Lightweight Privacy-Preserving Anomaly Detection for Time Series Based on Federated Learning

Bin Jiang, Guangfeng Wang, Xuerong Cui, and Jian Wang

***Abstract*—Anomaly monitoring in distributed computing scenarios is facing significant challenges due to issues of data privacy and security. To enhance the efficiency of federated learning, it is critical to develop a lightweight anomaly detection model. We propose a lightweight anomaly detection model based on federated learning, which ensures detection efficiency while protecting data privacy. In the local model, the model reduces network structure and accelerates inference speed through model fusion. And through the decomposition of the attention mechanism, the local features of the data are enhanced, improving the detection accuracy. In the federated framework, model compression is used to reduce the number of communication rounds and the number of parameters per round. Finally, comparative experiments on several publicly available datasets have shown a significant improvement in model performance compared to the baseline.**

***Index Terms*—Federated Learning, Anomaly Detection, Time Series, Privacy Protection.**

I. Introduction

DUE to the continuous improvement of distributed computing technology and the advancement of processing algorithms, the demand for data anomaly detection is constantly increasing. Therefore, anomaly detection has emerged as a vital data mining task [1]. Moreover, the complexity of industrial control systems (ICS) and Internet of Things (IoT) devices has significantly escalated. Anomaly detection in privacy protection mode can quickly identify unusual system behavior, ensuring the continuity of system operation and safeguarding the privacy and security of enterprises and individuals [2].

At the same time, the lack of labeled data, the diversity of abnormal phenomena, and the uniqueness of data in various fields make it more difficult to establish a robust and privacy based anomaly detection model [3]. Traditional statistical modeling methods, such as SAND, use clustering and statistical analysis for anomaly detection. Contemporary methods, such as LSTM-NDT [4], use short-term and long-term memory neural networks for anomaly detection. Transformers [5] use a multi-head attention mechanism to encode input sequences, which has higher parallelism and computational efficiency compared to traditional network models. The generate adversarial networks (GAN) [6] model can train a more accurate detection model without label data, but it is difficult to converge and prone to pattern collapse. In the field of user privacy protection, federated learning (FL) [7] has emerged as a widespread method and has been widely applied in IoT and healthcare.

We propose a lightweight anomaly detection model based on federated learning, which ensures detection efficiency while protecting data privacy. Our contributions of this work can be summarized as follows: 1) We integrate the autoencoder and transformer into local anomaly detection, reducing the overall network structure and achieving a lightweight model while accelerating inference speed. 2) We decompose the multi-head attention mechanism in transformer into a parallel structure consisting of attention blocks and convolution blocks, enhancing the local features of the data and improving detection accuracy. 3) We incorporate the model compression algorithm into federated framework to reduce the number of communication rounds and the number of parameters per round, achieving model lightweight.

II. Related Work

In the field of data mining, anomaly detection in time-series data has always been a very important research area. Currently, unsupervised machine learning methods are widely used for anomaly detection in time series data. LSTM-NDT proposed by Tuli *et al.* [4] took use of LSTM to predict data with input time series, and proposes an unsupervised and non parametric threshold method to explain the predictions generated by LSTM networks. Qi *et al.* [8] put forward DAGMM by a deep automatic encoding Gaussian mixture model for dimensionality reduction and feature extraction in the feature space, and uses recursive networks for modeling. ANNET proposed by Sivapalan *et al.* [9] proposed a lightweight machine learning for anomaly detection of edge sensors in the IoT framework. In this work, it combines multi-layer perceptrons with long-term and short-term memory networks to obtain global and instantaneous features respectively, improving computational efficiency while completing tasks. The core idea of Omni-Anomaly proposed by Su *et al.* [10] is to reconstruct data through the use of random recurrent neural networks and

Manuscript received July 5, 2024; revised December 8, 2024.

This work was supported in part by Young Expert Project of Taishan Scholar under Grant tsqnz20230602 and Natural Science Foundation of Shandong Province under Grant ZR2024MF115.

B. Jiang is an Associate Professor in the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao 266580, China (corresponding author phone: +86-0532-8618-8602; fax: +86-0532-8618-8602; e-mail: jiangbin@upc.edu.cn).

G. Wang is a graduate student in the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao 266580, China (e-mail: S22160037@s.upc.edu.cn).

X. Cui is a Professor in the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao 266580, China (e-mail: cxr@upc.edu.cn).

J. Wang is an Assistant Professor in the department of Computer Science, The University of Tennessee at Martin, Martin, TN 38238, USA (e-mail: jwang186@utm.edu).

planar normalized flows, and to propose a new threshold adjustment method (POT). Compared to previous methods, this method significantly improves detection accuracy, but significantly slows down inference speed.

Generative adversarial network is a structure with both good generative and judgmental abilities, which reduces the impact of insufficient label data. MAD-GAN proposed by Li *et al.* [11] combines LSTM and GAN, where the generator and discriminator play games with each other, while combining prediction error and discriminator loss. This method significantly improves performance and model robustness. USAD proposed by Audibert *et al.* [12] proposed a combination of automatic encoders and generative adversarial networks, which achieves unsupervised anomaly detection with high accuracy, fast speed, and low cost compared to previous methods. With the increasing rise of graph convolutional neural networks and transformer structures, GDN proposed by Deng *et al.* [13] considered the relationship between different dimensions of high-dimensional data. In order to obtain the correlation between different dimensions of data, attention mechanism is combined with graph convolutional neural networks, which significantly improves the performance of this method.TranAD proposed by Tuli *et al.* [14] combined the transformer structure with the generative adversarial network, which not only achieves parallel operation of data, greatly improves calculation speed, but also improves model training accuracy, making the overall model more stable. Based on the comprehensive summary of the existing methods mentioned above, we can find that mainstream methods require analysis of server data to obtain the results of anomaly detection. In this way, it puts forward new requirements for data privacy protection.

Federated learning is a distributed machine learning technology aimed at breaking down data silos and unleashing the potential of AI applications. It allows multiple participants to achieve joint modeling by exchanging encrypted machine learning intermediate results without disclosing the underlying data and encryption form of the underlying data. Federated learning gives consideration to AI application and privacy protection, and is applicable to business innovation scenarios in finance, consumer Internet and other industries. The existing federated learning methods mainly include: FedAvg [15], FedProx, SCAFFOLD, and MOON. FedAvg is the most fundamental federated learning algorithm, and other algorithms add regularization to FedAvg from different perspectives. In order to improve the real-time performance of anomaly detection for edge device failures, Liu *et al.* [16] combined federated learning with attention based convolutional neural networks to improve the real-time performance of industrial anomaly detection. Mothukuri *et al.* [17] proposed an anomaly detection method for the field of IoT security, which achieves strong privacy based anomaly detection by sharing central server parameters and updating local models. The recent advances in 5G and mobile edge computing facilitate the rapid development of the IOT, enabling collective intelligence with data support from a massive number of IoT devices. Meanwhile, federated leaning is prone to poor learning performance in large-scale IoT scenarios. Cui *et al.* [18] proposed a blockchain driven distributed asynchronous FL framework for anomaly detection in IoT systems, which ensures data integrity and prevents single points of failure while improving efficiency. Furthermore, an improved differential private FL based on generative adversarial networks was designed to optimize the data utility throughout the entire training process. Das *et al.* [19] highlight a high-level description of the current IoT architecture. In this article, the author provides a more in-depth discussion on the issue of anomaly detection in distributed learning.

III. Problem Statement

We address the issue of anomalies at the overall level of multivariate time series. Multi-dimensional time series can be defined as $X = [X_1, X_2, ..., X_n]$, where n is the amount of data collected, and $X^i = [X_1^i, X_2^i, ..., X_i^i]$ is an observation vector of the $i^(th)$ metric within a dimensionality of t. For entity-level multivariate time series anomaly detection, our target is to determine whether the observation $X_t = [X_t^1, X_t^2, ..., X_t^n]$ at time t is anomalous or normal.

Compared to traditional centralized learning methods, federated learning is an emerging machine learning method that allows multiple devices or computing nodes to train models without sharing raw data. In federated learning, model training is conducted on local devices, and only the updated model parameters are aggregated to a central server for aggregation. This distributed learning method can solve the problems of data privacy protection and data security, while reducing the need for data transmission and reducing the computational burden on the central server. Specifically, we assume that the data is owned by C clients, i.e., $X = [X(1), X(2), ..., X(C)]$, and that each client can only access their own data when collaborating to train a model. In a typical federated learning framework, each federated learning client uses its own private dataset to train the local model, and then each client sends its local model to the server. Finally, the server aggregates the local model into a global model, and this iterative cycle will continue several times until the model converges.

IV. Proposed Method

A. Lightweight Design for Proposed Model

We use the reconstructed transformer architecture for anomaly detection in multivariate time series. By reconstructing the data and setting the threshold, it is determined whether the data is abnormal or normal. Fig. 1 shows the framework of our lightweight anomaly detection model. First, the data is normalized to the range of [0,1]. In order to obtain the features of the data points within the local range, we use the local context window sequence W for training [10]. The length of each window sequence is set to K, and for window sequences that do not meet the length, they are filled by adding zeros before them. The encoder is divided into two parts: the first part encodes the complete sequence to obtain global features, the second part encodes the window sequence to obtain local features, and then undergoes data reconstruction through two decoders. The encoder converts the input sequence into a multi-modal matrix form and transforms the sequence using the scaled-dot product attention [20] containing Q (query), K (key value), and V (value) matrices, as shown in Eq. 1.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{m}}V) \qquad (1)$$

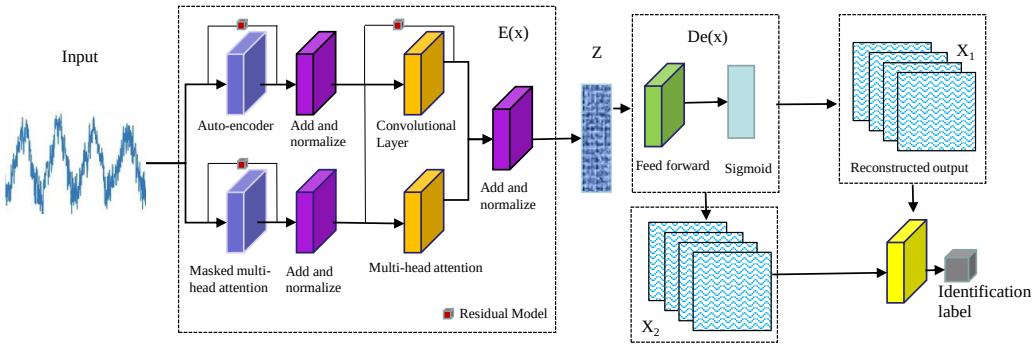

Fig. 1. Light-AD Model: a lightweight Anomaly Detection Model Based on Model Fusion and Attention Mechanism Decomposition

This process simplifies the inference operations of downstream neural networks by compressing the V-matrix into a smaller representative embedding using the softmax function. At the same time, the transformer uses a multi-head attention mechanism to process the input sequence. The multi-head attention mechanism refers to passing the input sequence through feedback layers with different number of heads (h_i) to obtain different Q_i, K_i, V_i, as shown in Eq. 2.

$$\begin{aligned} MultiHead(Q,K,V) &= Concat(H_1,...,H_h), \\ where H_i &= Attention(Q_i,K_i,V_i) \end{aligned} \tag{2}$$

Then, merging and passing them into a linear layer, and finally obtaining a matrix with the same dimension as the output of scaled dot product attention. Essentially, it involves parallel computation of multiple independent attention blocks, while simultaneously focusing on information from different subspaces at different positions in the input sequence.

B. Training Processing

We first perform positional encoding on the complete sequence(C) and window sequence(W). The encoded complete sequence extracts global information through an auto-encoder, and extracts local information through a convolutional layer [21]. The encoded window sequence is first subjected to a masked multi-head attention mechanism which purpose is to hide data from subsequent positions and prevent the model from viewing future data points as future timestamp values during the training process. This is because during model training, in order to achieve data parallel operations, all complete sequences C and window sequences W are inputted at once.

$$\begin{aligned} I_1^1 &= LayerNorm(I_1 + AutoEncoder(I_1) \\ I_2^1 &= LayerNorm(Mask(MultiHeadAtt(I_2,I_2,I_2)) \\ I &= LayerNorm(MultiHeadAtt(I_1^1,I_1^1,I_2^1) \\ &\quad + ConvLayer(I_1^1)) \end{aligned} \tag{3}$$

The processed window sequence(W) and the complete sequence(C) are simultaneously input into the attention mechanism, and then combined with the convolutional encoded representation to map to the hidden space. Finally, it is inputted into two decoders of the same architecture consisting of a feedforward network layer and a sigmoid activation function.

The sigmoid activation function is used to generate outputs within the range of [0,1]. The final result is that the model obtains two outputs O_1 and O_2 by inputting the complete sequence and window sequence, which is according to $O_i = Sigmoid(FeedForward(I))$.

Algorithm 1 The Light-AD Model Training Algorithm

Require: Encoder E, Decoders D_1 and D_2, Training dataset $\mathcal{W}$, Evolutionary hyperparameter ϵ, Iteration limit N

Initialize weights E, D_1, D_2
$n \leftarrow 0$
while n <N **do**
  for $t = 1$ to N **do**
    $O_1, O_2 \leftarrow D_1(E(W_t, \vec{0})), D_2(E(W_t, \vec{0}))$
    $L = \epsilon^{-n}||O_1 - W_t||_2 + (1-\epsilon^{-n}||O_2 - W_t||_2)$
    Update weights of E, D_1, D_2 using L
    $n \leftarrow n + 1$
  end for
end while

Algorithm 2 The Light-AD Model Testing Algorithm

Require: Trained Encoder E, Decoders D_1 and D_2 Test Dataset $\hat{\mathcal{W}}$

for $t = 1$ to $\hat{T}$ **do**
  $O_1, O_2 \leftarrow D_1(E(\hat{W}, \vec{0})), D_2(E(\hat{W}, \vec{0}))$
  $s = \frac{1}{2}||O_1 - \hat{W}||_2 + \frac{1}{2}||O_2 - \hat{W}||_2$
  $y_i = 1$ if $s_i \geq POT(s_i)$
  $y = \bigvee_i y_i$
end for

The above model inevitably faces the same challenges as other anomaly detection models, one of which is to maintain the stability of training. We use the L2-norm of the reconstructed output O_1, O_2, and the window sequence as the reconstruction loss. The first decoder aims to reconstruct the input as perfectly as possible, minimizing reconstruction errors, and deceiving the second decoder. The second decoder maximizes the same reconstruction error. As shown in Eq. 4, we use a loss function that combines reconstruction loss and adversarial loss. The entire model training and testing process is shown in Algorithms 1 and 2.

$$L = \epsilon^{-n}||O_1 - W||_2 + (1-\epsilon^{-n}||O_2 - W||_2 \tag{4}$$

C. The Model in Distributed Computing

We elaborate on the combination of federated learning framework and the aforementioned anomaly detection model, as shown in Fig. 2. Model aggregation can be achieved through various algorithms, including simple average aggregation algorithms, which as the name suggests are directly averaged on all client side training data, as shown in Eq. 5.

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{k,t} \tag{5}$$

There are also weighted average aggregation algorithms, which are weighted according to the quality of the model or the amount of its training data before averaging each model, as shown in Eq.6.

$$w_{w,T+\Delta T} = \sum_{k=1}^{k} \frac{n_k}{n} \left(w_T^K - \eta \sum_{t=T}^{T+\Delta T} G_t^k \right) \tag{6}$$

And gradient descent algorithms, *etc*, as shown in Eq.7.

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} (w_{k,t} - \eta \nabla f_k(w_{k,t}) + \mu(w_{k,t} - w_t)) \tag{7}$$

Before the formal start of training, the central server first distributes the anomaly detection model with initial parameters to each client, and then each client trains the obtained model using their own private dataset.

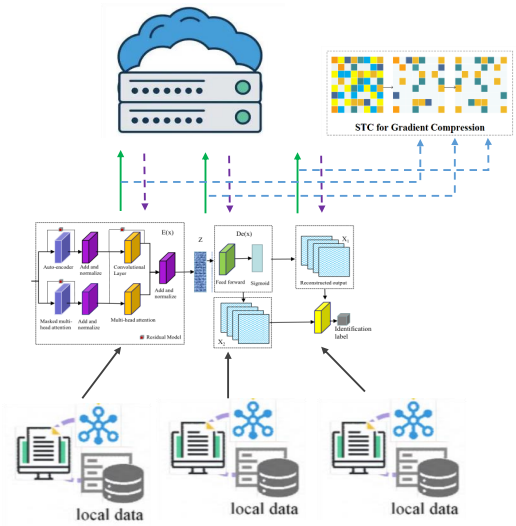

Fig. 2. Combination of Federated Framework and Light-AD Model

The parameter aggregation method we use is the FedAvg algorithm [15]. It also uses the sparse ternary compression method (STC) for gradient compression [22]. The conventional Topk sparsity method transfers sparse elements with full accuracy. Sattler *et al.* demonstrated that when sparsification is combined with quantization of non-zero elements, higher compression gains can be achieved as shown in Algorithm 3, when the Topk sparse element T^{masked} is obtained, it is quantified as the average of the sparse elements, so only a ternary tensor containing three values $\{\text{-}\mu,0,\mu\}$ needs to be passed in the end. If the gradients of each layer are treated as a matrix, the results obtained using Topk and sparse ternary compression are shown in Fig.2. The original gradient is a dense matrix, and the color depth represents the size of the values. By using the Topk method, a sparse matrix with larger values is obtained, while the values with smaller values are set to 0. Sparse ternary compression quantizes based on Topk, further improving the compression rate.

Sattler *et al.* used sparse ternary compression in FL to bi-directionally compress the communication gradients, and used an error feedback mechanism to accumulate errors for the next round of training.

$$\begin{aligned} \hat{g}_i^t &= STC\{g_i^t + error^{t-1}\} \\ error^t &= g_i^t - \hat{g}_i^t \end{aligned} \tag{8}$$

Among them, g_i^t is the original gradient obtained from the t-th round of training for the i-th client, $\hat{g}_i^t$ is the compressed gradient, and $error^t$ is the error before and after compression. This method achieves convergence speed similar to non-compression algorithms and greatly reduces communication volume in each round. Therefore, we will also use sparse ternary compression for gradient compression.

Algorithm 3 Sparse Ternary Compression

Input:$tensor\ T \in R^n, sparsness\ ratio : p$
Output:T^*
1.$k \leftarrow max(np, 1)$
2.$v \leftarrow top_k(|T|)$
3.$mask \leftarrow (|T| \geq v) \in \{0,1\}^n$
4.$T^{masked} \leftarrow mask \bigodot T$
5.$\mu \leftarrow \frac{1}{k} \sum_{i=1}^{n} |T_i^{masked}|$
6.$T^* \leftarrow \mu \times sign(T^{masked})$

V. Experimental Results and Analysis

A. Datasets, Experiment Settings and Evaluation Metrics

We use three publicly available datasets for experimental comparison, as shown in Table I.

1) PSM: a dataset proposed by eBay consisting 26 dimensional time series data from application servers.

2) MBA: a collection of electrocardiogram recordings from four patients,containing multiple instances of two different kinds of anomalies.

3) SMD: a five-week dataset of stacked traces of the resource utilizations of 28 machines from a compute cluster.

We uses the POT method in experiments to ensure smaller errors and more stable results [23], and uses the AdamW optimizer to train our model with an initial learning rate of 0.01 and step-scheduler with step size of 0.5. The hyperparameter values set in the experiment are as follows: Window size = 10, Number of layers in transformer encoders = 1, Number of layers in feed-forward unit of encoders = 2, Hidden units in encoder layers = 64, Dropout in encoders = 0.1. The only hyperparameter specific to the dataset is the number of headers in the multi header attention, which remains the same size as the dimension of the dataset. The time series is divided into 80% training data and 20% validation data in experiment. To avoid overfitting the model, we use early stopping criteria to train the model, which means that once

TABLE I
DETAILED INFORMATION OF DATASETS AND PERFORMANCE COMPARISON BETWEEN THE NEW MODEL AND TRANAD ON NON-IID DATA

Dataset	Train	Test	Dimensions	Clients	Method	Avg time	AUC-ROC	AUC-PR	Precision	Recall	F1
MBA	100000	100000	2	24	TranAD [14] + Fedavg [15]	10.05	0.40	0.29	0.24	0.06	0.10
					AE-Global	8.44	0.50	0.39	0.35	0.23	0.27
					Attention Split	10.93	0.49	0.40	0.51	0.13	0.21
					Light-AD	9.45	0.42	0.31	0.27	0.07	0.10
					C-Fed	9.20	0.45	0.33	0.35	0.10	0.15
PSM	2653	2611	25	24	TranAD + Fedavg	47.93	0.53	0.31	0.34	0.09	0.15
					AE-Global	39.45	0.56	0.33	0.38	0.09	0.15
					Attention Split	50.77	0.56	0.33	0.39	0.09	0.15
					Light-AD	44.67	0.56	0.34	0.39	0.13	0.19
					C-Fed	42.50	0.60	0.36	0.46	0.05	0.10
SMD	708405	708420	38	28	TranAD + Fedavg	233.67	0.69	0.23	0.60	0.10	0.18
					AE-Global	190.04	0.68	0.23	0.81	0.10	0.18
					Attention Split	247.56	0.68	0.23	0.81	0.10	0.18
					Light-AD	210.43	0.68	0.18	0.24	0.21	0.23
					C-Fed	238.24	0.69	0.28	0.70	0.19	0.29

the validation accuracy begins to decrease, the training stops. All dimensions of the dataset are scaled to the range of [0,1] to standardize the data.

We choose Average time of a global epoch (Avg time), AR (AUC-ROC:Area under the ROC Curve) and AP (AUC-PR:Area under the Precision-Recall Curve), Precision, Recall and F1-score as evaluation metrics. They are calculated as follows: $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, $F1 = \frac{2\times(Precision\times Recall)}{Precision+Recall}$. We search for the best threshold that has the highest F1-score for each experiment.

TABLE II
PERFORMANCE COMPARISON ON IID-DATA

Dataset	Method	Avg time	AUC-ROC	AUC-PR	Precision	Recall	F1
MBA	TranAD +Fedavg	10.94	0.39	0.30	0.24	0.06	0.10
	AE-Global	6.98	0.50	0.39	0.35	0.22	0.27
	Attention Split	9.52	0.49	0.39	0.49	0.13	0.20
	Light-AD	10.68	0.41	0.30	0.26	0.07	0.11
	C-Fed	10.77	0.43	0.32	0.32	0.08	0.13
PSM	TranAD+Fedavg	126.75	0.61	0.36	0.42	0.13	0.20
	AE-Global	116.66	0.58	0.36	0.42	0.15	0.23
	Attention Split	154.96	0.60	0.39	0.70	0.05	0.10
	Light-AD	118.81	0.58	0.36	0.43	0.14	0.21
	C-Fed	118.28	0.58	0.36	0.43	0.14	0.21
SMD	TranAD+Fedavg	233.35	0.68	0.21	0.42	0.11	0.17
	AE-Global	195.89	0.68	0.23	0.81	0.10	0.18
	Attention Split	247.26	0.68	0.19	0.51	0.06	0.11
	Light-AD	205.82	0.68	0.18	0.24	0.21	0.23
	C-Fed	237.17	0.69	0.28	0.70	0.19	0.29

B. Numerical Results and Analysis

We compared the performance of our model on three publiclyavailabledatasetsforiidandnon-iiddata,and theresultsareshowninTableIandTableII. We mainly analyze non-iid data in experiments. As shown in the table, replacing the global encoder with an auto-encoder (AE-Global) can significantly reduce the average training time of the model, and there is a certain improvement in various accuracy indicators. For each global epoch in the MBA dataset, the average run time is reduced by 16.02%, with 25.00% increase in precision metric. For the PSM dataset, the average runtime is reduced by 17.69% and the precision metric increases by 5.66%. Only decomposing the Transformer multi-head attention blocks (Attention Split) has a slight impact on the average running time of the model, but there is a certain improvement in the accuracy of various aspects of the model. For each global epoch in the MBA dataset, the average run time increases by 8.76%, with 22.50% increase in precision metric. For the PSM dataset, the average runtime increases by 5.93% and results in 5.66% increase in the precision metric.

By combining auto-encoder and attention mechanism decomposition (Light-AD), for each global epoch in the MBA dataset, the average run time is reduced by 5.97%, with 12.50% increase in precision metric. For the PSM dataset, the average runtime is reduced by 6.80% and the precision metric increases by 14.71%. After compression federated framework (C-Fed), for each global epoch in the MBA dataset, the average run time is reduced by 8.36%, resulting in 45.83% in precision metric. For the PSM dataset, the average run time is reduced by 11.33%, resulting in 35.29% in precision metric.

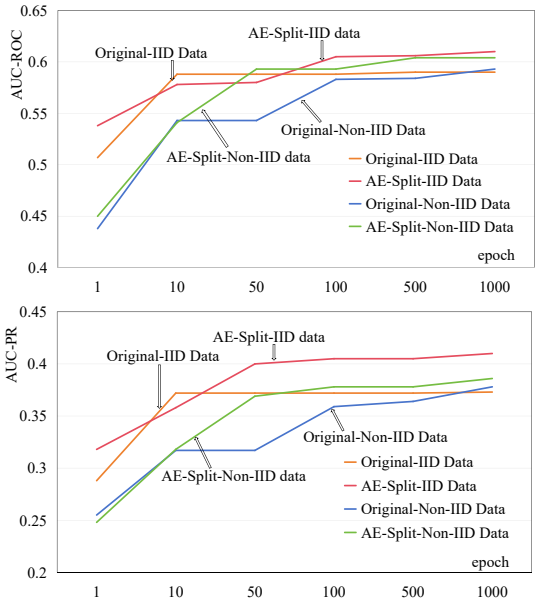

Fig. 3. Comparison of Model Performance Before and After Local Model Improvement Based on IID Data and Non-IID Data

For datasets of different sizes, the training time of the model is significantly different, but compared with the o-

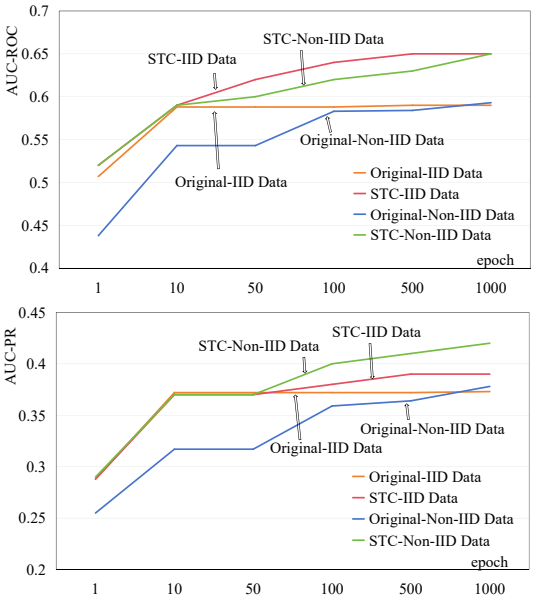

Fig. 4. Comparison of Model Performance Before and After Global Model Improvement Based on IID Data and Non-IID Data

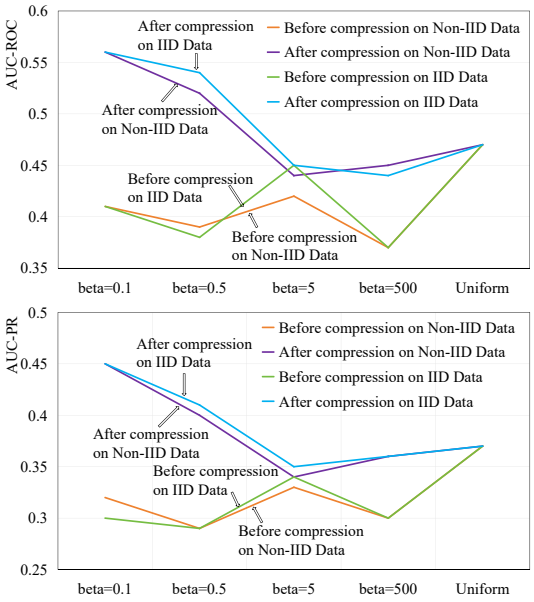

Fig. 6. Performance Comparison of Models Before and After Compression under Different Client Partitions Based on IID Data and Non-IID Data

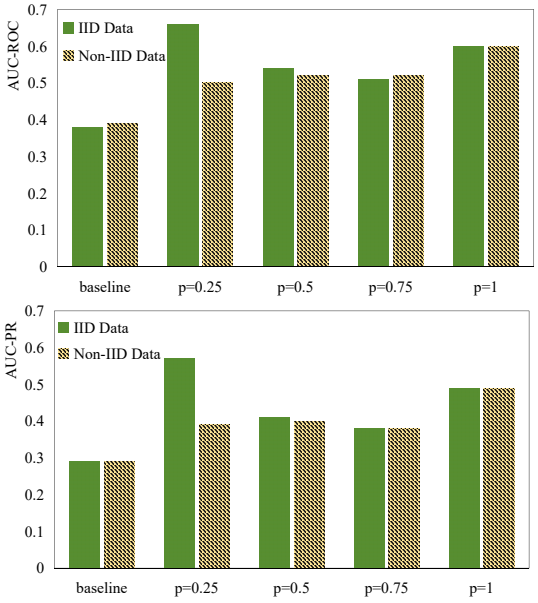

Fig. 5. Comparison of Model Performance under Different Sparsity Rates (p) for Non-IID Data and IID Data

riginal model, the new model reduces the running time and do not affect the accuracy of anomaly detection, ensuring the normal function of anomaly detection. From the current experimental results, we can demonstrate that the new model has significant effects in terms of lightweight models.

Fig. 3 displays the variation in model performance with the number of global iterations before and after improving the local model based on non-iid and iid data. Experimental results show that the model metrics continuously improve until they tend to stabilize as the number of iterations increases. The improved model has a certain improvement in performance compared to the baseline, whether based on iid or non-iid data. When the iteration reaches 1000 rounds, the baseline AR reaches 0.59 based on iid and non-iid data, while the improved model AR can reach 0.61. Similar to the AR curves, the overall curves show that the improved model performs better than the baseline. The good performance of the model on independent data is due to its statistical independence, which makes the training process faster and the effect better.

Fig. 4 shows the variation of various model indicators with the number of global iterations before and after introducing STC based on non-iid and iid data. When the number of iterations reaches 1000 rounds, the baseline AR reaches 0.59, while the improved model AR can reach 0.65. Similar to the AR curves, the baseline AP reaches 0.37, while the AR of the improved model can reach 0.39. Based on non-iid data, the baseline AP reaches 0.38, while the improved model can reach 0.42. Overall, the overall curves show that the improved model performs better than the baseline. The model's better performance on independent data is due to its statistical independence, which makes the training process faster and the effect better.

As shown in Fig. 5, comparison of models at different p based on iid and non-iid data shows that the STC algorithm improves the detection accuracy of the model overall. Compared to baseline, C-Fed accuracy improved significantly at different sparsity rates. For example,when p is equal to 0.25, the AR of C-Fed on iid data reaches0.66, which is about 73.68% higher than the baseline,and the AP of C-Fed on iid data reaches 0.57, whichis about 96.55% higher than the baseline. AR on C-Fed reaches 0.50 and AP on C-Fed reaches 0.39 based on non-iid data. Moreover, compared to iid data, the algorithm performs better on independently distributed data, which is due to the influence of dataset features.

As shown in Fig. 6, experimental results based on iid and non-iid data under different client partitions (beta) show that C-Fed has a significant improvement over baseline, and different beta has no significant impact on overall performance, with results fluctuating within a certain range. In addition, under a specific client partition, C-Fed performs better on iid data than on non-iid data, due to the characteristics of the

client data. Overall, a large number of experimental results indicate that compared to the baseline, the improved model achieves light weight while ensuring the accuracy of various indicators.

VI. Conclusion

We propose a lightweight FL-based anomaly detection model that ensures accuracy in anomaly detection while protecting data privacy. First, we reduce the network structure and computational complexity of the local model through model fusion and attention mechanism decomposition. Secondly, we compress the model on the federated framework using gradient compression algorithm, reducing the number of communications required for convergence and the amount of data transmitted per round, without affecting the accuracy of anomaly detection. Finally, the results of various experiments show that our model has significant optimization compared to the baseline.

References

[1] L. Shen, W. Cui, Y. Tao, T. Shi, and J. Liao, "Surface defect detection algorithm of hot-rolled strip based on improved yolov7," *IAENG International Journal of Computer Science*, vol. 51, no. 4, pp. 345–354, 2024.

[2] M. C. Belavagi and B. Muniyal, "Intrusion detection using rule based approach in rpl networks," *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp. 988–999, 2023.

[3] Z. Chen, C. Bai, Y. Zhu, and X. Lu, "Tut: Template-augmented u-net transformer for unsupervised anomaly detection," *IEEE Signal Processing Letters*, vol. 31, pp. 780–784, 2024.

[4] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. Association for Computing Machinery, 2018, p. 387395.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[7] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[8] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.

[9] G. Sivapalan, K. K. Nundy, S. Dev, B. Cardiff, and D. John, "Annet: A lightweight neural network for ecg anomaly detection in iot edge sensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 1, pp. 24–35, 2022.

[10] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, 2019, p. 28282837.

[11] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*. Cham: Springer International Publishing, 2019, pp. 703–716.

[12] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020, pp. 3395–3404.

[13] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," *CoRR*, vol. abs/2106.06947, pp. 4027–4035, 2021.

[14] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *CoRR*, vol. abs/2201.07284, 2022.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282.

[16] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.

[17] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2022.

[18] L. Cui, Y. Qu, G. Xie, D. Zeng, R. Li, S. Shen, and S. Yu, "Security and privacy-enhanced federated learning for anomaly detection in iot infrastructures," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3492–3500, 2022.

[19] T. Das, R. M. Shukla, and S. Sengupta, "What could possibly go wrong? identification of current challenges and prospective opportunities for anomaly detection in internet of things," *IEEE Network*, vol. 37, no. 3, pp. 194–200, 2023.

[20] S. Tuli, S. Tuli, R. Verma, and R. Tuli, "Modelling for prediction of the spread and severity of covid-19 and its association with socioeconomic factors and virus types," *MedRxiv*, pp. 2020–06, 2020.

[21] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with long-short range attention," in *8th International Conference on Learning Representations, ICLR 2020,Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[22] F. Sattler, S. Wiedemann, K.-R. Mller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, 2020.

[23] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17, 2017, p. 10671075.