applied sciences

MDPI

Article

Feature-Attended Federated LSTM for Anomaly Detection in the Financial Internet of Things †

Yunlong Li [1], Rongguang Zhang [1], Pengcheng Zhao [2,3] and Yunkai Wei [2,3,*]

[1] College of Management Science, Chengdu University of Technology, Chengdu 610059, China; liyunlong@stu.cdut.edu.cn (Y.L.); zhangrg6880@163.com (R.Z.)

[2] Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Chengdu 611731, China; 202221011129@std.uestc.edu.cn

[3] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

* Correspondence: ykwei@uestc.edu.cn

† This paper is an extended version of our paper published in IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York, NY, USA, 2–5 May 2022; Feature-Attended Multi-Flow LSTM for Anomaly Detection in Internet of Things, pp. 1–6.

Abstract: Recent years have witnessed the fast development of the Financial Internet of Things (FIoT), which integrates the Internet of Things (IoT) into financial activities. At the same time, the FIoT is facing an increasing number of stealthy network attacks. Long short-term memory (LSTM) can be used as an anomaly-detecting method to perceive such attacks since it specializes in discovering anomaly behaviors through the time correlation in FIoT traffic. However, current LSTM-based anomaly detection schemes have not considered the specific correlations among the features of the whole traffic. In addition, current schemes are usually trained based on local traffic with rare cooperation among different detecting nodes, leading to the result that current schemes usually suffer from insufficient adaptability and low coordination. In this paper, we propose a feature-attended federated LSTM (FAF-LSTM) for FIoT to address the above issues. FAF-LSTM combines feature-attended LSTM and federated learning to make full use of the deep correlation in data and enhance the accuracy of the trained model via cooperation among different detecting nodes. In FAF-LSTM, the features are grouped so that the model can learn the time–spatial correlation inner the flows of each group as well as their impact on the output. Meanwhile, the parameter aggregation is optimized based on feature correlation analysis. Simulations are conducted to verify the effect of FAF-LSTM. The results show that FAF-LSTM has good performance in anomaly detection. Compared with independently trained LSTM and traditional federated learning-based LSTM, FAF-LSTM can improve the detection accuracy by up to 39.22% and 334.36%, respectively.

Keywords: feature-attended; anomaly detection; long short-term memory; federated learning; Financial Internet of Things

check for updates

Citation: Li, Y.; Zhang, R.; Zhao, P.; Wei, Y. Feature-Attended Federated LSTM for Anomaly Detection in the Financial Internet of Things. *Appl. Sci.* **2024**, *14*, 5555. https://doi.org/10.3390/app14135555

Academic Editor: Baraq Ghaleb

Received: 12 May 2024
Revised: 4 June 2024
Accepted: 24 June 2024
Published: 26 June 2024

1. Introduction

In recent years, financial products have made a closed loop involving multiple market entities and are usually based on the information of practical objects, such as a car, a house or a famous painting in a mortgage loan. Financial institutions generally need more information about the objects to reduce risks and improve efficiency. The Internet of Things (IoT) can provide full-process and full-time information about the objects. The combination of finance and IoT leads to a new pattern, i.e., the Financial Internet of Things (FIoT), which has great potential in the information era [1,2]. However, potential attacks may seriously threaten the security and efficiency of the FIoT. The detection of potential attacks has become a vital issue in FIoT. Actually, the anomalies in the network provide us with a way to detect potential threats in a timely manner, thus making anomaly detection an important method of identifying network attacks [3].

At present, there are many studies on anomaly detection in IoT, but barely specialized for FIoT. The literature [4] proposes an anomaly detection scheme based on fuzzy theory. In [5], principal component analysis (PCA) is used to detect abnormal behaviors in aging Industrial IoT. A Squeezed Convolutional Variational Auto Encoder (SCVAE) for anomaly detection in Industrial IoT is proposed in [6]. Based on an adaptive learning rate and momentum, a method for trustworthy network anomaly detection is proposed in [7], whereas these methods ignore the time correlation of the network traffic and have limitations in the anomaly detection performance. Motivated by the excellent performance of long short-term memory recurrent neural network (LSTM-RNN) in natural language processing and speech recognition, Kim et al. [8] proposed an anomaly detection method with the LSTM network, and Meng et al. [9] trained the model based on kernel PCA and LSTM-RNN to detect anomalies. Nevertheless, these methods leave the time correlation of each feature itself out of consideration, which is a significant characteristic of the FIoT traffic.

At the same time, financial entities, such as persons, banks, or related companies, have stricter rules on using the data generated from their FIoTs. They cannot freely exchange data to train more powerful anomaly-detecting models. The anomaly detection in FIoT faces problems such as data isolation and data shortage during independent training. This may lead to impaired accuracy of the obtained models. Thus, federated learning as a new technology was applied in anomaly detection. Li et al. [10] proposed a federated learning-based anomaly detection framework called Deepfed, for identifying threats in cyber–physical systems. Chen et al. [11] presented federated learning-based attention gated recurrent unit (FedAGRU), an anomaly detection algorithm for wireless edge networks. However, the environmental difference of different nodes exists objectively, and the traditional federated learning model is not highly adaptable to the different environments.

To solve the problems above, this paper combines the LSTM network and federated learning based on the correlation of data features. We propose multi-channel LSTM based on data association, apply federated learning architecture as the solution to the lack of data and optimize parameter aggregation strategy based on the data correlation. Therefore, the adaptability and performance of the model are improved in different environments. Our contributions are summarized as follows:

- We propose the architecture of FAF-LSTM, so as to improve the utilization of the time correlations, the coordination among detection nodes, and the adaptability to a dynamic changing environment. FAF-LSTM is composed of the feature-attended LSTM and the correlation-based federated learning.
- We propose the structure of the feature-attended LSTM, design a novel algorithm to classify the features into different groups, and present the detailed schemes to extend LSTM from a single training channel into multiple training channels so that the correlation among the features can be fully utilized.
- We apply the federated learning architecture to solve problems like the lack of training data faced by single detection devices and enhance the synergy of detection devices. According to the traffic characteristics of each detecting node, the correlation is analyzed, and the parameter aggregation strategy in cooperative training is optimized to improve the detection models.

The remainder of this paper is organized as follows. The related work is introduced in Section 2. In Section 3, the architecture of feature-attended federated LSTM is demonstrated. Section 4 shows the detailed schemes in the proposed architecture about feature-attended LSTM, while the specific framework for feature-attended federated learning is described in Section 5. In Section 6, the UNSW-NB15 dataset is simulated as a case to verify the performance of the proposed architecture which is compared with standard LSTM in individual training and in conventional federated average training. Finally, Section 7 concludes this paper.

2. Related Work

In the FIoT, anomaly detection is critical for the security and integrity of financial transactions, as it handles sensitive data and faces significant risks from data breaches and fraud [12]. Current anomaly detection research primarily targets general IoT systems, overlooking the distinct challenges posed by financial environments. This section reviews anomaly detection algorithms that are potentially applicable to FIoT. To date, the research for anomaly detection could be roughly divided into non-machine learning studies and machine learning studies. Further, machine learning methods include mathematical-model-based and deep learning-based methods.

2.1. Non-Machine Learning-Based Anomaly Detection

Ye and Chen [13] chose to apply the chi-square theory in anomaly detection. Based on this technique, a profile of normal events in the information system can be created to distinguish and detect abnormal events and intrusion events which are quite different from normal events. In order to detect anomalies, Altaher et al. [14] proposed a solution by analyzing the relative entropy changes in traffic characteristics: firstly, calculate the entropy distribution of each time interval, then calculate the average entropy value of a specific time interval, and finally distinguish the normal and abnormal traffic behaviors of the network by comparing the variance of entropy and the adaptive threshold. However, the algorithms based on statistical rules can detect fewer types of exceptions, and their learning ability and adaptability to the environment are not high.

2.2. Machine Learning-Based Anomaly Detection

Detection algorithms based on machine learning are widely studied. Shyu M L et al. [15] performed PCA on the correlation matrix of the normal group to realize a novel scheme using a robust principal component classifier in intrusion detection problems, based on an assumption that the number of normal instances has to be much larger than the number of anomalies. Yang et al. [16] proposed a new mixed model of information entropy and support vector machine "Ent-SVM", which classifies abnormal network behaviors based on the normalized eigenvalues of six characteristics of the traffic. But the above methods are not applicable to dynamic and time-varying anomaly detection for the reason that they are based on static, time-invariant models.

To solve that problem of anomaly detection with dynamic and time-varying characteristics, the LSTM [17,18] network was adopted for anomaly detection in a dynamic environment, somehow combined with support vector machine (SVM) [19]. Based on the predictive error, LSTM was combined with the Gaussian Bayes model in [20] for anomaly detection. The literature [21] proposed a novel LSTM model, combining attention mechanism and convolutional neural network (CNN) to detect anomalies. Similarly, a new hybrid deep learning (DL) approach based on CNN to classify the flow traffic into normal or attack classes is mentioned in [22]. Bontemps et al. [23] proposed a proposal to detect collective anomalies based on the LSTM network and improved detection accuracy by predicting and modeling the correlation between stationary and non-stationary time so as to realize effective detection of time abnormal structures. These methods take full advantage of the temporal relevance of IoT to LSTM. However, in these methods, the time correlations inner each feature (or a feature group including several features) are ignored. This leads to the result that such an important character in IoT traffic, which may greatly improve anomaly detection performance, is not utilized.

2.3. Federated Learning in IoT

The above discussion is based on a single node and ignores the variations between different nodes, which are characteristic of the FIoT. This oversight results that current anomaly detection methods suffer from insufficient adaptability and low coordination. H. Brendan et al. [24] combined federated learning and LSTM to train a global model in multiple node networks. In federated learning, the parameter aggregator collects the parameters of the trained model from each distributed training node and sets the average value of each parameter to be the parameter value in the converged model. Therefore, traditional federated learning can be abbreviated as Fed-Avg. But statistical heterogeneity due to the non-IID distribution of data across IoT devices often leads to the local models trained solely on their private data performing better than the global shared model. To solve this problem, Y. Mansour et al. [25] presented a systematic learning–theoretic study of personalization in learning and proposed and analyzed three algorithms. M. Zhang et al. [26] presented a flexible personalized federated learning framework that achieves strong performance across various non-IID settings. However, the federated learning approaches proposed in these studies focus only on general IoT applications and do not consider the specific requirements of the FIoT nor conduct experimental analysis within the FIoT context.

It can be seen that most of the current research focuses on the improvement of a single detection model or cooperative training among nodes in IoT. There is still a lack of anomaly detection methods that combine financial application scenarios and implement improvement and improvement from the model's own performance and training cooperation in the FIoT.

3. Our Proposed Architecture

In order to solve the problems such as low accuracy of the detection algorithm, insufficient coordination among detection nodes and poor adaptability to a dynamic changing environment in abnormal traffic detection in FIoT, an architecture of feature-attended federated LSTM is proposed in this paper.

3.1. Architecture of Feature-Attended Federated LSTM

As shown in Figure 1, the proposed architecture includes two parts, a feature-attended LSTM algorithm and a correlation-based federated learning scheme.

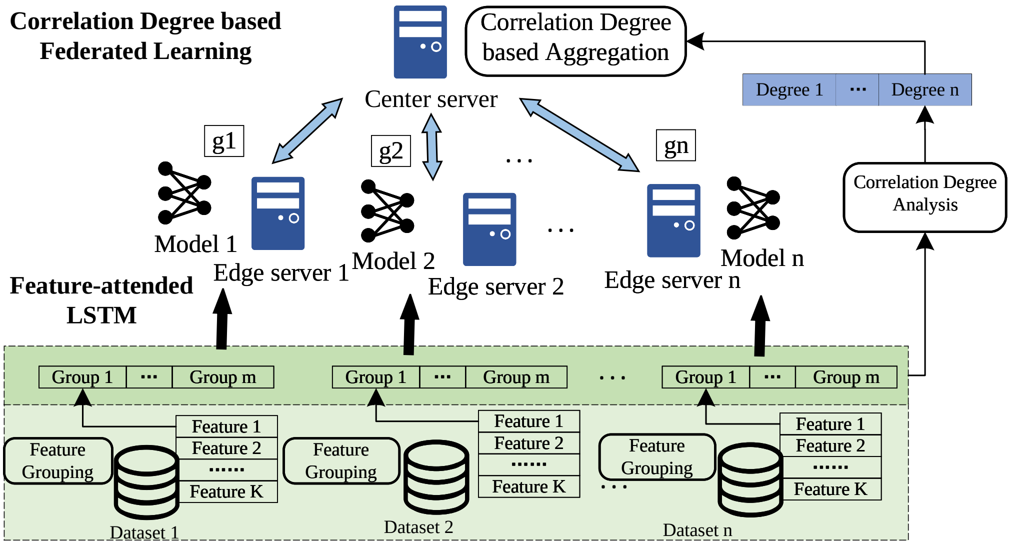


Figure 1. Architecture of FAF-LSTM.

The part of feature-attended LSTM is designed based on the time correlation among different features. In this part, the features of input data will be divided into multiple groups according to the correlation, and then the grouped features will be used to train the local model in edge servers. With this part, the model can make full use of the training data in the training process as well as balance the cost of training calculation while improving the model performance.

The other part is a correlation-based federated learning scheme, which is based on the correlation of training data belonging to different detection nodes. In this part, edge servers send the local model parameters to the center server, and then the correlation-based federated learning technology we proposed is used to realize the cooperative training of each detection node and solve problems like lack of data or imbalance in the independent training process. Finally, the central server sends the aggregation model parameter to each edge server, and the aggregation model parameters based on the correlation degree could be used to optimize the local model as well as improve the adaptability of the model in the differential environment and optimize the model accuracy.

3.2. Feature-Attended LSTM

The structure of feature-attended LSTM includes two parts, feature grouping and training process. First of all, the FIoT devices receive traffic data as a time series and form a dataset, then the dataset will be divided into several proper groups according to the feature grouping algorithm based on the correlation of features. In the training process, feature-attended LSTM firstly constructs multiple streams as inputs. Next, a corresponding number of Information Extraction Units (IEUs) with the same hidden layer structure as LSTM are established on each stream. These IEUs can extract the information according to the grouped features. Then, each stream sends the output to the reconstruction layer. The final output result is at last achieved based on the weighted impact factors.

In this way, the time correlation of each feature and its influence on the final model is fully considered to optimize the model performance. Furthermore, the grouping strategy of features is optimized to achieve a balance between computational cost and training efficiency.

3.3. Correlation-Based Federated Learning

The correlation-based federated learning includes two parts, cooperative training and aggregation optimization. In the cooperative training part, each detection node first uses local traffic data for model training to obtain local parameters and then uploads model parameters to the parameter aggregation node for federated optimization. Then, the model is trained in the cooperative scheme according to the model parameters updated by the parameter aggregation node, and the model parameters of each detection node are updated by the parameter aggregation node according to the weight of parameter aggregation so as to provide the next round of training for each detection node. Aggregation optimization is mentioned to find a trade-off between using global data features and local data features, and a parameter aggregation weight optimization algorithm is proposed based on the above cooperative process. The local parameter weight and global parameter weight of each detection node were determined by feature extraction and correlation degree and weight calculation.

In this way, the global data feature is indirectly used to improve the problem of insufficient data through the coordination of the parameter aggregation node to each detection node. By updating the model parameters of detection nodes in a weighted way, the global data features can be utilized while the local data features can be appropriately retained to improve the detection accuracy and adaptability of the model to different financial scenarios.

4. Detail of Feature-Attended LSTM

This section will introduce the feature grouping algorithm and model training process under the feature-attended LSTM scheme in detail.

4.1. Model Training

As shown in Figure 2, the dataset is noted as D, which consists of $L(L \in N_+)$ flows with $K(K \in N_+)$ dimension features. Let $f_i(i \in N_+, i \leq K)$ represent the i-th feature, and the feature set could be noted as $F = \{f_1, f_2, \cdots, f_k\}$. Then, according to the feature grouping algorithm, K data features will be divided into $m(m \in N_+, m \leq K)$ groups. Given time step parameter is $v(v \in N_+)$ dataset D will be transformed into $s(s \in N_+)$ samples $D_i(i \in N_+, i \leq s)$ each sample X_i has a length of m. As s is the largest integer not greater than $\frac{L}{v}$, the remaining data will be ignored. Thus, the processed dataset is represented as $D^{'} = \{D_1, D_2, \cdots, D_s\}$.

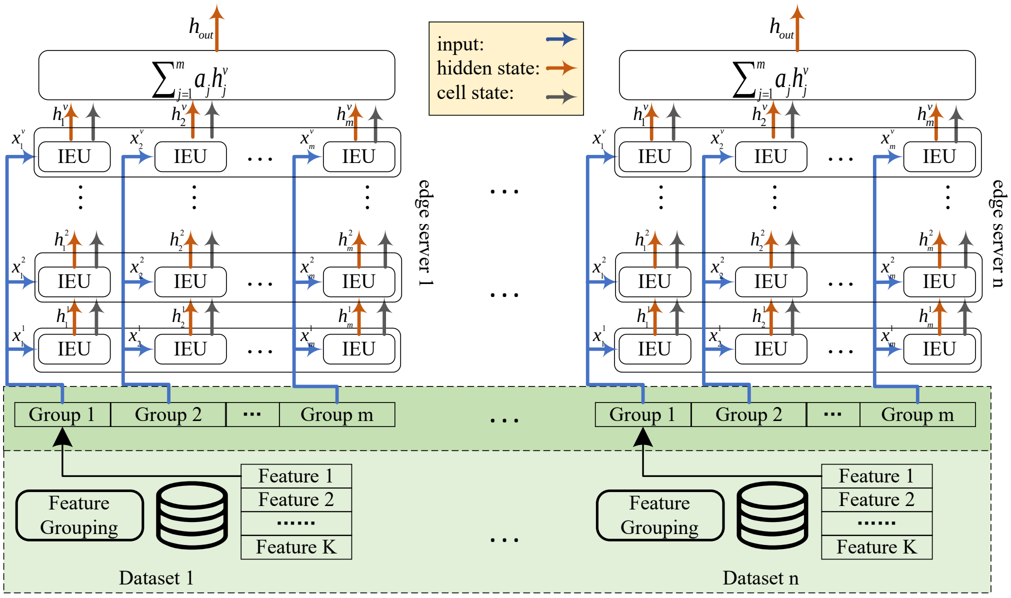


Figure 2. Structure of feature-attended LSTM.

After processing the dataset, the feature-attended LSTM model will construct m training channels, and each training channel contains v IEU to extract the feature information. These channels will focus on extracting feature information from the corresponding group. Then, the extracted information will be contributed to the reconstruction layer for fusion so as to achieve delicate perceptual extraction from local to global. IEU is a neural network with the same structure as LSTM, consisting of an input gate (Ig), a forget gate (Fg), and an output gate (Og).

Let $G = \{g_1, g_2, \cdots, g_m\}(m \in N_+, m \leq K)$ be the group set and $x_j(t)$ be the input data at the t-th time step of the j-th group g_i, and the input gate, output gate and forget gate be Ig, Fg and Og, respectively. Then, the calculation formulas of the input gate, forget gate and output gate of the j-th flow at the t-th time step are Formulas (1)–(3):

$$Ig_j(t) = \sigma\Big(W_{(Ig,xj)}x_j(t) + W_{(Ig,hj)}h_j(t-1) + b_{(Ig,j)}\Big) \tag{1}$$

$$Fg_j(t) = \sigma\Big(W_{(Fg,xj)}x_j(t) + W_{(Fg,hj)}h_j(t-1) + b_{(Fg,j)}\Big) \tag{2}$$

$$Og_j(t) = \sigma\Big(W_{(Og,xj)}x_j(t) + W_{(Og,hj)}h_j(t-1) + b_{(Og,j)}\Big) \tag{3}$$

where $W_{(Ig,xj)}$, $W_{(Ig,hj)}$ and $b_{(Ig,j)}$ represent the input weight, the hidden weight and the bias of input gate, respectively. $W_{(Fg,xj)}$, $W_{(Fg,hj)}$ and $b_{(Fg,j)}$ represent the input weight, the hidden weight and the bias of forget gate, respectively. $W_{(Og,xj)}$, $W_{(Og,hj)}$ and $b_{(Og,j)}$ represent the input weight, the hidden weight and the bias of output gate, respectively. Then, the output $h_j(t)$ of the j-th flow at the t-th time step is calculated as Formulas (4)–(6):

$$z_j(t) = tanh(W_{xj}x(t) + W_{hj}h(t-1) + b_j) \tag{4}$$

$$c_j(t) = Fg_i(t) \odot c_j(t-1) + Ig_j(t) \odot z_j(t) \tag{5}$$

$$h_j(t) = Og_j(t) \odot tanh(c_j(t)) \tag{6}$$

where z is the input node with a tanh activation function and c stands for the memory cell's state.

As shown in Figure 2, the j-th IEU output $h_j^v (j \leq m)$ will serve as the input of the reconstruction layer, and the reconstruction layer will be calculated based on the output data of each channel as Formula (7):

$$h_{out} = \sum_{j=1}^{m} a_j h_j^v \tag{7}$$

where a_j is the impact factor of h_j^v and the coefficient used to adjust the output weight of the channel. It can be obtained by the back-propagation algorithm based on the final result of the model. The influence factor can evaluate the influence of each feature on the overall training result from a global perspective so that the model can focus on extracting the internal time correlation of each feature while considering the correlation between various features and accelerating the convergence of the model.

After the h_{out} is obtained, the model parameters can be adjusted by comparing them with the label information of the training data. Assume that $\widehat{y}$ and y represent the predicted value and label value, respectively, the predicted value will be calculated according to Formula (8) and (9):

$$h_y = net(h_{out}) \tag{8}$$

$$\widehat{y} = softmax(h_y) \tag{9}$$

where net represents the fully connected neural network and $softmax$ represents the SoftMax activation function.

4.2. Feature Grouping

In the financial traffic data set, data features have a certain correlation, that is, features are not completely orthogonal. If the features with low correlation are divided into a group as input, the feature information cannot be fully extracted, thus affecting the model performance. If a corresponding training channel is constructed for each feature, the computational overhead in the model training process will increase greatly.

In a simple case, we can put each feature into one group. That is to say, we have k groups. Assume S_{max} denotes the anomaly detection accuracy in this simple case. Then, we can put two features f_i and f_j into one group, and we have $k-1$ groups. Assume $S_{i,j}$ denotes the anomaly detection accuracy in this case. Therefore, given a pre-defined threshold P, γ_{f_i,f_j} can be obtained from Formula (10):

$$\gamma_{f_i,f_j} = \begin{cases} true, & \left(\frac{S_{max}-S_{i,j}}{S_{max}} - P\right) \geq 0 \\ false, & \left(\frac{S_{max}-S_{i,j}}{S_{max}} - P\right) < 0 \end{cases} \tag{10}$$

Then, the eigen correlation matrix Γ could be expressed as Equation (11):

$$\Gamma = \begin{bmatrix} \gamma_{f_1,f_1} & \cdots & \gamma_{f_1,f_K} \\ \cdots & \ddots & \vdots \\ \gamma_{f_K,f_1} & \cdots & \gamma_{f_K,f_K} \end{bmatrix} \tag{11}$$

When executing the feature grouping algorithm, the matrix Γ is traversed. For each element $\gamma_{f_i,f_j}(i,j \leq K)$, if $\left(\gamma_{f_i,f_j} = true\right)$ and $((i == K)\&(j == K) = false)$, we will then choose one of the following actions:

- When f_i and f_j are already in one same group, switch to $\gamma_{f_i,f_{j+1}}$ (when $j < K$) or γ_{f_{i+1},f_1} (when $j = K$).
- When f_i is in a group g_a, and f_j is in another group g_b, g_a and g_b should be merged into one new group g'. Then, switch to $\gamma_{f_i,f_{j+1}}$ (when $j < K$) or γ_{f_{i+1},f_1} (when $j = K$).

- When f_i (or f_j) is in a group, but f_j (or f_i) not, we should put f_j (or f_i) into the group that f_i (or f_j) is in. Then, switch to $\gamma_{f_i,f_{j+1}}$ (when $j < K$) or γ_{f_{i+1},f_1} (when $j = K$).
- When f_i is not in any group, and f_j neither, we should create a new group, and put both f_i and f_j into this group. Then, switch to $\gamma_{f_i,f_{j+1}}$ (when $j < K$) or γ_{f_{i+1},f_1} (when $j = K$).

Algorithm 1 shows the detailed feature-based iterative grouping algorithm.

Algorithm 1 Feature-based iterative grouping algorithm

```
Require: correlation matrix $\Gamma$, group set $G$
Initialize $G = \varnothing$
for $i = 1; i \leq K; i++$ do
    for $j = 1; j \leq K; j++$ do
        if $\gamma_{f_i,f_j} == true$ then
            if $\exists f_i, f_j \in g, g \in G$ then
                continue to next loop;
            else if $\exists f_i \in g_a, f_i \in g_b$, and $g_a \in G, g_b \in G (g_a \cap g_b = \varnothing)$ then
                create a new group $g' = g_a \cup g_b$, then set $g' \in G$;
            else if $\exists f_i \in g_a, g_a \in G$ and $\forall f_j \notin g_b (b \in N_+, b \leq m)$ then
                set $f_j \in g_a$;
            else if $\exists f_j \in g_a, g_a \in G$ and $\forall f_i \notin g_b (b \in N_+, b \leq m)$ then
                set $f_i \in g_a$;
            else
                create a new group $g_l = \{f_i, f_j\}$, then set $g_l \in G$;
            end if
        end if
    end for
end for
```

5. Detail of Correlation-Based Federated Learning

This section will specifically introduce and explain the cooperative training process of multi-detection nodes in the cooperative anomaly detection architecture based on federated learning, as well as the aggregation weight optimization algorithm of key parameters in the training process.

5.1. Cooperative Training

In the proposed architecture, the center server is expressed as C, then the parameter calculated by C, which is also called the global parameter, could be expressed as p_c. The set of edge servers is noted as E, while the numbers of edge servers is n, then, $E = \{e_1, e_2, \cdots, e_n\}$. The corresponding parameters, which are also called local parameters, could be recorded as $p_j (j \leq n, j \in N_+)$. In parameter aggregation, the weight of the global parameter and local parameter are noted as $w^C_{e_(i)}$, where $w^C_{e_(i)} + w_{e_i} = 1$. Under the cooperative training scheme, the edge servers begin training with an organization of the center server in a cooperative. The detection model is co-trained by e_j under the organization of C.

Before cooperative training, the center server C sets the global parameter weight ($w_{e^C_j}$) and local parameter weight (w_{e_j}) of each edge server e_j used in parameter aggregation according to the aggregation algorithm, which has been introduced in part III. At first, the edge server e_j trains its model with a local dataset to obtain its local parameter p_j, which will be sent to the center server C. Next, C calculates the global parameter according to Formula (12):

$$p_C = \frac{1}{n} \sum_{j=1}^{n} p_j \tag{12}$$

Then, C calculates the new parameter $p_j^{'}$ of e_j according to Formula (13) and sends the new parameter to the corresponding edge server helping to update its model.

$$p_j^{'} = w_{e_j}^{C} p_C + w_{e_j} p_j \tag{13}$$

The above steps are repeated until the loss function converges or the maximum number of iterations is reached, then the training is stopped and the current detection model is saved. In this way, the federated learning architecture is used to realize the cooperative training and updating of the multi-detection-node detection model, and the model optimization is realized by combining local parameters and global parameters in a weighted way.

The detailed feature grouping algorithm is shown in Algorithm 2.

Algorithm 2 Cooperation training algorithm

```
Require: model parameter $p_j$, iteration rounds $R$
Initialize: parameter weight $w_{d_i}^{C}$, $w_{d_i}$, iteration variable r
// aggregation node executes
while $r \leq R$ do
    receive local model parameter $p_j$ from $d_j$
    calculate global parameter $p_C$
    calculate new parameter $p_j^{'}$ for $d_j$
    send $p_j^{'}$ to $d_j$
    $r = r + 1$
end while
// detecting nodes execute
while $r < R$ do
    send local model parameter $p_j$ to aggregation node
    //waiting for new parameter $p_j^{'}$
    if Receive a timeout then
        using original parameter $p_j$ for training
    else
        using new parameter $p_j^{'}$ for training
    end if
    $r = r + 1$
end while
```

5.2. Aggregation Optimization

To seek the tradeoff between global parameters and local parameters, a parameter aggregation weight optimization algorithm was proposed under the coordination of multiple detection nodes based on federated learning by associating data feature changes in each detection node with model training optimization.

The parameter aggregation weight optimization algorithm could be divided into two stages: feature extraction and correlation degree analysis. In the feature extraction stage, each detection node forms a feature sequence by calculating the information entropy of flow data. In the calculation stage of correlation degree and weight, the relative entropy of each detection node is calculated by the parameter aggregation node based on the feature sequence. This section will introduce and explain the two parts in detail, respectively.

5.2.1. Feature Extraction

In this stage, the information entropy using t as a constant interval is used for extracting traffic features. In this paper, source IP address and destination IP address are used as examples of entropy calculation since the historical financial behavior of IP devices and the financial interactions between them are crucial. While traffic characteristics can be selected according to specific requirements in practical financial applications, the specific values of t and T should be determined based on the actual network situation.

The total flow of the edge server e_j in a unit time is expressed as $N_j^{flow}(N_j^{flow} \in N_+)$, and e_j has $N_j^{src}(N_j^{src} \in N_+)$ sorts of source IP address and $N_j^{dst}(N_j^{dst} \in N_+)$ sorts of destination IP address. While the source IP address is expressed as a random variable Y, the number of occurrences of a certain source IP address could be expressed as $y_u(u \in N_+)$. Then, the information entropy of edge server e_j, which is recorded as $b_{e_j}^i(src), (i \leq T, i \in N_+)$ could be calculated according to Formula (14).

$$b_{e_j}^i(src) = \sum_{u=1}^{N_j^{src}} \frac{y_u}{N_j^{flow}} \log_2 \frac{y_u}{N_j^{flow}} \tag{14}$$

In the same way, the destination IP address could be expressed as a random variable Z, and the number of occurrences of a certain destination IP address could be expressed as z_u. Then, the information entropy of edge server e_j, which is recorded as $b_{e_j}^i(dst)$, could be calculated according to Formula (15).

$$b_{e_j}^i(dst) = \sum_{u=1}^{N_j^{dst}} \frac{z_u}{N_j^{flow}} \log_2 \frac{z_u}{N_j^{flow}} \tag{15}$$

Then, the feature values $b_{e_j^i}$ of the edge server e_j should be calculated according to Formula (16), in order to retrain the trend of entropy change.

$$b_{e_j}^i = \left| b_{e_j}^i(src) - b_{e_j}^i(dst) \right| \tag{16}$$

Finally, the center server C could calculate the feature value in the i-th unit time at a global vision according to Formula (17).

$$b_C^i = \frac{1}{n} \sum_{j=1}^{n} b_{e_j}^i \tag{17}$$

5.2.2. Correlation Degree Analysis

Over T unit time intervals, the center server C obtains a feature sequence named B_C, and $B_C = \{b_C^1, b_C^2, \cdots, b_C^T\}$. The edge servers also obtain their feature sequences in the same way, which could be expressed as B_{e_j}, and $B_{e_j} = \left\{B_{e_i}^1, B_{e_i}^2, \cdots, B_{e_i}^T\right\}$. Then, the relative entropy $L(B_{e_i}||B_C)$ based on the edge server feature sequence B_{e_j} and center server feature sequence B_C could be calculated according to Formula (18).

$$L(B_{e_i}||B_C) = \sum_{i=1} Tb_{e_j^i} \ln \frac{b_{e_j}^i}{h_C^i} \tag{18}$$

In the next step, the relative entropy $L(B_{e_i}||B_C)$ is used to analyze and quantify the correlation degree according to Formulas (19) and (20).

$$w_{e_j} = \tanh(L(B_{e_i}||B_C)) \tag{19}$$

$$w_{e_j}^C = 1 - \tanh(L(B_{e_i}||B_C)) \tag{20}$$

where the larger the value of $w_{e_j}^C$ indicates that in the cooperative update, the larger the proportion of the global parameter in parameter aggregation. Conversely, the larger the proportion of local parameters in parameter aggregation.

The detailed feature grouping algorithm is shown in Algorithm 3.

Algorithm 3 Parameter optimization algorithm

```
Require: eigenvalue sequence length $T$, number of detecting nodes $m$
// detecting nodes execute
Initialize: current number of eigenvalues $i$
for $i = 0;i < T$ do
    calculate eigenvalues for the current feature
end for
end eigenvalue sequence to aggregation node
// aggregation node executes
for $j = 0; j < m$ do
    receive eigenvalue sequence $H_{d_j}$ from $d_j$
end for
calculate eigenvalue sequence $H_C$ based on $H_{d_j}$
calculate correlation degree of $H_C$ and $H_{d_j}$
calculate parameter weight $w_{d_j}^{C}$, $w_{d_j}$ for $d_j$
```

6. Simulation

In this section, we conduct simulations to evaluate the anomaly detection performance of FAF-LSTM based on the UNSW-NB15 dataset, utilizing its characteristics of network intrusions as proxies for unusual transactions and financial intrusions and theft in FIoT. The result shows that FAF-LSTM can improve the classification effect considerably in anomaly detection.

6.1. Simulation Environment

The simulation is conducted on the UNSW-NB15 dataset. The reason for using this dataset is that it contains up-to-date attacked data, which makes it an ideal dataset for detecting contemporary attacks in FIoT, such as device intrusion and data theft. For more details, the total number of records is 2,540,044 which are stored in the CSV files and every record has 49 features [27]. Although this dataset has nine types of attacks, we just intend to distinguish between normal and abnormal traffic behavior.

We divide the dataset into four parts, pretending that the data collected by four detecting nodes are located at different areas in the network, which are named Node I to Node IV. To simulate the difference in each device in IoT, the number of records is different among the four nodes. Specifically, the four nodes have 137,485, 30,829, 9428, 123,986 records, respectively.

The proposed system was developed by Python version 3.8 using the Pytorch and Keras packages. The specification of the computer includes an Intel-based processor core i-5 @ 2.30 GHz, a 64-bit operating system, and 16 GB memory.

6.2. Evaluation Metrics

In order to verify the anomaly detection results of the proposed algorithm, there are three metrics of classification tasks are comprehensively used in the simulation. The first is accuracy, denoted as A, which can directly measure the overall correctness including the correct identification of both anomalies and normal instances. The second metric is the area under the curve (AUC), representing the area under the receiver operating characteristic (ROC) curve, which assesses the algorithm's ability to discriminate between classes at various thresholds. A higher AUC indicates better algorithm performance, crucial for effective anomaly detection in dynamic scenarios. The last metric is the true positive rate, which indicates the proportion of actual anomalies correctly identified by the algorithm. This is crucial for financial systems, where missing an anomaly can have severe consequences. The accuracy is calculated as follows:

$$A = \frac{TP + TN}{TP + FP + FN + TN} \tag{21}$$

where TP represents the number of correctly classified target samples, TN represents the number of correctly classified other samples, FP represents the number of incorrectly identified target samples and FN represents the number of target samples that were missed.

6.3. Simulation Results and Analysis

Based on the above evaluation criteria, we compare our proposed FAF-LSTM with conventional Federated Averaging algorithm and independent training LSTM. For the models obtained under different training modes training by four nodes, their accuracy is shown in Table 1. While the independent LSTM models trained by Node I to Node IV obtain the accuracy of 0.9401, 0.2209, 0.6540 and 0.9416, and the Fed-Avg LSTM models [24] obtain the accuracy of 0.2287, 0.2225, 0.2170 and 0.2209, our proposed FAF-LSTM models arrive at an accuracy of 0.9648, 0.9315, 0.9680 and 0.9738. The accuracy of models trained by different nodes has slightly better performance.

Table 1. Simulation accuracy on different training modes.

Training Mode	Node I	Node II	Node III	Node IV
Independent LSTM	0.9401	0.2209	0.6540	0.9416
Fed-Avg LSTM	0.2287	0.2225	0.2170	0.2209
FAF-LSTM	0.9648	0.9315	0.9680	0.9738

Furthermore, we evaluated the accuracy of these models as the scale of flow increased, the result is shown in Figure 3. Compared to the test result of three different training modes on four nodes, independent LSTM can obtain higher accuracy in Node I and Node IV which have sufficient training data, but this training mode performed terribly in Node II and Node IV which lack enough data. When we compared FedAvg LSTM, it did not have a better result, while the FAF-LSTM model we proposed always has the best performance among the four nodes.

The average accuracy of Node I to Node IV is shown in Figure 4, it can be observed that our proposed FAF-LSTM model has a distinct advantage over the FedAvg LSTM and independent LSTM.

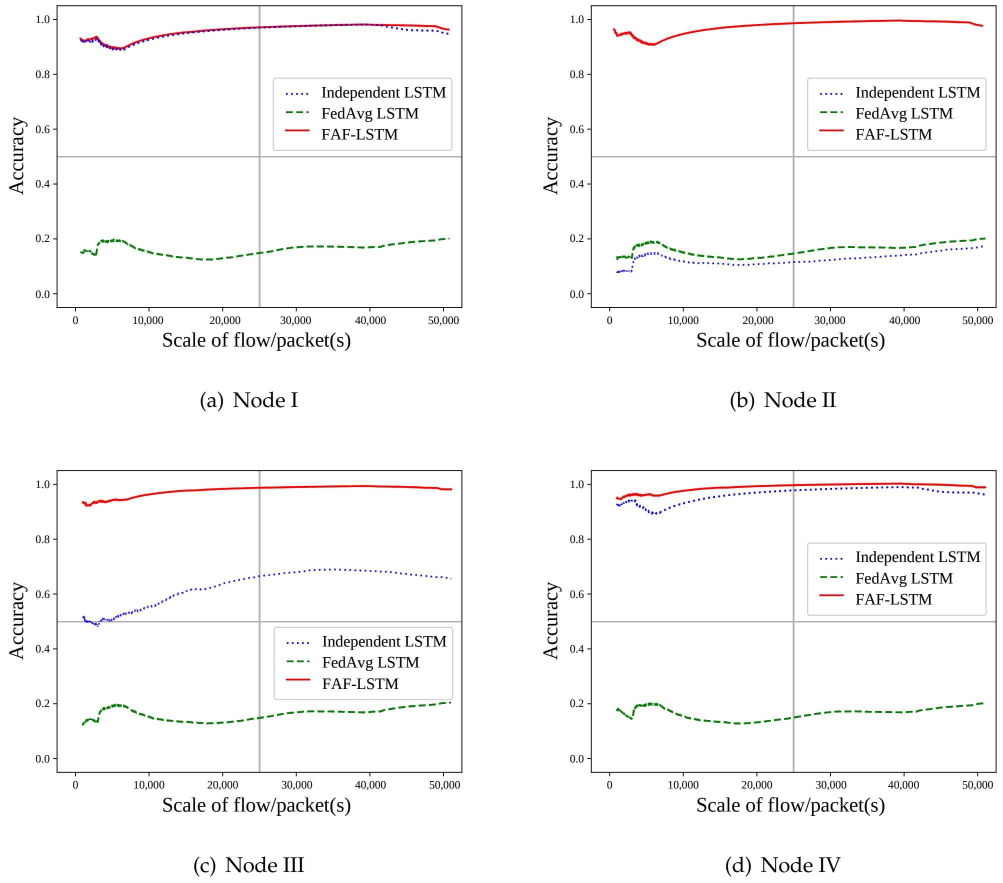
The ROC is shown in Figure 5. As the AUCs of FAF-LSTM are 0.8159, 0.7451, 0.8040 and 0.8688, the AUCs of Fed-Avg LSTM are 0.5508, 5457, 0.5491, 0.5495, and the AUCs of independent LSTM are 0.6513, 0.4095, 0.6468, 0.6472, respectively. The FAF-LSTM is slightly better in all these evaluation criteria.

Generally, compared with independent LSTM, Fed-Avg LSTM cannot improve the performance of each model. But our proposed FAF-LSTM has improved the accuracy and AUC value. The average improvements are shown in Table 2.

Table 2. Improvement compared with different training modes.

Indicator	Compare with Independent LSTM	Compare with Fed-Avg LSTM
Accuracy Increase	39.22%	334.36%
AUC Increase	24.92%	47.13%

As can be seen from Figures 4 and 5, Tables 1 and 2, the proposed schemes in this paper not only improve the performance of a single model but also improve the performance of each model as a whole.



(a) Node I

(b) Node II

(c) Node III

(d) Node IV

Figure 3. Accuracy of Node I to Node IV.

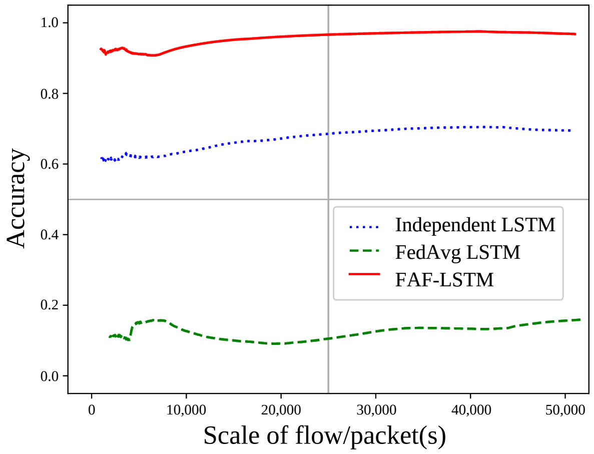
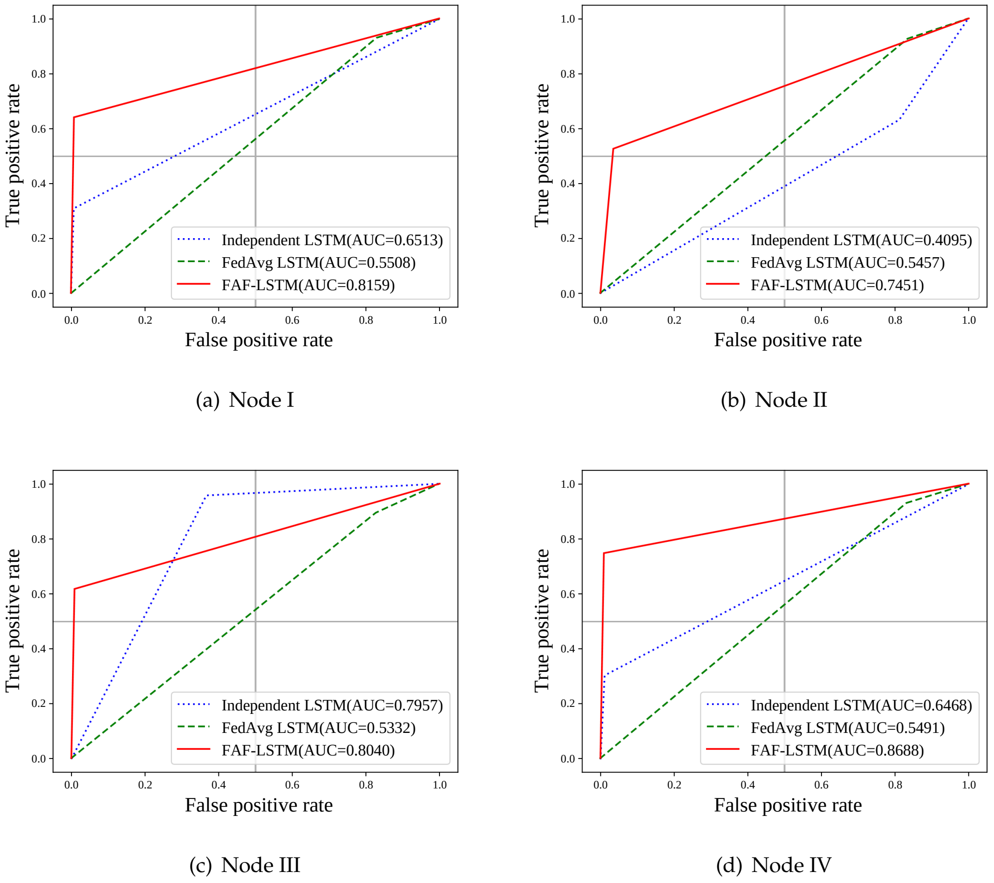


Figure 4. Average accuracy of four nodes.



(a) Node I

(b) Node II

(c) Node III

(d) Node IV

Figure 5. ROC of Node I to Node IV.

7. Conclusions

In this paper, we put forward a novel scheme named FAF-LSTM to improve the anomaly-detection accuracy of the FIoT. In FAF-LSTM, the input data are processed and divided into multi-flows to train the model according to the time correlation among the features. Based on this, FAF-LSTM fully utilizes the time correlation in the FIoT traffic in the aspect of the feature level. Then, in feature-attended federated learning, the parameter aggregation optimization strategy is proposed based on the correlation of flow entropy calculation, and federated learning architecture is applied to realize cooperative training at multiple detection nodes. We conduct simulations to evaluate FAF-LSTM based on the UNSW-NB15 dataset. The simulation results show that our proposed scheme can fundamentally improve the anomaly detection performance in FIoT and outperform the benchmark schemes by up to 334.36% in aspect of the detection accuracy. For future work, the performance of FAF-LSTM can be verified across different datasets and environments. And other models, such as CNN or SVM, can also be combined with federated learning and compared with FAF-LSTM.

Author Contributions: Conceptualization, Y.W., R.Z. and Y.L.; methodology, Y.L. and R.Z.; software, P.Z.; validation, P.Z., R.Z. and Y.L.; formal analysis, Y.L.; investigation, Y.L. and Y.W; resources, Y.L.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.W. and R.Z.; visualization, Y.L.; supervision, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by by Innovation Ability Construction Project for Sichuan Provincial Engineering Research Center of Communication Technology for Intelligent IoT (2303-510109-04-03-318020), the Municipal Government of Quzhou under Grant No. 2022D027 and No. 2023D003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are unavailable due to privacy.

Acknowledgments: The authors want to thank Luhan Zou, Lixiang Ma and Supeng Leng for valuable discussions and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Pecori, R.; Tayebi, A.; Vannucci, A.; Veltri, L. Iot attack detection with deep learning analysis. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
2. Peng, H.; Shen, X. Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 2416–2428. [CrossRef]
3. Ahmed, M.; Mahmood, A.N.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]
4. Peng, Y.; Tan, A.; Wu, J.; Bi, Y. Hierarchical edge computing: A novel multi-source multi-dimensional data anomaly detection scheme for industrial internet of things. *IEEE Access* **2019**, *7*, 111257–111270. [CrossRef]
5. Genge, B.; Haller, P.; Enăchescu, C. Anomaly detection in aging industrial internet of things. *IEEE Access* **2019**, *7*, 74217–74230. [CrossRef]
6. Kim, D.; Yang, H.; Chung, M.; Cho, S.; Kim, H.; Kim, M.; Kim, K.; Kim, E. Squeezed convolutional variational autoencoder for unsupervised anomaly detection in edge device industrial internet of things. In Proceedings of the 2018 International Conference on Information and Computer Technologies (ICICT), DeKalb, IL, USA, 23–25 March 2018; pp. 67–71.
7. Yan, X.; Xu, Y.; Xing, X.; Cui, B.; Guo, Z.; Guo, T. Trustworthy network anomaly detection based on an adaptive learning rate and momentum in iiot. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6182–6192. [CrossRef]
8. Kim, J.; Kim, J.; Thu, H.L.T.; Kim, H. Long short term memory recurrent neural network classifier for intrusion detection. In Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), Jeju, Republic of Korea, 15–17 February 2016; pp. 1–5.
9. Meng, F.; Fu, Y.; Lou, F.; Chen, Z. An effective network attack detection method based on kernel PCA and LSTM-RNN. In Proceedings of the 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), Dalian, China, 25–27 December 2017.
10. Li, B.; Wu, Y.; Song, J.; Lu, R.; Li, T.; Zhao, L. Deepfed: Federated deep learning for intrusion detection in industrial cyber–physical systems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5615–5624. [CrossRef]
11. Chen, Z.; Lv, N.; Liu, P.; Fang, Y.; Chen, K.; Pan, W. Intrusion detection for wireless edge networks based on federated learning. *IEEE Access* **2020**, *8*, 217463–217472. [CrossRef]
12. Vanini, P.; Rossi, S.; Zvizdic, E.; Domenig, T. Online payment fraud: From anomaly detection to risk management. *Financ. Innov.* **2023**, *9*, 66. [CrossRef]
13. Ye, N.; Chen, Q. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Qual. Reliab. Eng. Int.* **2001**, *17*, 105–112. [CrossRef]
14. Altaher, A.; Ramadass, S.; Thuraisingham, B.; Mehedy, M. On-line anomaly detection based on relative entropy. In Proceedings of the 2011 4th IEEE International Conference on Broadband Network and Multimedia Technology, Shenzhen, China, 28–30 October 2011; pp. 33–36.
15. Shyu, M.-L.; Chen, S.-C.; Sarinnapakorn, K.; Chang, L. A novel anomaly detection scheme based on principal component classifier. In Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, Melbourne, FL, USA, 19–22 November 2003; pp. 172–179.
16. Yang, C. Anomaly network traffic detection algorithm based on information entropy measurement under the cloud computing environment. *Clust. Comput.* **2019**, *22*, 8309–8317. [CrossRef]
17. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
18. Shao, H. Delay-dependent stability for recurrent neural networks with time-varying delays. *IEEE Trans. Neural Netw.* **2008**, *19*, 1647–1651. [CrossRef] [PubMed]
19. Lindemann, B.; Maschler, B.; Sahlab, N.; Weyrich, M. A survey on anomaly detection for technical systems using lstm networks. *Comput. Ind.* **2021**, *131*, 103498. [CrossRef]
20. Wu, D.; Jiang, Z.; Xie, X.; Wei, X.; Yu, W.; Li, R. Lstm learning with bayesian and gaussian processing for anomaly detection in industrial iot. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5244–5253. [CrossRef]
21. Liu, Y.; Garg, S.; Nie, J.; Zhang, Y.; Xiong, Z.; Kang, J.; Hossain, M.S. Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach. *IEEE Internet Things J.* **2020**, *8*, 6348–6358. [CrossRef]
22. ElSayed, M.S.; Le-Khac, N.-A.; Albahar, M.A.; Jurcut, A. A novel hybrid model for intrusion detection systems in sdns based on cnn and a new regularization technique. *J. Netw. Comput. Appl.* **2021**, *191*, 103160. [CrossRef]

23. Bontemps, L.; Cao, V.L.; McDermott, J.; Le-Khac, N.-A. Collective anomaly detection based on long short-term memory recurrent neural networks. In *Future Data and Security Engineering: Third International Conference, FDSE 2016, Can Tho City, Vietnam, 23–25 November 2016, Proceedings 3*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 141–152.
24. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; PMLR: New York, NY, USA, 2017; pp. 1273–1282.
25. Mansour, Y.; Mohri, M.; Ro, J.; Suresh, A.T. Three approaches for personalization with applications to federated learning. *arXiv* **2020**, arXiv:2002.10619.
26. Zhang, M.; Sapra, K.; Fidler, S.; Yeung, S.; Alvarez, J.M. Personalized federated learning with first order model optimization. *arXiv* **2020**, arXiv:2012.08565.
27. Moustafa, N.; Slay, J. Unsw-nb15: A comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In Proceedings of the 2015 military communications and information systems conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.