be compatible with the existing networks in the form of a sub-card, which facilitates a smooth upgrade of the OID tool.

4 Application Overview

The demands for intelligent OAM in OCh of the current networks focus on three aspects:

1) Real-time: To query the channel power and OSNR, optical performance monitoring (OPM)-type boards are widely deployed in optical terminal multiplexer (OTM) or reconfigurable optical add/drop multiplexer (ROADM) stations. However, the total measurement time is always at the minute-level, especially in multi-dimensional ROADM systems, as it usually shares an OPM among multiple directions using a time division multiplexing mechanism.

2) Refined: Optical line amplifier (OLA) sites are not equipped with OPM, so it can only monitor the total power without channel power details. Moreover, the channel power inside the ROADM site can hardly be obtained even with OPM.

3) Low-cost: The deployment of large-scale OPM boards increases network costs and occupies more sub-rack space.

Thanks to the basic properties of the OID tools, the above issues can be fully solved.

4.1 OCh Performance Monitoring and Management: Real-Time and Low-Cost

As illustrated in Fig. 4, benefitting from the shorter response time, the real-time performance of OID is better than that of traditional OPM. OID can be integrated into the board due to its small size, which can be deployed in more diverse boards. Each board equipped with an OID detection point has the channel-level monitoring capabilities and there is no need to share a detection point in multiple directions through optical switch multiplexing like traditional OPM. The resource conflict is resolved, so the parallelism could be improved. Furthermore, the power flatness can be adjusted according to all the channel-power sensed by the OID through the detection point. The following is an overview of the application of OID.

1) Power monitoring and optimization: As mentioned before, OPM boards are rarely deployed at OLA sites, because the flatness degradation induced by single-span transmission is relatively small in C band and the power flatness issue is not urgent. However, it should not be ignored in C++ and C+L band scenarios. Including the OLA site in the scope of OID tool deployment provides an efficient solution to the above issues. Our experiment results indicate that the error of OCh power monitoring is less than 1.5 dB under the 80-ch 20-span transmission scenario. We should also pay attention to whether ASE-shaped channels are used in the C+L band scene, as it brings some difficulties to OID loading.

2) OSNR estimation: OSNR of an optical amplifier (OA) can be calculated through OID channel-power monitoring combined with noise figure (NF) model[15] calibration, gain spectrum model[16] calibration and SRS power transfer model[17], which can achieve real-time dynamic monitoring of OSNR over the optical path. Furthermore, the detection range has also been expanded from ROADM and/or OTM sites to any site or even any detection point within a site. The experiment

results prove that the OSNR estimation accuracy of our OID scheme is within 1.5 dB and suitable for most OA types, which could stably replace the existing OPM boards and obtain high integration and low-cost benefits.

3) Channel LOS alarm: LOS alarm based on the OID tool can achieve ms-level response, which is vital for fault location, protection and restoration[18]. The OCh power will replace the total power as the analysis criterion. Such applications as insertion loss and connection loss detection can provide important support for real-time monitoring of network health.

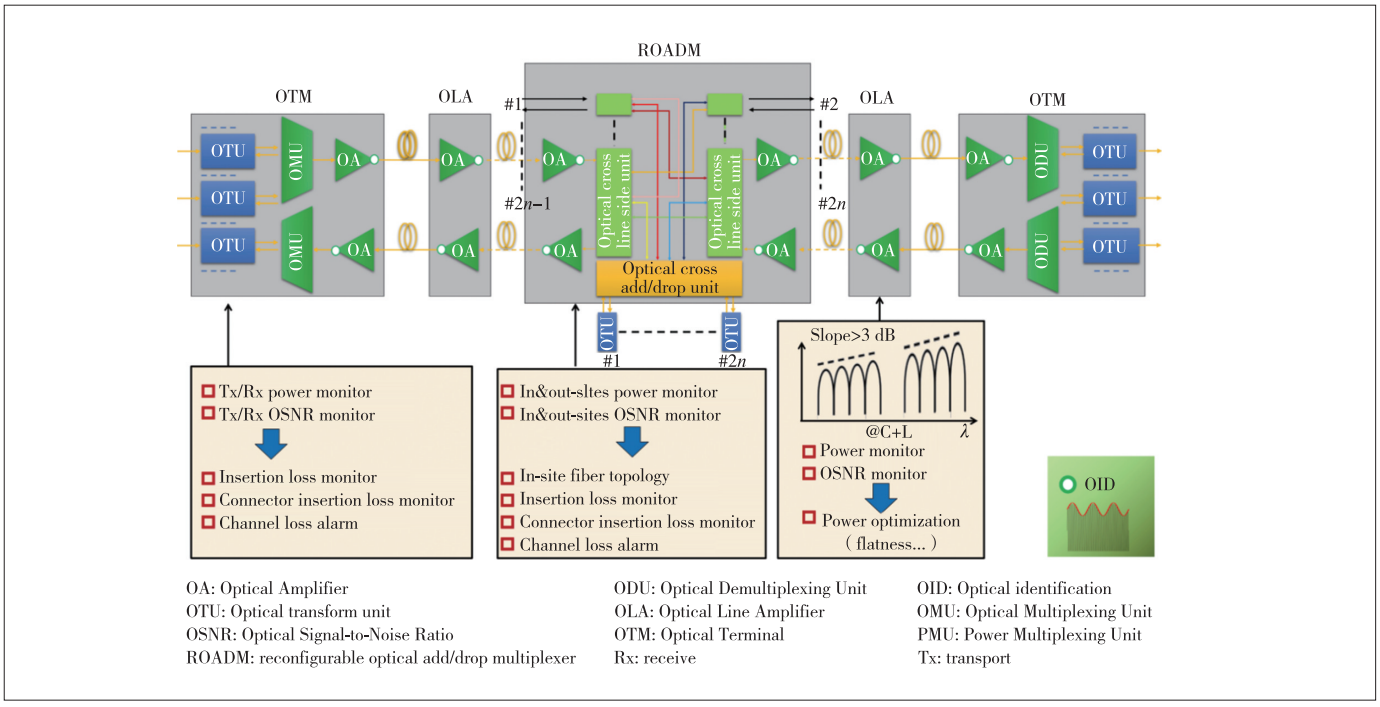
4.2 Optical Layer Channel-Associated Overhead Management: Refined and Intelligent

As shown in Fig. 5, all wavelengths are marked with unique channel IDs and transmitted together with the main optical signal. All the channel IDs can be extracted in each site equipped with the OID detection board. Simplified OAM such as TD in capacity expansion, new service creation/removal, rerouting, service path tracking and fiber misconnection recognition, can be realized by integrating the message of all OID detection points over the entire network. In the past, the topology connection within the site cannot be obtained owing to the limited coverage of the optical supervisory channel (OSC), while the OID tool can just make up for this shortcoming. In the case of TD between ROADM and OTM sites, it is determined whether each network node is in the same topology based on the detection of the same designated channel identifier. Meanwhile, the order of each network node in the topology can be judged by detecting the sequence of the initial time information. For the TD within a site, by tuning the channel attenuation of wavelength selective switch (WSS)-type boards, the OCh power in different sites should be changed accordingly. Once the change occurs, it means that the site is behind the WSS-type board, and thus the complete topological connection sequence within the site can be determined. Based on the OID unique channel identification of the entire network, the practical transmission path of the service wavelength can be realized, which solves another shortcoming of traditional OPM, that is, it can only distinguish wavelengths from the spectrum domain and cannot distinguish different services of the same wavelength. With the compression of channel spacing, it is even difficult to distinguish effective wavelength channels through spectral detection technique.

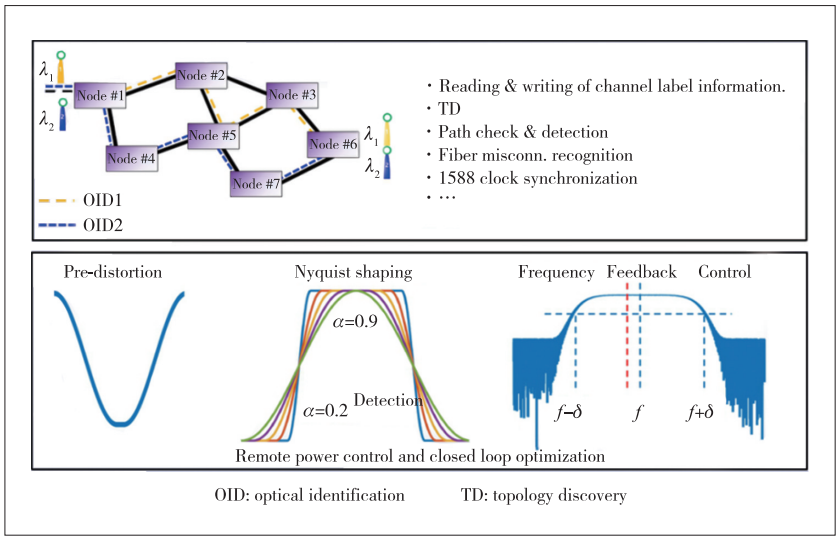
If the planned path is known, the service path check can be achieved by querying whether all the nodes on the planned path have detected the expected identification overhead. The service path check is a simplified version of the service path detection that adds an OCh power verification stage to further improve the accuracy of path detection and fiber connection judgment.

The OID channel can also be used to achieve 1588 clock synchronization. The 1588 clock synchronization system has large asymmetric delay through the optical module DSP, while the DSP for OID is relatively simpler and the delay for 1588 clock is fixed, which can ensure stable results.

For remote power control and closed-loop optimization, the channel-associated overhead could also include performance and remote control signaling information. Especially, the performance of the receiver is fed back to the transmitter in real



▲ **Figure 4. Schematic diagram of optical channel layer (OCh) performance monitoring and management based on OID tool**



▲ **Figure 5. Schematic diagram of channel-associated overhead management based on OID tool**

time through the reverse channel to form a closed loop mechanism, and then the best system configuration can be selected. With the closed-loop mechanism, it is possible to realize automatic optimization and opening up of new services or rerouting, as well as real-time performance optimization. For example, we can optimize the signal spectrum through predistortion[19], Nyquist shaping[20], central wavelength adjustment[21] and so on, and then select the best configuration based on the *Q* value of the opposite end in real-time feedback loop.

5 Conclusions and Outlook

In this paper, the basic concepts, applications and challenges of OID for optical layer intelligent OAM are discussed. We propose a low-cost OID implementation scheme and verify its applications in both optical performance monitoring and optical layer channel-associated overhead management. The proposed OID tool is proved to be effective and helpful for intelligent OAM in optical networks. However, there still remain some challenges that require further investigation: how to gradually upgrade and replace current network monitoring equipment such as OPM boards, and to make a tradeoff among the various negative factors discussed in the paper. In the future, this scheme is expected to evolve mainly towards two directions:

1) Faster optical layer sensing network needs to be constructed. The OID tool integrated on the board offers a sensitive detection way for real-time performance monitoring. The highly integrated feature facilitates its wide deployment in optical networks. With the full deployment of optical layer sensing networks based on this OID tool, the perception of OCh performance will become ubiquitous, and the collection of massive data should provide more real-time reliability for the optical layer intelligent engine basis for decision-making.

2) A more complete optical layer overhead system should be constructed. The OID tool provides a transmission channel for the overhead data in the optical layer, opening up new prospects for the improvement of the optical layer overhead system with diverse OAM functions.

It is expected that the wide deployment and application of OID tools will elevate the operation and maintenance capabilities of optical networks to a new level.

References

[1] FREY D, REDPATH I. Optical networks forecast spreadsheet: 2020 – 25 [EB/OL]. (2020-07-24)[2021-12-29]. https://omdia. tech. informa. com/OM011452/Optical-Networks-Forecast-Spreadsheet-202025
[2] HILL G R, CHIDGEY P J, KAUFHOLD F, et al. A transport network layer based on optical network elements [J]. Journal of lightwave technology, 1993, 11(5/6): 667 – 679. DOI: 10.1109/50.233232
[3] HEISMANN F, FATEHI M T, KOROTKY S K, et al. Signal tracking and performance monitoring in multi-wavelength optical networks [C]//Proceedings of European Conference on Optical Communication. IEEE, 1996: 47 – 50
[4] WAN P W, REMEDIOS D, JIN D, et al. Channel identification in communications networks: US7054556 B2 [P]. 2006
[5] SEDDIGH N, NANDY B, OBEDA P D, et al. Method and system for light path monitoring in an optical communication network: CA2451888 A1 [P]. 2009
[6] NANDY B, SEDDIGH N, ANSARI S, et al. Method and system for network topology discovery: CA2669435 A1 [P]. 2012
[7] PARK P K J, JUN S B, CHUNG Y C. Chromatic dispersion monitoring technique based on chirped pilot tones [J]. Optics communications, 2006, 266(1): 280 – 283. DOI: 10.1016/j.optcom.2006.04.080
[8] JI H C, PARK K J, LEE J H, et al. Optical performance monitoring techniques based on pilot tones for WDM network applications [J]. Journal of optical networking, 2004, 3(7): 510. DOI: 10.1364/jon.3.000510
[9] LI W J, LU X, YE Z Y, et al. Pilot aided OSNR monitoring in optical Nyquist transmission system [C]//The tenth international conference on information optics and photonics. Tsinghua University and Chinese Laser Press, 2018, 10964: 1219 – 1222. DOI: 10.1117/12.2506451
[10] JIANG Z P, TANG X F. Nonlinear noise monitoring in coherent systems using amplitude modulation pilot tone and zero-power gap [C]//Proceedings of Optical Fiber Communication Conference (OFC). OSA, 2019. DOI: 10.1364/ofc.2019.th2a.34
[11] JIANG Z P, TANG X F. Multiband pilot tone based optical performance monitoring and its application in mitigating chromatic dispersion fading [C]//Proceedings of Asia Communications and Photonics Conference/International Conference on Information Photonics and Optical Communications 2020 (ACP/IPOC). OSA, 2020. DOI: 10.1364/acpc.2020.t3c.4
[12] DU J H, YANG T, SHI S P, et al. Optical label-enabled low-cost DWDM optical network performance monitoring using novel DSP processing [C]//Proceedings of Asia Communications and Photonics Conference/International Conference on Information Photonics and Optical Communications 2020 (ACP/IPOC). OSA, 2020. DOI: 10.1364/acpc.2020.m4a.296
[13] ROPPELT M, LAWIN M, EISELT M. Experimental demonstration of a new pilot tone generation method [C]//Proceedings of Optical Fiber Communication Conference/National Fiber Optic Engineers Conference. OSA, 2013: 1 – 3. DOI: 10.1364/nfoec.2013.jw2a.72
[14] JUN S B, KIM H, PARK P K J, et al. Pilot-tone-based WDM monitoring technique for DPSK systems [J]. IEEE photonics technology letters, 2006, 18(20): 2171 – 2173. DOI: 10.1109/LPT.2006.884237
[15] ZHANG X P, MITCHELL A. A simple black box model for erbium-doped fiber amplifiers [J]. IEEE photonics technology letters, 2000, 12(1): 28 – 30. DOI:

10.1109/68.817458
[16] BURGEMEIER J, CORDS A, MARZ R, et al. A black box model of EDFA's operating in WDM systems [J]. Journal of lightwave technology, 1998, 16(7): 1271 – 1275. DOI: 10.1109/50.701405
[17] ZIRNGIBL M. Analytical model of Raman gain effects in massive wavelength division multiplexed transmission systems [J]. Electronics letters, 1998, 34(8): 789. DOI: 10.1049/el: 19980555
[18] MUKHERJEE B. WDM optical communication networks: progress and challenges [J]. IEEE journal on selected areas in communications, 2000, 18(10): 1810 – 1824. DOI: 10.1109/49.887904
[19] RAFIQUE D, RAHMAN T, NAPOLI A, et al. Digital pre-emphasis in optical communication systems: on the nonlinear performance [J]. Journal of lightwave technology, 2015, 33(1): 140 – 150. DOI: 10.1109/JLT.2014.2378374
[20] WU Q, ZHU Y X, YIN L J, et al. 50GBaud PAM-4 IM-DD transmission with 24% bandwidth compression based on polybinary spectral shaping [C]//Proceedings of 2021 European Conference on Optical Communication (ECOC). IEEE, 2021: 1 – 4. DOI: 10.1109/ECOC52684.2021.9606146
[21] POIRIER M, BOUDREAU M, LIN Y M, et al. InP integrated coherent transmitter for 100 Gbit/s DP-QPSK transmission [C]//Proceedings of Optical Fiber Communication Conference. OSA, 2015: 1 – 3. DOI: 10.1364/ofc.2015.th4f.1

Biographies

YAN Baoluo (yan.baoluo@zte.com.cn) received his MS degree in optical engineering from Nankai University, China in 2021. He joined ZTE Corporation in 2021 and is engaged in pre-research of optical transmission and monitoring system. His current research interests include intelligent optical network monitoring and optical fiber communication. He has published over 20 papers in peer-reviewed journals and conference proceedings.

WU Qiong received his PhD from Huazhong University of Science and Technology, China in 2017. He works with ZTE Corporation and has been engaged in the research and development of pilot tone technique and optical performance monitoring since 2017.

SHI Hu received his MS degree in optical engineering from Beijing University of Posts and Telecommunications, China in 2015. He works with the WDM System Department of Wireline Product R&D Institute of ZTE Corporation and has been engaged in pre-research of high-speed coherent optical transmission systems since 2015.

ZHAO Yan received his PhD from Beijing University of Posts and Telecommunications, China in 2021. He has been engaged in pre-research for optical performance monitoring at ZTE Corporation since 2021.

JIA Yinqiu received his PhD from Tsinghua University, China in 2018. He works with ZTE Corporation and has been engaged in the research and development of DWDM, OTN, WASON and SDON products since 2018.

FENG Zhenhua received his PhD from Huazhong University of Science and Technology, China in 2017. He has been engaged in the research and development of optical fiber communication systems and algorithms for about five years with more than 50 journal and conference publications and about 10 granted patents. His research interests mainly lie in optical transmission system and DSP algorithms design and verification.

CHEN Weizhang received his bachelor's degree in electronic engineering from Wuhan University, China in 1994. He works with ZTE Corporation and has been engaged in the research and development of DWDM and OTN hardware division.

ZHU Mo received his MS degree in optical engineering from Harbin Institute of Technology, China in 2014. He has been engaged in the development and pre-research of DWDM products since 2014, involving EDFA power control algorithm, OID demodulation algorithm, OTDR noise reduction, recognition algorithm, etc.

ZHAO Zhiyong received his MS degree in radio electronics from Wuhan University, China in 1997. He works with ZTE Corporation and has been engaged in the research and development of DWDM, OTN, WASON, SDON, IP Switch, Router and Access products since 1998.

FANG Yu received his MS degree from the Institute of Electronics, Chinese Academy of Science, China. He has been engaged in long haul optical transmission technology and OTN product developing at ZTE Corporation since 2004.

CHEN Yong received his PhD from Beijing Jiaotong University, China. He has been engaged in the research and development of long haul optical transmission technology and OTN products at ZTE Corporation since 2007.

Spectrum Sensing for OFDMA Using Multicarrier Covariance Matrix Aware CNN

ZHANG Jintao[1], HE Zhenqing[1], RUI Hua[2,3], XU Xiaojing[2,3]

(1. National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China;
2. ZTE Corporation, Shenzhen 518057, China;
3. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China)

DOI: 10.12142/ZTECOM.202203008

https://kns.cnki.net/kcms/detail/34.1294.TN.20220718.1423.002.html, published online July 19, 2022

Manuscript received: 2022-03-06

Abstract: We consider spectrum sensing problems in the orthogonal frequency division multiplexing access (OFDMA) cognitive radio scenario, where a secondary user with multiple antennas detects several consecutive subcarriers of an entire OFDM symbol occupied by multiple primary users. Specifically, an OFDM multicarrier covariance matrix convolutional neural network (CNN)-based approach is proposed for simultaneously detecting the occupancy of all OFDM subcarriers, where the multicarrier sample covariance matrix array is specially set as the input of the CNN. The proposed approach can efficiently learn the energy information and correlation information between antennas and between subcarriers to significantly improve the spectrum sensing performance. Numerical results demonstrate that the proposed method has a substantial performance advantage over the state-of-the-art spectrum sensing methods in an OFDMA scenario under the 5G new radio network.

Keywords: cognitive radio; spectrum sensing; OFDMA; deep learning; 5G new radio

Citation (IEEE Format): J. T. Zhang, Z. Q. He, H. Rui, et al., "Spectrum sensing for OFDMA using multicarrier covariance matrix aware CNN," *ZTE Communications*, vol. 20, no. 3, pp. 61 - 69, Sept. 2022. doi: 10.12142/ZTECOM.202203008.

1 Introduction

Spectrum resources are key strategic resources to constructing new competitive advantages in the global information technology, technological innovation, and economic development. The scarce spectrum resources have become an important factor limiting the high speed and large capacity of the 5G networks[1]. The cognitive radio (CR) technology[2], which allows secondary users (SUs) to opportunistically access primary users' (PUs) licensed bands without affecting the communication quality of the PU, has become a reliable method to make efficient use of spectrum resources. Spectrum sensing, as an important part of CR, needs to continuously detect and determine whether the PU occupies the frequency band for communication through spectrum data before SU accesses this frequency band[3]. A large amount of research has been conducted on spectrum sensing both in academia and the industry over recent years[4].

Energy detection (ED)[5] is the commonest and simplest method of spectrum sensing, but it requires prior knowledge of noise energy and its performance is vulnerable to noise uncertainty (NU)[6-7]. An eigenvalue-based sensing method[8] was proven to be stable under the influence of NU by using antenna correlation information rather than energy information to perform sensing. Various methods based on eigenvalues, such as the maximum-minimum eigenvalue (MME) detection[8], arithmetic to geometric mean (AGM)[9] detection, mean-to-square extreme eigenvalue (MSEE)[10] detection, maximum eigenvalue-to-arithmetic mean (ME-AM) detection and maximum eigenvalue-to-geometric mean (ME-GM) detection[11-12], were proposed to calculate a good test statistic with improved performance, but the performance of these methods varies with different channel models and it is difficult to build an accurate model in the practical wireless environment. To further improve the sensing performance, many spectrum sensing methods based on deep learning (DL)[13] have been proposed, which are motivated by the powerful potential of DL to learn the data-driven features[14]. In Ref. [15], energy information and the statistics of likelihood ratio were treated as the input of the artificial neural network (ANN) to perform spectrum sensing. In Ref. [16], the CNN-based sensing method using a covariance matrix as the input was proposed to obtain the optimal test statistic. Besides the correlation and the energy information, other hidden features like PU's activity pattern could be learned by CNN[17] and the Long Short-Term Memory (LSTM) network[18] to assist spectrum sensing.

Nowadays, most of the 5G wireless communication networks are built under the orthogonal frequency division multiplexing (OFDM) system for high speed transmission of signals

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No.HC-CN-2020120002.

over broadband wireless channels. OFDM has been adopted in several wireless standards, such as IEEE 802.11, IEEE 802.16, 3GPP-LTE, and LTE-Advanced, and various enhanced OFDM schemes have been developed in the 5G new radio (NR) network. How to efficiently perform spectrum sensing in such broadband scenarios to detect the usage of the idle subcarriers has become a key issue still worth investigating. Most of current researchers like in Refs. [19 – 20] are devoted to exploring how to detect the existence of an entire OFDM symbol, rather than detecting which subcarrier is occupied in the entire OFDM symbol. It is therefore difficult to directly migrate these methods to the wideband multi-user systems. Note that many traditional multiband spectrum sensing methods can be applied to the OFDM access (OFDMA) scenarios. For example, the multiband sensing frameworks based on narrowband ED were proposed in Refs. [21 – 22]. In Ref. [23], eigenvalue-based methods were performed on each subchannel in the OFDMA scenarios. In Ref. [24], a DL-based method was proposed to combine the decisions of all SUs on all subchannels. However, these methods ignore the correlation between subchannels and are vulnerable to interference from the frequency selective channel fading, the diversity of PUs' signal power, and the noise uncertainty in practical 5G wireless communication networks.

We propose a CNN-based spectrum sensing algorithm for an OFDMA system with multiple PUs and a multi-antenna SU, which aims to detect the occupancy of subcarriers in an entire OFDM symbol. Specifically, based on the received OFDM symbols, we utilize a multicarrier covariance matrix array as the input of the proposed CNN, ending up with an OFDM multicarrier covariance matrix-CNN (OMCM-CNN) algorithm. The proposed OMCM-CNN algorithm enjoys the following features.

1) In addition to the energy information and antenna correlation information on each subcarrier in the OFDMA system, it can simultaneously learn the correlation information between subcarriers to assist spectrum sensing.

2) It can simultaneously detect the occupancy of all subcarriers in an entire OFDM symbol, while most researchers concern only the detection of the whole OFDM symbol but not the busy state of subcarriers in an OFDM symbol.

3) It can achieve satisfactory spectrum sensing accuracy over the existing methods and its performance is evaluated by simulations under the 5G NR frame structure where the frequency selective channel fading, the diversity of PUs' signal power, and the noise uncertainty are considered.

2 System Model

We consider an OFDMA CR system with N_s subcarriers, K PUs, and an SU, where each PU is equipped with a single antenna and the SU has M antennas to receive the entire OFDM symbols emitted by PUs. SU aims to detect which subcarriers of the radio frequency spectrum of PUs are occupied or sensed idle, so that the SU can utilize the idle subcarriers for communication. In an OFDMA system, an entire block of frequency bands with multiple sets of subcarriers is assigned to one PU for a period each time. A resource block (RB), which contains N_f consecutive subcarriers in the frequency domain and a slot (N_t OFDM symbols) in the time domain, is a minimum time-frequency resource unit allocated to one PU. Taking N_r RB as a subchannel, the k-th PU selects B_k consecutive subchannels with a random location for communication at each sensing time. In addition to the location of the occupied subchannels, the activity pattern of PUs is assumed to be varied with time. The probability of PU k accessing the subchannels for communication is set to be P_k. Furthermore, the receiving signal-to-noise ratios (SNR) of different PU signals are different because of different locations and transmit powers. For simplicity, we assume that the SNRs are uniformly distributed within $[c - w, c + w]$, where c is the average SNR of all PUs and w is the SNR fluctuation factor. Fig. 1 depicts an example of PUs' occupancy in the OFDMA CR system. To this end, we can model the multiband spectrum sensing problem in OFDMA CR network as a binary hypothesis testing problem on multiple channels, which can be expressed as:

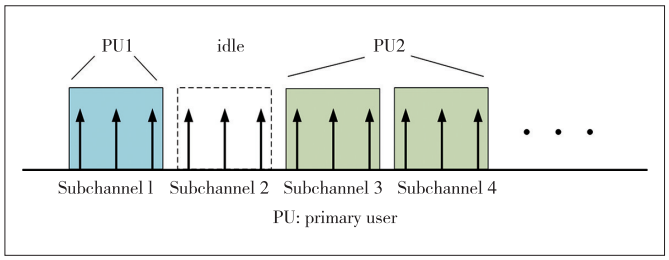
$$\boldsymbol{Y}_b = \begin{cases} \boldsymbol{H}_b \boldsymbol{X}_b + \boldsymbol{W}_b, & H_1 \\ \boldsymbol{W}_b, & H_0 \end{cases}, \tag{1}$$

where $b = 1, \cdots, B$ represents the b-th subchannel and B is the total number of subchannels; $\boldsymbol{Y}_b$ and $\boldsymbol{X}_b$ are the received OFDM symbol and the transmitted OFDM symbol in the frequency domain, respectively; $\boldsymbol{H}_b$ is the channel frequency domain response on subchannel b; H_0 and H_1 represent the binary hypothesis to indicate that subchannel b is idle and occupied, respectively. The popular detector employing signal magnitude information is energy detection[5], which is to detect whether the signal energy on the subchannel is greater than the noise energy threshold.

$$E\left\{\left\| Y_b \right\|^2\right\} \underset{H_0}{\overset{H_1}{\gtrless}} \sigma_W^2. \tag{2}$$

Note that energy detection has the following problems: 1) It requires prior information about the value of noise energy σ_W^2. 2) It relies entirely on energy information for sensing and its detection performance is significantly declined under noise uncertainty. 3) Correlation information between subchannels, such as user occupancy of consecutive subchannels and channel response correlation, is ignored as each subchannel is separately detected. Therefore, we need to jointly detect all subchannels based on the energy information, antenna correlation information, and subchannel correlation information to improve the spectrum sensing performance.

In an OFDM system, the receiver samples the OFDM symbol in the whole frequency band at each time. After sampling and removing the cyclic prefix (CP), we get the received sig-



▲ **Figure 1. PUs' occupancy on an orthogonal frequency division multiplexing (OFDM) symbol**

nal $\boldsymbol{y}(n) = [y_1(n),\cdots,y_M(n)]^{\mathrm{T}}$ in the time domain and $(\cdot)^{\mathrm{T}}$ denotes the transpose operation, which is denoted as:

$$\boldsymbol{y}(n) = \sum_{l=1}^{L} \boldsymbol{h}(l)x(n-l) + \boldsymbol{w}(n), \quad n = 0,\cdots,N_s - 1, \tag{3}$$

where $x(n)$ is the N_s point transmit OFDM symbol, and $\boldsymbol{w}(n)$ is the additive white Gaussian noise (AWGN). In addition, the NU is considered where the actual noise power is changing with time. In the NU scenario, the actual noise power is denoted by $\sigma_\omega^2 = \varepsilon\widehat{\sigma}_\omega^2$, where $\widehat{\sigma}_\omega^2$ is the estimate noise power, ε is the NU factor, and ε (dB) is uniformly distributed within $[-D, D]$. The baseband equivalent channel $\boldsymbol{h}(l) \in \mathbb{C}^M$ $(l = 0,\cdots, L-1)$ is the discrete-time impulse response of the channel with L resolvable paths and is assumed to be the multipath frequency-selective fading channel. To model the correlation between antennas, $\boldsymbol{h}(l)$ is modeled as an exponential correlated zero mean Gaussian random vector, with $M \times M$ statistical covariance matrix $\boldsymbol{R}_{h(l)}$ and the (p, q)-th element of $\boldsymbol{R}_{h(l)}$ is defined as $\boldsymbol{R}_{h(l)} = \sigma_h^2(l) \cdot \rho^{|p-q|}$, where $\sigma_h^2(l)$ is the channel gain power at impulse l and $\rho \in (0,1)$ denotes the correlation coefficient. Through N_s point FFT demodulation, the received OFDM signal $\boldsymbol{Y}(n)$ in the frequency domain can be expressed as

$$\boldsymbol{Y}(n) = \frac{1}{\sqrt{N_s}} \sum_{i=1}^{N_s} \boldsymbol{y}(i) e^{-\frac{j2\pi in}{N_s}}. \tag{4}$$

The objective of spectrum sensing in the OFDMA CR system is to detect the occupancy on all B subchannels based on the available $\boldsymbol{Y}(n)$.

3 OFDM Multicarrier Covariance Matrix Aware CNN

We propose an OMCM-CNN based sensing method to solve the multicarrier spectrum sensing problem in the OFDMA CR system, which consists of sampling, preprocessing, offline training, and online sensing, as illustrated in Fig. 2. In the sampling stage, the multi-antenna SU samples the whole OFDM frequency band and performs FFT demodulation to get the OFDM symbol $\boldsymbol{Y}(n)$ of length N_s. Then, the offline labeled dataset, where the occupancy is known in advance and the online (unlabeled) samples are the data we prepare to detect, can be constructed from the multi-antenna system. Next, we preprocess the raw data and transform it into a data form so that significant features can be readily learned via CNN, and then construct the training set for offline training. Finally, we perform spectrum sensing based on the well-trained CNN using the test data to get its occupancy on each subcarrier. In the following, the construction of a multicarrier covariance matrix array based on the OFDM symbol, the structure of the proposed CNN, the offline training module, and the online sensing module will be elaborated in detail respectively.

3.1 OFDM Multicarrier Covariance Matrix Array

At each sensing time, we can get N_{sym} OFDM symbol $\boldsymbol{Y}(n)$, where $n = 0,\cdots, N_s - 1$, to perform sensing. Expanding the vector $\boldsymbol{Y}(n)$ at n from $n = 0$ to $n = N_s - 1$, we obtain

$$\boldsymbol{Y} = \begin{pmatrix} Y_{1,0} & Y_{1,1} & \cdots & Y_{1,N_s-1} \\ Y_{2,0} & Y_{2,1} & \cdots & Y_{2,N_s-1} \\ \vdots & \ddots & \ddots & \vdots \\ Y_{M,0} & Y_{M,1} & \cdots & Y_{M,N_s-1} \end{pmatrix}, \tag{5}$$

where $Y_{m,n}$ denotes the m-th element of $\boldsymbol{Y}(n)$. By splitting $\boldsymbol{Y}$ as per column, the OFDM symbol at subchannel b becomes

$$\boldsymbol{Y}_b = \begin{pmatrix} Y_{1,(b-1)N_c} & Y_{1,(b-1)N_c+1} & \cdots & Y_{1,bN_c-1} \\ Y_{2,(b-1)N_c} & Y_{2,(b-1)N_c+1} & \cdots & Y_{2,bN_c-1} \\ \vdots & \ddots & \ddots & \vdots \\ Y_{M,(b-1)N_c} & Y_{M,(b-1)N_c+1} & \cdots & Y_{M,bN_c-1} \end{pmatrix}, \tag{6}$$

where $N_c = N_f \cdot N_r$ represents the number of subcarriers in a subchannel. At each sensing time, the receiver (i.e., SU) can get N_{sym} OFDM symbol $\boldsymbol{Y}$ and N_{sym} $\boldsymbol{Y}_b$ based on Eq. (5). Then, we can concatenate N_{sym} $\boldsymbol{Y}_b$ in the column to get the observation matrix $\widehat{\boldsymbol{Y}}_b$ on subchannel b.

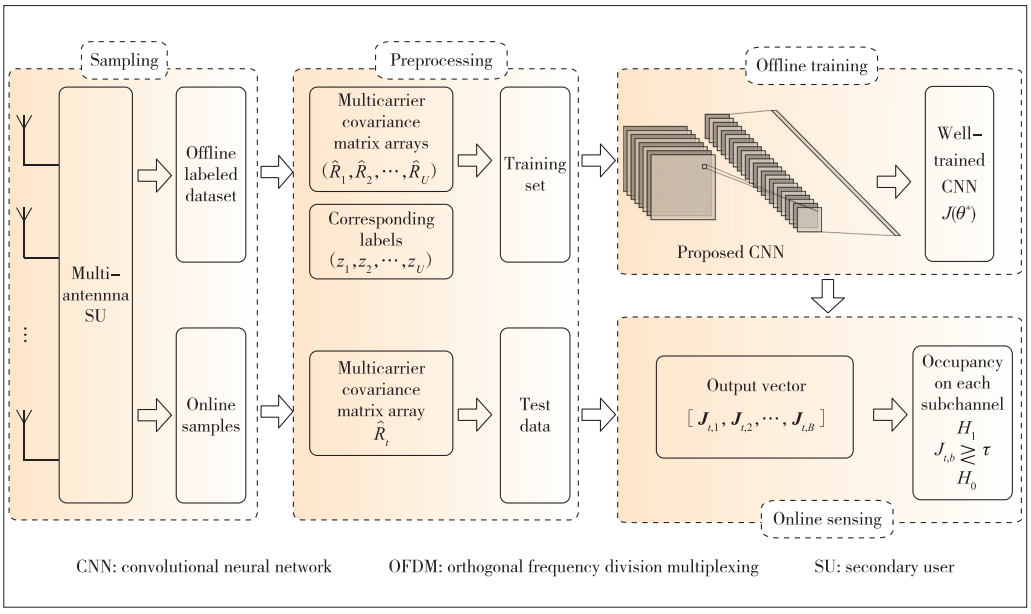
$$\widehat{\boldsymbol{Y}}_b = \left[\boldsymbol{Y}_b^0, \boldsymbol{Y}_b^1, \cdots, \boldsymbol{Y}_b^{N_{\mathrm{sym}}}\right], \tag{7}$$

where $\boldsymbol{Y}_b^{n_{\mathrm{sym}}}$ denotes the n_{sym}-th $\boldsymbol{Y}_b$ within one sensing time.

After obtaining the observation matrix of each subchannel at each sensing time, we need to construct a good CNN model to fit the practical system model by learning appropriate features of the raw data. The statistical sample covariance matrix is considered here, which is calculated as:

$$\boldsymbol{R}_b = \frac{1}{N_0} \widehat{\boldsymbol{Y}}_b \widehat{\boldsymbol{Y}}_b^{\mathrm{H}}, \tag{8}$$

where $N_0 = N_c \cdot N_{\mathrm{sym}}$ stands for the number of observation samples at subchannel b and $(\cdot)^{\mathrm{H}}$ denotes the conjugate transpose operation. The reason for choosing the sample covariance



▲ **Figure 2. OFDM multicarrier covariance matrix-CNN sensing workflow for spectrum sensing**

matrix $\boldsymbol{R}_b$ in Eq. (8) as the input of CNN is that the sample covariance matrix contains not only the energy information of the received signal but also the correlation information between antennas. It has been shown in Ref. [16] that excellent performance of spectrum sensing based on sample covariance matrix could be obtained in the narrowband (single band) sensing scenario. However, such methods which leverage the covariance matrix on each subchannel separately do not utilize the correlation information between the subchannels. To this end, we consider designing a new algorithm that can simultaneously make use of all $\boldsymbol{R}_b$ to detect all subchannels together. Without loss of generality, we concatenate the sample covariance matrices on all subchannels into a big multicarrier covariance matrix array $\widehat{\boldsymbol{R}}$ of $PM \times QM$ size, denoted as

$$\widehat{\boldsymbol{R}} = \begin{pmatrix} \boldsymbol{R}_1 & \boldsymbol{R}_2 & \cdots & \boldsymbol{R}_Q \\ \boldsymbol{R}_{Q+1} & \boldsymbol{R}_{Q+2} & \cdots & \boldsymbol{R}_{2Q} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{R}_{(P-1)Q+1} & \boldsymbol{R}_{(P-1)Q+2} & \cdots & \boldsymbol{R}_{PQ} \end{pmatrix}, \tag{9}$$

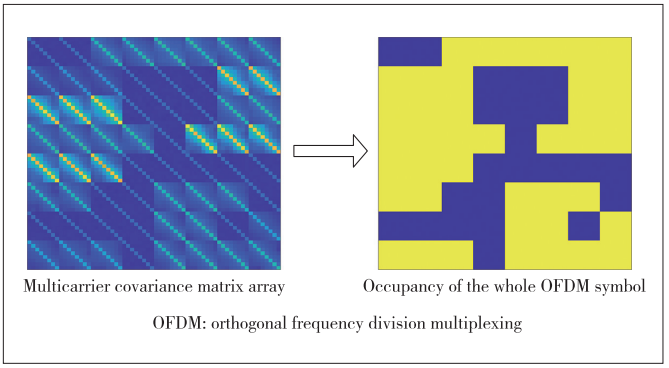
where $PQ = B$. Fig. 3 depicts the characteristics of the multicarrier covariance matrix array example where $M = 8$, $B = 64$, $c = 0$ dB, $w = 2$ dB and $K = 16$. We concatenate the 64 subcarrier covariance matrices into the multicarrier covariance matrix array with $P = 8$ and $Q = 8$. The left subplot is the multicarrier covariance matrix array of the received OFDM symbols. The right subplot is the corresponding OFDM symbol occupancy of the multicarrier covariance matrix array example of given data, where the yellow part represents the occupancy of the received signal and the blue part means it is not occupied or it is idle. By comparison with the two subplots, we see that the multicarrier covariance matrix array in the left subplot can obviously characterize the practical occupancy in the right subplot, and the correlation information between matrix elements and between covariance matrices is obvious to human eyes. Such correlation information between the row and column elements and the specific pattern of the multicarrier covariance matrix array picture can be easily learned by the convolution calculation of CNN, and then yield a decent spectrum sensing performance for online testing.

3.2 CNN Model Selection

In this section, we propose a CNN to detect the occupancy of the OFDMA system based on the OFDM covariance matrix $\widehat{\boldsymbol{R}}$ in Eq. (9). The motivation for choosing CNN includes the following two aspects:

1) CNN is a class of deep neural networks that is widely employed in image classification and recognition and has the powerful potential for extracting hidden features of the matrix-shaped data. Thus, we consider using CNN to learn the energy information and the correlation feature including antenna correlation and subchannel correlation between the row and column elements of $\widehat{\boldsymbol{R}}$, so that we can decide the occupancy on all subchannels based on these distinguishable features.

2) Traditional model-driven methods generally exploit the data features for spectrum sensing based on its models, such as the energies and various expressions of eigenvalues. The performance of model-driven methods depends on the accurate model assumption. However, there does not exist an accurate model for the practical wireless environment. In contrast, CNN is a data-driven method that can obtain the optimal test



▲ **Figure 3. OFDM multicarrier covariance matrix array and its corresponding subchannel occupancy**

statistic based on the sensing data without any accurate model assumption for the wireless environments and thus keep a good spectrum sensing performance under different wireless environments.

For the considered OFDMA CR scenario, we propose a CNN with eight layers, which consists of four convolutional layers, three max-pooling layers, and a fully connected layer. Too few convolutional layers will result in a simple CNN structure and cannot effectively fit the relationship between the raw data and the label, while too many convolutional layers will cause the problem of gradient disappearance and gradient explosion. We use four convolutional layers in the proposed CNN based on many empirical attempts, which can avoid the problem of gradient disappearance and gradient explosion, and achieve a good sensing performance. The size of the convolution kernel of all convolution layers is set as 3×3, since the small size of convolution kernel can learn the correlation information between antennas from the convolution calculation of the elements in a single covariance matrix, while learning the correlation between subchannels from the elements in different covariance matrices. As for the activation function, the rectifier linear unit (Relu) is used as the activation function of all convolutional layers, which is to increase the nonlinearity of the proposed CNN model. After the convolution calculation and down-sampling operation in convolutional layers and max-pooling layers, the feature map which contains the occupancy information in the OFDM symbol can be obtained. Then the proposed CNN flattens the feature map into a feature vector and connects it with a fully connected layer to convert the feature vector into the output vector with B elements. Finally, we connect the fully connected layer with the sigmoid function to limit the value range of the output vector within (0,1) and get the final output vector. The sigmoid function is expressed as:

$$S(x) = \frac{1}{1 + e^{-x}}. \tag{10}$$

In this way, the divergence value of the output vector is converted to (0,1), which can be considered as the probability of the occupancy on each subchannel. The hyper-parameter setting is detailed in Section 4.

3.3 Offline Training

After the CNN model selection, we need to optimize the specific parameters of the proposed CNN, which includes the weight and the bias of all convolution kernels and the fully connected layer. Based on the training data set, the objective of the offline training is to fit the relationship between the multicarrier covariance matrix array and the occupancy on all subchannels.

In the offline training stage, numerous labeled OFDM symbols $\boldsymbol{Y}$ in Eq. (5) can be obtained from the offline labeled database, where "label" means that the occupancy of the training OFDM symbol is given. We obtain the training data set with U OFDM multicarrier covariance matrix arrays $\widehat{\boldsymbol{R}}$ and the corresponding labels $\boldsymbol{z}$ via Eqs. (6) – (9). The training data set can be denoted as:

$$\Omega = \left\{\left(\widehat{\boldsymbol{R}}_1, z_1\right), \left(\widehat{\boldsymbol{R}}_2, z_2\right), \cdots, \left(\widehat{\boldsymbol{R}}_U, z_U\right)\right\}, \tag{11}$$

where $z_u \in \{0,1\}^B$ $(u \in \{1,\cdots,U\})$ represents the occupancy of B subchannels in the whole OFDM symbol, "1" means the associated subchannel is occupied and "0" means idle, and U is the total number of the training data set. In this way, we take $\widehat{\boldsymbol{R}}_u$ as the input of CNN and z_u as the label for CNN training. Note that CNN generally does not support the input with complex values. Thus, we should overlap the real and imaginary parts of $\widehat{\boldsymbol{R}}_u$ on the third dimension, and then input this three-dimensional matrix with real values into the CNN. After nonlinear operations of the CNN layers, the output vector, denoted by $\boldsymbol{J}(\boldsymbol{\theta}, \widehat{\boldsymbol{R}}_u)$ with B elements can be obtained, where $\boldsymbol{J}(\cdot,\cdot)$ represents the total function of the CNN and $\boldsymbol{\theta}$ denotes the whole CNN model parameters. The purpose of offline training is to make the output of CNN approximate the label data more accurately. To measure the accuracy of the output vector, we use the mean square error criterion, and the loss function is defined as:

$$L(\boldsymbol{\theta}) = \frac{1}{U}\sum_{u=1}^{U}\left\| z_u - \boldsymbol{J}\left(\boldsymbol{\theta}, \widehat{\boldsymbol{R}}_u\right)\right\|^2, \tag{12}$$

where $\|\cdot\|^2$ denotes the L_2 norm.

The smaller $L(\boldsymbol{\theta})$ indicates a smaller gap between the output and the label set. To obtain appropriate CNN parameters for online sensing, in the training stage we optimize the loss function $L(\boldsymbol{\theta})$ in Eq. (12) by solving the following minimization problem:

$$\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}). \tag{13}$$

Note that Eq. (13) is non-convex and is generally hard to obtain an analytical solution. Thus, the stochastic gradient descent scheme can be used here to get a sub-optimal solution of Eq. (13). In the simulation, we adopt the optimizer Adam instead of stochastic gradient descent to solve this problem, which is proven to have excellent performance in the deep learning training work[25].

3.4 Online Sensing

After the offline training process, we now use the well-trained CNN with parameter $\boldsymbol{\theta}^*$ to obtain the online sensing results. In the online sensing stage, the SU samples N_0 time-domain received signal sequences $y(n)$ at each sensing time. After data pre-processing via Eqs. (4) – (9), SU can get $\widehat{\boldsymbol{R}}_t$ as the input of the trained CNN to detect the occupancy on all subchannels at sensing time t, where $\widehat{\boldsymbol{R}}_t$ denotes the t-th

OFDM multicarrier covariance matrix array at sensing time t. The output vector can be obtained by the non-linear calculation of the well-trained CNN, denoted as

$$\boldsymbol{J}(\boldsymbol{\theta}^*, \widehat{\boldsymbol{R}}_t) = [J_{t,1}, J_{t,2}, \cdots, J_{t,B}]^{\mathrm{T}}, \tag{14}$$

where $J_{t,b}$ $(b = 1, \cdots, B)$ denotes the b-th output at the t-th sensing time.

Then, we propose a certain coding scheme to transform the output vector into the occupancy vector of all subchannels. As mentioned above, the value of $J_{t,b}$ is limited to (0,1) with the sigmoid function, which can be considered as the probability of the occupancy of subchannel b. Therefore, we can treat $J_{t,b}$ as the test statistic on subchannel b to determine whether subchannel b is occupied or not. Consistent with the decision of traditional sensing algorithms, the occupancy result on subchannel b can be obtained based on the following decision criterion.

$$J_{t,b} \underset{H_0}{\overset{H_1}{\gtrless}} \tau, \tag{15}$$

where τ is the detection threshold, which is determined for the desired probability of false alarm (PFA). We denote the probability of detection (PD) and PFA in our proposed method as follows:

$$P_d = \frac{1}{B}\sum_{b=1}^{B} P\left\{J_{t,b} > \tau | H_1\right\}, \tag{16}$$

$$P_{fa} = \frac{1}{B}\sum_{b=1}^{B} P\left\{J_{t,b} > \tau | H_0\right\}, \tag{17}$$

where P_d and P_{fa} are defined as the averaged probability on all subchannels. Thus, according to the definition of PFA, we can get the estimated value of τ by the the Monte Carlo method. We define $J|H_0$ as the test statistic in the unoccupied situation and $\Omega_{J|H_0}$ as the Monte Carlo dataset of $J|H_0$, where all $J|H_0$ is sorted in descending order. Then, the detection threshold τ with the desired PFA value α is defined as:

$$\tau = \Omega_{J|H_0}\left(\left\lfloor \alpha U_J \right\rfloor\right), \tag{18}$$

where $\lfloor . \rfloor$ represents the round down symbol, $\Omega_{J|H_0}(u)$ denotes the u-th elements of $\Omega_{J|H_0}$, and U_J represents the size of the dataset that indicates the number of Monte Carlo realizations.

3.5 Computational Complexity Analysis

We now discuss the computational complexity of the proposed OMCM-CNN method with a comparison of the traditional model-driven methods including energy detection[5] and the Eigenvalue-based methods[8-12]. The specific complexity analysis of respective algorithms is given in Table 1, where "×" means that the corresponding method does not need any computational operation. For the energy detection method, $\boldsymbol{O}(\boldsymbol{BMN})$ denotes the complexity of calculating the energy information of B subchannels. For the eigenvalue-based methods, $\boldsymbol{O}\left(\boldsymbol{BM}^2\boldsymbol{N}\right)$ denotes the complexity of calculating the covariance matrix of B subchannels from the observation matrix and $\boldsymbol{O}\left(\boldsymbol{BM}^3\right)$ is the complexity of the eigenvalue decomposition of $\boldsymbol{B}$ covariance matrices. The computation of OMCM-CNN comes from the offline training stage and the online sensing stage. For OMCM-CNN, $\boldsymbol{O}\left(\boldsymbol{BM}^2\boldsymbol{N} + \boldsymbol{B}\right)$ denotes the complexity of calculating B subband covariance matrices and converting them into the multicarrier covariance matrix array in the preprocessing stage; $\boldsymbol{O}\left(\sum_{l=1}^{D} n_{l-1} s_l^2 n_l m_l^2\right)$ denotes the complexity of obtaining the output vector from the well-trained CNN, where D, s_l, n_l, m_l, N_t, and N_e denote the number of CNN layers, the spatial size of the convolution kernel of the l-th layer, the number of channels of the l-th layer, the spatial size of the output feature map, the numbers of training examples, and the number of epochs in the offline training stage, respectively. In summary, the main computational complexity of the proposed OMCM-CNN algorithm comes from the offline training stage, which needs a relatively high computational complexity to construct the well-trained CNN model. However, the sensing efficiency depends on the computational complexity of online sensing, which aims to get the test statistics directly based on the well-trained CNN model. After the training stage, the complexity of OMCM-CNN in online sensing is greatly reduced, even lower than the complexity of eigenvalue-based methods. That is to say, the proposed method can avoid the computation of eigenvalue decomposition and directly calculate the test statistic based on the well-trained CNN network parameters.

Table 1. Computational complexity of respective algorithms

Algorithms	Online Sensing	Offline Training
Energy detection[5]	$\boldsymbol{O}(\boldsymbol{BMN})$	×
Eigenvalue-based methods[8-12]	$\boldsymbol{O}\left(\boldsymbol{BM}^2\boldsymbol{N} + \boldsymbol{BM}^3\right)$	×
OMCM-CNN	$\boldsymbol{O}\left(\boldsymbol{BM}^2\boldsymbol{N} + \boldsymbol{B}\right)+$ $\boldsymbol{O}\left(\sum_{l=1}^{D} n_{l-1} s_l^2 n_l m_l^2\right)$	$\boldsymbol{O}\left(N_t N_e\left(\boldsymbol{BM}^2\boldsymbol{N} + \boldsymbol{B}\right)\right)+$ $\boldsymbol{O}\left(N_t N_e \sum_{l=1}^{D} n_{l-1} s_l^2 n_l m_l^2\right)$

OMCM-CNN: OFDM multicarrier covariance matrix-convolutional neural network

4 Simulation Results

In this section, the performance of the proposed OMCM-CNN algorithm is evaluated. In order to explore the practical application significance of our proposed algorithm, we consider the OFDMA sensing problem under 5G NR system parameters. We consider an OFDMA system with $K = 16$ PUs and 4 096 subcarriers, where 3 072 subcarriers are used for

communication and 512 subcarriers on each band side are considered as the guard interval. The bandwidth of a single subcarrier is set to W_{sub} = 30 kHz and the total bandwidth of the system is W_B = 30 kHz × 4 096 = 122.88 MHz. At each sensing time, an SU with M = 8 antennas can sample the received signal to get N_{sym} = 100 OFDM symbol $\boldsymbol{Y}$ for spectrum sensing. According to the Third Generation Partnership Project (3GPP) 38.211 standard, each resource block (RB) contains N_f = 12 subcarriers and a single subchannel contains N_r = 4 RB. Thus, the total number of subchannels can be calculated by $B = 3\,072/(N_r \times N_f) = 64$. PU k occupies B_k consecutive subchannels with probability P_k = 50% at each sensing time where B_k is randomly selected from the integer set $\{2, 3, 4, 5, 6\}$ and randomly chooses BPSK, QPSK or 4QAM as its modulation mode. The average SNR is set to c = −10 dB and the SNR fluctuation factor is set to w = 2 dB. As for the channel model, the channel gain power $\sigma_h^2(l)$ at impulse l is set according to Tapped Delay Line (TDL)-B model in 3GPP 38.901 standard, where the normalized time delay is set to 100 us. Furthermore, we assume that the length of CP is 1 000, ρ = 0.75 and D = 0.5 dB. In the preprocessing stage, we concatenate the subcarrier covariance matrices with B = 64 into the OFDM multicarrier covariance matrix array where P = 8 and Q = 8. The specific hyperparameters of the proposed covariance matrix-aware CNN are given in Table 2.

Table 2. Hyper parameters of the proposed CNN

Input: Multicarrier Covariance Matrix Array (64 × 64 × 2)	
Layers	Convolution Kernel Size
C1+ ReLu	128@(3 × 3), padding, stride = 1
M1	2 × 2, stride = 2
C2+ ReLu	128@(3 × 3), padding, stride = 1
M2	2 × 2, stride = 2
C3+ ReLu	256@(3 × 3), padding, stride = 1
C4+ ReLu	256@(3 × 3), padding, stride = 1
M3	2 × 2, stride = 2
F+ Sigmoid	16 384 × 64
Output: Feature Vector (64 × 1)	

CNN: convolutional neural network ReLu: rectifier linear unit

We take the ED[5], MME[8], AGM[9], MSEE[10], ME-AM, and ME-GM[11-12] as the benchmark to evaluate our proposed method. Note that these baseline algorithms are introduced for narrowband spectrum sensing and applied to the OFDMA system in each subband separately. The PD and PFA in the simulation results are defined as the averaged value of all subchannels by 10 000 Monte Carlo realizations.

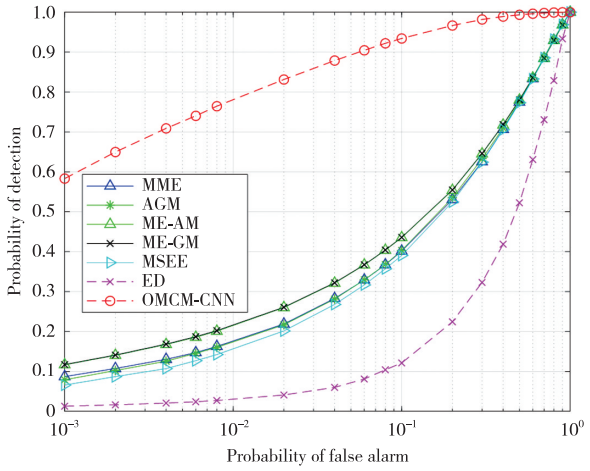
Fig. 4 depicts the receiver operating characteristic (ROC) curves of respective algorithms, i.e., PD versus PFA. It can be observed from Fig. 4 that the ED which relies on only the energy information has the lowest performance essentially due to the presence of NU. Eigenvalue-based methods (MME, AGM, MSEE, ME-AM, and ME-GM) also achieve an unsatisfactory performance since these methods ignore the energy information and the correlation information between subchannels. The performance of the proposed OMCM-CNN method is significantly better than those of other baseline methods since it comprehensively combines the energy information of the received signal and the correlation information between the antennas and subchannels. Fig. 5 depicts the ROC curves with the number of antennas M = 64 and the other parameter setting is kept the same as Fig. 4. We see that the proposed method still exhibits the best performance, and with the increase in the number of antennas, the spectrum sensing performance of all algorithms is further improved.

Next, we explore the influence of different OFDM symbol sampling numbers N_{sym}, i.e., PD versus N_{sym}. We set M = 8, w = 2 dB, and c = −10 dB, and PFA is set to 0.1 according to IEEE 802.22 standard. Fig. 6 shows the PD of each algorithm under different sampling numbers with N_{sym} from 20 to 200. In practice, we can achieve a high N_{sym} by extending the sampling time or setting multiple sensors to sample. It can be seen from Fig. 6 that the performance of the proposed algorithm is better than other traditional algorithms under all N_{sym}. As expected, the performance of the proposed algorithm and the eigenvalue-based methods improves as the sampling number N_{sym} increases. This is because in the case of large N_{sym}, the statistical characteristics of the received signal have a more accurate estimate at a higher sampling number. However, the ED has a poor performance and has no improvement under different sampling numbers. This phenomenon is incurred by NU and SNR fluctuation, which causes huge interference to the energy information. That is to say, even with a large number of sampling numbers, the statistical characteristic of the energy information still cannot be correctly estimated.

Finally, we plot the PD versus different averaged SNR c in Fig. 7 to test the robustness of the proposed algorithm. We set M = 8, N_{sym} = 100, and PFA is set to 0.1 according to IEEE 802.22 standard. It can be observed from Fig. 7 that the proposed CNN method achieves the best performance under different SNRs. The performance of all algorithms improves significantly as the SNR increases. The ED still achieves an unsatisfactory sensing performance with SNR increasing since it is greatly disturbed by the NU, which shows that it is not feasible to perform multicarrier sensing based on only the energy information.

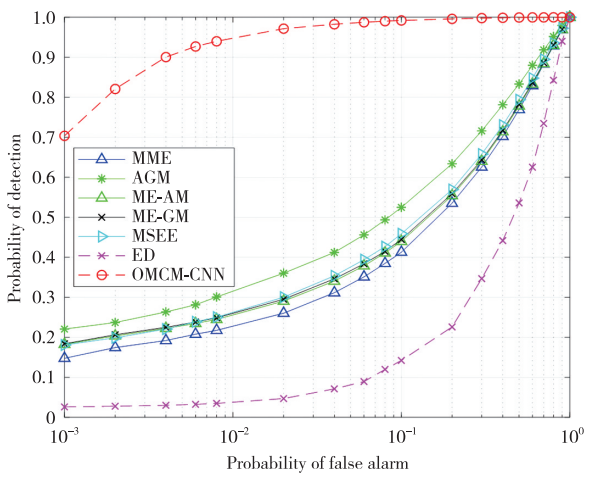
5 Conclusions

In this paper, we have investigated the spectrum sensing problem in the OFDMA scenario under the 5G NR network and developed a spectrum sensing method based on the multicarrier covariance matrix aware-CNN. The proposed approach can effectively learn the energy and the correlation informa-



AGM: arithmetic to geometric mean ED: energy detection
ME-AM: maximum eigenvalue-to-arithmetic mean
ME-GM: maximum eigenvalue-to-geometric mean
OMCM-CNN: OFDM multicarrier covariance matrix-convolutional neural network
MME: maximum-minimum eigenvalue
MSEE: mean-to-square extreme eigenvalue

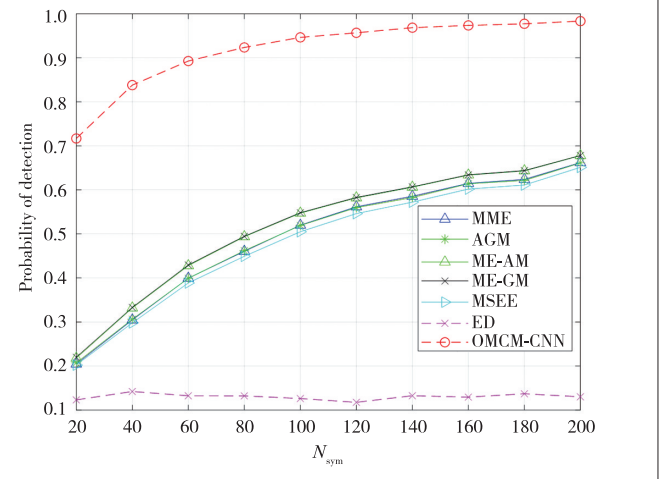
▲ **Figure 4. Receiver operating characteristic (ROC) curves of different algorithms with M = 8**



AGM: arithmetic to geometric mean ED: energy detection
ME-AM: maximum eigenvalue-to-arithmetic mean
ME-GM: maximum eigenvalue-to-geometric mean
MME: maximum-minimum eigenvalue
MSEE: mean-to-square extreme eigenvalue
OMCM-CNN: OFDM multicarrier covariance matrix-convolutional neural network

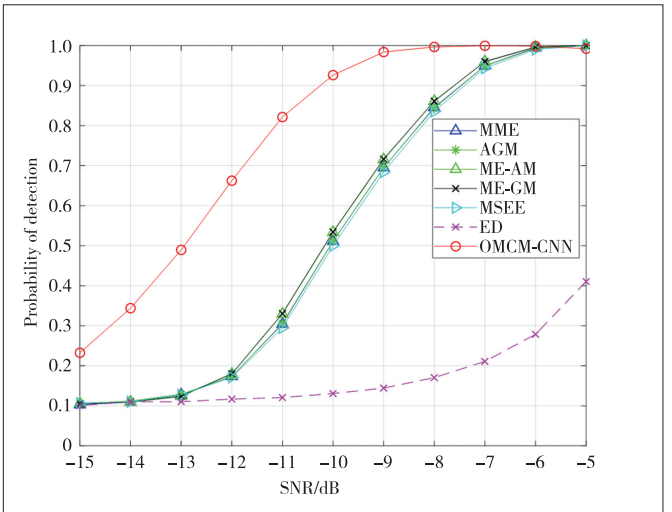
▲ **Figure 5. Receiver operating characteristic (ROC) curves of different algorithms with M = 64**

tion between antennas and between subcarriers to further improve the sensing performance. Simulation results in the OFDMA scenarios under 5G NR network have illustrated the superior performance of the proposed method over several state-of-the-art algorithms.



AGM: arithmetic to geometric mean ED: energy detection
ME-AM: maximum eigenvalue-to-arithmetic mean
ME-GM: maximum eigenvalue-to-geometric mean
MME: maximum-minimum eigenvalue
MSEE: mean-to-square extreme eigenvalue
OMCM-CNN: OFDM multicarrier covariance matrix-convolutional neural network

▲ **Figure 6. Probability of detection with different number of samples N_{sym} with probability of false alarm (PFA) = 0.1**



AGM: arithmetic to geometric mean ED: energy detection
ME-AM: maximum eigenvalue-to-arithmetic mean
ME-GM: maximum eigenvalue-to-geometric mean
MME: maximum-minimum eigenvalue
MSEE: mean-to-square extreme eigenvalue
OMCM-CNN: OFDM multicarrier covariance matrix-convolutional neural network
SNR: signal-to-noise ratio

▲ **Figure 7. Probability of detection with different average SNR c under probability of false alarm (PFA) = 0.1**

References

[1] TIAN F, FENG Z, CHEN X. Spectrum occupancy measurement and analysis [J]. ZTE communications, 2009, 7(2): 16 – 20
[2] MITOLA J, MAGUIRE G Q. Cognitive radio: making software radios more personal [J]. IEEE personal communications, 1999, 6(4): 13 – 18. DOI: 10.1109/

98.788210
[3] LIANG Y C, CHEN K C, LI G Y, et al. Cognitive radio networking and communications: an overview [J]. IEEE transactions on vehicular technology, 2011, 60 (7): 3386 – 3407. DOI: 10.1109/TVT.2011.2158673
[4] WANG H, SU X, WANG J. Cooperative spectrum sensing techniques in cognitive radio [J]. ZTE communications, 2009, 7(2): 11 – 15
[5] DIGHAM F F, ALOUINI M S, SIMON M K. On the energy detection of unknown signals over fading channels [J]. IEEE transactions on communications, 2007, 55(1): 21 – 24. DOI: 10.1109/TCOMM.2006.887483
[6] SONNENSCHEIN A, FISHMAN P M. Radiometric detection of spread-spectrum signals in noise of uncertain power [J]. IEEE transactions on aerospace and electronic systems, 1992, 28(3): 654 – 660. DOI: 10.1109/7.256287
[7] TANDRA R, SAHAI A. Fundamental limits on detection in low SNR under noise uncertainty [C]//Proceedings of 2005 International Conference on Wireless Networks, Communications and Mobile Computing. IEEE, 2005: 464 – 469. DOI: 10.1109/WIRLES.2005.1549453
[8] ZENG Y, LIANG Y C. Eigenvalue-based spectrum sensing algorithms for cognitive radio [J]. IEEE transactions on communications, 2009, 57(6): 1784 – 1793. DOI: 10.1109/TCOMM.2009.06.070402
[9] ZHANG R, LIM T J, LIANG Y C, et al. Multi-antenna based spectrum sensing for cognitive radios: A GLRT approach [J]. IEEE transactions on communications, 2010, 58(1): 84 – 88. DOI: 10.1109/TCOMM.2010.01.080158
[10] BOUALLEGUE K, DAYOUB I, GHARBI M, et al. Blind spectrum sensing using extreme eigenvalues for cognitive radio networks [J]. IEEE communications letters, 2018, 22(7): 1386 – 1389. DOI: 10.1109/LCOMM.2017.2776147
[11] ZHAO W J, LI H, JIN M L, et al. Eigenvalues-based universal spectrum sensing algorithm in cognitive radio networks [J]. IEEE systems journal, 2021, 15 (3): 3391 – 3402. DOI: 10.1109/JSYST.2020.3002941
[12] LI H, ZHAO W J, JIN M L, et al. Improved spectrum sensing algorithm combining energy and eigenvalue [J]. International journal of future computer and communication, 2020: 27 – 32. DOI: 10.18178/ijfcc.2020.9.2.561
[13] LECUN Y, BENGIO Y, HINTON G. Deep learning [J]. Nature, 2015, 521 (7553): 436 – 444. DOI: 10.1038/nature14539
[14] GUO D, ZHENG Q, PENG X, et al. Face detection, alignment, quality assessment and attribute analysis with multi-task hybrid convolutional neural networks [J]. ZTE Communications, 2019, 17(3): 15 – 22. DOI: 10.12142/ZTECOM.201903004
[15] VYAS M R, PATEL D K, LOPEZ-BENITEZ M. Artificial neural network based hybrid spectrum sensing scheme for cognitive radio [C]//Proceedings of 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications. IEEE, 2017: 1 – 7. DOI: 10.1109/PIMRC.2017.8292449
[16] LIU C, WANG J, LIU X M, et al. Deep CM-CNN for spectrum sensing in cognitive radio [J]. IEEE journal on selected areas in communications, 2019, 37 (10): 2306 – 2321. DOI: 10.1109/JSAC.2019.2933892
[17] XIE J D, LIU C, LIANG Y C, et al. Activity pattern aware spectrum sensing: a CNN-based deep learning approach [J]. IEEE communications letters, 2019, 23(6): 1025 – 1028. DOI: 10.1109/LCOMM.2019.2910176
[18] XIE J D, FANG J, LIU C, et al. Deep learning-based spectrum sensing in cognitive radio: a CNN-LSTM approach [J]. IEEE communications letters, 2020, 24(10): 2196 – 2200. DOI: 10.1109/LCOMM.2020.3002073
[19] CHEN H S, GAO W, DAUT D G. Spectrum sensing for OFDM systems employing pilot tones [J]. IEEE transactions on wireless communications, 2009, 8 (12): 5862 – 5870. DOI: 10.1109/TWC.2009.12.080777
[20] ZENG Y H, LIANG Y C, PHAM T H. Spectrum sensing for OFDM signals using pilot induced auto-correlations [J]. IEEE journal on selected areas in communications, 2013, 31(3): 353 – 363. DOI: 10.1109/JSAC.2013.130303
[21] QUAN Z, CUI S G, SAYED A H, et al. Optimal multiband joint detection for spectrum sensing in cognitive radio networks [J]. IEEE transactions on signal processing, 2009, 57(3): 1128 – 1140. DOI: 10.1109/TSP.2008.2008540
[22] HAMDAOUI B, KHALFI B, GUIZANI M. Compressed wideband spectrum sensing: concept, challenges, and enablers [J]. IEEE communications magazine, 2018, 56(4): 136 – 141. DOI: 10.1109/MCOM.2018.1700719
[23] GUIMARÃES D, DA SILVA C, DE SOUZA R. Cooperative spectrum sensing using eigenvalue fusion for OFDMA and other wideband signals [J]. Journal of sensor and actuator networks, 2013, 2(1): 1 – 24. DOI: 10.3390/jsan2010001
[24] LEE W, KIM M, CHO D H. Deep cooperative sensing: cooperative spectrum sensing based on convolutional neural networks [J]. IEEE transactions on vehicular technology, 2019, 68(3): 3005 – 3009. DOI: 10.1109/TVT.2019.2891291
[25] KINGMA D P, BA J. Adam: a method for stochastic optimization [EB/OL]. [2022-03-01]. https://arxiv.org/abs/1412.6980

Biographies

ZHANG Jintao received his BS and MS degrees in communication engineering from the University of Electronic and Science Technology of China, in 2019 and 2022, respectively. His research interests include spectrum sensing and deep learning.

HE Zhenqing (zhengqinghe@uestc.edu.cn) received his PhD degree in communication and information system from the University of Electronic Science and Technology of China (UESTC) in 2017. From 2015 to 2016, he was a visiting PhD student with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, USA. Since 2018, he has been with the National Key Laboratory of Science and Technology on Communications, UESTC, where he is currently an associate professor. His main research interests include statistical signal processing, wireless communications, and machine learning. He was a recipient of the IEEE Communications Society Heinrich Hertz Prize Paper Award in 2022.

RUI Hua received his BS, MS, and PhD degrees from Nanjing University of Aeronautics and Astronautics, China in 1999, 2002, and 2005, respectively. He currently works as a senior pre-research expert and the head of the 6G Future Wireless Lab in ZTE Corporation. He has been engaged in wireless communication product and new technology pre-research, including 3G/4G/WIFI/5G/6G network architecture and key technologies. At present, his main research direction is the 6G wireless communication technology, including new receiver research integrated with communication-sensing-computing, NB-NTN narrowband low-orbit satellite system and key technologies, 6G network architecture and protocol standardization research, digital twin wireless network technology, network intelligence, regional block chain network, etc. He has published more than 20 invention patents and papers in related fields. He has participated in more than 10 industry technical standards and white papers including 3GPP 3G/4G/5G series standards and IEEE 802.11 series standards.

XU Xiaojing received her BS and MS degrees in communication and information system from Northeastern University, China in 2006 and 2008, respectively. At present, she works in ZTE Corporation as a senior algorithm engineer in the Algorithm Department. She has been engaged in wireless communication technology pre-research and product algorithm research. Her research interests include the 6G wireless communication physical layer technology and wireless AI technology. She has published more than 10 invention patents and papers in related fields.

Synthesis and Design of 5G Duplexer Based on Optimization Method

WU Qingqiang[1], CHEN Jianzhong[1], WU Zengqiang[2], GONG Hongwei[2]

(1. National Key Laboratory of Antennas and Microwave Technology, Shaanxi Joint key Laboratory of Graphene, Xidian University, Xi'an 710071, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202203009

https: //kns.cnki.net/kcms/detail/34.1294.TN.20220712.1122.002.html, published online July 12, 2022

Manuscript received: 2021-10-21

Abstract: A new optimization method is proposed to realize the synthesis of duplexers. The traditional optimization method takes all the variables of the duplexer into account, resulting in too many variables to be optimized when the order of the duplexer is too high, so it is not easy to fall into the local solution. In order to solve this problem, a new optimization strategy is proposed in this paper, that is, two-channel filters are optimized separately, which can reduce the number of optimization variables and greatly reduce the probability of results falling into local solutions. The optimization method combines the self-adaptive differential evolution algorithm (SADE) with the Levenberg-Marquardt (LM) algorithm to get a global solution more easily and accelerate the optimization speed. To verify its practical value, we design a 5G duplexer based on the proposed method. The duplexer has a large external coupling, and how to achieve a feed structure with a large coupling bandwidth at the source is also discussed. The experimental results show that the proposed optimization method can realize the synthesis of higher- order duplexers compared with the traditional methods.

Keywords: optimization; self-adaptive differential evolution algorithm; LM optimization algorithm; filter synthesis; duplexer

Citation (IEEE Format): Q. Q. Wu, J. Z. Chen, Z. Q. Wu, et al, "Synthesis and design of 5G duplexer based on optimization method," *ZTE Communications*, vol. 20, no. 3, pp. 70 - 76, Sept. 2022. doi: 10.12142/ZTECOM.202203009.

1 Introduction

With the rapid development of the 5G technology, microwave duplexers have been widely used in wireless and satellite communications. The earliest method of synthesizing duplexers was connecting two channel filters directly to a common cavity, and then modifying the parameters of each filter to compensate for the interaction between the two channels[1-2]. This method was limited by the number of channels and the coupling topology of the channel filter, and the synthesis results would get worse as the frequency band approaches.

In recent years, MACCHIARELLA and TAMIAZZO have proposed a more efficient and flexible method for synthesizing duplexers[3]. In this method, the transmission function and reflection function of each channel filter are derived from the relationship between the global parameters of the duplexer and the parameters of the independent channel filter. Recently, ZHAO Ping and WU Keli proposed a new duplexer synthesis method[4-6], by which all the channel filters are synthesized separately under the consideration of the influence of other channels and the process is repeated to ensure that the final results meet the requirements.

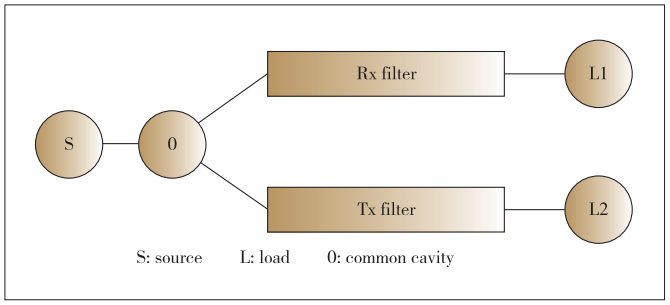
In this paper, the self-adaptive differential evolution algorithm[7-8] and the LM optimization algorithm[9-10] are used instead of the analytical method to obtain the coupling matrix of a single channel filter. The self-adaptive differential evolution algorithm can reduce the probability of convergence to local solutions while the LM method can improve the optimization speed. Compared with the analytical methods mentioned above, an optimization algorithm has a higher degree of freedom, and the optimization algorithm proposed in this paper can achieve a higher order than the traditional optimization algorithm. Because the optimized duplexer requires a larger coupling structure between the source and the common cavity, how to achieve a larger port coupling is also discussed in this paper.

2 Design of Duplexer Optimization Algorithm

Fig. 1 illustrates the structure of the star-junction duplexer, which consists of two filters connected in parallel to the same common cavity. The parameters S, L1, and L2 in the figure represent the source and load of the duplexer, while 0 repre-

This work was supported by the National Natural Science Foundation of China (NSFC) under project no. 62071357 and the Fundamental Research Funds for the Central Universities.

sents the common cavity. Before the duplexer optimization, the transmission and reflection polynomials of a single channel filter should be obtained by the traditional generalized Chebyshev synthesis method[11], and then we need to transform the polynomials of these filters so that they correspond to the passband of the duplexer. Finally, the corresponding coupling matrix is extracted according to the polynomial after transformation, which is taken as the initial value of optimization. The source and the common cavity are generally 1.4 according to experience.



▲ **Figure 1. Structure of star-junction duplexer**

As a global optimization algorithm, the adaptive differential evolution algorithm needs to determine the upper and lower limits of the variables to be optimized according to the initial values in order to improve the optimization speed and success rate. Since the influence of other channels is mainly reflected in the first cavity of the filter, the upper and lower limits of the coupling between the source and the common cavity are taken as "initial value +/−0.1", and the limits of the coupling between the common cavity and the first cavity of two channels are taken as "initial value +/−0.2". In addition, the self-coupling of the first cavity of each channel are taken as "initial value +/−0.2" as well. Finally, the limits of the rest couplings are taken as "initial value +/−0.05". The advantages of the adaptive differential evolution algorithm are as follows. 1) An adaptive control mechanism is adopted for parameters in optimization; 2) In order to avoid falling into the local solution, a new operator, called the self-adaptive return operator, is activated when the optimization is judged to be trapped in local optima, which can often be observed in earlier iterations if it happens. That is the algorithm searches again according to the initial value and its range, when trapped in a local solution.

After the upper and lower limits of the optimized variables are obtained, it is the choice of the objective function, which also plays a key role in the success of optimization. For coupling matrix synthesis, the objective function is formed by *S*-parameter specifications which can be calculated according to Eqs. (1) and (2). In this paper, the objective function of optimizing a single channel filter for the first time is given in Eq. (3). When optimizing another channel filter, its objective function is given in Eq. (4).

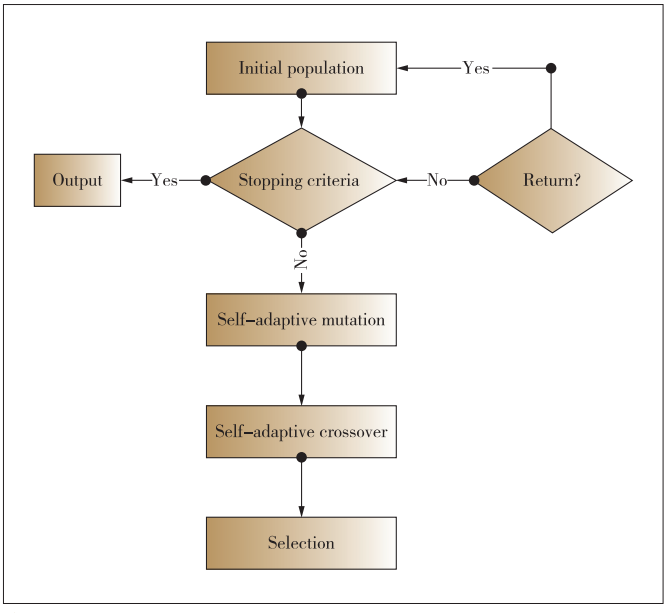
$$S_{pp} = \pm\left(1 - 2[A]_{pp}^{-1}\right), \tag{1}$$

$$S_{pq}|_{p \neq q} = 2[A]_{pq}^{-1}. \tag{2}$$

$$f(x) = \frac{\left|\max\left[S_{11}(\mathrm{PB}_1)\right] - \mathrm{RL}\right|}{|\mathrm{RL}|} + \frac{\left|\max\left[S_{21} - \mathrm{Attenu}\right]\right|}{|\mathrm{Attenu}|}, \tag{3}$$

$$f(x) = \frac{\left|\max\left[S_{11}(\mathrm{PB}_1)\right] - \mathrm{RL}\right|}{|\mathrm{RL}|} + \frac{\left|\max\left[S_{11}(\mathrm{PB}_2)\right] - \mathrm{RL}\right|}{|\mathrm{RL}|} + \frac{\left|\max\left[S_{21} - \mathrm{Attenu}\right]\right|}{|\mathrm{Attenu}|} + \frac{\left|\max\left[S_{31} - \mathrm{Attenu}\right]\right|}{|\mathrm{Attenu}|}. \tag{4}$$

Among them, PB_k denotes the k-th (k = 1, 2) passband. Return loss (RL) and Attenu are the desired return loss and restraint outside the band, respectively. Fig. 2 shows the general steps of the adaptive differential evolution algorithm.



▲ **Figure 2. Flow diagram of the adaptive differential evolution algorithm**

After the approximate response curve is obtained by the adaptive differential evolution algorithm, the LM algorithm is adopted in the optimization algorithm, and the reflection zero, passband edge return loss, and transmission zero of the corre-

sponding channel filter are selected as the sampling points. In summary, the basic steps of the algorithm can be obtained as follows, and the algorithm is realized by Matlab[12].

1) Initial selection. According to the requirements of the index, the transmission and reflection polynomials of the two-channel filters are obtained by using the generalized Chebyshev synthesis method, then the corresponding duplexer polynomials are obtained by frequency transformation, and the initial values of the optimized variables are obtained after the coupling matrix is extracted and rotated. The variables to be optimized include the coupling M_{s0} between source and common cavities, the mutual coupling M_{ij} (i≠j) between cavities, and the self-coupling M_{ii} of cavities.

2) Determining the range of variables. According to the obtained initial value, we can define the value range of the corresponding variable. The specific method has been introduced and will not be repeated here.

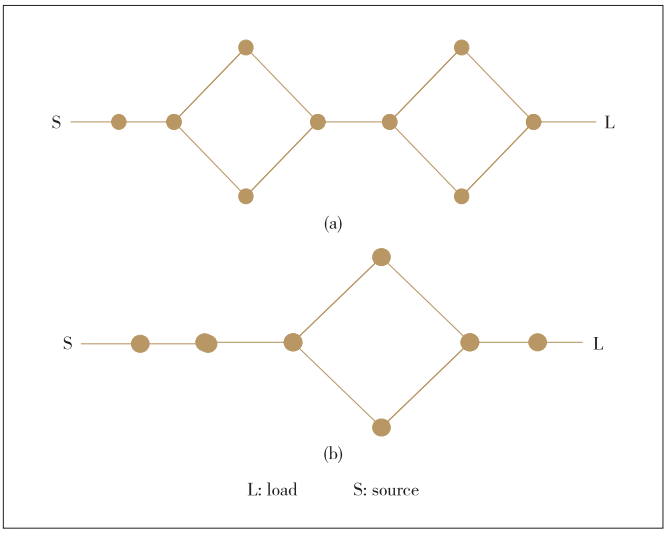
3) Optimization of the low-frequency channel filter. The coupling between the source and the common cavity and the non-zero elements of the low-frequency channel filter coupling matrix is selected as optimization variables, and the elements of the high-frequency channel filter coupling matrix are kept unchanged. The differential evolution algorithm and Eq. (1) are used to optimize the low-frequency channel filter.

4) Optimization of the high-frequency channel filter. Similarly, the coupling between the source and the common cavity and the non-zero elements of the high-frequency channel filter coupling matrix is selected as optimization variables, and the elements of the low-frequency channel filter coupling matrix remain unchanged. The differential evolution algorithm and Eq. (2) are used to optimize the high-frequency channel filter.

5) Optimization of the coupling matrix of the duplexer with the LM algorithm. The duplexer model obtained by the differential evolution algorithm can meet the requirements of the index, but still there is room for optimization. LM algorithms as a gradient optimization algorithm can make the final frequency response better meet the requirements. Different from the adaptive differential evolution algorithm, the LM algorithm uses the reflection zeros of the two-channel filters, the points on the channel edge, and the transmission zeros as the sampling points.

3 Experiments and Results Discussion

To verify the above design, an example of a duplexer that has a low-frequency channel of order 9 and a high frequency channel of order 7 is used. Fig. 3 shows the specific topology of this duplexer and the following tables list the specifications of the duplexer (Table 1 shows the passband range and return loss and Table 2 shows the restraint outside the band). The parameters S and L in the figure represent the source and load of the filter respectively. The box topology in Fig. 3 is chosen mainly for the reason that the box topology is easier to realize



▲**Figure 3. Diplexer topology used in the example: (a) low frequency topology and (b) high frequency topology**

▼**Table 1. Passband indicators**

Passbard Range/MHz	Return Loss/dB
1 710 – 1 785	−17
1 920 – 1 980	−17

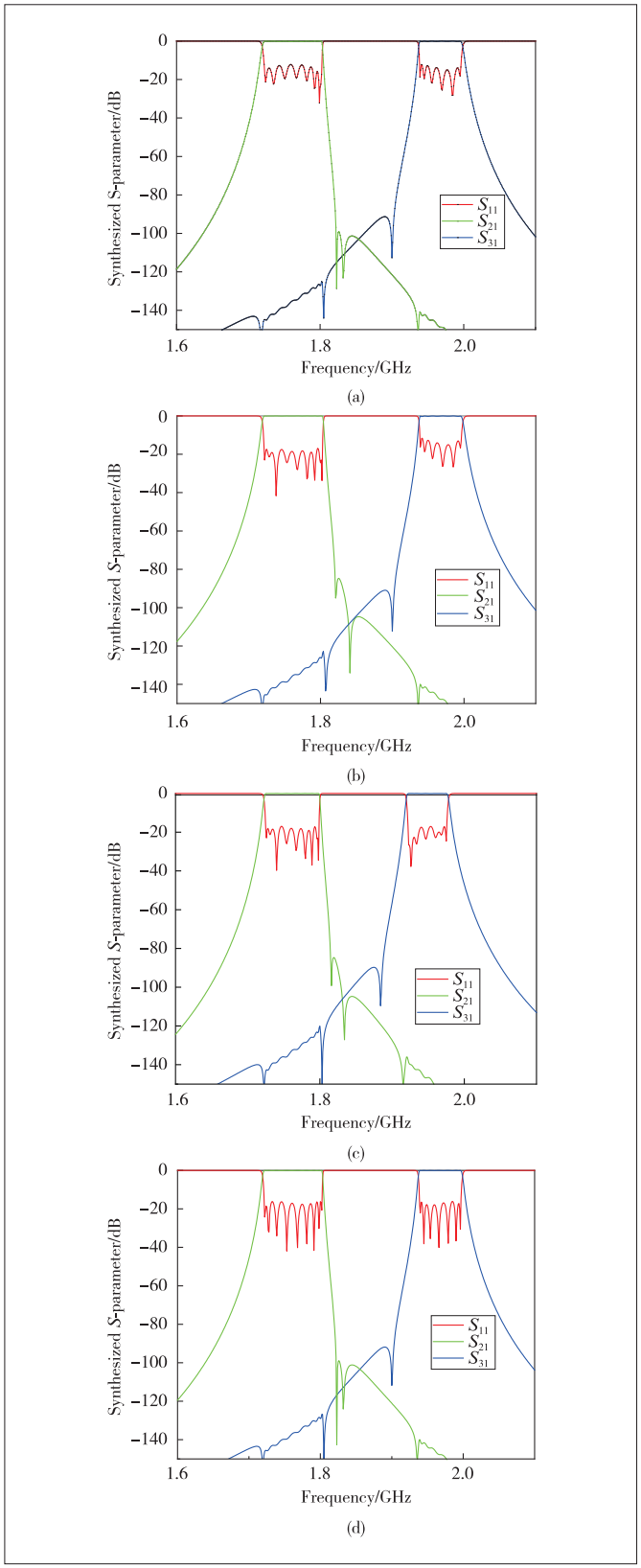
▼**Table 2. Restraint outside the band**

Low-Frequency Channel Filter		High-Frequency Channel Filter	
1 805 – 1 880 MHz	−80 dB	1 805 – 1 880 MHz	−80 dB

in simulation and machining, and the parasitic influence between cavities can be reduced due to the lack of diagonal coupling. It can be seen that the passband range of the low-frequency channel filter is 1 710 – 1 785 MHz, the passband range of the high-frequency channel filter is 1 920 – 1 980 MHz, and the return loss in both passbands is −17 dB. Both of the two-channel filters require −80 dB out-of-band suppression in 1 805 – 1 880 MHz. To achieve this goal, two additional transmission zeros should be introduced for the low frequency channel filter and one transmission zero for the high frequency channel filter. How to use optimization algorithms to achieve these indicators will be described below.

First, we use the generalized Chebyshev synthesis method to get the initial value of the duplexer, and the corresponding response curve is shown in Fig. 4(a). It can be seen that due to the interaction between the two channels, the response in the passband of the duplexer becomes very poor.

fter the initial value of the coupling variable of the duplexer is obtained, according to the above theory and experience, we can design the initial values of the variables and their optimization intervals. Then we keep the value of the high-frequency channel filter coupling matrix unchanged. According to Eq. (1) and the index requirements in Tables 1 and 2, we use the adaptive differential evolution algorithm to optimize the coupling variables of the low-frequency channel. The results of optimization are shown in Fig. 4(b), where the response in the



▲ **Figure 4. (a) Initial response of the duplexer, (b) corresponding result after optimizing low channel filter in the first iteration, (c) corresponding result after optimizing high channel filter in the first iteration, and (d) final result**

passband of the low-frequency channel filter has been greatly improved after optimization.

Similarly, we keep the value of the low-frequency channel filter coupling matrix unchanged. According to Eq. (2) and the index requirements in Tables 1 and 2, the adaptive differential evolution algorithm is used to optimize the coupling variables of the high-frequency channel. The results of optimization are shown in Fig. 4(c). It can be seen that although the final curve meets the requirements of the index, there is still room for optimization. We do gradient optimization on the final result to make it converge to the optimal solution. The optimization algorithm is the LM algorithm, and the sampling point is the reflection zero and transmission zero of the two channel filters. Fig. 4(d) shows that after gradient optimization, the final result curve further meets our requirements, and the values of the coupling matrix corresponding to each figure in Fig. 4 are listed in Table 3.

After the coupling matrix of the duplexer is obtained by the optimization algorithm, a simulation analysis is needed. Because the relative bandwidth of the duplexer is 14.67%, the coverage band is wide, the two passband bands of the duplexer are far apart, and the intermediate interval bandwidth accounts for 50% of the coverage band of the whole duplexer, which puts forward a great demand for the coupling bandwidth of the feed. According to the optimization results, the port needs to provide a coupling between the source and the common cavity of 1.391. According to Eq. (5), the required external Q value is 3.524. Under such requirements, it is difficult for the traditional coupling structure to achieve such a large coupling bandwidth for the duplexer realized by the dielectric waveguide. Therefore, before simulation, it is necessary to discuss how to realize the coupling structure design of the large coupling feed.

$$Q_e = \frac{1}{FBW \times M_{s0}^2}. \tag{5}$$

In order to solve this problem, this paper introduces a new type of joint structure, the model of which is shown in Fig. 5. We can see that the whole is a dielectric waveguide cavity fed by coaxial taps. A cuboid groove is dug just below the tap, a through hole is used to connect the tap with the groove, and the through hole is covered with metal.

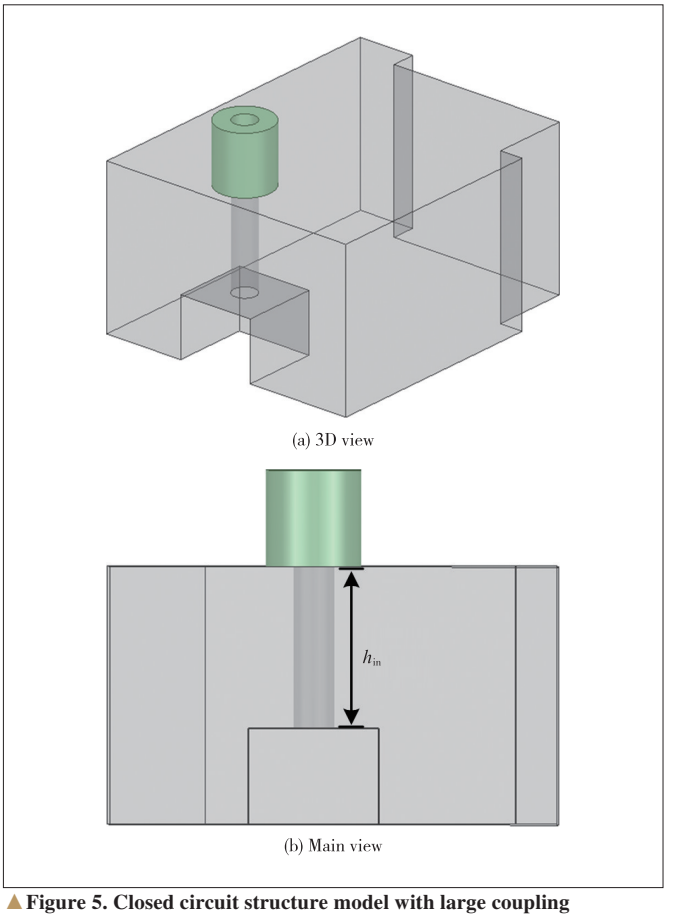
The cavity form from the groove to the tap becomes a quarter wavelength resonant unit. The metal-coated through hole in the inner wall is equivalent to the inner conductor of the coaxial resonant unit. The resonant frequency of the coaxial resonant unit can be adjusted by adjusting h_{in}. The resonant element is essentially the common cavity part of the common cavity type circuit structure, while the rest of the waveguide cavity is not necessary, which can be retained or not retained according to the overall model of the duplexer. Although eliminating redundant waveguide cavities can effectively reduce

▼Table 3. Values of the coupling matrix

	Initial	First	Second	Final
M(s,0)	1.400 0	1.388 4	1.374 2	1.391 0
M(0,1)	0.658 7	0.704 4	0.690 8	0.673 1
M(0,10)	0.658 7	0.644 9	0.651 8	0.584 2
M(9,L1)	0.501 3	0.517 4	0.517 4	0.501 3
M(16,L2)	0.427 2	0.427 2	0.428 7	0.427 2
M(1,1)	0.708 9	0.661 7	0.661 7	0.656 5
M(1,2)	0.225 4	0.224 5	0.224 5	0.211 1
M(2,2)	0.709 1	0.704 7	0.704 7	0.703 4
M(2,3)	0.092 4	0.091 3	0.091 3	0.092 4
M(2,4)	−0.139 8	−0.145 1	−0.145 1	−0.139 3
M(3,3)	0.490 6	0.478 2	0.478 2	0.489 8
M(3,5)	0.083 2	0.080 0	0.080 0	0.083 0
M(4,4)	0.805 4	0.798 5	0.798 5	0.804 0
M(4,5)	0.130 9	0.137 2	0.137 2	0.131 0
M(5,5)	0.712 8	0.715 2	0.715 2	0.712 5
M(5,6)	0.155 1	0.158 3	0.158 3	0.155 5
M(6,6)	0.714 1	0.717 4	0.717 4	0.714 0
M(6,7)	0.086 5	0.084 8	0.084 8	0.086 5
M(6,8)	0.132 2	0.139 3	0.139 3	0.132 1
M(7,7)	0.489 3	0.476 9	0.476 9	0.489 2
M(7,9)	−0.131 4	−0.133 2	−0.133 2	−0.132 0
M(8,8)	0.822 1	0.821 4	0.821 4	0.822 0
M(8,9)	0.183 1	0.190 8	0.190 8	0.183 2
M(9,9)	0.709 3	0.704 9	0.704 9	0.708 7
M(10,10)	−0.790 9	−0.790 9	−0.743 2	−0.748 4
M(10,11)	0.163 8	0.163 8	0.178 5	0.155 9
M(11,11)	−0.791 1	−0.791 1	−0.790 2	−0.787 7
M(11,12)	0.122 2	0.122 2	0.130 5	0.122 0
M(12,12)	−0.791 6	−0.791 6	−0.792 7	−0.790 8
M(12,13)	0.070 7	0.070 7	0.071 7	0.092 0
M(12,14)	−0.091 9	−0.091 9	−0.097 2	0.070 7
M(13,13)	−0.647 5	−0.647 5	−0.642 7	−0.647 3
M(13,15)	0.075 1	0.075 1	0.077 3	0.075 0
M(14,14)	−0.879 1	−0.879 1	−0.884 2	−0.878 9
M(14,15)	0.096 3	0.096 3	0.107 9	0.096 35
M(15,15)	−0.791 1	−0.791 1	−0.793 5	−0.791 0
M(15,16)	0.163 8	0.163 8	0.170 4	0.163 8
M(16,16)	−0.791 0	−0.791 0	−0.783 5	−0.790 8

the volume of the joint structure, in most cases, the coupling between the common cavity and the channel filter still needs to be realized by window coupling, and the physical connection between the joint structure and the channel filter can be realized by reserving the waveguide cavity.

In addition to the coupling between the source and the common cavity, the coupling bandwidth between the common cavity and the channel filter is also relatively high. Large inter-cavity coupling of waveguide cavities located in the same layer is easy to achieve by opening windows, but for duplexers

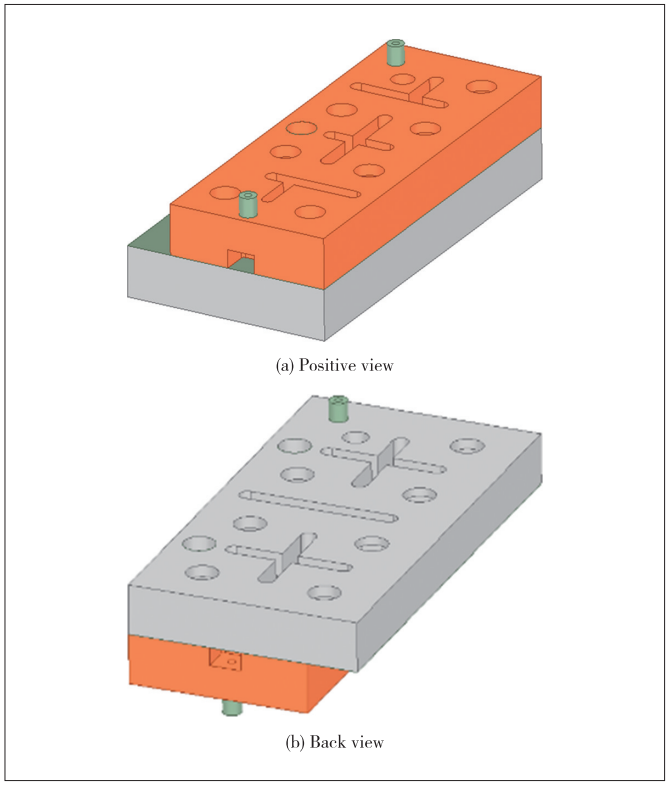


▲Figure 5. Closed circuit structure model with large coupling

with more orders, placing all cavities in the same layer will lead to too much device area. Therefore, in practice, a two-layer structure is preferred.

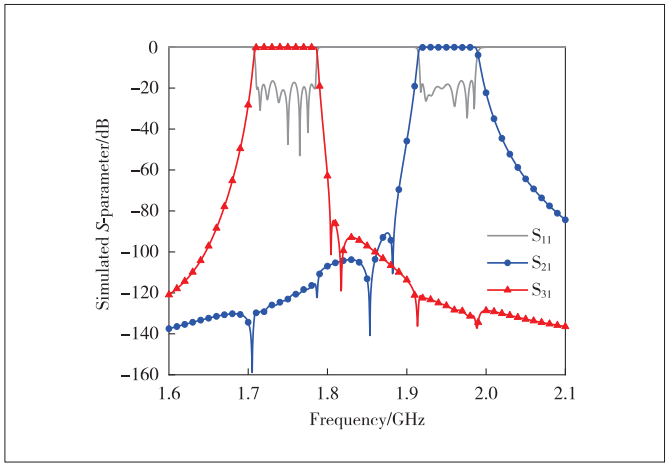
After the coupling structure of the feed part is obtained, the overall model of the duplexer is simulated as is shown in Fig. 6.

We can see that the waveguide duplexer is filled with ceramic, and the whole duplexer is divided into two layers, in which the joint structure is located in the first layer. In order to avoid this problem, the order of channel 1 is increased to 9 in this scheme. As shown in Fig. 6(a), the input port, namely the junction structure, is located at the window between the two cavities. Its left and right sides are the first cavity of channel 2 and the first cavity of channel 1 respectively. The coupling amount of the common cavity to the two channels can be adjusted by the size of the window and the relative position of the input port from the two ports in the horizontal direction. The cavities 2 – 7 of channel 2 are located in the first layer, and the cavities 2 – 9 of channel 1 are located in the second layer. The inter-cavity coupling between cavity 1 and cavity 2 in channel 1 requires a small coupling bandwidth, which is achieved by opening a circular window between layers. The simulation results of the model are shown in Fig. 7. The simulation results in Fig. 7 show that the design meets the require-



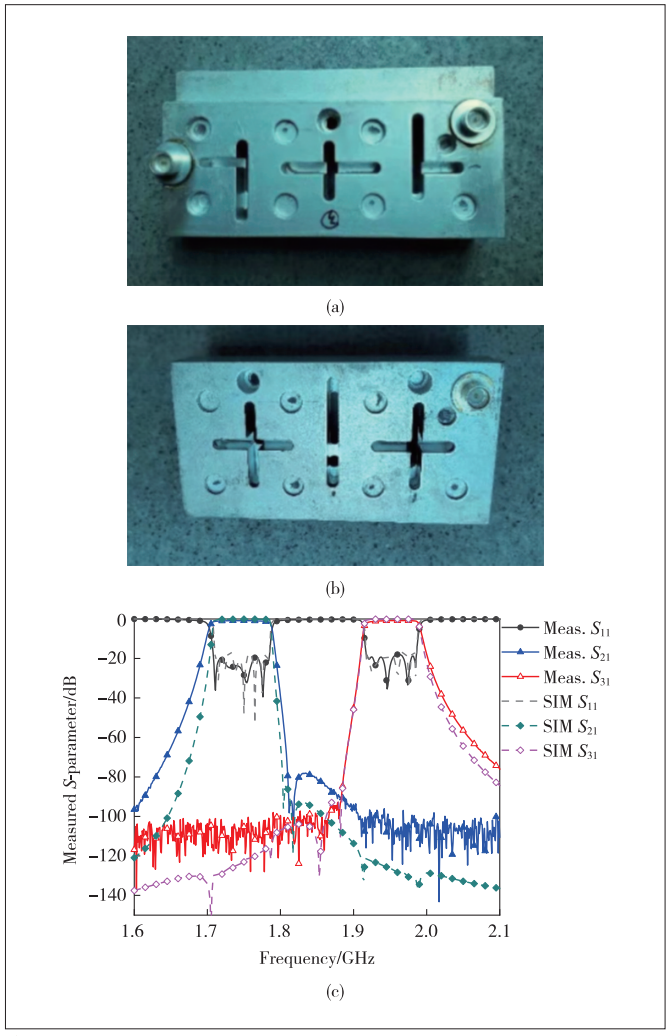
▲ **Figure 6. Simulation model of double-layer duplexer**

ments of the indicators.



▲ **Figure 7. Electromagnetic simulation (EM) simulation results of the model**

Finally, we made physical processing of the simulated model and measured the processed physical object. The finished product is shown in Figs. 8(a) and 8(b). The comparison between the frequency response results and the simulation results is shown in Fig. 8(c) and the optimization results we hope to obtain can be referred to in Fig. 4(d). Among them, the measured results of the duplexer are solid lines in the Fig. 8 (a), which meet the requirements of the index, and are basically consistent with the simulation results represented by dotted lines, which verifies the effectiveness of the algorithm and the circuit combination structure proposed in this paper.



▲ **Figure 8. (a) Positive view, (b) back view, and (c) comparison between the measured results and the simulation results**

4 Conclusions

In this paper, a star junction duplexer synthesis method based on the adaptive differential evolution algorithm (SADE) and LM optimization algorithm is proposed. As a global optimization algorithm, the adaptive differential evolution algorithm can effectively avoid the convergence of optimization results to local solutions, while the LM algorithm as a gradient optimization algorithm can not only accelerate the optimization speed, but also make the results more in line with the requirements of the index. In order to verify the effectiveness of the algorithm, a duplexer with large port coupling is designed, and the structure of realizing large port coupling is also given in this paper.

References

[1] RHODES J D, LEVY R. A generalized multiplexer theory [J]. IEEE transactions on microwave theory and techniques, 1979, 27(2): 99 – 111. DOI: 10.1109/tmtt.1979.1129570
[2] RHODES J D, LEVY R. Design of general manifold multiplexers [J]. IEEE transactions on microwave theory and techniques, 1979, 27(2): 111 – 123. DOI: 10.1109/ tmtt.1979.1129571
[3] MACCHIARELLA G, TAMIAZZO S. Novel approach to the synthesis of microwave duplexers [J]. IEEE transactions on microwave theory and techniques, 2006, 54(12): 4281 – 4290. DOI: 10.1109/TMTT.2006.885909
[4] MENG H, WU K L. Direct optimal synthesis of a microwave bandpass filter with general loading effect [J]. IEEE transactions on microwave theory and techniques, 2013, 61(7): 2566 – 2573. DOI: 10.1109/TMTT.2013.2264682
[5] ZHAO P, WU K L. An analytical approach to synthesis of duplexers with an optimal lumped-element junction model [C]//2014 IEEE MTT-S international microwave symposium (IMS2014). IEEE, 2014: 1 – 3. DOI: 10.1109/MWSYM.2014.6848399
[6] ZHAO P, WU K L. An iterative and analytical approach to optimal synthesis of a multiplexer with a star-junction [J]. IEEE transactions on microwave theory and techniques, 2014, 62(12): 3362 – 3369. DOI: 10.1109/TMTT.2014.2364222
[7] LIU B, YANG H, LANCASTER M J. Synthesis of coupling matrix for duplexers based on a self-adaptive differential evolution algorithm [J]. IEEE transactions on microwave theory and techniques, 2018, 66(2): 813 – 821. DOI: 10.1109/tmtt.2017.2772855Jan. 2018.
[8] YU Y, LIU B, WANG Y, et al. A general coupling matrix synthesis method for all-resonator duplexers and multiplexers [J]. IEEE transactions on microwave theory and techniques, 2020, 68(3): 987 – 999. DOI: 10.1109/TMTT.2019.2957430
[9] SKAIK T F, LANCASTER M J, HUANG F. Synthesis of multiple output coupled resonator circuits using coupling matrix optimisation [J]. IET microwaves, antennas & propagation, 2011, 5(9): 1081. DOI: 10.1049/iet-map.2010.0447
[10] XIA W L. Duplexers and multiplexers design by using coupling matrix optimization [D]. Birmingham, U.K.: Birmingham University, 2015
[11] CAMERON R J, KUDSIA C M, MANSOUR R R. Microwave filters for communication systems [M]. Hoboken, USA: John Wiley & Sons, Inc., 2018. DOI: 10.1002/9781119292371
[12] MATLAB. MATLAB optimization toolbox. user's guide [EB/OL]. (2018-05-12) [2021-09-18]. https://www.mathworks.com

Biographies

WU Qingqiang received his BS degree in electromagnetic field and wireless technology and MS degree in electronic science and technology from Xidian University, China in 2019 and 2022, respectively. He joined ZTE Corporation in 2022 after graduation,. During his postgraduate study, he mainly engaged in the research related to filters in the National Key Laboratory of Antennas and Microwave Technology at Xidian University, and participated in the publication of papers in academic journals and international conferences and completed a number of filter related projects. In addition, he won the first-class scholarship of the university for many times, and won the second prize of northwest division in the 15th China Graduate Electronic Design Competition.

CHEN Jianzhong (jianzhong.chen@xidian.edu.cn) received his BE degree in information engineering from Xi'an Jiaotong University, China in 2007 and the PhD degree in microwave technology from Xidian University, China in 2013. Currently, he is working with the Key Laboratory of Antennas and Microwave Technology at Xidian University as a professor. His research interests include electromagnetic compatibility, design of antennas, and microwave circuits.

WU Zengqiang received his BE degree and MS degree in microwave technology from Xidian University, China in 2015 and 2018, respectively. Currently, he is an RF engineer in ZTE Corporation. His research interests include microwave engineering design, modeling, and optimization.

GONG Hongwei joined ZTE Corporation as an RF&MW filter design director in 2015. He was in charge of all the RF filter design in ZTE and has made excellent progress in this domain. His research interests include electromagnetic compatibility, optimization and microwave device design.

Alarm-Based Root Cause Analysis Based on Weighted Fault Propagation Topology for Distributed Information Network

LYU Xiaomeng[1], CHEN Hao[1], WU Zhenyu[1], HAN Junhua[2], GUO Huifeng[2]

(1. Engineering Research Center for Information Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China; 2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202203010

https://kns.cnki.net/kcms/detail/34.1294.TN.20220805.1618.002.html, published online August 8, 2022

Manuscript received: 2021-10-31

Abstract: A distributed information network with complex network structure always has a challenge of locating fault root causes. In this paper, we propose a novel root cause analysis (RCA) method by random walk on the weighted fault propagation graph. Different from other RCA methods, it mines effective features information related to root causes from offline alarms. Combined with the information, online alarms and graph relationship of network structure are used to construct a weighted graph. Thus, this approach does not require operational experience and can be widely applied in different distributed networks. The proposed method can be used in multiple fault location cases. The experiment results show the proposed approach achieves much better performance with 6% higher precision at least for root fault location, compared with three baseline methods. Besides, we explain how the optimal parameter's value in the random walk algorithm influences RCA results.

Keywords: distributed information network; alarm; graph; root cause analysis; random walk

Citation (IEEE Format): X. M. Lyu, H. Chen, Z. Y. Wu, et al., "Alarm-based root cause analysis based on weighted fault propagation topology for distributed information network," *ZTE Communications*, vol. 20, no. 3, pp. 77 - 84, Sept. 2022. doi: 10.12142/ZTECOM.202203010.

1 Introduction

Distributed information networks have been widely used in the Internet, government, military and other important fields because of its reliability, scalability, resource sharing and high performance. However, due to its large-scale system configuration, complex graph structure and operation logic, the frequent occurrences of faults and fault propagation increase the difficulties for locating faults' root causes and troubleshooting the distributed information network.

In recent years, many root cause analysis (RCA) methods have been proposed, which can be divided into two types: knowledge-based and data-driven methods.

1) Knowledge-based: The fault diagnosis methods based on the rules of knowledge generally use the expert experiences to guide the fault diagnosis. ZENG et al.[1] constructed fault reasoning rules with the empirical knowledge of IT operation and maintenance, and then built fault trees to deduce fault root causes. The authors in Ref. [2] proposed an RCA tool inspired by the pattern matching technology. This tool uses the automata built online and the space-time causal relationship between the symbols observed in the log is stored. Its construction does not need annotation and has some interpretability. However, it cannot be used directly and flexibly because of a complex structure.

2) Data-driven: These methods are implemented by multiple technologies including machine learning, causality graph and real graph.

• Machine learning: Bayesian networks (BN) are often used for fault root cause analysis because they contain causal information. LIU et al.[3] proposed a BN construction algorithm based on the alarm seriality, which could reduce the alarm preprocessing time while considering the effectiveness. However, training the network needs a large amount of labeled data to improve the performance generalization of the model. ZHANG et al.[4] trained an attention based autoencoder to predict fault signals. In the case of no labeled samples, this method considered the time dependence, but it is difficult to explain the fault mechanism to some extent.

• Causality graph: A causality graph is a graph based on event co-occurrence or conditional independence test with each event as a node. It locates the root causes by random walk in a causality graph. KALANDER et al.[5] proposed an embedding algorithm based on a causal propagation graph to

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. HC-CN-20201120009.

infer the weight of the edge, and applied the impact maximization algorithm to determine the root cause alarm. Although it explains the fault mechanism between alarms, the trimming of opposite edges in causal graphs usually requires some expert experience and does not adapt well in a variety of scenarios.

• Real graph: It is more intuitive for random walk in a real relationship graph that is not like the causality graph. ZHAO et al.[6] used performance indicators such as key performance indicators (KPIs) to calculate the similarity of the edges in an anomaly propagation graph, formed the transition probability matrix, and located the fault root by random walk. This method requires such a large amount of performance indicator data for calculation and analysis that the RCA takes a long time.

Compared with the traditional methods based on empirical knowledge, the data-driven methods can better realize real-time analysis with more accuracy and do not need to be greatly adjusted due to the updates of environment configuration. However, the existing data-driven methods often need a large amount of labeled data for supervised training[7]. TraceRCA[8] mined the suspicious nodes by KPIs, which could reduce the locating noise. It inspired us to propose the idea of locating the root causes by alarms. Because common alarms cannot be used to mine more fault root cause information. ZHANG et al.[9] proposed the anomaly propagation graph using system data and used two optional algorithms to locate root causes. This inspires us to construct the fault propagation with alarms to explain the mechanism of fault propagation. Those methods without graphs cannot intuitively explain the mechanism of fault propagation. One the other hand, the other methods of using constructed fault propagation graphs are almost based on KPIs[10 - 12] or other metrics collected from the database. However, these methods have to use acquisition tools and set the collection locations to acquire various kinds of data, which may cost too much labor. A causal graph in alarms also needs expert experience, which cannot adapt well in distributed environments with frequent updates.

For the above deficiencies, we propose an alarm-based method for root cause analysis of distributed information networks based on a weighted fault propagation topology (WFPT-RCA). It is inspired by the previous work, mainly Refs. [8 - 9]. It trains the classifier using a few historical labeled alarms to mine the effective information of root causes. When a fault occurs, based on the character of alarms, the WFPT-RCA immediately extracts a subgraph from the real graph of the distribute network. Then combined with the information of root causes and alarms' features, our method calculates the weights of nodes and edges in the subgraph. Based on the random walk in the weighted subgraph, it not only explains the behaviors of fault propagation, but also outputs the nodes' list about root causes' scores to help operators to repair the fault. We evaluate WFPT-RCA in two datasets in different scenarios (an e-commerce platform and a transport network). The results show that WFPT-RCA achieves a good performance result, with 90% in precision and 92.7% in mean average precision. It outperforms several other state-of-the-art methods.

In summary, the contributions of this paper are threefold:

1) We propose a two-stage RCA approach. In the offline phase, a few labeled alarms are used to train the classifier for digging more information associated with root causes in order to guide the fault location in the online phase.

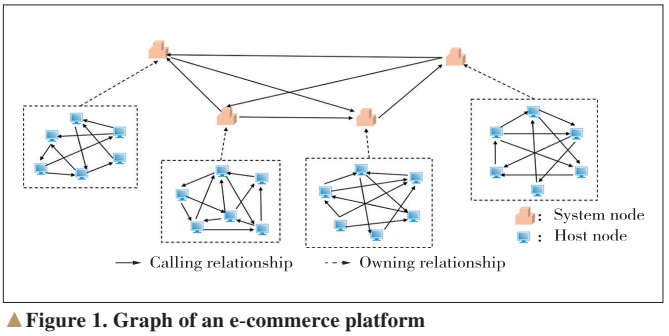
2) We provide a method based on alarms to calculate the nodes' weights as the scores of root causes and edges' weights as the probabilities of the fault propagation in the real graph which adapts well in distribute information network.

3) We evaluate WFPT-RCA in two datasets. The results demonstrate that WFPT-RCA localizes root causes correctly and has a better generalization ability. Our method pays more attention to features related to root causes and does not rely on the experience knowledge of operators.

The remaining of this paper is organized as follows. The framework and details of WFPT-RCA are mainly introduced in Section 2. In Section 3, we show the related experiments' results and conclusion analysis to prove the efficiency of our approach. Finally, Section 4 concludes the paper.

2 Framework of WFPT-RCA

Static topological relationships in a distributed information network are often complex and hierarchical (Fig. 1). An e-commerce platform is often composed of multiple system nodes to achieve efficient work. And there are more host nodes that belong to the system nodes to offer different services. The real lines in nodes represent the calling relationships between the nodes, while the dashed lines represent the owning relationships between system nodes and host nodes. Similarly, Fig. 1 can also be regarded as a graph of the transport network where the host nodes can be represented as the network element (NE) and the links and pseudo-wires are expressed as real edges. Moreover, the transport network includes the core layer, convergence layer and access layer. There are various NEs to transmit data through multiple links in each layer to represent the hierarchy of graph. In real scenarios, such complex and hierarchical relationships often lead to faults due to resource usage and response timeout of a system node. If we directly locate faults based on performance



▲Figure 1. Graph of an e-commerce platform

indicators in the original graph, noise interference may occur, resulting in low accuracy. Meanwhile, alarms usually reflect node status. Using alarms to identify abnormal nodes in the graph and extract abnormal subgraphs, noise interference can be reduced and fault location accuracy can be improved.

The framework of WFPT-RCA is shown in Fig. 2, which is mainly divided into offline analysis and online diagnosis. We make full use of the collected and labeled historical alarms of each fault event. Taking the occurrence location as the research object, feature extraction is carried out for the alarms in each location. The root location is identified by the binary classifier training model, and the key features are determined by feature importance analysis. In the online phase, alarms and network graph configuration data are firstly collected if the fault occurs after the fault work order is obtained from the operators. After the features of the nodes where alarms have occurred in the offline phase are extracted from the alarms, an abnormal subgraph (ASG) based on the location of the alarm and an original network graph are extracted and the weights of nodes and edges based on the alarm features of nodes are then calculated to generate a weighted abnormal subgraph called Weighted Fault Propagation Graph. Then, a random walk is carried out in ASG. After iteration convergence, the node with the highest score is output and regarded as the root node according to the ranking of root cause score of each abnormal node.

2.1 Data Collection

The collected data are mainly from the alarms and graph generated in the distributed information network. After a system fault occurs, a surge in the number of alarms occurs within a few minutes, namely alarm storms[13]. In the online phase, we collect statistics on the number, type and severity of alarms generated in the distributed system every minute. According to the occurrence time sequence, WFPT-RCA constitutes the corresponding time series, respectively adopting S-H-ESD anomaly detection[14] to find outlier points and integrating the occurrence time corresponding to detected outlier points, so as to determine the occurrence time range of faults. The graph is usually extracted from system configuration data when a fault occurs. It analyzes the owning and association relationships of each location based on the location where an alarm occurs.

2.2 Feature Analysis

Feature analysis is to mine and analyze the alarm information at the offline stage and find the features related to the root cause. It is mainly divided into four steps: data cleaning, feature extraction, classifier training and feature importance analysis.

1) Data cleaning. WFPT-RCA first collect the alarms based on the operators' fault repair experience and fault work in order to obtain the labeled alarm dataset. The content of the alarms is mainly consisted of the timestamp, location and rich concrete content. Alarm pretreatment is a usual practice to enable the alarm content to become a standard template, such as removal of the IP address and request ID. This approach can reduce the alarm type space and noise, and facilitate subsequent cutting word analysis. The content of the warning words is cut to get rid of some stop words such as "for" and "is", and then text information will be extracted more accurately.

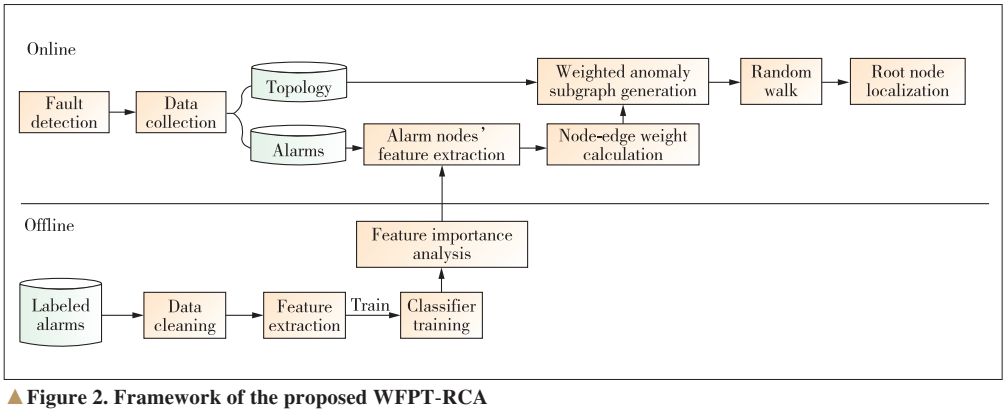
2) Feature extraction. Main features include text, frequency and time.

• Text: After cutting alarm words, we analyze the alarm information based on words to find important words related to faults. Inverse Document Frequency (IDF)[15] is a key feature used to measure the importance of words in text mining, reducing the weight of frequent words and increasing the weight of unfamiliar words; $\mathrm{IDF}(w) = \log\left[N/(N_w + 1)\right]$, where N is the total number of words in all alarms and N_w is the number of alarms containing the word w. After the IDF for the words contained in each alarm is calculated, the information entropy of each alarm will be calculated as $\sum_m IDF(\mathrm{w})/m$, where m is the total number of words in each alarm.

• Frequency: This feature is extracted based on the statistics for the number of alarms, the total number of species, the number of alarms per minute on average from every node, the number of serious alarms and so on. More serious faults and richer node types are to determine a more serious alarm type, such as failure and downtime.

• Time: The occurrence of faults often has a certain time rule. Therefore, the statistics on relative occurrence time (the time difference between the earliest alarm of a node and the earliest alarm of a fault event) and on alarm duration of each node is collected.

3) Classifier training. Based on the occurrence location, WFPT-RCA inputs the extracted alarm features with the la-



▲ **Figure 2. Framework of the proposed WFPT-RCA**

bels 0 (not root cause) and 1 (root cause) into XGBoost[16] for training until the model has the optimal effect to classify the root cause samples.

4) Feature importance analysis. When we train the binary XGBoost model, the importance of features can be analyzed in the meantime. It is implemented by employing the F score to evaluate the influence of each feature in the dataset on classification decision. The F score is used to measure the discrimination ability of the features to model classification. The higher the F score is, the stronger the distinguishing ability of the feature is. Moreover, the results of feature importance will play a great role in the subsequent root location.

2.3 ASG Generation

As shown in Fig. 3, ASG is constructed based on the actual graph of the distributed information network. Due to the nature of alarms, we select the set of candidate abnormal nodes $V_a = \{v_{a1}, v_{a2}, \cdots, v_{an}\}$, where n is the number of abnormal nodes and v_{a1} is one of the anomaly nodes. The filter rules are based on whether alarms are generated at each location in the graph during the fault occurrence. The ASG is expressed as ASG(V_a, E), where E is the set of e_{ij} that shows the directed real edge where v_{ai} points to v_{aj}. V_a and E have different physical meanings in different distributed information networks, which can assign different meanings to them based on the graph and alarm location. The ASG corresponding to each fault event varies according to the locations of the alarms. The weights of the nodes and edges of the extracted ASG must be defined to provide physical significance in the scenario of root cause locating and more explanatory for root cause diagnosis. The following is the definitions:

1) Node weight w_v: It calculates nodes' weights based on the alarms of nodes. It can be regarded as the initial root cause score of node failure. The weight of v_{ai} is calculated as follows:

$$w_{vi} = \theta_1 \cdot f_i(1) + \theta_2 \cdot f_i(2) + \cdots + \theta_l \cdot f_i(l), \tag{1}$$

where l is the number of features, k is the k-th feature of the feature set, $k \in [1, l]$, and θ_k and f_k are respectively the normalized feature importance score and the value of k. Finally, all calculated node weights are normalized again. The larger the weight value is, the higher the empirical root score or probability value of the node is considered.

2) Edge weight w_{ij}: It is the weight of the edge between v_{ai} and v_{aj}. The calculation formula is:

$$w_{ij} = \max \left| \mathrm{corr}\left(f_i(k), f_j(k) \right) \right|, \tag{2}$$

where corr(·) is Pearson correlation calculation; $w_{ij} \in [0, 1]$ and its physical meaning is the probability of fault propagation, which is the maximum similarity degree of each feature between nodes. That is, if there are edges between a node and multiple nodes, by calculating the weights of all adjacent edges connected, it can be considered that the edge with a larger weight is more likely to have fault propagation. The edge weights are calculated in order to construct the transition probability matrix in the random walk. By calculating the weights of nodes and edges, we obtain the weighted ASG. The specific process is shown in Algorithm 1.

Algorithm 1 : Weighted ASG

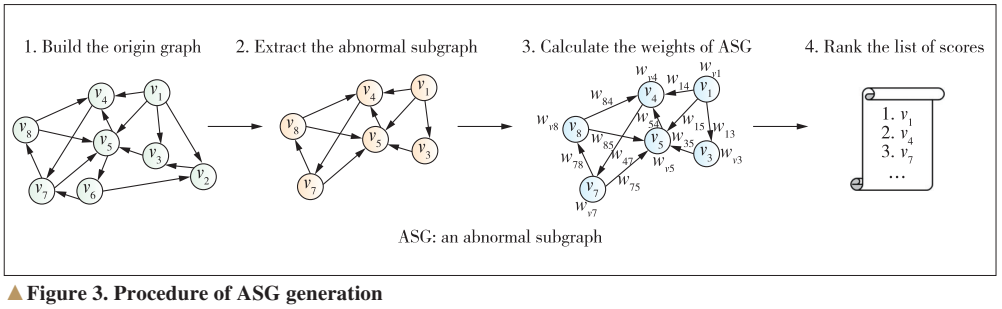
Input: anomalous subgraph ASG, anomalous edge set E, anomalous node set V_a, alarm feature vector $\boldsymbol{f}$, and weight parameters of feature importance obtained by offline training $\boldsymbol{\theta}$
Output: weighted ASG
1: **for** node $\boldsymbol{v}_{aj}$ in V_a **do**
2: Assign $\theta_1 \cdot f_i(1) + \theta_2 \cdot f_i(2) + \cdots + \theta_l \cdot f_i(l)$ to w_{vi};
3: **for** e_{ij} in E of v_{aj} **do**
4: **for** k in l **do**
5: Assign $\left| corr\left(f_i(k), f_j(k) \right) \right|$ to $w_{ij}(k)$;
6: **end**
7: **Assign** $\max(w_{ij}(k))$ to w_{ij};
8: **end**
9: **end**
10: **return** weighted SG

2.4 Root Cause Localization

Root cause localization refers to locating the root node by random walk on the weighted ASG. We define the vector $\boldsymbol{v}$[17] in PageRank as the root score of each node of V_a. Before calculating the root scores, we define the matrix of transition probability $\boldsymbol{P}$ among the nodes of V_a. For instance, v_{ai} points to v_{aj}, and the transfer probability between v_{ai} and v_{aj} is calculated as follows:

$$P_{ij} = \frac{w_{ij}}{\Sigma_j w_{ij}}. \tag{3}$$

If there is no edge between v_{ai} and v_{aj}, $P_{ij} = 0$. $\Sigma_j w_{ij}$ is the sum of the weights of all the out-edges from v_{ai}. The formula of PageRank is shown in Eq. (4).



▲ **Figure 3. Procedure of ASG generation**

$$\boldsymbol{v}_m = \frac{1-q}{n} + q \cdot \boldsymbol{P} \cdot \boldsymbol{v}_{m-1}, \quad (4)$$

where $\boldsymbol{P}$ is the transition probability matrix made up of P_{ij}, n is the number of nodes, q is the damping factor that means that the node jumps back to a random node with the probability of q in each step and continues to advance along the directed edge in the graph with the probability of 1-q, and $\boldsymbol{v}_m$ is the vector composed of root score from each node obtained by iterating m times. Finally, the abnormal nodes are sorted according to the root score to obtain the list. Operators can check and repair alarms reported by the abnormal nodes and their locations in sequence, which improves the locating efficiency and reduces labor costs.

3 Experimental Evaluation

In this section, we mainly introduce the experimental setup, show experimental results, compare the results with other state-of-the-art methods, and analyze the advantages of our method.

3.1 Experimental Setup

In order to verify the effectiveness of the proposed WFPT-RCA, we totally choose two different types of datasets in two distribute scenarios.

The former called Dataset A is adopted in the experiment of an e-commerce platform to release the actual production in a scenario of the real dataset[1] that contains the topological relationship and the alarms of 50 failure events. The topological relationship refers to the invocation relationship data between systems, between systems and hosts, and between hosts. Table 1 lists the format of alarms. Alarms of each fault event are sorted by timestamp and stored in a csv file, in which root cause alarms (system/host/alarm content) are labeled and only one root cause exists.

The latter called Dataset B is from a transport network in the telecommunication system provided by ZTE Corporation. It also has the system configurations to describe the topology relationship and alarms. Unlike the former dataset, its graph includes the NEs, links, tunnels and pseudo-wires, and presents the data transmission in L2/L3VPN. The difference of the two datasets also reflects in the content of alarms: Dataset B has alarm codes and types instead of content as shown in Table 2. In Dataset B, there are 38 fault events and the root cause location (NE) labeled by the operators who have rich experience.

We compare WFPT-RCA with three baseline methods as follows.

1) MicroRCA: It is a way to locate root causes in microservices and uses the metrics to construct the weighted graph for random walk. Different from our method, it uses the anomaly detection confidence to calculate weights of edges in the graph.

2) Microscope[18]: It is another graph-based approach to identify faults in microservice environment. To implement it, we construct the causality graph with alarms and then use cause inference to find the root causes.

3) Association Rules[19]: It is a traditional method to mine the rules between alarms for assisting the operators to locate root causes.

To implement the proposed WFPT-RCA method, we adapt the frequent item mining to outputing the association rules for potential alarms.

3.2 Evaluation Metrics

In order to evaluate the effectiveness of the RCA methods, the following indicators are adopted in the fault event set A:

1) Precision at the top k: The precision is denoted as $PR@k$ which means the real root is in the the top k output results. When k is small, the bigger the value is, the higher the accu-

▼**Table 1. Examples of alarms generated during a fault in Dataset A**

Timestamp	System	Host	Alarm content	Is_root
2019/6/14 1:14	SYS_5	Host_14	I/O wait load exceeds 10% for 15 minutes	0
2019/6/14 1:14	SYS_4	Host _9	The log generates ERROR information	0
2019/6/14 1:14	SYS_9	Host _92	On CPU Steal Time lasts 5 minutes over 10%	0
2019/6/14 1:14	SYS_9	Host _75	Free swap space is less than 50%	0
2019/6/14 1:14	SYS_5	Host _60	The communication on port 80 is abnormal	1
2019/6/14 1:14	SYS_5	Host _76	The upper I/O wait load is greater than 50%	0
2019/6/14 1:14	SYS_4	Host _23	Ping packet loss rate is 100%, and the server breaks down	0
2019/6/14 1:14	SYS_9	Host _75	The Slot00 status of the hard disk is failed	0
2019/6/14 1:14	SYS_9	Host _60	Number of FullGC: 32 (greater than threshold: 10)	0
2019/6/14 1:14	SYS_5	Host _97	Average heap memory usage: 94.61% (greater than threshold: 90%)	0
2019/6/14 1:14	SYS_5	Host _32	Average FullGC time: 2 118 ms (greater than threshold: 1 000 ms)	0
2019/6/14 1:14	SYS_4	Host _3	Nic traffic unknown	0

1. http://www.cnsoftbei.com/plus/view.php?aid=479.

▼Table 2. Examples of alarms generated during a fault in Dataset B

Timestamp	NE	Duration	System Type	Code	Severity	Alarm Type	Root
2020/2/27 10:01	4 167	1 000	4 198	964	1	0	0
2020/2/27 10:01	4 715	12 000	4 590	18 956	2	3	0
2020/2/27 10:01	4 167	15 000	4 197	43	4	0	1
2020/2/27 10:01	4 167	11 000	4 590	18 956	4	3	0
2020/2/27 10:01	4 166	5 000	4 590	18 956	3	3	0
2020/2/27 10:01	4 595	10 000	4 198	964	1	0	0
2020/2/27 10:01	5 496	6 000	4 590	18 956	3	1	0
2020/2/27 10:01	5 497	5 000	4 197	43	4	4	0

NE: network element

racy of location becomes. The detail is shown in Eq. (5).

$$\mathrm{PR@}k = \frac{1}{|A|}\sum_{a\in A}\frac{\sum_{i<k}(R[i]\in v_c)}{(\min(k,|v_c|))}, \quad (5)$$

where $R[i]$ is the results of the top k obtained by root score sorting in each fault event and v_c is a set of real causes in fault events.

2) Mean average precision (MAP): It measures the average location performance of the algorithm and the equation is shown in Eq. (6):

$$\mathrm{MAP} = \frac{1}{|A|}\sum_{a\in A}\sum_{1\leq k\leq N}\mathrm{PR@}k. \quad (6)$$

3.3 Experimental Results

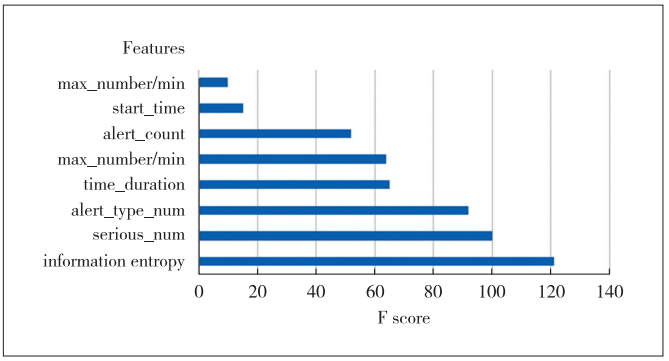
3.3.1 Feature Importance

The details of the offline analysis in Dataset A are represented to show how our method extracts the information of root causes. The alarm features of nodes are extracted from each fault event, and the detailed features and meanings are shown in Table 3.

▼Table 3. Features used for feature importance analysis in Dataset A

Feature	Meaning
information entropy	Average IDF of the system node
max_number/min	Maximum number of alarms per minute
node_num	Number of nodes in same systems
alert_count	Total number of alarms
alert_type_num	Number of alarm types
start_time	Relative start time
time_duration	Time span (minutes)
serious_num	Number of serious type alarms

These features labeled by the root cause of alarms are input into the classifier for training, so as to obtain the analysis results of the feature importance (Fig. 4).



▲Figure 4. Results of feature importance analysis

It shows that the IDF and the number of serious alarms mined from the alarm information are most related to root causes. The information entropy describes the richness of alarm content on each node. A higher value states the more information about root causes in the nodes. Serious alarms usually indicate the severity of faults and the root causes may have more serious alarms. The F score of each feature is normalized and used as the feature weight parameter θ. The parameter not only completes the subsequent node weight calculation that can be seen in Algorithm 1, but also helps us understand the root causes reflected on alarms without the operational experience.

3.3.2 RCA Results

Table 4 shows the performance of the compared methods. WFPT-RCA (no ASG) directly locates root causes without extracting abnormal subgraphs. The compared results prove that the ASG can effectively reduce the noise of fault location and improve the accuracy and efficiency. The results of WFPT-RCA (no feature analysis) illustrate the importance and effectiveness of the offline analysis to obtain the feature weight parameter θ. It also shows that the analysis of feature samples of historical alarms in the offline phase can affect the initial root scores of nodes, thus determining the accuracy of location. MicroRCA is also based on random walk. Different from our method, the prior knowledge is added in the calculation of node edge weights. However, the prior knowledge often does

▼Table 4. Performance in Datasets A and B

Metrics	Dataset A				Dataset B			
	PR@1	PR@3	PR@5	MAP	PR@1	PR@3	PR@5	MAP
WFPT-RCA	**0.90**	**0.92**	**0.96**	**0.927**	**0.89**	**0.95**	**1.00**	**0.947**
WFPT-RCA (no ASG)	0.64	0.70	0.84	0.727	0.53	0.63	0.74	0.633
WFPT-RCA (no feature analysis)	0.28	0.54	0.90	0.573	—	—	—	—
MicroRCA	0.84	0.92	0.94	0.900	0.79	0.84	0.89	0.840
Microscope	0.82	0.88	0.90	0.867	0.74	0.79	0.84	0.790
Association rules	0.36	0.56	0.78	0.567	0.47	0.58	0.63	0.560

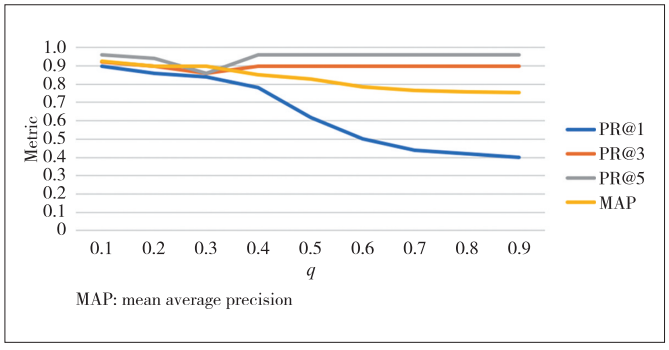
ASG: anomaly subgragh MAP: mean average precision RCA: root cause analysis
PR: precision WFPT: weighted fault propagation topology

not have good generalization and the effect may vary greatly in different scenarios. This shows the operational experience may not adapt well in different distribute information networks. Microscope uses the causal graph to explore the relationship between alarms, so as to locate the root causes. The reason for its unsatisfactory effect is that the nodes downstream of the root cause is often located in the random walk of the causal graph rather than real adjacent nodes. The method lacks of the certain interpretability compared with the fault propagation in a real graph. The performance of the association rules based on frequent item mining mainly lies in the fact that different faults present different behaviors, and the rules are difficult to be used in multiple scenarios. Unless they are updated with the change for environments. Through the comparison in two datasets, it can be found that our method has great advantages in the RCA. Because the features are extracted and analyzed offline, the offline feature analysis effectively reduces the impact of environmental changes on locating accuracy in different scenarios. The method of locating faults based on the real graph as fault propagation is able to help operators understand the propagation way of faults. In a word, WFPT-RCA has wider usage, higher precision, efficient computation and some comprehensibility.

3.3.3 Experimental Results of Parameter Adjustment

Because the damping factor q in PageRank has its unique physical meaning, its value also straightly impacts the metrics of RCA. Therefore, we analyze and evaluate the influence of the value of q on the WFPT-RCA final results in Dataset A.

As can be seen from Fig. 5, the trends of PR@3 and PR@5 are similar, which shows the change in q does not make much difference to them. PR@1 decreases gradually with the increase of the q value until the results of each index reach the optimal level when q = 0.1. We can see that PR@1 decreases obviously at $q \in [0.1, 0.2, 0.3, 0.4]$, which indicates that the transition probability of random jump back to a node has a great influence on fault location. If the q value is too large, it directly interferes with the random walk on the ASG. As a result, the constraints of the real graph on the location result are reduced and the random transfer between nodes plays a leading role in the location. Therefore, we generally keep the q value in the range of 0.1 – 0.15 to ensure that our method achieve better performance.



▲ **Figure 5. Results of each metric for fault location at different q values in Dataset A**

3.4 Discussion

Here we discuss the significance of the proposed approach.

1) Generalization performance: The weighted fault propagation graph is constructed without the operational experience. As system configuration is updated, it does not need to adjust the method completely. In addition, the experimental results in two different datasets also present better adaption. The characteristic reduces the operators' pressure of work and improves the availability of distributed information networks.

2) Intelligibility: Unlike the other compared methods, WPFT-RCA mines the features of root causes from alarms. The alarms directly filter the nodes from the real graph to construct a weighted fault propagation graph, which can decrease the complexity of fault location. Therefore, operators can analyze the behaviors of fault propagation caused by the root cause with the weighted graph. For example, the higher the root score is, the more related the root cause fault is. The larger the edge weight is, the more likely fault propagation will occur. Based on the above rules, it shows that WPFT-RCA has better intelligibility in the cases of fault propagation.

4 Conclusions

In this paper, we propose an alarm-based method for root cause analysis of distributed information networks based on a weighted fault propagation topology, which is constructed in real graph relationship and calculates weights of nodes and edges in the ASG by the features using historical offline analysis. The experimental results on public datasets in real scenarios show that our method can achieve 90% precision and 92.7% mean average precision. Our method is based on the analysis of historical alarms and real graphs, which can effectively reduce the impact of environmental configuration changes on fault location results. In addition, the location based on real graph helps operators understand the mechanism of fault propagation. Verification in various kinds of large, dynamic environments are our main future work.

References

[1] ZENG M F, XIE P Y. Research on fault location of information system based on CMDB and rule inference [J]. Journal of Guangxi academy of sciences, 2017, 33 (1): 53 – 58. DOI: 10.46960/2658-6754_2019_3_4

[2] BOUILLARD A, BUOB M-O, RAYNAL M, et al. Log analysis via space-time pattern matching [C]//14th International Conference on Network and Service

Management (CNSM). IEEE, 2018: 303 – 307
[3] LIU M L, QI X G, LIU L F, et al. Roots-tracing of communication network alarm: A real-time processing framework [J]. Computer networks, 2021, 192: 108037. DOI: 10.1016/j.comnet.2021.108037
[4] ZHANG C X, SONG D J, CHEN Y C, et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data [C]//33th AAAI Conference on Artificial Intelligence. AAAI, 2019: 1409 – 1416. DOI: 10.1609/aaai.v33i01.33011409
[5] ZHANG K L, KALANDER M, ZHOU M, et al. An influence-based approach for root cause alarm discovery in telecom networks [C]//Service-Oriented Computing ICSOC 2020 workshops, 2021: 124 – 136. DOI: 10.1007/978-3-030-76352-7_16
[6] ZHANG L Y, ZHAO J B, ZHANG M. Root cause analysis of concurrent alarms based on random walk over anomaly propagation graph [C]//IEEE International Conference on Networking, Sensing and Control. IEEE, 2020: 1 – 6. DOI: 10.1109/ICNSC48988.2020.9238084
[7] YUAN Y N, YANG J L, DUAN R, et al. Anomaly detection and root cause analysis enabled by artificial intelligence [C]//IEEE Globecom Workshops. IEEE, 2020: 1 – 6. DOI: 10.1109/GCWkshps50303.2020.9367508
[8] LI Z Y, CHEN J J, JIAO R, et al. Practical root cause localization for microservice systems via trace analysis [C]//29th International Symposium on Quality of Service(IWQOS). IEEE, 2021: 1 – 10. DOI: 10.1109/IWQOS52092.2021.9521340
[9] ZHANG L Y, ZHAO J B, ZHANG M. Root cause analysis of concurrent alarms based on random walk over anomaly propagation graph [C]//IEEE International Conference on Networking, Sensing and Control. IEEE, 2020: 1 – 6. DOI: 10.1109/ICNSC48988.2020.9238084
[10] SHARMA B, JAYACHANDRAN P, VERMA A, et al. CloudPD: problem determination and diagnosis in shared dynamic clouds [C]//43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 2013: 1 – 12. DOI: 10.1109/DSN.2013.6575298
[11] LIN J Y, ZHANG Q, BANNAZADEH H, et al. Automated anomaly detection and root cause analysis in virtualized cloud infrastructures [C]//IEEE/IFIP Network Operations and Management Symposium. IEEE, 2016: 550 – 556. DOI: 10.1109/NOMS.2016.7502857
[12] CHEN P F, QI Y, HOU D. CauseInfer: automated end-to-end performance diagnosis with hierarchical causality graph in cloud environment [J]. IEEE transactions on services computing, 2019, 12(2): 214 – 230. DOI: 10.1109/TSC.2016.2607739
[13] ZHAO N W, CHEN J J, PENG X, et al. Understanding and handling alert storm for online service systems [C]//42nd International Conference on Software Engineering: Software Engineering in Practice. ACM, 2020: 162 – 171. DOI: 10.1145/3377813.3381363
[14] GOLIĆ M, ŽUNIĆ E, ĐONKO D. Outlier detection in distribution companies business using real data set [C]//18th International Conference on Smart Technologies. IEEE, 2019: 1 – 5. DOI: 10.1109/EUROCON.2019.8861526
[15] MANNING C D, RAGHAVAN P, SCHUTZE H. Introduction to information-retrieval [J]. Information retrieval, 2010, 13: 192 – 195. DOI: 10.1007/s10791-009-9115-y
[16] CHEN T Q, GUESTRIN C. XGBoost: a scalable tree boosting system [C]//22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016: 785 – 794. DOI: 10.1145/2939672.2939785
[17] WU L, TORDSSON J, ELMROTH E, et al. MicroRCA: root cause localization of performance issues in microservices [C]//IEEE/IFIP Network Operations and Management Symposium. IEEE, 2020: 1 – 9. DOI: 10.1109/NOMS47738.2020.9110353
[18] LIN J J, CHEN P F, ZHENG Z B. Microscope: pinpoint performance issues with causal graphs in micro-service environments [C]//ICSOC 2018: Service-Oriented Computing. ICSOC, 2018: 3 – 20. DOI: 10.1007/978-3-030-03596-9_1
[19] HRYCEJ T, STROBEL C M. (2008) Extraction of maximum support rules for the root cause analysis [M]//Computational Intelligence in Automotive Applications. Berlin Heidelberg, Germany: Springer. 2008: 117 – 131. DOI: 10.1007/978-3-540-79257-4_6

Biographies

LYU Xiaomeng (lvxiaomeng@bupt.edu.cn) is studying for her master's degree at Beijing University of Posts and Telecommunications (BUPT), China and received her bachelor's degree in information and communication engineering from BUPT in 2019. Her main research interests include fault prediction and fault diagnosis. She has published one paper in disks fault prediction and two patents in the AIOps.

CHEN Hao is studying for his master's degree at Beijing University of Posts and Telecommunications (BUPT), China and received his bachelor's degree in 2020 at the Faculty of the Information and Communication Engineering, BUPT. His main research interests are fault recognition and prediction.

WU Zhenyu received his BS and PhD degrees from Beijing University of Posts and Telecommunications (BUPT), China in 2008 and 2013. He is currently an associate professor of School of Information and Communication Engineering at BUPT. His research interests include AIOps, intelligent fault diagnostics, machine learning and prognostics and health management (PHM) technology.

HAN Junhua received his master's degree from Graduate School of the Chinese Academy of Sciences (now University of the Chinese Academy of Sciences) in 2005. He is currently an engineer of ZTE Corporation. His research interests include intelligent operation and maintenance, intelligent fault diagnosis, knowledge graph and graph neural network.

GUO Huifeng received her master's degree from Huazhong University of Science and Technology (HUST), China. She is currently an engineer of ZTE Corporation. Her research interests include intelligent network and fault management.

Approach to Anomaly Detection in Microservice System with Multi-Source Data Streams

ZHANG Qixun[1], HAN Jing[2], CHENG Li[2],
ZHANG Baisheng[2], GONG Zican[2]
(1. Peking University, Beijing 100091, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202203011
https://kns.cnki.net/kcms/detail/34.1294.TN.20220728.1549.002.html, published online July 28, 2022
Manuscript received: 2022-01-24

Abstract: Microservices have become popular in enterprises because of their excellent scalability and timely update capabilities. However, while fine-grained modularity and service-orientation decrease the complexity of system development, the complexity of system operation and maintenance has been greatly increased, on the contrary. Multiple types of system failures occur frequently, and it is hard to detect and diagnose failures in time. Furthermore, microservices are updated frequently. Existing anomaly detection models depend on offline training and cannot adapt to the frequent updates of microservices. This paper proposes an anomaly detection approach for microservice systems with multi-source data streams. This approach realizes online model construction and online anomaly detection, and is capable of self-updating and self-adapting. Experimental results show that this approach can correctly identify 78.85% of faults of different types.

Keywords: anomaly detection; data stream; microservice; monitored indicator; system log

Citation (IEEE Format): Q. X. Zhang, J. Han, L. Cheng, et al., "Approach to anomaly detection in microservice system with multi-source data streams," *ZTE Communications*, vol. 20, no. 3, pp. 85 – 92, Sept. 2022. doi: 10.12142/ZTECOM.202203011.

1 Introduction

In recent years, the microservice architecture has been widely used in enterprises. Its core ideas are fine-grained module division, service-oriented interface encapsulation, and lightweight communication interaction. The architecture splits a tightly coupled application into several independent services that have their own functions and run in independent development and deployment processes. The services coordinate and cooperate with each other based on a lightweight communication mechanism. Compared with traditional software systems, microservice systems are characterized by finer granularity towards the division of service, more flexible expansion, more frequent program update iterations, etc. At the same time, in order to improve resource utilization, services are often deployed in a lightweight containerized manner. In the microservice system, besides the defects in an application itself, system failures may often be caused by configuration errors and resource contention problems. When a failure inside or outside the system is activated, it may cause errors and failures, which will further spread between services to produce a chain reaction, affecting the service performance or even making it impossible to run the service normally.

Existing microservice anomaly detection approaches often acquire the behavior features of the system through analyzing system runtime data such as monitored system indicator data or log data, identifying the abnormal behavior of the system, diagnosing the type of system failure, and locating the root cause of the failure. Some methods have key limitations and shortcomings. Firstly, these methods often use offline training and online detection methods, which are not efficient and cannot adapt to system updates or data changes, resulting in poor anomaly detection results. Secondly, the data are often output as a stream when the system is running. The existing methods usually apply batch processing for data analysis, which cannot adapt to the real-time characteristics of streaming data, leading to a high degree of lag in anomaly detection. Therefore, how to process and analyze these data streams and how to construct an online anomaly detection model have become important issues. This paper will focus on how to use both log data and monitored system indicator data for anomaly detection and simultaneously to improve the model's capabilities of self-updating and self-adapting for streaming data.

1) An anomaly detection method with multiple data streams

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. HF-CN-202008200001.

is proposed. Based on the data flow of the runtime system, the microservice anomaly features in the data stream are mined, and online model construction and online anomaly detection are realized with the capability of self-updating and self-adapting.

2) A rule-based fault identification method is proposed, which can synthesize abnormal information online, filter noise and identify faults.

Experiments are conducted in Sock-Shop, an open source microservice application system, to verify the effectiveness of the method in this paper through fault injection. The experimental results show that the proposed method can identify different types of faults with a correctness of over 81%.

2 Related Work

Formerly, anomaly detection is mostly achieved by monitoring indicator data or learning the features of system behavior. The related work can be classified as anomaly detection approaches based on monitored indicators or based on system log analysis. The anomaly detection based on monitored indicators include approaches based on rules, statistical methods, or machine learning. Rule-based approaches usually define rules by analyzing historical data and expert experience, which helps to accurately detect anomalies that meet the rules. However, limited to the fixed rules, it requires the in-time updating of rules. Otherwise, an anomaly belonging to the new cluster would not be able to be detected. Statistical methods-based approaches assume the data obeys a certain distribution, and then use statistical data to estimate, which heavily relies on the assumption. The approaches based on machine learning are usually classified by supervised learning or unsupervised learning. Supervised learning uses plenty of sample data with labels to train a classifier. Unsupervised learning detects the anomaly using mathematical approaches such as distance, density and clustering. To detect anomalies on multiple dimension with causality, an indicator dependency graph can be depicted to discover the abnormal indicator. The graph-based approach generally consists of two steps, graph representation and abnormal indicator detection. Based on observed performance indicators, CloudRanger[1] uses the PC algorithm to construct an influence diagram, then uses Pearson Correlation Function to calculate the correlation between services, and finally uses Customized Second-Order Random Walk Heuristic Survey Algorithm in the influence diagram to detect anomalies. Through causal analysis, MS-Rank[2] extracts the impact diagram between services from various indicators, and then uses Customized Random Walk Algorithm in the impact diagram based on the confidence of service indicators to obtain the abnormal service level to achieve the result.

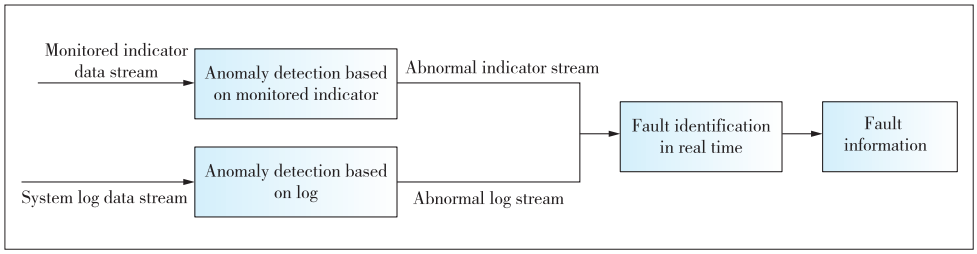
Log-based anomaly detection includes anomaly detection based on graph models, probability distributions, and machine learning[3]. A graph-based anomaly detection technique constructs a model for log sequence relationship, association relationship, and log text content. The anomaly detection based on the probability distribution calculates the correlation probability between the log and the anomaly. The approach based on machine learning is to extract the features of the log, and use machine learning algorithms such as clustering for feature correlation. CHUAH et al.[4] proposed a log diagnosis tool, which extracts log information through a structured template and calculates the similarity of the log to detect the anomaly log. CHEN et al.[5] proposed a log analysis approach that analyzes the trace log of a large-scale system. It aims to calculate and analyze the frequency of the log template in each time window. The time window in which the frequency suddenly changes is a fault window and the corresponding log is an anomaly log. ZHOU et al.[6] proposed an anomaly detection approach for microservice applications called Microservice Error Prediction and Fault Localization (MEPFL), which trains the model through supervised learning, uses the features and injected faults on the tracking log in the system as the training set, and then uses the model in the production environment to capture potential anomalies.

3 Anomaly Detection Based on Multi-Source Data Streams

This section provides a detailed description of the approach proposed in this paper. As mentioned earlier, this approach performs real-time analysis on the multi-source data streams to find anomalies and diagnose the root cause of indicators that characterize anomalies. This approach includes three key steps: the anomaly detection based on monitored indicators, anomaly detection based on system logs, and real-time fault identification (Fig. 1). Anomaly detection based on monitored indicators, which integrates multiple time series models, is responsible for analyzing monitored data streams. The model captures a variety of different features of the monitored indicators, finds abnormal points in the indicator data in an all-direction way, and outputs the abnormal indicator data stream. Log-based anomaly detection is responsible for analyzing the system log stream, constructing a real-time time-weighted control flow, finding different types of anomalies in the log, and outputting the abnormal log data stream. Online fault identification is responsible for integrating the abnormal indicator data stream and the abnormal log data stream, filtering the abnormal noise, and finally identifying the fault and providing feedback on the fault information in real time.

3.1 Anomaly Detection Based on Monitored Indicator Data Stream

Monitored indicators can be formalized as time series streams in the form of $\{x_t\}$, where x is a specific indicator type and t is the corresponding time for collecting. With the most recent T indicator values (that is, selecting a sliding time window with length T), the anomaly detection problem of moni-



▲ **Figure 1. Overview of the proposed approach**

tored indicators can be regarded as a historical time series $\{x_{t-T}, x_{t-T+1}, \cdots, x_{t-1}\}$ with length T to determine whether the current indicator value x_t is abnormal or not.

This paper proposes an anomaly diagnosis approach based on monitored indicators. Specifically, the kernel density estimation and weighted moving average approaches are selected, and the anomaly detection results obtained are quantified and normalized. The final anomaly score is obtained by integration and used for subsequent root cause diagnosis.

Kernel density estimation[7] is a non-parametric test approach, mainly used to estimate the unknown probability distribution of a sample. The probability density function based on the sample frequency is smoothed by the kernel density estimation to obtain the derivable density function. A major advantage of the kernel density estimation is that there is no need to make any assumptions about the distribution of the sample data. In the scenario of monitored indicator anomaly detection, we aim to establish the distribution function model of each indicator through the kernel density estimation of historical data. When a new monitored data point is received, the quantified degree of abnormality can be measured by using the idea of hypothesis testing and verifying the probability that the new data point conforms to the existing distribution function. Specifically, we can estimate a probability distribution $f_X(x)$ from the historical data.

$$f_X(x) = \frac{1}{T-1}\sum_{i=t-T}^{t-1} G(x; x_i). \tag{1}$$

Based on this distribution, we can use hypothesis testing to calculate the degree of anomaly parameter p quantified by the latest indicator value x_t. This value will be used to calculate the overall degree of anomaly of the data point.

Another effective lightweight unsupervised time series predicting approach is the weighted moving average[8]. The main idea is to assign higher weights to the nodes that are closer to the current moment and perform a weighted average, thereby obtaining the predictive value of the current indicator.

$$\widehat{x_t} = \alpha x_{t-1} + \alpha(1-\alpha)x_{t-2} + \alpha(1-\alpha)^2 x_{t-3} + \cdots. \tag{2}$$

We use the difference between the predicted value and the real value at the current moment as an evaluation of the degree of the indicator's anomaly.

The overall degree of the indicator's anomaly (denoted as A) is the weighted integration of the above two statistical values. Before weighting, the above statistical values must be normalized in advance (mapped to the interval of 0 – 1). Then, statistical values are assigned with different weights to obtain the overall anomaly score.

$$A = \omega_1 p_{\text{value}} + \omega_2 \left| x_t - \widehat{x_t} \right|. \tag{3}$$

Indicators with an overall anomaly score higher than the threshold are considered anomaly indicators. These indicators will be performed with subsequent fault identification.

3.2 Anomaly Detection Based on Log Data Stream

This approach converts the log stream into a log template stream, uses a network inference algorithm to construct and update the control flow graph model in real time, and finally detects anomalies in real time based on the control flow graph model.

3.2.1 Time-Weighted Control Flow Graph

The time-weighted control flow graph (TCFG) is a directed graph composed of edges, nodes and time weights. The nodes represent log templates, the edges represent the transfer relationship between log templates, and the time-weight records the transfer time between log templates. The time-weight is calculated by the difference between the timestamps of two adjacent logs of the log sequence belonging to the same request.

The formalized definition of TCFG is as follows:

$$\text{TCFG} = (V, E, W), \tag{4}$$

where $V=\{v_1, v_2, \cdots, v_n\}$ represents the nodes (log templates) in the graph model, and the total number is n. $E=\{e_{ij} | 1 \leqslant i, j \leqslant n\}$ represents the edge from v_i to v_j in the graph model. $W=(w_{ij} | e_{ij} \in E)$ represents the time weight of each edge in the graph model.

The TCFG model describes the request execution logic of the healthy system and is the basis for fault diagnosis. When the system fails, the requested log sequence will show a difference from the TCFG model. For example, a request outputs an ERROR-level log that is not recorded in the TCFG model, which indicates the system has a fault and this fault-sensitive log is accurately located. Furthermore, a TCFG model can diagnose system request latency exceptions at a fine-grained level. When a request latency exception occurs in the system, the execution time between adjacent logs in the same request log sequence increases. By comparing with the time weight in the TCFG model, we can accurately locate the log with high latency where the request latency exception occurs, and even the program fragment.

3.2.2 Anomaly Detection Model Construction

In this paper, the log template mining algorithm, Drain[9], is used to convert the log stream into a log template stream p. The core idea is to use a transition probability function parameter $\alpha_{j,i}$ to model the transition probability-time distribution from template j to template i. The transition probability function is formalized as $f\left(t_i|t_j,\alpha_{j,i}\right)$, representing the probabilities of log template j to log template i appearing at the time t_j and t_i respectively. Through the analysis of the real log data in this paper, a power law distribution is used to fit the function, which is

$$f\left(t_i|t_j,\alpha_{j,i}\right)=\begin{cases}\dfrac{\alpha_{j,i}}{\delta}\left(\dfrac{t_i-t_j}{\delta}\right)^{-1-\alpha_{j,i}} & \text{if } t_j+\delta<t_i\\ 0 & \text{otherwise,}\end{cases} \tag{5}$$

where δ represents the minimum transition time from template j to template i. Based on this function, the occurrence probability of the entire log template stream is calculated. By adjusting the parameters to maximize the occurrence probability of the real log template stream, the log stream is fitted.

In the log template stream p, the occurrence probability of any log template i at time t_i is the sum of the transition probabilities of all previous log templates at time $(t_1,\cdots,t_N|\,t_k\leqslant t_i)$. For any log template transfer $j\rightarrow i$, the probability that the transfer does not occur is $S\left(t_i|t_k,\alpha_{k,i}\right)$ (non-transfer probability).

$$S\left(t_i|t_k,\alpha_{k,i}\right)=1-F\left(t_i|t_k,\alpha_{k,i}\right), \tag{6}$$

where $F\left(t_i|t_k,\alpha_{k,i}\right)=\int_{t_j}^{t_i}f\left(t|t_k,\alpha_{k,i}\right)\mathrm{d}t$. The transfer probability of the log template transfer $j\rightarrow i$ is multiplied by the transfer probability of $j\rightarrow i$ and non-transfer probability towards other log templates $k\rightarrow i$, where $k\in\{1,\cdots,N\},k\neq j,t_k<t_i$ and $A=\{\alpha_{i,i}|i,j=1,\cdots,N,i\neq j\}$.

$$f\left(t_i|t_j,A\right)=f\left(t_i|t_j,\alpha_{j,i}\right)\times\prod_{k:k\neq j,t_k<t_i}S\left(t_i|t_k,\alpha_{k,i}\right). \tag{7}$$

The occurrence probability of the entire log template stream p is

$$f\left(t^{\leqslant T},A\right)=\prod_{t_i\leqslant T}f\left(t_i|t_1,\cdots,t_{N_{t_i}},A\right), \tag{8}$$

which is

$$f\left(t^{\leqslant T},A\right)=\prod_{t_i\leqslant T}\left(\prod_{t_k<t_i}S\left(t_i|t_k,\alpha_{k,i}\right)\times\sum_{j:t_j<t_i}\frac{f\left(t_i|t_j,\alpha_{j,i}\right)}{S\left(t_i|t_j,\alpha_{j,i}\right)}\right). \tag{9}$$

More specific simplification steps can be found in Ref. [10].

Finally, the TCFG model construction is transformed into inferring the most likely graph structure so that the graph structure can fit the log template flow $\boldsymbol{p}$ with the greatest probability. Given a TCFG, the matrix composed of transition probability function parameters between any two log templates in the graph is A. The problem can be formalized as

$$\begin{aligned}&\text{maximize}_A && \log f\left(t,A\right)\\ &\text{subject to} && \alpha_{j,i}\geqslant 0,i,j=1,\cdots,N,i\neq j,\end{aligned} \tag{10}$$

where $A=\left\{\alpha_{j,i}|I,j=1,\cdots,N,i\neq j\right\}$.

This approach uses the random gradient descending for training. In each iteration during the training process, A is updated. The updating calculation is as follows.

$$\alpha_{j,i}^{k}(t)=\left(\alpha_{j,i}^{k-1}(t)-\gamma\nabla_{\alpha_{j,i}}L_c\left(A^{k-1}(t)\right)\right)^{+}, \tag{11}$$

where k is the number of iterations and $\nabla_{\alpha_{j,i}}L_c(\cdot)$ is the gradient of $L_c(\cdot)$. In each iteration, only the TCFG subgraph related to the log template that appears in the current time period is updated. Finally, if the transition probability of the two log templates is high enough, a corresponding edge is added to TCFG.

3.2.3 Anomaly Detection Based on TCFG

The anomaly detection based on the control flow graph identifies the difference between the control flow graph and the log sequence. There are three types of anomalies that serve as the basis for fault diagnosis, including sequence anomalies, redundancy anomalies, and latency anomalies. The sequence anomaly refers to any child node of the log template T that does not appear in the log template sequence in the expected time window t after T. The redundancy anomaly is defined as a new log template that has never appeared in the expected time window after T. The latency anomaly means that the time interval between T and the most recent child node in the log template sequence is greater than that recorded in TCFG.

3.2.4 Real-Time Fault Identification

Our anomaly detection approach, which is based on metrics data and log data, outputs anomaly information in real time. However, due to data noise and the inference accuracy of the algorithm, not all anomalies are system failures. Therefore, a rule-based fault identification approach that combines characteristics including anomaly density and anomaly duration is proposed to determine system failures. The calculation function is expressed as follows

$$f\left(\text{density, time}\right)=\begin{cases}1\\ 0,\end{cases} \tag{12}$$

where 1 signifies a fault has occurred and 0 signifies no fault has occurred. The anomaly density refers to the number of anomalies output in real time based on monitored data and log data over a period of time. It has been verified in Ref. [11] that anomalies with a higher frequency are more likely to characterize a failure. Therefore, higher anomaly density leads to a greater possibility of system failure. The duration of an anomaly is an important factor in determining system faults. Generally, if there is no intervention from external factors, such as manual processing, critical faults will hardly become weaker or disappear over time. On the contrary, some system states often fluctuate instantaneously and these instantaneous fluctuations will produce anomalies that are not system failures.

For these characteristics, two parameter thresholds $\{\gamma, \varepsilon\}$ are set for fault determination. For anomaly density, the number of anomalies per minute is used as the determination parameter. When the parameter value exceeds the threshold, it is determined as a fault. The anomaly distribution is evaluated by the standard deviation of the number of anomalies per minute for each service. If the number of anomalies in the duration exceeds the threshold, it will be determined as a fault:

$$f(\text{density}, \text{distribution}, \text{time}) = \begin{cases} 1, \text{desity} > \gamma \vee \text{time} > \varepsilon \\ 0, \text{otherwise} \end{cases}. \tag{13}$$

3.2.5 Experiment Environment

In order to verify the effectiveness of the proposed approach, we built a microservice system based on Kubernetes as an experimental environment. The hardware platform used in the experiment is 2 Dell R740 Server, configured with 2 Intel Xeon Gold 5220R processors (2.2 GHz, 48 core, and 96 threads), 128 G physical memory, a 4 TB SSD hard disk, and a Gigabit Ethernet card. For each sever , we installed the Ubuntu LTS operating system, created a virtual machine through Kernel-Based Virtual Machine (KVM), and then built the Kubernetes cluster on the virtual machine. The cluster contains 2 master nodes and 3 worker nodes, with the Istio Service Grid System installed. We also deployed supportive software for analysis such as Jaeger, Kiali, Node-exporter, Filebeat, ELK (Elasticsearch, Logstash, and Kibana), Zabbix, and Prometheus for log and monitoring data collection in the Kubernetes cluster. The resource configuration information of the virtual machine used in the experimental environment is shown in Table 1.

We selected the open source microservice application system Sock-Shop as the experimental object. Sock-Shop is an electronic business system that simulates selling socks. The development environment includes Java, Golang and NodeJS. The system is divided into eight application services, including Front-end (user interaction interface), Users (user registration and login), Catalogue (product classification), Carts (shopping cart), Orders (submitting orders), Queue Master (processing order queue), Payment (payment), and Shipping (delivery), besides the database service MongoDB and message middleware service RabbitMQ. Each service mainly communicates and interacts using the HTTP protocol. Thus, the coupling between services is low and the development and deployment are convenient. Sock-Shop has been widely used in Refs. [12 – 13] as a typical representation.

The resource configuration of each application container in the microservice application system Sock-Shop deployed in the cluster is set according to the official reference. The specific configuration information is listed in Table 2.

3.2.6 Fault Injection

In an actual production environment, the failure probability of a running system is extremely low, and the failures are often uncertain. A common approach is to inject specified types of faults into the system to verify its ability of the microservice system to handle failures and observe the operating status of the system. We used a series of tools (Stress-ng, traffic control, etc.) to inject faults into the Sock-Shop system and a load testing tool (Locust) to simulate multiple users sending a series of requests to the system at the same time, real user login, query, order and other operations, and collected logs, metrics and service KPI data generated during the running period.

By investigating the faults that often occur in the microservice system, two representative faults are identified: application faults and system resource faults. The fault description is shown in Table 3.

▼Table 1. Virtual machine resource configuration

Virtual Machine Number	Virtual Machine Function	Resource Configuration	Virtual Machine Location
1	Master 1	8 Core CPU, 16 G Memory, 200 G Disk Space	Server_1
2	Master 2	8 Core CPU, 32 G Memory, 200 G Disk Space	Server_2
3	Worker 1	8 Core CPU, 32 G Memory, 200 G Disk Space	Server_1
4	Worker 2	8 Core CPU, 16 G Memory, 200 G Disk Space	Server_1
5	Worker 3	4 Core CPU, 16 G Memory, 200 G Disk Space	Server_2
6	ELK	8 Core CPU, 32 G Memory, 1 T Disk Space	Server_2
7	Others	4 Core CPU, 8 G Memory, 200 G Disk Space	Server_2

ELK: Elasticsearch, Logstash, and Kibana

▼Table 2. Container resource configuration

	Front-End	User	Catalogue	Carts	Orders	Queue-Master	Payment	Shipping
CPU/m	300	300	200	300	500	300	200	300
Memory/Mi	1000	200	200	500	500	500	200	500

▼Table 3. Two representative fault types in microservice system

Fault Type	Fault Description
Application fault	Caused by null value errors in the code, short-circuit of exception statements, condition reversal, default values missing in switch statements, etc.
System resource fault	Caused by high node CPU utilization, insufficient memory, network delay or packet loss, disk I/O blocking, etc.

3.2.7 Application Faults

Application faults mainly refer to software bugs introduced during the software development process by developers, such as the direct use of uninitialized objects in the code and the incorrect boundary of a conditional statement. When the bugs within the application are activated during the system running, exceptions or even service failures might appear. For application faults, we directly modify the application source code, inject faults into the source code, and trigger them by sending a request to the microservice system. The application faults involve the null value, unexpected value, short-circuit in the exception statement, condition reversed, switch statement lacking a default value, exception uncaught, requested memory unreleased, and middleware upgrade.

• Null value: When the program encounters an uninitialized object during the running period, the error log will be printed if there is a null value judgment statement block; an exception error will be thrown if there is no null value judgment. Therefore, abnormal characteristics will appear in the application log. The corresponding service outputs a failing request.

• Unexpected value: The variable value in the process of a program is of the wrong type. If there is a corresponding type judgment statement block, the error log will be printed; otherwise, an abnormal error will be thrown. Therefore, the abnormal characteristics will be shown in the application log. The corresponding service outputs a failing request.

• Short-circuit in the exception statement: The exception statement in the program is directly triggered and the corresponding exception is thrown. If it is not caught, there will be an error output in the log. If the exception is caught, the program logic will change. The service outputs a request failure or an incorrectly return result.

• Condition reversal: The judgment condition of the conditional judgment statement used in the program running process is reversed. In some cases, it will cause the wrong variable value or even an exception thrown directly. The service outputs a request failure.

• Switch statement lacking a default value: The switch statement used in the running program lacks the default branch and the existing branch cannot cover the current situation. In some cases, it will cause variable value initialization errors, null value errors, etc. The service outputs a request failure.

• Exception uncaught: An undeclared exception is thrown when the program is running, and there is no corresponding capturing and processing statement block in the code. The service outputs a request failure.

• Requested memory unreleased: When the requested memory resources fail to release in the program code, a memory leak occurs. When the memory occupation reaches its upper limit, the process will be killed and the pod restarted. The service will output time-outs or request failures.

• Middleware upgrade: The upgrade of middleware which the application is relied on causes compatibility issues. This further causes system failures or request failures.

3.2.8 System Resource Faults

System resource faults refer to system faults in the actual production environment due to resource contention or incorrect configurations. When this type of fault appears, service response time becomes longer, leading to service instability or unavailability. We use third-party tools to simulate service resource faults in the experiment. To simulate CPU, memory and disk I/O exceptions, we use the open-source tool stress-ng under the Linux operating system to seize the system's CPU, memory, and disk I/O resources. For network anomalies, we use the traffic control command in the Linux system to control network traffic to simulate network delays and network packet loss. The system resource faults include the high node CPU load, high container CPU load, insufficient node memory, insufficient container memory, node disk I/O obstructed, and network delay in the node/packet loss.

• High node CPU load: As other pods on the node seize CPU resources, resource competition is caused, and the response time of some services is affected. Faults can be found through the memory resource monitored data on the node.

• High container CPU load: If the deployment is configured improperly, the container memory is insufficient and the service cannot run. As the dynamic expansion strategy is not set, CPU resource load is too high under high concurrent requests. The service request response is therefore abnormal or the service request fails.

• Insufficient node memory: Due to insufficient node memory, the node fails and all services on the node are unavailable.

• Insufficient container memory: Pod start-up failure or continuous restart of pod under high load occurs due to insufficient memory resource allocation. The service will be unavailable or unstable.

• Node disk I/O obstructed: Due to a large number of disk I/O requests from other pods on the node, disk read and write competition occurs, which leads to a prolonged service request time.

• Network delay in the node/package loss: Packet loss or network latency occurs on the network due to switch failure or server network card failure, which affects the request time of the service on the node.

3.3 Analysis of Anomaly Detection Results

We conducted a large number of random fault injection experiments on the system. The injected applications include the front-end (user interaction interface), users (user registration and login), catalogue (product classification), carts (shopping carts), orders (order submit), payment (payment) and shipping (delivery). At least 20 successful activation cases were randomly selected for each failure, and indicator data, log data and service KPI data were collected to serve as the experimental data set. Recall was used to evaluate the fault diagnosis results of injected faults. The calculation of recall is shown in Eq. (14), where TP is the number of correctly identified faults and FN is the number of unrecognized faults.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} . \tag{14}$$

The anomaly detection approach based on monitored indicators and the anomaly detection approach based on logs proposed in this paper were individually performed on the experimental data set to detect the anomalies and identify the faults. The results are shown in Table 4.

▼Table 4. Recall rate of anomaly detection and fault recognition (recall)

	Indicator-Based Anomaly Detection	Log-Based Anomaly Detection	Fault Identification
Null value	1.00	1.00	1.00
Unexpected value	1.00	1.00	1.00
Short circuit of the exception statement	0.35	0.45	0.45
Condition reversed	0.10	0.55	0.55
Switch statement lacking a default value	0.15	0.75	0.45
Exception uncaught	1.00	1.00	1.00
Requested memory unreleased	1.00	1.00	1.00
High node CPU load	1.00	0.00	0.70
High container CPU load	1.00	0.00	0.75
Insufficient node memory	1.00	1.00	1.00
Insufficient container memory	1.00	1.00	1.00
Node disk I/O obstructed	0.65	0.00	0.65
Network delay in the node/package loss	0.90	0.00	0.70

According to the experimental results in Table 4, the method proposed in this paper can correctly identify about 78.85% of the fault types. Although the proposed method can accurately detect and locate most anomalies, there are still some faults that cannot be detected effectively. Through manual analysis, the key reasons include:

1) The sampling time interval of indicator monitoring data is too long and some instantaneous peak data cannot be collected, thus some faults do not output abnormal values and cannot be detected by the algorithm.

2) Some detected anomalies are false alarms, because noises are hidden in system running data.

3) There are many kinds of application logic faults. These faults only output abnormal values of calculation results, however, the degree of anomaly in the running data is not obvious. Therefore, the algorithm proposed in this paper cannot solve these problems.

4 Conclusions

This paper proposes an anomaly detection method based on streaming runtime data for microservices. First, an anomaly detection method based on monitored indicators is used to analyze the streaming system running data monitored. By analyzing a variety of different features of the monitored indicators, the abnormal points in the indicator data are found. Then the log-based anomaly detection method is used to analyze the system log stream, and the time-weighted control flow graph is built online to detect anomalies in log data. Finally, the results of the previous two anomaly detection methods are integrated and a filtering method is applied to these results to output the final anomalies.

We simulated a microservice system based on Kubernetes as our lab environment. Fault injection is utilized to simulate multiple faults including system changes, applications faults and system resources faults. Logs, monitored indicators, and service PKI data are collected as datasets for evaluation. The experimental results show that the proposed method can identify different types of faults with over 78% accuracy. In the future, we consider to design more sophisticated models to capture features of multi-source data such as logs, monitoring data, KPI and tracing data.

References

[1] WANG P, XU J M, MA M, et al. CloudRanger: root cause identification for cloud native systems [C]//18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE, 2018: 492 – 502. DOI: 10.1109/CCGRID.2018.00076
[2] MA M, LIN W L, PAN D S, et al. MS-rank: multi-metric and self-adaptive root cause diagnosis for microservice applications [C]//IEEE International Conference on Web Services. IEEE, 2019: 60 – 67. DOI: 10.1109/ICWS.2019.00022
[3] JIA T, LI Y, WU Z H. Survey of state-of-the-art log-based failure diagnosis (in Chinese) [J]. Journal of software, 2020, 31(7): 1997 – 2018. DOI: 10.13328/j.cnki.jos.006045
[4] CHUAH E, KUO S H, HIEW P, et al. Diagnosing the root-causes of failures from cluster log files [C]//International Conference on High Performance Computing. IEEE, 2010: 1 – 10. DOI: 10.1109/HIPC.2010.5713159
[5] CHEN C, SINGH N, YAJNIK S. Log analytics for dependable enterprise telephony [C]//Ninth European Dependable Computing Conference. IEEE, 2012: 94 – 101. DOI: 10.1109/EDCC.2012.14
[6] ZHOU X, PENG X, XIE T, et al. Latent error prediction and fault localization

for microservice applications by learning from system trace logs [C]//27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ACM, 2019: 683 – 694
[7] SILVERMAN B. Density estimation for statistics and data analysis [M]. London, UK: Routledge, 2018
[8] PARZEN E, BROWN R G. Smoothing, forecasting and prediction of discrete time series [J]. Journal of the American statistical association, 1964, 59(307): 973. DOI: 10.2307/2283122
[9] HE P J, ZHU J M, ZHENG Z B, et al. Drain: an online log parsing approach with fixed depth tree [C]//IEEE International Conference on Web Services. IEEE, 2017: 33 – 40. DOI: 10.1109/ICWS.2017.13
[10] JIA T, WU Y F, HOU C J, et al. LogFlash: real-time streaming anomaly detection and diagnosis from system logs for large-scale software systems [C]//IEEE 32nd International Symposium on Software Reliability Engineering. IEEE, 2021: 80 – 90. DOI: 10.1109/ISSRE52982.2021.00021
[11] ZHAO N W, CHEN J J, WANG Z, et al. Real-time incident prediction for online service systems [C]//28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ACM, 2020: 315 – 326. DOI: 10.1145/3368089.3409672
[12] WU L, TORDSSON J, BOGATINOVSKI J, et al. MicroDiag: fine-grained performance diagnosis for microservice systems [C]//IEEE/ACM International Workshop on Cloud Intelligence (CloudIntelligence). IEEE, 2021: 31 – 36. DOI: 10.1109/CloudIntelligence52565.2021.00015
[13] YANG Y, LI Y, WU Z H. Survey of state-of-the-art distributed tracing technology (in Chinese) [J]. Journal of software, 2020, 31(7): 2019 – 2039

Biographies

ZHANG Qixun is currently an assistant professor in School of Software and Microelectronics in Peking University, China. He received his PhD in 2022. His research interests include distributed systems, AIOps, etc.

HAN Jing (han.jing28@zte.com.cn) joined ZTE Corporation in 2000. She is an expert in AIOps. She has been putting effort into natural language process for over 10 years and has published several papers.

CHENG Li joined ZTE Corporation in 2006. He is an expert in AIOps and wireless communications. He has much experience in analyzing variety of types of data. He has a lot of experience in problem-solving and methodology.

ZHANG Baisheng joined ZTE Corporation in 2011. His work has been devoted to cell-phone terminal techniques for over 10 years. Besides, he is interested in the research of auto-driving technology.

GONG Zican joined ZTE Corporation in 2020. He received his master's degree in computing from Australian National University, Australia in 2019. His research interests include AIOps and natural language processing.

Symbiotic Radio Systems: Detection and Performance Analysis

CUI Ziqi[1], WANG Gongpu[1], WANG Zhigang[2], AI Bo[1], XIAO Huahua[3]

(1. Beijing Jiaotong University, Beijing 100044, China;
2. Guangdong Communications and Networks Institute, Guangzhou 510070, China;
3. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202203012

https://kns.cnki.net/kcms/detail/34.1294.TN.20220704.1911.002.html, published online July 6, 2022

Manuscript received: 2021-11-28

Abstract: Symbiotic radio (SR) is an emerging green technology for the Internet of Things (IoT). One key challenge of the SR systems is to design efficient and low-complexity detectors, which is the focus of this paper. We first drive the mathematical expression of the optimal maximum-likelihood (ML) detector, and then propose a suboptimal iterative detector with low complexity. Finally, we show through numerical results that our proposed detector can obtain near-optimal bit error rate (BER) performance at a low computational cost.

Keywords: bit error rate; data detection; Internet of Things; symbiotic radio system; wireless communication

Citation (IEEE Format): Z. Q. Cui, G. P. Wang, Z. G. Wang, et al., "Symbiotic radio systems: detection and performance analysis," *ZTE Communications*, vol. 20, no. 3, pp. 93 - 98, Sept. 2022. doi: 10.12142/ZTECOM.202203012.

1 Introduction

The Internet of Things (IoT) enables large-scale connection of IoT devices and is regarded as one of the major applications for the fifth-generation (5G) communication networks. However, due to the huge number of IoT devices, traditional IoT communication technologies inevitably face huge energy consumption problem and a shortage of spectrum resources. The two factors have become bottlenecks in the development and implementation of IoT[1-2].

Ambient backscatter communication (AmBC) is a promising technology to address the above bottlenecks. It enables passive backscatter devices (like passive tags) to harvest energy from ambient signals such as WiFi, broadcast TV, or cellular signals, and to modulate information by dynamically adjusting the impedance inside the circuits without using any specific RF components[3-4]. Unfortunately, as a result of the spectrum-sharing nature and the double attenuation of the backscatter link, the prime signal (from RF sources) is usually stronger than the backscattered signal (transmitted by the backscatter device) and is usually taken as interference to the reader, leading to a severe error floor problem[5-7].

Recently, symbiotic radio (SR) system has been proposed to tackle prime signal interference[2, 8-9]. The simplest SR system consists of a prime transmitter (PT), a tag, and a reader, in which the PT not only serves as an energy source to support backscatter communication but also transmits its own information. Thus, the reader needs to recover the information from the PT and the tag. Benefitting from the cooperative communication property, the SR system converts the prime signal interference to useful information, yielding a higher achievable rate than the traditional AmBC system[10-11].

A key challenge of the SR system is to design effective and efficient detectors. The main difficulties are as follows. First, the prime signal and the backscattered signal are mutually dependent (formulated in Section 2), making it difficult to recover the information carried by received signals. Second, the computational complexity of the optimal detectors is relatively high, and grows rapidly as the modulation order of the PT and the tag increases, causing non-negligible decoding time consumption that may exceed the strict response time constraint of some specific IoT protocols like the industrial RFID Gen2 protocol[12].

In order to realize effective and efficient signal detection for the SR system, it is worth studying detectors that provide desirable performance but with low complexity. Related research is still in its infancy. In Ref. [13 - 14], the authors proposed low-complexity linear detectors and successive interference cancellation (SIC) based detectors to recover bits from the PT and the tag.

The effective and efficient detector design is a practical constraint for the SR system but is rarely studied at present, which motivates our current work. In this paper, we investigate the signal detection problem of the SR system. The contributions of our work are summarized as follows:

First, we formulize the system model of the SR system, and derive the optimal maximum-likelihood (ML) detector which is

optimal when all the symbols are equiprobable.

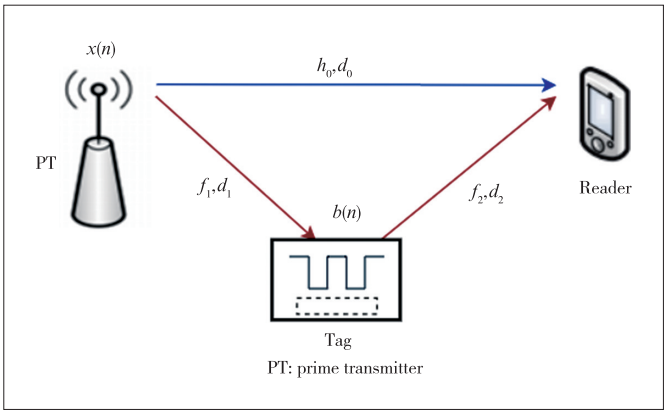
Then, considering that the prime signal is much stronger than the backscatter signal, we propose a suboptimal iterative detector with polynomial complexity. In each iteration, the suboptimal iterative detector detects the prime signal, and then detects the backscatter signal using the knowledge from the detected prime signal.

Finally, we numerically analyze the bit error rate (BER) performance of the proposed suboptimal detector. Simulation results demonstrate that the performance of the proposed detector becomes stable after about two rounds of iteration, and its performance is close to the optimal ML detector for typical application scenarios.

The rest of this paper is organized as follows. Section 2 introduces the system model under consideration. Section 3 derives the optimal ML detector and proposes a suboptimal iterative detector that can reduce the computational complexity while obtaining near-optimal detection performance. Section 4 numerically evaluates the BER performance of the proposed detector. Finally, Section 5 summarizes this paper.

2 System Model

Fig. 1 depicts a symbiotic radio system that consists of a PT, a passive tag, and a reader. In this system, the tag harvests RF energy from the PT, and then modulates its information by varying the antenna load impedance. Compared with the traditional ambient backscatter system, the PT is designed to provide power to the tag and to enable communications. Therefore, the reader receives superposed signals from the PT and the tag. Thus, it needs to recover the information from both of the two devices.



▲ **Figure 1. Symbolic radio system model**

Let d_0 represent the distance between the PT and the reader, d_1 represent the distance between the PT and the tag, and d_2 represent the distance between the tag and the reader. In this paper, we consider the block fading channel model, which means the channel states are static during one time slot. Let h_0, f_1, and f_2 be the channel coefficients from the PT to the reader, the PT to the tag, and the tag to the reader, respectively. We assume h_0, f_1 and f_2 are mutually independent, each of which follows the Rician distribution as

$$q \sim CN\left(\sqrt{\frac{k_q}{k_q+1}}\sigma^2, \frac{\sigma_q^2}{k+1}\right), q \in \{h_0, f_1, f_2\}, \tag{1}$$

where k_q is the ratio of the energy in the specular path to the energy in the scattered path[15-16]. Let $k_q = 0$, and Rayleigh fading is obtained. Noting that the channel state can be estimated using pilot signals[17], we assume that perfect channel state information (CSI) is available.

Let $x(n) \in \mathcal{A}_x$, and $b(n) \in \mathcal{A}_b$ denote the transmitted signal of the PT and the tag at the n-th time slot, respectively, where $\mathcal{A}_x$ stands for the modulation alphabet set of the PT and $\mathcal{A}_b$ stands for the modulation alphabet set of the tag. The signal power of the PT is represented by P_x.

In this paper, we assume the data rate of the tag equals that of the reader. Therefore, the signal received by the reader in the n-th time slot is

$$y(n) = \frac{h_0}{\sqrt{d_0^\alpha}}x(n) + \eta\frac{f_1}{\sqrt{d_1^\alpha}}\frac{f_2}{\sqrt{d_2^\alpha}}x(n)b(n) + \omega(n), \tag{2}$$

where η is the attenuation inside the tag, and $\omega(n)$ is the complex additive white Gaussian noise (AWGN) with zero-mean and σ^2 variance.

Eq. (2) can be simplified as:

$$y(n) = \mu x(n) + \nu x(n)b(n) + \omega(n), \tag{3}$$

where

$$\mu = \frac{h_0}{\sqrt{d_0^\alpha}}, \tag{4}$$

$$\nu = \eta\frac{f_1}{\sqrt{d_1^\alpha}}\frac{f_2}{\sqrt{d_2^\alpha}}. \tag{5}$$

For convenience, we define the signal-to-noise ratio (SNR) of the prime link as $\gamma_P \triangleq \frac{P_x\mathbb{E}[|\mu|^2]}{\sigma^2}$, and the SNR of the backscatter link as $\gamma_B \triangleq \frac{P_xP_b\mathbb{E}[|\nu|^2]}{\sigma^2}$.

Strictly speaking, the arrival of $b(n)$ is delayed by time $\tau(\tau \geqslant 0)$ compared with $x(n)$. However, such delay can be ignored in most scenarios for the following reasons[3, 13, 18]. First, the tag is close to the reader with a distance usually less than 10 feet (3.048 m). Second, the signal transmission speed inside the tag circuit is so fast that the transmission delay is too short to impact the signal detection.

3 Data Detection for Symbiotic Radio Systems

In this section, we first introduce the optimal ML detector for the SR system utilizing the perfect CSI, and then propose a suboptimal iterative detector.

3.1 Optimal ML Detector

The ML detector is the most suitable when all the symbols are equiprobable. The ML detection rule is given by:

$$\left\{\hat{x}(n),\hat{b}(n)\right\} = \arg \max_{\substack{x(n)\in\mathcal{A}_x,\\ b(n)\in\mathcal{A}_b}} f\left(y(n)|x(n),b(n)\right), \tag{6}$$

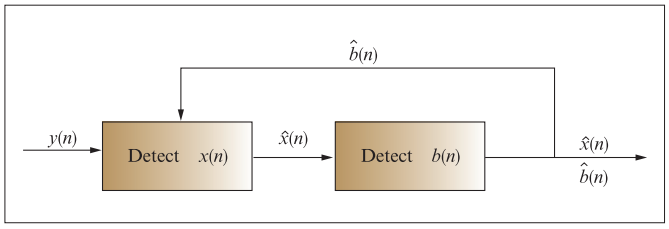
where $\hat{x}(n)$ and $\hat{b}(n)$ are detected bits of the PT and the tag, respectively. Eq. (6) can also be written as:

$$\left\{\hat{x}(n),\hat{b}(n)\right\} = \arg \min_{\substack{x(n)\in\mathcal{A}_x,\\ b(n)\in\mathcal{A}_b}} \left|y(n)-\mu x(n)-\nu x(n)b(n)\right|^2. \tag{7}$$

Note that the searching time of the optimal ML detector is $|\mathcal{A}_x|\cdot|\mathcal{A}_b|$, and it will increase rapidly when the PT and the tag use high-order modulation. We then propose a suboptimal iterative detector with relatively low complexity.

3.2 Suboptimal Iterative Detector

In the SR system, the prime signal $\mu x(n)$ is stronger than the backscatter signal $\nu x(n)b(n)$ due to the double attenuation of the backscatter link. Therefore, $x(n)$ can be recovered first with desirable performance by taking $\nu x(n)b(n)$ as noise. Then the detector subtracts $\mu\hat{x}(n)$ from the received signal $y(n)$ to construct new statistic to recover $\hat{b}(n)$. In this process, some recovered symbols may be erroneous. However, $\hat{b}(n)$ can be fed back to the detector to improve the detection performance. The same is true for $\hat{x}(n)$. After repeating this process several times, the detector will obtain desirable performance. Fig. 2 shows the block diagram of this procedure. Next, we introduce the detail process of the suboptimal iterative detector.



▲ **Figure 2. Suboptimal iterative detector scheme**

1) Detect $x(n)$

Considering that the prime signal is stronger than the backscatter signal, we first use the ML detector to recover $x(n)$ while taking $\nu x(n)b(n)$ as noise. Under this circumstance, the detection rule is:

$$\hat{x}(n) = \arg \min_{x(n)\in\mathcal{A}_x} \left|y(n)-\mu x(n)\right|^2. \tag{8}$$

2) Detect $b(n)$

After obtaining $\hat{x}(n)$, the prime signal inference can be removed by subtracting $\mu\hat{x}(n)$ from the received signal $y(n)$. This can be written as

$$y_2(n) = y(n)-\mu\hat{x}(n). \tag{9}$$

Then with the use of the ML detector, $\hat{b}(n)$ is given by

$$\hat{b}(n) = \arg \min_{b(n)\in\mathcal{A}_b} \left|y_2(n)-\nu\hat{x}(n)b(n)\right|^2. \tag{10}$$

3) Iterative manner

The detection results of process (1) and process (2) may be erroneous. Fortunately, the existing work shows that the detection performance can be improved using the iterative detection manner[19]. Therefore, we can redetect $x(n)$ as Eq. (11), and then substitute redetected $\hat{x}(n)$ into Eq. (10) to update $\hat{b}(n)$. After a few iterations, the suboptimal detector can get desirable performance.

$$\hat{x}(n) = \arg \min_{x(n)\in\mathcal{A}_x} \left|y(n)-\mu x(n)-\nu x(n)\hat{b}(n)\right|^2. \tag{11}$$

We summarize the algorithm of the suboptimal detector in Algorithm 1. In each iteration, the detector needs to search $|\mathcal{A}_x|$ possible values to recover $x(n)$ and $|\mathcal{A}_b|$ possible values to recover $b(n)$. Moreover, this algorithm repeats K times, so the overall search time of the suboptimal iterative detector is $K(|\mathcal{A}_x|+|\mathcal{A}_b|)$. Considering K is a preset constant and usually a small value, the computational complexity of the suboptimal detector is $O(|\mathcal{A}_x|+|\mathcal{A}_b|)$.

Algorithm 1. Suboptimal Iterative Detection

Input: $\mathcal{A}_x, \mathcal{A}_b, y(n), \mu, \nu$
Output: $\hat{x}(n), \hat{b}(n)$
$\hat{x}(n) = \arg \min_{x(n)\in\mathcal{A}_x} \left|y(n)-\mu x(n)\right|^2$
While $K \neq 0$
    $y_2(n) = y(n)-\mu\hat{x}(n)$
    $\hat{b}(n) = \arg \min_{b(n)\in\mathcal{A}_b} \left|y_2(n)-\nu\hat{x}(n)b(n)\right|^2$
    $\hat{x}(n) = \arg \min_{x(n)\in\mathcal{A}_x} \left|y(n)-\mu x(n)-\nu x(n)\hat{b}(n)\right|^2$
    $K \leftarrow K-1$
End While

4 Numerical Results

In this section, we analyze the detection performance of the proposed suboptimal iterative detector. We first compare the

suboptimal detector with the optimal ML detector and the linear minimum mean-square-error (LMMSE) detector, and then investigate the performance of the suboptimal detector in different iterations. The expression of the LMMSE detector can be found in Ref. [20]. We summarize the computational complexity of the three detectors in Table 1.

▼Table 1. Computational complexity of different detectors

Detector	Computational Complexity
Optimal ML	$O(\lvert\mathcal{A}_x\rvert \cdot \lvert\mathcal{A}_b\rvert)$
LMMSE	$O(\lvert\mathcal{A}_x\rvert \cdot \lvert\mathcal{A}_b\rvert)$
Suboptimal iterative	$O(\lvert\mathcal{A}_x\rvert + \lvert\mathcal{A}_b\rvert)$

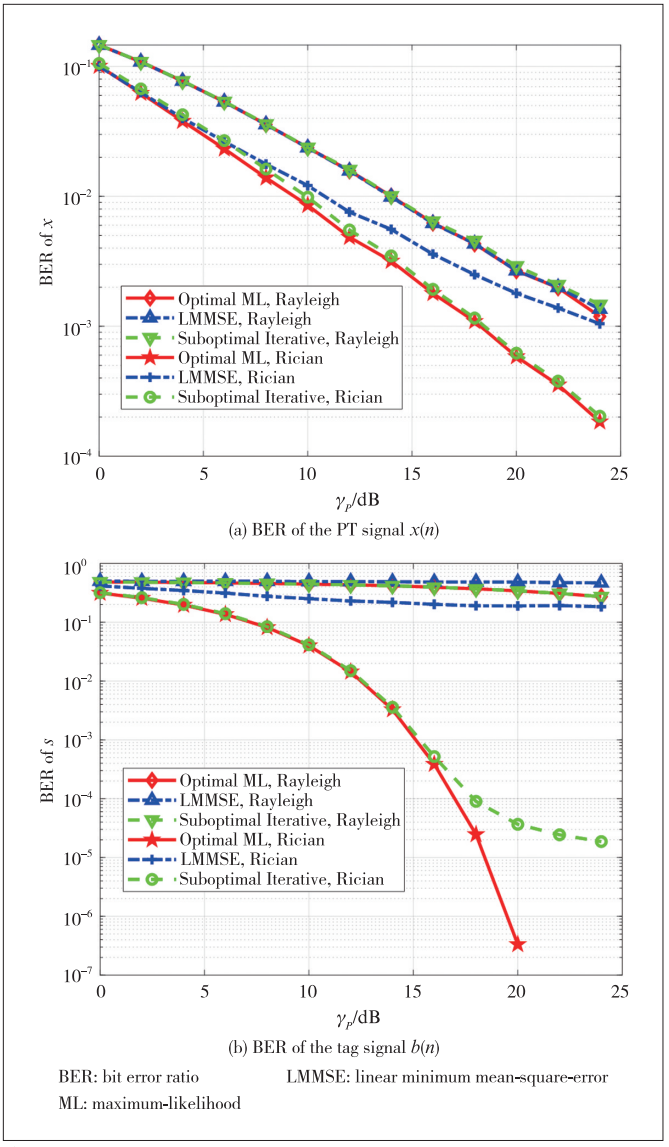
LMMSE: linear minimum mean-square-error ML: maximum-likelihood

We consider that the PT uses the binary phase shift keying (BPSK) modulation and the tag uses on-off keying (OOK) modulation. Therefore, $\mathcal{A}_x = \left\{\sqrt{P_x}, -\sqrt{P_x}\right\}$, and $\mathcal{A}_b = \{0, 1\}$. Specifically, $b(n) = 1$ means that the tag backscatters signals, and $b(n) = 0$ means that the tag does not backscatter signals. We assume that the tag attenuation η = 0.8, path loss factor α = 2, channel parameter k_q = 20, and the noise variance σ^2 = 1. Totally 10^5 Monte Carlo runs are adopted for average.

Fig. 3 shows the BER of $x(n)$ and $b(n)$ versus prime link SNR γ_P when setting $d_0 = d_1 = d_2 = 1$ m. According to the figure, the BER of all the detectors shows a downtrend. This is because the backscatter link SNR γ_B increases with the increase of γ_P in our experiment setting, so all the detectors have satisfactory performance. It is also obvious from Fig. 3 that the suboptimal detector performs better than the LMMSE detector under both Rayleigh and Rician channels. Besides, the proposed detector obtains near-optimal performance when γ_P is less than 16 dB under the Rician channel. When γ_P >16 dB, its BER of $b(n)$ is limited by the BER of $x(n)$, and the suboptimal detector is inferior to the ML detector.

Fig. 4 shows the BER of $x(n)$ and $b(n)$ versus d_1 when setting γ_P = 8 dB and $d_0 = d_2 = 1$ m. It is clear from the figure that the suboptimal detector and the optimal ML detector have the same performance when d_1 is greater than 2 m, and the suboptimal detector always performs better than the LMMSE detector. This proves the effectiveness of the proposed detector. We can also find that the BER of $x(n)$ decreases as d_1 gets larger while the BER of $b(n)$ increases. This is because γ_B decreases as d_1 increases. When d_1 is large enough, the backscattered signal gets too weak to be detected, so all the detectors have poor performance in detecting $b(n)$. However, as the power of the backscattered signal decreases, its inference to the prime signal alleviates, resulting in better BER performance of $x(n)$.

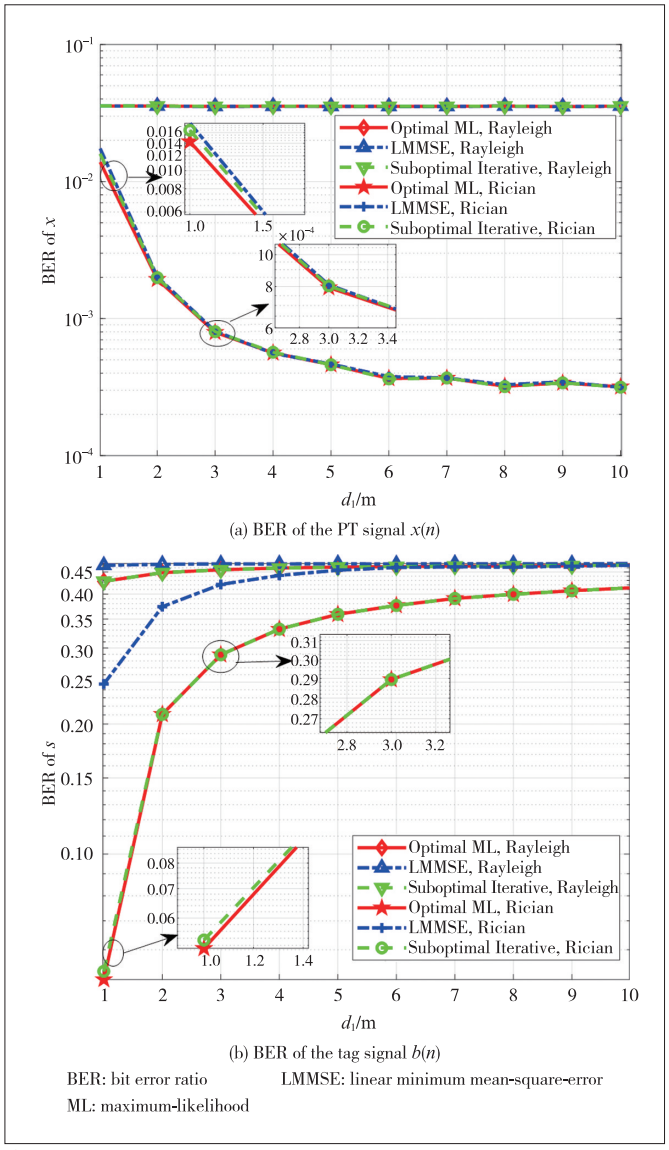
Fig. 5 depicts the BER performance versus γ_P in different iterations. In this figure, it is worth noting that "iteration 0" refers to step 1 in Algorithm 1. It can be found that the iteration manner can improve the detection performance under both Rayleigh and Rician channels. With the increase of γ_P, the improvement of iterative performance gets more significant, especially for the Rician channel. It can be seen that when γ_P is greater than 20 dB, the BER performance of detecting $x(n)$ and $b(n)$ increases by 79.5% and 50%, respectively. Moreover, it is clear from the figure that the curves of iteration two and iteration three are essentially indistinguishable, which proves that the performance of the suboptimal detector is stable after two iterations and is consistent with the conclusion in Ref. [19].



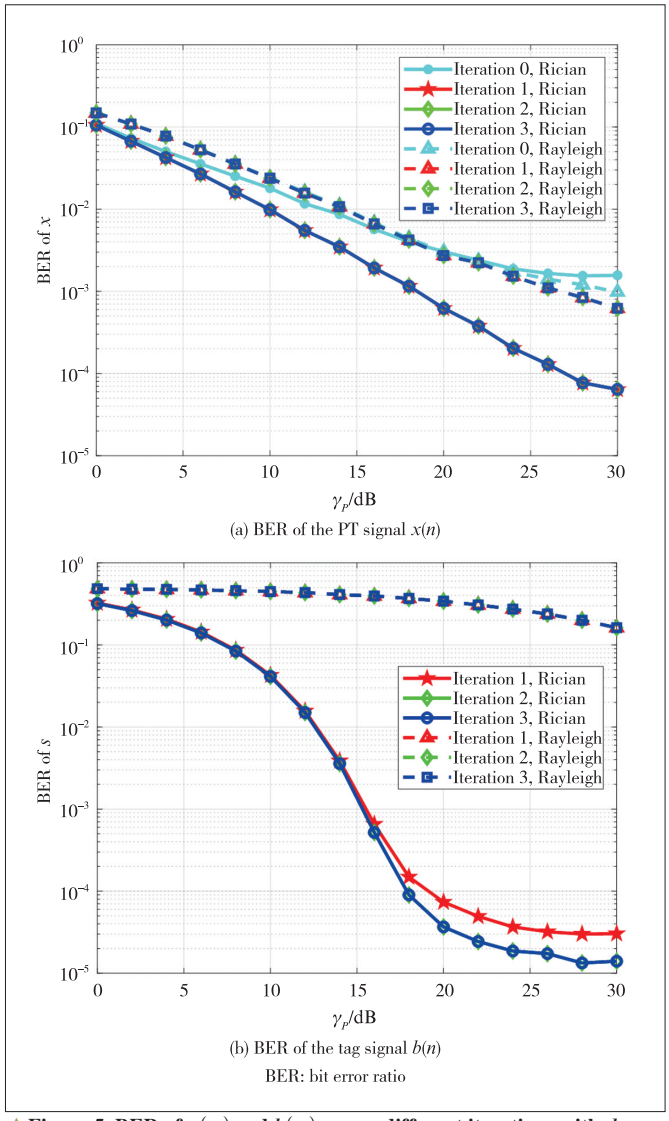
▲Figure 3. BER of $x(n)$ and $b(n)$ versus γ_P with $d_0 = d_1 = d_2 = 1$ m

5 Conclusions

This paper studied the data detection problem of the SR system. First, we derived the mathematical expression of the opti-



▲ **Figure 4. BER of $x(n)$ and $b(n)$ versus d_1 with $d_0 = d_2 = 1$ m and $\gamma_P = 8$ dB**



▲ **Figure 5. BER of $x(n)$ and $b(n)$ versus different iterations with $d_0 = d_1 = d_2 = 1$ m**

mal ML detector. Then, based on the fact that the power of the prime signal is much stronger than that of the backscatter signal, we therefore proposed a low-complexity suboptimal iterative detector. Numerical results showed that the proposed detector could achieve near-optimal BER performance after about two iterations.

References

[1] ZHANG L, LIANG Y C, XIAO M. Spectrum sharing for Internet of Things: a survey [J]. IEEE wireless communications, 2019, 26(3): 132 – 139. DOI: 10.1109/MWC.2018.1800259
[2] LONG R Z, LIANG Y C, GUO H Y, et al. Symbiotic radio: a new communication paradigm for passive Internet of Things [J]. IEEE Internet of Things journal, 2020, 7(2): 1350 – 1363. DOI: 10.1109/JIOT.2019.2954678
[3] LIU V, PARKS A, TALLA V, et al. Ambient backscatter [J]. ACM SIGCOMM computer communication review, 2013, 43(4): 39 – 50. DOI: 10.1145/2534169.2486015
[4] ZHAO W J, WANG G P, FAN R F, et al. Ambient backscatter communication systems: Capacity and outage performance analysis [J]. IEEE access, 2018, 6: 22695 – 22704. DOI: 10.1109/ACCESS.2018.2828021
[5] QIAN J, GAO F F, WANG G P, et al. Noncoherent detections for ambient backscatter system [J]. IEEE transactions on wireless communications, 2017, 16(3): 1412 – 1422. DOI: 10.1109/TWC.2016.2635654
[6] TAO Q, ZHONG C, LIN H., et al. Symbol detection of ambient backscatter systems with Manchester coding [J]. IEEE transactions on wireless communications, 2018, 17(6): 4028 – 4038
[7] GUO H Y, ZHANG Q Q, XIAO S, et al. Exploiting multiple antennas for cognitive ambient backscatter communication [J]. IEEE Internet of Things journal, 2019, 6(1): 765 – 775. DOI: 10.1109/JIOT.2018.2856633
[8] LIANG Y C, ZHANG Q Q, LARSSON E G, et al. Symbiotic radio: cognitive backscattering communications for future wireless networks [J]. IEEE transac-

tions on cognitive communications and networking, 2020, 6(4): 1242 – 1255. DOI: 10.1109/TCCN.2020.3023139
[9] GUO H Y, LIANG Y C, LONG R Z, et al. Cooperative ambient backscatter system: a symbiotic radio paradigm for passive IoT [J]. IEEE wireless communications letters, 2019, 8(4): 1191 – 1194. DOI: 10.1109/LWC.2019.2911500
[10] CHU Z, HAO W M, XIAO P, et al. Resource allocations for symbiotic radio with finite blocklength backscatter link [J]. IEEE Internet of Things journal, 2020, 7(9): 8192 – 8207. DOI: 10.1109/jiot.2020.2980928
[11] LIU W C, LIANG Y C, LI Y H, et al. Backscatter multiplicative multiple-access systems: fundamental limits and practical design [J]. IEEE transactions on wireless communications, 2018, 17(9): 5713 – 5728. DOI: 10.1109/TWC.2018.2849372
[12] EPC Radio-Frequency Identity Protocols. UHF RFID Protocol for Communications at 860 MHZ-960 MHZ [S]. 2015
[13] YANG G, ZHANG Q Q, LIANG Y C. Cooperative ambient backscatter communications for green Internet-of-Things [J]. IEEE Internet of Things journal, 2018, 5(2): 1116 – 1130. DOI: 10.1109/JIOT.2018.2799848
[14] YANG G, LIANG Y C, ZHANG Q Q. Cooperative receiver for ambient backscatter communications with multiple antennas [C]//Proceedings of 2017 IEEE International Conference on Communications. IEEE, 2017: 1 – 6. DOI: 10.1109/ICC.2017.7997479
[15] TSE D, VISWANATH P. Fundamentals of wireless communication [M]. Cambridge: Cambridge University Press, 2005. DOI: 10.1017/cbo9780511807213
[16] OUROUTZOGLOU M, VOUGIOUKAS G, KARYSTINOS G N, et al. Multistatic noncoherent linear complexity miller sequence detection for Gen2 RFID/IoT [J]. IEEE transactions on wireless communications, 2021, 20(12): 8067 – 8080. DOI: 10.1109/TWC.2021.3089910
[17] BHARADIA D, JOSHI K R, KOTARU M, et al. BackFi: high throughput WiFi backscatter [J]. ACM SIGCOMM computer communication review, 2015, 45(4): 283 – 296. DOI: 10.1145/2829988.2787490
[18] WANG G P, GAO F F, DOU Z Z, et al. Uplink detection and BER analysis for ambient backscatter communication systems [C]//IEEE Global Communications Conference. IEEE, 2015: 1 – 6. DOI: 10.1109/GLOCOM.2015.7417704
[19] BUZZI S, LOPS M, SARDELLITTI S. Performance of iterative data detection and channel estimation for single-antenna and multiple-antennas wireless communications [J]. IEEE transactions on vehicular technology, 2004, 53(4): 1085 – 1104. DOI: 10.1109/TVT.2004.830144
[20] SENGIJPTA S K. Fundamentals of statistical signal processing: estimation theory [J]. Technometrics, 1995, 37(4): 465 – 466. DOI: 10.1080/00401706.1995.10484391

Biographies

CUI Ziqi received her BE degree in computer science and technology from the North University of China in 2018, the MS degree from the Beijing Jiaotong University, China in 2021, where she is currently pursuing the PhD degree with the Department of Computer Science and Technology. Her research interests include the Internet of Things, performance analysis theories, and signal processing technologies.

WANG Gongpu received his BE degree from Anhui University, China in 2001, and MS degree from Beijing University of Posts and Telecommunications, China in 2004. From 2004 to 2007, he was an assistant professor in the School of Network Education, Beijing University of Posts and Telecommunications. He received his PhD degree from University of Alberta, Canada in 2011. Currently, he is a professor in School of Computer and Information Technology, Beijing Jiaotong University, China. His research interests include wireless communication theories, signal processing technologies, and the Internet of Things.

WANG Zhigang (wangzhigang@gdcni.cn) is currently with Guangdong Communications and Networks Institute, China. His research interests include Internet of Things, autonomous aerial vehicles, and MIMO communication.

AI Bo received his MS and PhD degrees from Xidian University, China in 2002 and 2004, respectively. He was with Tsinghua University, China, where he was an Excellent Post-Doctoral Research Fellow in 2007. He is currently a professor and a PhD supervisor with Beijing Jiaotong University, China. He is also the Deputy Director of the State Key Laboratory of Rail Traffic Control and Safety. He has published six Chinese academic books, three English books, over 110 IEEE journal articles, five ESI highly cited articles, and one ESI hot article. He is mainly engaged in the research and application of the theory and core technology of broadband mobile communication and rail transit dedicated mobile communication systems (GSM-R, LTE-R, 5G-R, and LTE-M). He is a Fellow of the IET. He received the National Science Fund for Distinguished Young Scholars, the Outstanding Youth Science Fund, the Ministry of Science and Technology's Young and Middle-Aged Science and Technology Innovation Leaders, the China Association for Science and Technology's Seeking Outstanding Youth Award, the Ministry of Education's New Century Excellence Talent, the Zhan Tianyou Railway Science and Technology Youth Award, the Beijing Science and Technology Star Winner, and the Honorary Title of Beijing Excellent Teacher. He has obtained nine international conference paper awards and 26 invention patents (18 proposals adopted by the ITU, 3GPP, etc.), and eight provincial and ministerial-level science and technology awards. He is also the president of the IEEE BTS Xi'an Branch, the vice president of the IEEE VTS Beijing Branch, the IEEE VTS Distinguished Lecturer, an Associate Editor of the *IEEE Transactions on Consumer Electronics* and the *IEEE Transactions On Antennas and Propagation*, and a Guest Editor of the *IEEE Antennas and Wireless Propagation Letters, the IEEE Transactions on Vehicular Technology*, the *IEEE Transactions on Industrial Electronics*, and other SCI journals.

XIAO Huahua received his MS degree in computer software and theories from Sun Yat-Sen University, China. He is currently with ZTE Corporation, as a senior engineer in the field of antenna algorithm pre-research. He has applied for more than 150 Chinese and foreign patents in the multi-antenna field. His research interests include MIMO communication, cellular radio, precoding, and Long Term Evolution.

ZTE Communications Guidelines for Authors

Remit of Journal

ZTE Communications publishes original theoretical papers, research findings, and surveys on a broad range of communications topics, including communications and information system design, optical fiber and electro-optical engineering, microwave technology, radio wave propagation, antenna engineering, electromagnetics, signal and image processing, and power engineering. The journal is designed to be an integrated forum for university academics and industry researchers from around the world.

Manuscript Preparation

Manuscripts must be typed in English and submitted electronically in MS Word (or compatible) format. The word length is approximately 3 000 to 8 000, and no more than 8 figures or tables should be included. Authors are requested to submit mathematical material and graphics in an editable format.

Abstract and Keywords

Each manuscript must include an abstract of approximately 150 words written as a single paragraph. The abstract should not include mathematics or references and should not be repeated verbatim in the introduction. The abstract should be a self-contained overview of the aims, methods, experimental results, and significance of research outlined in the paper. Five carefully chosen keywords must be provided with the abstract.

References

Manuscripts must be referenced at a level that conforms to international academic standards. All references must be numbered sequentially intext and listed in corresponding order at the end of the paper. References that are not cited in-text should not be included in the reference list. References must be complete and formatted according to *ZTE Communications* Editorial Style. A minimum of 10 references should be provided. Footnotes should be avoided or kept to a minimum.

Copyright and Declaration

Authors are responsible for obtaining permission to reproduce any material for which they do not hold copyright. Permission to reproduce any part of this publication for commercial use must be obtained in advance from the editorial office of *ZTE Communications*. Authors agree that a) the manuscript is a product of research conducted by themselves and the stated co-authors; b) the manuscript has not been published elsewhere in its submitted form; c) the manuscript is not currently being considered for publication elsewhere. If the paper is an adaptation of a speech or presentation, acknowledgement of this is required within the paper. The number of co-authors should not exceed five.

Content and Structure

ZTE Communications seeks to publish original content that may build on existing literature in any field of communications. Authors should not dedicate a disproportionate amount of a paper to fundamental background, historical overviews, or chronologies that may be sufficiently dealt with by references. Authors are also requested to avoid the overuse of bullet points when structuring papers. The conclusion should include a commentary on the significance/future implications of the research as well as an overview of the material presented.

Peer Review and Editing

All manuscripts will be subject to a two-stage anonymous peer review as well as copyediting, and formatting. Authors may be asked to revise parts of a manuscript prior to publication.

Biographical Information

All authors are requested to provide a brief biography (approx. 100 words) that includes email address, educational background, career experience, research interests, awards, and publications.

Acknowledgements and Funding

A manuscript based on funded research must clearly state the program name, funding body, and grant number. Individuals who contributed to the manuscript should be acknowledged in a brief statement.

Address for Submission

http://mc03.manuscriptcentral.com/ztecom

ZTE COMMUNICATIONS
中兴通讯技术(英文版)

ZTE Communications has been indexed in the following databases:

- Abstract Journal
- Cambridge Scientific Abstracts (CSA)
- China Science and Technology Journal Database
- Chinese Journal Fulltext Databases
- Index of Copurnicus
- Ulrich's Periodicals Directory
- Wanfang Data
- WJCI 2021

ZTE COMMUNICATIONS
Vol. 20 No. 3 (Issue 80)
Quarterly
First English Issue Published in 2003

Supervised by:
Anhui Publishing Group

Sponsored by:
Time Publishing and Media Co., Ltd.
Shenzhen Guangyu Aerospace Industry Co., Ltd.

Published by:
Anhui Science & Technology Publishing House

Edited and Circulated (Home and Abroad) by:
Magazine House of ZTE Communications

Editorial Correspondence:
Add: 12F Kaixuan Building, 329 Jinzhai Road,
Hefei 230061, P. R. China
Tel: +86-551-65533356
Email: magazine@zte.com.cn
Website: http://zte.magtechjournal.com

Annual Subscription: RMB 120
Printed by:
Hefei Tiancai Color Printing Company
Publication Date: September 25, 2022

China Standard Serial Number: ISSN 1673−5188
CN 34−1294/TN