Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach

Yi Liu, *Student Member, IEEE*, Sahil Garg, *Member, IEEE*, Jiangtian Nie, *Graduate Student Member, IEEE*, Yang Zhang, *Member, IEEE*, Zehui Xiong, *Member, IEEE*, Jiawen Kang, and M. Shamim Hossain, *Senior Member, IEEE*

***Abstract*—Since edge device failures (i.e., anomalies) seriously affect the production of industrial products in Industrial IoT (IIoT), accurately and timely detecting anomalies are becoming increasingly important. Furthermore, data collected by the edge device contain massive user's private data, which is challenging current detection approaches as user privacy has attracted more and more public concerns. With this focus, this article proposes a new communication-efficient on-device federated learning (FL)-based deep anomaly detection framework for sensing time-series data in IIoT. Specifically, we first introduce an FL framework to enable decentralized edge devices to collaboratively train an anomaly detection model, which can improve its generalization ability. Second, we propose an attention mechanism-based convolutional neural network-long short-term memory (AMCNN-LSTM) model to accurately detect anomalies. The AMCNN-LSTM model uses attention mechanism-based convolutional neural network units to capture important fine-grained features, thereby preventing memory loss and gradient dispersion problems. Furthermore, this model retains the advantages of the long short-term memory unit in predicting time-series data. Third, to adapt the proposed framework to the timeliness of industrial anomaly detection, we propose a gradient compression mechanism based on Top-*k* selection to improve communication efficiency. Extensive experimental studies on four real-world data sets demonstrate that our framework accurately and timely detects anomalies and also reduces the communication overhead by 50% compared to the FL framework that does not use the gradient compression scheme.**

***Index Terms*—Deep anomaly detection (DAD), federated learning (FL), gradient compression, Industrial Internet of Things.**

Manuscript received April 16, 2020; revised June 24, 2020; accepted July 14, 2020. Date of publication July 24, 2020; date of current version April 7, 2021. This work was supported in part by the Alibaba Group through the Alibaba Innovative Research (AIR) Program and the Alibaba-NTU Singapore Joint Research Institute (JRI), NTU, Singapore; in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China, under Grant ICT20044; in part by the National Natural Science Foundation of China under Grant 51806157; in part by the Young Innovation Talents Project in Higher Education of Guangdong Province, China, under Grant 2018KQNCX333; and in part by the Researchers Supporting Project under Grant RSP-2020/32, King Saud University, Riyadh, Saudi Arabia. This article was presented in part at the International Conference on Blockchain and Trustworthy Systems, Aug. 2020. *(Corresponding author: Jiawen Kang.)*

Yi Liu is with the School of Data Science and Technology, Heilongjiang University, Harbin 150080, China (e-mail: 97liuyi@ieee.org).

Sahil Garg is with the Electrical Engineering Department, École de technologie supérieure, Université du Québec, Montreal, QC H3C 1K3, Canada (e-mail: sahil.garg@ieee.org).

Jiangtian Nie is with the Energy Research Institute, Interdisciplinary Graduate Programme, and School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: jnie001@e.ntu.edu.sg).

Yang Zhang is with the Hubei Key Laboratory of Transportation Internet of Things, School of Computer Science and Technology, Wuhan University of Technology, Wuhan 639798, China (e-mail:yangzhang@whut.edu.cn).

Zehui Xiong is with the Alibaba-NTU Joint Research Institute and School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: zxiong002@e.ntu.edu.sg).

Jiawen Kang is with the Energy Research Institute, Nanyang Technological University, Singapore (e-mail: kavinkang@ntu.edu.sg).

M. Shamim Hossain is with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: mshossain@ksu.edu.sa).

Digital Object Identifier 10.1109/JIOT.2020.3011726

I. Introduction

THE WIDESPREAD deployment of edge devices in the Industrial Internet of Things (IIoT) paradigm has spawned a variety of emerging applications with edge computing, such as smart manufacturing, intelligent transportation, and intelligent logistics [1]. The edge devices provide powerful computation resources to enable real time, flexible, and quick decision making for the IIoT applications, which has greatly promoted the development of Industry 4.0 [2]. However, the IIoT applications are suffering from critical security risks caused by abnormal IIoT nodes that hinder the rapid development of IIoT. For example, in smart manufacturing scenarios, industrial devices acting as IIoT nodes, e.g., engines with sensors, that have abnormal behaviors (e.g., abnormal traffic and irregular reporting frequency) may cause industrial production interruption thus resulting in huge economic losses for factories [3], [4]. Edge devices (e.g., industrial robots) generally collect sensing data from IIoT nodes, especially time-series data, to analyze and capture the behaviors and operating conditions of IIoT nodes by edge computing [5]. Therefore, these sensing time-series data can be used to detect the anomaly behaviors of IIoT nodes [6].

To solve the abnormality problems from IIoT devices, typical methods are to perform abnormal detection for the affected IIoT devices [7]–[10]. The previous work focused on utilizing deep anomaly detection (DAD) [11] approaches

to detect abnormal behaviors of IIoT devices by analyzing sensing time-series data. From historical time-series data, DAD techniques can learn hierarchical discriminative features. Malhotra *et al.* [12], Lu *et al.* [13], and Lu *et al.* [14] proposed a long short-term memory (LSTM) networks-based deep learning model to achieve anomaly detection in sensing time-series data. Munir *et al.* [15] designed a novel DAD approach, called DeepAnT, to achieve anomaly detection by utilizing the deep convolutional neural network (CNN) to predict anomaly value. Although the existing DAD approaches have achieved success in anomaly detection, they cannot be directly applied to the IIoT scenarios with distributed edge devices for timely and accurate anomaly detection. The reasons are twofold: 1) the most of detection models are not flexible enough in traditional approaches, the edge devices lack dynamic and automatically updated detection models for different scenarios, and hence, the models fail to accurately predict frequently updated time-series data [8] and 2) due to privacy concerns, the edge devices are not willing to share their own collected time-series data with each other, thus the data exist in the form of "islands." The data islands significantly degrade the performance of anomaly detection. In addition, people often overlook the sensitive private information of training data, resulting in potential privacy leaks. The anomaly detection context has some privacy issues. For instance, the anomaly detection model will reveal the patient's heart disease history when detecting the patient's abnormal pulse [16], [17].

To address the above challenges, a promising on-device privacy-preserving distributed machine learning paradigm, called on-device federated learning (FL), was proposed for edge devices to train a global DAD model while keeping the training data sets locally without sharing raw training data [18]. Such a framework allows edge devices to collaboratively train an on-device DAD model without compromising privacy. For example, Ito *et al.* [19] proposed an on-device FL-based approach to achieve collaborative anomaly detection. Tsukada *et al.* [20] utilized the FL framework to propose a backpropagation neural networks (i.e., BPNNs) based approaches for anomaly detection. However, previous researches ignore the communication overhead during model training using FL among large-scale edge devices. Expensive communication overhead may cause excessive overhead and long convergence time for edge devices so that the on-device DAD model cannot quickly detect anomalies. Therefore, it is necessary to develop a communication-efficient on-device FL framework to achieve accurate and timely anomaly detection for edge devices.

In this article, we present a communication-efficient on-device FL framework that leverages an attention mechanism-based CNN-LSTM (AMCNN-LSTM) model to achieve accurate and timely anomaly detection for edge devices. First, we introduce an FL framework to enable distributed edge devices to collaboratively train a global DAD model without compromising privacy. Second, we propose an AMCNN-LSTM model to detect anomalies. Specifically, we use attention-based CNNs to extract fine-grained features of historical observation-sensing time-series data and use LSTM modules for time-series prediction. Such a model can prevent memory loss and gradient dispersion problems. Third, for further enhancing the communication efficiency of our framework, a gradient compression mechanism (GCM) based on Top-k selection is proposed to reduce the number of gradients uploaded by edge devices. We evaluate the proposed framework on real-world data sets as follows: space shuttle, power demand, engine, and ECG. The experimental results show that the proposed framework can achieve high-efficiency communication and achieve accurate and timely anomaly detection. The contributions of this article are summarized as follows.

1) We introduce an FL framework to develop an on-device collaborative DAD model for edge devices in IIoT.
2) We propose an AMCNN-LSTM model to detect anomalies, which uses CNN to capture the fine-grained features of time-series data and uses the LSTM module to accurately and timely detect anomalies.
3) We propose a Top-k selection-based gradient compression scheme to improve the proposed framework's communication efficiency. Such a scheme decreases communication overhead by reducing the exchanged gradient parameters between the edge devices and the cloud aggregator.
4) We carry out extensive experiments on four real-world data sets to demonstrate our framework can accurately detect anomalies with low communication overhead.

II. Related Work

A. Deep Anomaly Detection

DAD has always been a hot issue in IIoT, which serves as a function of detecting anomalies. Previous researches about DAD generally are divided into three categories: 1) *unsupervised DAD* approaches; 2) *supervised DAD*; and 3) *semisupervised DAD*.

Supervised Deep Anomaly Detection: A supervised DAD typically trains a deep-supervised binary or multiclass classifier by using the labels of normal and abnormal data. For example, Erfani *et al.* [21] proposed a supervised support vector machine (SVM) classifier for high-dimensional data to classify normal and abnormal data. Despite the success of supervised DAD methods in anomaly detection, these methods are not as popular as unsupervised or semisupervised methods because of the lack of labeled training data [22]. Furthermore, the supervised DAD method has poor performance for data with class imbalance (the total number of negative class data is much less than the total number of positive class data) [12].

Semisupervised Deep Anomaly Detection: Since normal instances are easier to obtain the labels than that of anomalies, semisupervised DAD techniques are proposed to utilize a single (normally positive class) existing label to separate outliers [11]. For example, Wulsin *et al.* [23] detected abnormalities in the electroencephalogram (EEG) waveforms by using deep belief nets (DBNs) in a semisupervised manner. The semisupervised DBN performance is comparable to the standard classifier on the EEG data set. The semisupervised DAD approach is popular because it can use only a single class of labels to detect anomalies.

Unsupervised Deep Anomaly Detection: Unsupervised DAD techniques use the inherent properties of data instances to detect outliers [11]. For example, Zong *et al.* [24] proposed a

deep automatic coding Gaussian mixture model (DAGMM) for unsupervised anomaly detection. Schlegl *et al.* [25] proposed a deep convolutional generative adversarial network, called AnoGAN, which detects abnormal anatomical images by learning a variety of normal anatomical images. They trained such a model in an unsupervised manner. The unsupervised DAD is widely used since it does not require the characteristics of labeled training data.

B. Communication-Efficient Federated Learning

Google proposed a privacy-preserving distributed machine learning framework, called FL, to train machine learning models without compromising privacy [26]. Inspired by this framework, different edge devices can contribute to the global model training while keeping the training data locally. However, communication overhead is the bottleneck of FL being widely used in IIoT [18]. The previous work has focused on designing efficient stochastic gradient descent (SGD) algorithms and using model compression to reduce the communication overhead of FL. Agarwal *et al.* [27] proposed an efficient cpSGD algorithm to achieve communication-efficient FL. Reisizadeh *et al.* [28] used periodic averaging and quantization methods to design a communication-efficient FL framework. Jeong *et al.* [29] proposed a federated model distillation method to reduce the communication overhead of FL.

However, the above methods do not substantially reduce the number of gradients exchanged between edge devices and the cloud aggregator. The fact is that a large number of gradients exchanged between the edge devices and the cloud aggregator may cause excessive communication overhead for FL [30]. Therefore, in this article, we propose a Top-k selection-based gradient compression scheme to improve the communication efficiency of FL.

III. Preliminary

In this section, we briefly introduce anomalies, federated deep learning, and gradient compression as follows.

A. Anomalies

In statistics, anomalies (also called outliers and abnormalities) are the data points that are significantly different from other observations [11]. We assume that N_1, N_2, and N_3 are regions composed of most observations, so they are regarded as normal data instance regions. If data points O_1 and O_2 are far from these regions, they can be classified as anomalies. To define anomalies more formally, we assume that an n-dimensional data set $\vec{x}_i = (x_{i,1}, \ldots, x_{i,n})$ follows a normal distribution and its mean μ_j and variance σ_j for each dimension where $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$. Specifically, for $j \in \{1, \ldots, n\}$, under the assumption of the normal distribution, we have

$$\mu_j = \sum_{i=1}^{m} x_{i,j}/m,\ \sigma_j^2 = \sum_{i=1}^{m} \left(x_{i,j} - \mu_j\right)^2/m. \tag{1}$$

If there is a new vector $\vec{x}$, the probability $p(\vec{x})$ of anomaly can be calculated as follows:

$$p(\vec{x}) = \prod_{j=1}^{n} p\left(x_j; \mu_j, \sigma_j^2\right) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{\left(x_j - \mu_j\right)^2}{2\sigma_j^2}\right). \tag{2}$$

We can then judge whether vector $\vec{x}$ belongs to an anomaly according to the probability value.

B. Federated Learning

Traditional distributed deep learning techniques require a certain amount of private data to be aggregated and analyzed at central servers (e.g., cloud servers) during the model training phase by using a distributed SGD (D-SGD) algorithm [31]. Such a training process suffers from potential data privacy leakage risks for IIoT devices. To address such privacy challenges, a collaboratively distributed deep learning paradigm, called federated deep learning, was proposed for edge devices to train a global model while keeping the training data sets locally without sharing raw training data [18]. There exist the following phases in the procedure of FL: the initialization phase, the aggregation phase, and the update phase. In the initialization phase, we consider that FL with N edge devices and a parameter aggregator, i.e., a cloud aggregator, distributes a pretrained global model ω_t on the public data sets (e.g., MNIST [32] and CIFAR-10 [33]) to each edge devices. Following that each device uses local data set $\mathcal{D}_k$ of size D_k to train and improve the current global model ω_t in each iteration. In the aggregation phase, the cloud aggregator collects local gradients uploaded by the edge nodes (i.e., edge devices). To do so, the local loss function to be optimized is defined as follows:

$$\min_{x \in \mathbb{R}^d} F_k(x) = \frac{1}{D_k} \sum\nolimits_{i \in D_k} \mathbb{E}_{z_i \sim D_k} f(x; z_i) + \lambda h(x) \tag{3}$$

where $f(\cdot;\cdot)$ is the local loss function for edge device k $\forall \lambda \in [0, 1]$, $h(\cdot)$ is a regularizer function for edge device k, and $\forall i \in [1, \ldots, n]$, z_i is sampled from the local data set $\mathcal{D}_k$ on k device. In the update phase, the cloud aggregator uses the federated averaging (FedAVG) algorithm [26] to obtain a new global model ω_{t+1} for the next iteration, thus we have

$$\omega_{t+1} \leftarrow \omega_t + \frac{1}{n} \sum_{n=1}^{N} F_{t+1}^n \tag{4}$$

where $\sum_{n=1}^{N} F_{t+1}^n$ denotes model updates aggregation and $(1/n) \sum_{n=1}^{N} F_{t+1}^n$ denotes the average aggregation (i.e., the FedAVG algorithm). Both the cloud aggregator and the edge devices repeat the above process till the global model reaches convergence. This paradigm significantly reduces the risks of privacy leakage because of no need to directly access to the raw training data on edge nodes.

C. Gradient Compression

Large-scale FL training requires significant communication bandwidth for gradient exchange, which limits the scalability of multinodes training [34]. In this context, Lin *et al.* [34]

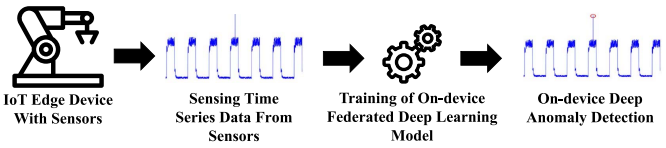


Fig. 1. Workflow of the on-device DAD in IIoT.

stated that 99.9% of the gradient exchange in D-SGD is redundant. To avoid expensive communication bandwidth limiting large-scale distributed training, gradient compression is proposed to greatly reduce communication bandwidth. Researchers generally use gradient quantization [35] and gradient sparsification [30] to achieve gradient compression. Gradient quantization reduces communication bandwidth by quantizing gradients to low-precision values. Gradient sparsification uses threshold quantization to reduce communication bandwidth.

For a fully connected (FC) layer in a deep neural network, we have: $b = f(W*a+v)$, where a is the input, v is the bias, W is the weight, f is the nonlinear mapping, and b is the output. This formula is the most basic operation in a neural network. For each specific neuron i, the above formula can be simplified to the following: $b_i = \text{ReLU}(\sum_{j=0}^{n-1} W_{ij}a_j)$, where ReLU is the activation function. Gradient compression compresses the corresponding weight matrix into a sparse matrix, and hence, the corresponding formula is given as follows:

$$b_i = \text{ReLU}\left(\sum_{j\in X_i\cap Y} \text{Sparse}\left[I_{ij}\right]a_j\right) \tag{5}$$

where $\sum_{j\in X_i\cap Y} S[I_{ij}]$ represents the compressed weight matrix and i, j represent the position information of the gradient in the weight matrix W. Such a method reduces the communication overhead by sparsing the gradient in the weight matrix W.

IV. System Model

We consider the generic setting for on-device DAD in IIoT, where a cloud aggregator and edge devices work collaboratively to train a DAD model by using a given training algorithm (e.g., LSTM) for a specific task (i.e., anomaly detection task), as illustrated in Fig. 1. The edge devices train a shared global model locally on their own local data set (i.e., sensing time-series data from IIoT nodes) and upload their model updates (i.e., gradients) to the cloud aggregator. The cloud aggregator uses the FedAVG algorithm or other aggregation algorithms to aggregate these model updates and obtains a new global model. In the end, the edge devices will receive the new global model sent by the cloud aggregator and use it to achieve accurate and timely anomaly detection.

A. System Model Limitations

The proposed framework focuses on a DAD model learning task involving N distributed edge devices and a cloud aggregator. In this context, this framework has two limitations: 1) *missing labels* and 2) *communication overhead.*

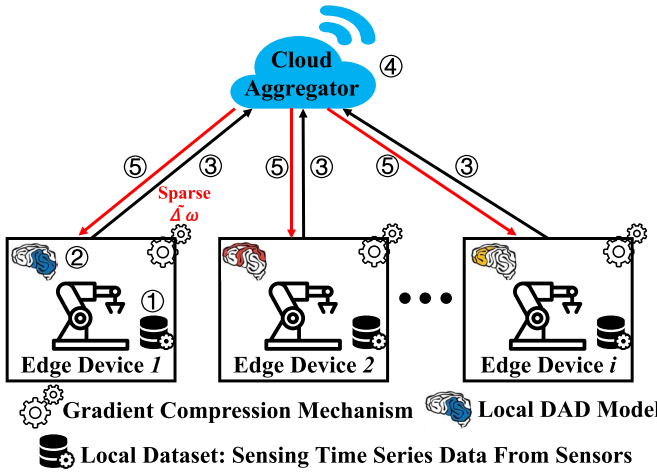


Fig. 2. Overview of the on-device communication-efficient DAD framework in IIoT. This framework's workflow consists of five steps as follows: 1) the edge device uses the sensing time-series data collected from IIoT nodes as a local data set (as shown in ①); 2) the edge device performs local model (i.e., the AMCNN-LSTM model) training on the local data set (as shown in ②); 3) the edge device uploads the sparse gradients $\tilde{\Delta}\omega$ to the cloud aggregator by using a GCM (as shown in ③); 4) the cloud aggregator obtains a new global model by aggregating sparse gradients uploaded by the edge device (as shown in ④); and 5) the cloud aggregator sends the new global model to each edge device. The above steps are executed cyclically until the global model reaches optimal convergence (as shown in ⑤). Decentralized devices can use this optimal global model to perform anomaly detection tasks.

For missing-label limitation, we assume that the labels of the training sample with proportion p $(0 < p < 1)$ are missing. The lack of the label of the sample will cause the problem of class imbalance, thereby reducing the accuracy of the DAD model. For communication-overhead limitation, we consider that there exists an excessive communication overhead when a large number of gradients exchanged between edge devices and the cloud aggregator, which may make the model fail to converge [29].

The above restrictions hinder the deployment of the DAD model in edge devices, which motivates us to develop a communication-efficient FL-based unsupervised DAD framework to achieve accurate and timely anomaly detection.

B. Proposed Framework

We consider an on-device communication-efficient DAD framework that involves multiple edge devices for collaborative model training in IIoT, as illustrated in Fig. 2. In particular, this framework consists of a cloud aggregator and edge devices. Furthermore, the proposed framework also includes two mechanisms: 1) an anomaly detection mechanism and 2) a GCM. More details are described as follows.

1) *Cloud Aggregator:* The cloud aggregator is generally a cloud server with strong computing power and rich computing resources. The cloud aggregator contains two functions: a) it initializes the global model and sends the global model to all edge devices and b) it aggregates the gradients uploaded by edge devices until the model converges.
2) *Edge Devices:* Edge devices are generally agents and clients, such as whirlpool, wind turbine, and vehicle,

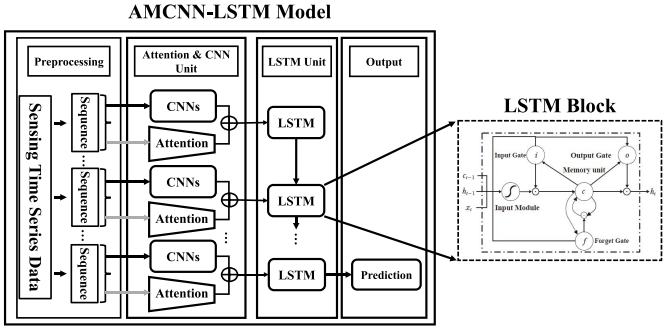


Fig. 3. Overview of the AMCNN-LSTM model.

which contain local models and functional mechanisms (see below for more details). Each edge device uses the local data set (i.e., sensing time-series data from IIoT nodes) to train the global model sent by the cloud aggregator and uploads the gradients to the cloud aggregator until the global model converges. The local model is deployed in the edge device and it can perform anomaly detection. In this article, we use the AMCNN-LSTM model to detect anomalies, which uses CNN to capture fine-grained features of sensing time-series data and uses an LSTM module to accurately and timely detect anomalies.

The functions of mechanisms are described as follows.

1) *DAD Mechanism:* The DAD mechanism is deployed in the edge devices, which can detect anomalies to reduce economic losses.
2) *GCM:* The GCM is deployed in the edge devices, which can compress the local gradients to reduce the number of gradients exchanged between the edge devices and the cloud aggregator, thereby reducing communication overhead.

C. Design Goals

In this article, our goal is to develop an on-device communication-efficient FL framework for DAD in IIoT. First, the proposed framework needs to detect anomalies accurately in an unsupervised manner. The proposed framework uses an unsupervised AMCNN-LSTM model to detect anomalies. Second, the proposed framework can significantly improve communication efficiency by using a GCM. Third, the performance of the proposed framework is comparable to traditional FL frameworks.

V. Communication-Efficient On-device Deep Anomaly Detection Framework

In this section, we first present the AMCNN-LSTM model. This model uses CNN to capture the fine-grained features of sensing time-series data and uses an LSTM module to accurately and timely detect anomalies. We then propose a deep GCM to further improve the communication efficiency of the proposed framework.

A. Attention Mechanism-Based CNN-LSTM Model

We present an unsupervised AMCNN-LSTM model, including an input layer, an attention mechanism-based CNN unit, an LSTM unit, and an output layer shown in Fig. 3. First, we use the preprocessed data as an input to the input layer. Second, we use CNN to capture the fine-grained features of the input and utilize the attention mechanism to focus on the important features of CNN captured features. Third, we use the output of the attention mechanism-based CNN unit as the input of the LSTM unit and use LSTM to predict future time-series data. Finally, we propose an anomaly detection score to detect anomalies.

Preprocessing: We normalize the sensing time-series data collected by IIoT nodes into [0, 1] to accelerate the model convergence.

Attention Mechanism-Based CNN Unit: First, we introduce an attention mechanism in the CNN unit to improve the focus on important features. Note that due to the bottleneck of information processing, people will selectively focus on important parts of the information and ignore other visible information [36]. Inspired by the above facts, the attention mechanism was proposed to improve the ability of feature extraction in various tasks, such as computer vision and natural language processing [36]–[38]. Therefore, the attention mechanism can improve the performance of the model by paying attention to important features. The formal definition of the attention mechanism is given as follows:

$$\begin{aligned} e_i &= a(\mathbf{u}, \mathbf{v}_i) \text{ (Compute Attention Scores)} \\ \alpha_i &= \frac{e_i}{\sum_i e_i} \text{ (Normalize)} \\ c &= \sum_i \alpha_i \mathbf{v}_i \text{ (Encode)} \end{aligned} \tag{6}$$

where $\mathbf{u}$ is the matching feature vector based on the current task and is used to interact with the context, $\mathbf{v}_i$ is the feature vector of a timestamp in the time series, e_i is the unnormalized attention score, α_i is the normalized attention score, and c is the context feature of the current timestamp calculated based on the attention score and feature sequence $\mathbf{v}$. In most instances, $e_i = \mathbf{u}^T W \mathbf{v}$, where W is the weight matrix.

Second, we use the CNN unit to extract fine-grained features of time-series data. The CNN module is formed by stacking multiple layers of 1-D CNN, and each layer includes a convolution layer, a batch normalization layer, and a nonlinear layer. Such modules implement sampling aggregation by using pooling layers and create hierarchical structures that gradually extract more abstract features through the stacking of convolutional layers. This module outputs m feature sequences of length n, and the size can be expressed as $(n \times m)$. To further extract significant time-series data features, we propose a parallel feature extraction branch by combining the attention mechanisms and CNN. The attention mechanism module is composed of feature aggregation and scale restoration. The feature aggregation part uses the stacking of multiple convolutions and pooling layers to extract key features from the sequence and uses a convolution kernel of size 1×1 to mine the linear relationship. The scale restoration part restores the

key features to $(n \times m)$, which is consistent with the size of output features of the CNN module, and then uses the sigmoid function to constrain the values to [0, 1].

Third, we multiply elementwise output features of the CNN module and the output of the important features by the corresponding attention mechanism module. We assume that the sequence $X^i = \{x_1^i, x_2^i, \ldots, x_n^i\}(0 \le i < I)$. The output of the sequence X^i processed by the CNN module is represented by W_{CNN}, and the output of the corresponding attention module is represented as $W_{\text{attention}}$. We multiply the two outputs element by element as follows:

$$W(i, c) = W_{\text{CNN}}(i, c) \odot W_{\text{attention}}(i, c) \tag{7}$$

where $\odot$ represents elementwise multiplication, i is the corresponding position of the time series in the feature layer, and c is the channel. We use the final feature layer $W(i, c)$ as the input of LSTM block.

We introduce the attention mechanism to expand the receptive field of the input, which allows the model to obtain more comprehensive contextual information, thereby learning the important features of the current local sequence. Furthermore, we use the attention module to suppress the interference of unimportant features to the model, thereby solving the problem that the model cannot distinguish the importance of the time-series data features.

LSTM Unit: In this article, we use a variant of a recurrent neural network, called LSTM, to support accurately predict the sensing time-series data to detect anomalies, as shown in Fig. 3. LSTM uses a well-designed "gate" structure to delete information or add information to the state of the cell. The "gate" structure is a method of selectively filtering information. LSTM cells include forget gates f_t, input gates i_t, and output gates o_t. The calculations on the three gate structures are defined as follows:

$$\begin{aligned} f_t &= \sigma_l\big(W_f \cdot \big[h_{t-1}, x_t\big] + b_f\big) \\ i_t &= \sigma_l\big(W_i \cdot \big[h_{t-1}, x_t\big] + b_i\big) \\ \tilde{C}_t &= \tanh\big(W_C \cdot \big[h_{t-1}, x_t\big] + b_C\big) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t &= \sigma_l\big(W_o \cdot \big[h_{t-1}, x_t\big] + b_o\big) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{8}$$

where W_f, W_i, W_C, W_o, and b_f, b_i, b_C, b_o are the weight matrices and the bias vectors for input vector x_t at time step t, respectively. σ_l is the activation function, $*$ represents elementwise multiplication of a matrix, C_t represents the cell state, h_{t-1} is the state of the hidden layer at time step $t-1$, and h_t is the state of the hidden layer at time step t.

Anomaly Detection: We use the AMCNN-LSTM model to predict real-time and future sensing time-series data in different edge devices

$$\big[x_{n-T+1}^i, x_{n-T+2}^i, \ldots, x_n^i\big] \overset{f(\cdot)}{\rightarrow} \big[x_{n+1}^i, x_{n+2}^i, \ldots, x_{n+T}^i\big] \tag{9}$$

where $f(\cdot)$ is the prediction function. In this article, we use the LSTM unit for time-series prediction. We use anomaly scores for anomaly detection, which is defined as follows:

$$A_n = (\beta_n - \mu)^T \sigma^{-1} (\beta_n - \mu) \tag{10}$$

where A_n is the anomaly score, $\beta_n = |x_n^i - x_n^{'i}|$ is the reconstruction error vector, and the error vectors β_n for the time series in sequences X^i are used to estimate the parameters μ and σ of a normal distribution $\mathcal{N}(\mu; \sigma)$ using maximum-likelihood estimation.

In an unsupervised setting, when $A_n \ge \varsigma$ ($\varsigma = \max F_\theta = ([(1+\theta^2) \times P \times R]/[\theta^2 P + R])$), where P is precision, R is recall, and θ is the parameter, a point in a sequence can be predicted to be "anomalous," otherwise, "normal."

B. Gradient Compression Mechanism

If the gradients reach 99.9% sparsity, only 0.1% gradients with the largest absolute value are useful for model aggregation [30]. Therefore, we only need to aggregate the gradient with a larger absolute value to update the model. This way reduces the byte size of the gradient matrix, which can reduce the number of gradients exchanged between the device and the cloud to improve communication efficiency, especially for distributed machine learning systems. Inspired by the above facts, we propose a GCM to reduce the gradients exchanged between the cloud aggregator and edge devices. We expect that this mechanism can further improve the communication efficiency of the proposed framework.

When we choose a gradient with a larger absolute value, we will meet the following situations: 1) all gradient values in the gradient matrix are not greater than the given threshold and 2) there are some gradient values in the gradient matrix that are very close to the given threshold. If we set these gradients that do not meet the threshold requirements to 0, it will cause information loss. Therefore, the device uses a local gradient accumulation scheme to prevent information loss. Specifically, the cloud returns smaller gradients to the device instead of filtering the gradients. The device keeps the smaller gradient in the buffer and accumulates all the smaller gradients until it reaches a given threshold. Note that we use D-SGD for iterative updates, and the loss function to be optimized is defined as follows:

$$F(\omega) = \frac{1}{D_k} \sum_{x \in D_k} f(x, \omega) \tag{11}$$

$$\omega_{t+1} = \omega_t - \eta \frac{1}{Nb} \sum_{k=1}^{N} \sum_{x \in \mathcal{B}_{k,t}} \nabla f(x, \omega_t) \tag{12}$$

where $F(\omega)$ is the loss function, $f(x, \omega)$ is the loss function for the local device, ω are the weights of the model, N is the total edge devices, η is the learning rate, $\mathcal{B}_{k,t}$ represents the data sample for the tth round of training, and each local data set size of b.

When the gradients' sparsification reaches a high value (e.g., 99%), it will affect the model convergence. By following [30], we use momentum correction and local gradient clipping to mitigate this effect. Momentum correction can make the accumulated small local gradients converge toward the gradients with a larger absolute value, thereby accelerating the model's convergence speed. Local gradient clipping is used to alleviate the problem of gradient explosions [30]. Next, we prove that the local gradient accumulation scheme will not affect

the model convergence. We assume that $g^{(i)}$ is the ith gradient, $u^{(i)}$ denotes the sum of the gradients using the aggregation algorithm in [26], $v^{(i)}$ denotes the sum of the gradients using the local gradient accumulation scheme, and m is the rate of gradient descent. If the ith gradient does not exceed threshold until the $(t-1)$th iteration and triggers the model update, we have

$$u_{t-1}^{(i)} = m^{t-2} g_1^{(i)} + \cdots + m g_{t-2}^{(i)} + g_{t-1}^{(i)} \tag{13}$$

$$v_{t-1}^{(i)} = \left(1 + \cdots + m^{t-2}\right) g_1^{(i)} + \cdots + (1+m) g_{t-2}^{(i)} + g_{t-1}^{(i)} \tag{14}$$

then we can update $\omega_t^{(i)} = w_1^{(i)} - \eta \times v_{t-1}^{(i)}$ and set $v_{t-1}^{(i)} = 0$. If the ith gradient reaches the threshold at the tth iteration, model update is triggered, thus we have

$$u_t^{(i)} = m^{t-1} g_1^{(i)} + \cdots + m g_{t-1}^{(i)} + g_t^{(i)} \tag{15}$$

$$v_t^{(i)} = m^{t-1} g_1^{(i)} + \cdots + m g_{t-1}^{(i)} + g_t^{(i)}. \tag{16}$$

Then, we can update $\omega_{t+1}^{(i)} = \omega_t^{(i)} - \eta \times v_t^{(i)} = \omega_1{}^{(i)} - \eta \times [(1 + \cdots + m^{t-1}) g_1^{(i)} + \cdots + (1+m) g_{t-1}^{(i)} + g_t^{(i)}] = w_1^{(i)} - \eta \times v_{t-1}^{(i)}$, so the result of using the local gradient accumulation scheme is consistent with the usage effect of the optimization algorithm in [26].

The specific implementation phases of the GCM are given as follows [39].

1) *Phase 1, Local Training:* Edge devices use the local data set to train the local model. In particular, we use the gradient accumulation scheme to accumulate local small gradients.
2) *Phase 2, Gradient Compression:* Each edge device uses Algorithm 1 to compress the gradients and upload sparse gradients (i.e., only gradients larger than a threshold are transmitted.) to the cloud aggregator. Note that the edge devices send the remaining local gradient to the cloud aggregator when the local gradient accumulation is greater than a threshold.
3) *Phase 3, Gradient Aggregation:* The cloud aggregator obtains the global model by aggregating sparse gradients and sends this global model to the edge devices.

The gradient compression algorithm is thus presented in Algorithm 1 [39].

Algorithm 1: GCM on Edge Node k

Input: $\mathcal{G} = \{g^1, g^2, \ldots, g^k\}$ is the edge node's gradient, B is the local mini-batch size, $\mathcal{D}_k$ is the local dataset, η is the learning rate, $f(\cdot,\cdot)$ is the edge node's loss function, and the optimization function SGD.

Output: Parameter ω.

Initialize parameter ω_t;
$g^k \leftarrow 0$;
for $t = 0, 1, \cdots$ **do**
  $g_t^k \leftarrow g_{t-1}^k$;
  for $i = 1, 2, \cdots$ **do**
    Sample data x from $\mathcal{D}_k$;
    $g_t^k \leftarrow g_{t-1}^k + \frac{1}{|\mathcal{D}_k|B} \nabla f(x; \omega_t)$;
if *Gradient Clipping* **then**
  $g_t^k \leftarrow$ Local_Gradient_Clipping (g_t^k);
foreach $g_t^{k_j} \in \{g_t^k\}$ *and* $j = 1, 2, \cdots$ **do**
  Thr $\leftarrow |\text{Top } \rho\% \text{ of } \{g_t^k\}|$;
  if $|g_t^{k_j}| \geq$ Thr **then**
    Send this gradient to the cloud aggregator;
  if $|g_t^{k_j}| <$ Thr **then**
    The edge node k uses the local gradient accumulation scheme to accumulate gradients until the gradient reaches Thr;
  Aggregate $g_t^k : g_t \leftarrow \sum_{k=1}^{N} (\text{sparse } \tilde{g}_t^k)$;
  $\omega_{t+1} \leftarrow \text{SGD}(\omega_t, g_t)$.
return ω.

VI. Experiments

In this section, the proposed framework is applied to four real-world data sets, i.e., power demand,[1] space shuttle,[2] ECG,[3] and engine[4] for performance demonstration. These data sets are time-series data sets collected by different types of sensors from different fields [6]. For example, the power demand data set is composed of electricity consumption data recorded by the electricity meter. There are normal subsequences and anomalous subsequences in these data sets. As shown in Table I, X, X_n, and X_a are the number of original sequences, normal subsequences, and anomalous subsequences, respectively. For the power demand data set, the anomalous subsequences indicate that the electricity meter has failed or stop working. Therefore, we need to use these data sets to train an FL model that can detect anomalies. We divide all data sets into a training set and a test set in a 7:3 ratio. We implement the proposed framework by using Pytorch and PySyft [40]. The experiment is conducted on a virtual workstation with the Ubuntu 18.04 operation system, Intel Core i5-4210M CPU, 16-GB RAM, 512-GB SSD.

TABLE I
Details of Four Real-World Data Sets

Datasets	Dimensions	X	X_n	X_a
Power Demand	1	1	45	6
Space Shuttle	1	3	20	8
ECG	1	1	215	1
Engine	12	30	240	152

A. Evaluation Setup

In this experiment, to determine the hyperparameter ρ of GCM, we first apply a simple CNN network (i.e., CNN with two convolutional layers followed by one FC layer) in the proposed framework to perform the classification task on

[1] https://archive.ics.uci.edu/ml/datasets/
[2] https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)
[3] https://physionet.org/about/database/
[4] https://archive.ics.uci.edu/ml/datasets.php

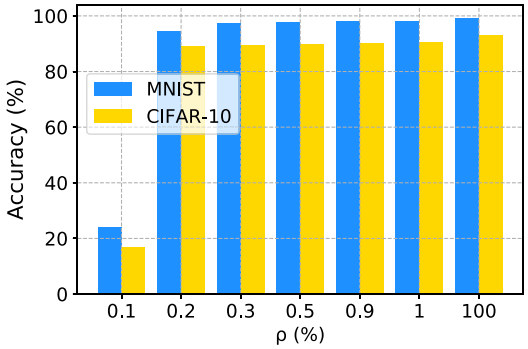


Fig. 4. Accuracy of the proposed framework with different ρ on MNIST and CIFAR-10 data sets.

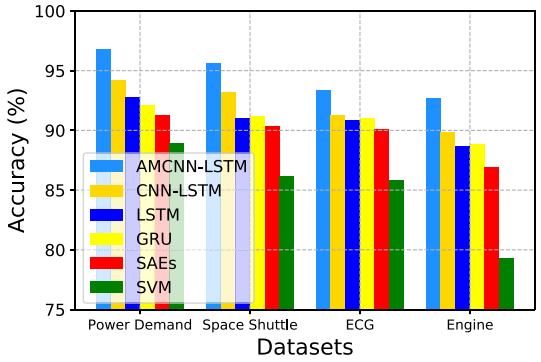


Fig. 5. Performance comparison of detection accuracy for AMCNN-LSTM, CNN-LSTM, LSTM, GRU, SAEs, and SVM on different data sets: power demand, space shuttle, ECG, and engine.

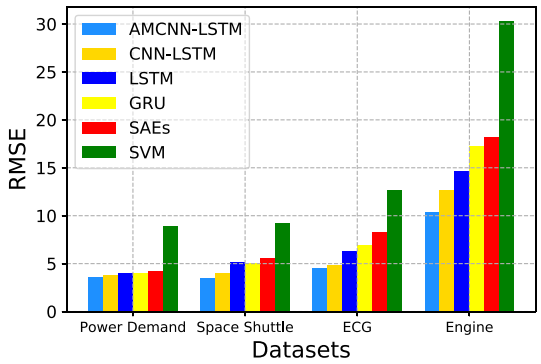


Fig. 6. Performance comparison of RMSE for AMCNN-LSTM, CNN-LSTM, LSTM, GRU, SAEs, and SVM on different data sets: power demand, space shuttle, ECG, and engine.

MNIST and CIFAR-10 data set. The pixels in all data sets are normalized into [0,1]. During the simulation, the number of edge devices is $N = 10$, the learning rate is $\eta = 0.001$, the training epoch is $E = 1000$, and the mini-batch size is $B = 128$, and we follow reference [5] and set θ as 0.05.

We adopt root mean square error (RMSE) to indicate the performance of the AMCNN-LSTM model as follows:

$$\text{RMSE} = \left[\frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_p|)^2\right]^{\frac{1}{2}} \tag{17}$$

where y_i is the observed sensing time-series data, and $\hat{y}_p$ is the predicted sensing time-series data.

B. Hyperparameters Selection of the Proposed Framework

In the context of the deep gradient compression scheme, proper hyperparameter selection, i.e., a threshold of absolute gradient value, is a notable factor that determines the proposed framework performance. In this section, we investigate the performance of the proposed framework with different thresholds and try to find a best-performing threshold for it. In particular, we employ $\rho \in \{0.1, 0.2, 0.3, 0.5, 0.9, 1, 100\}$ to adjust the best threshold of the proposed framework. We use MNIST and CIFAR-10 data sets to evaluate the performance of the proposed framework with the selected threshold [39]. As shown in Fig. 4, we observe that the larger ρ, the better the performance of the proposed framework. For the MNIST task, the results show that when $\rho = 0.3$, the accuracy is 97.25%; and when $\rho = 100$, the accuracy is 99.08%. This means that the model increases the gradient size by about 300 times, but the accuracy is only improved by 1.83%. Furthermore, we observe a tradeoff between the gradient threshold and accuracy. Therefore, to achieve a good tradeoff between the gradient threshold and accuracy, we choose $\rho = 0.3$ as the best threshold of our scheme.

C. Performance of the Proposed Framework

We compared the performance of the proposed model with that of CNN-LSTM [41], LSTM [5], gate recurrent unit (GRU) [42], stacked autoencoders (SAEs) [43], and SVM [21] method with an identical simulation configuration. Among these competing methods, AMCNN-LSTM is an FL-based model, and the rest of the methods are centralized ones. All models are popular DAD models for general anomaly detection applications. We evaluate these models on four real-world data sets, i.e., power demand, space shuttle, ECG, and engine.

First, we compare the accuracy of the proposed model with competing methods in anomaly detection. We determine $\max F_\theta$ and hyperparameter ς based on the accuracy and recall of the model on the training set. The hyperparameters ς of the data set power demand, space shuttle, ECG, and engine are 0.75, 0.80, 0.80, and 0.60. In Fig. 5, experimental results show that the proposed model achieves the highest accuracy on all four data sets. For example, for the data set power demand, the accuracy of the AMCNN-LSTM model is 96.85%, which is 7.87% higher than that of the SVM model. From the experimental results, AMCNN-LSTM has better robustness to different data sets. The reason is that we use the on-device FL framework to train and update the model, which can learn the time-series features from different edge devices as much as possible, thereby improving the robustness of the model. Furthermore, the FL framework provides opportunities for edge devices to update models in a timely manner. This helps the edge device owner to update the model on the edge devices in time.

Second, we need to evaluate the prediction error of the proposed model and competing methods. As shown in Fig. 6, experimental results show that the proposed model achieves

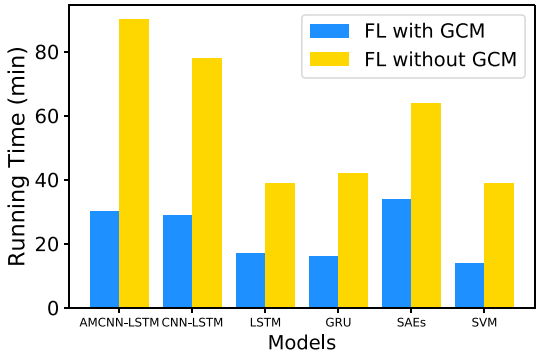


Fig. 7. Comparison of communication efficiency between FL with GCM and FL without GCM with different models.

the best performance on four real-world data sets. For the ECG data set, RMSE of the AMCNN-LSTM model is 63.9% lower than that of the SVM model. The reason is that the AMCNN-LSTM model uses AMCNN units to capture important fine-grained features and prevent memory loss and gradient dispersion problems. Memory loss and gradient dispersion problems often occur in encoder–decoder models, such as LSTM and GRU models. Furthermore, the proposed model retains the advantages of the LSTM unit in predicting time-series data. Therefore, the proposed model can accurately predict time-series data.

Therefore, the proposed model not only accurately detects abnormalities but also accurately predicts time-series data.

D. Communication Efficiency of the Proposed Framework

In this section, we compare communication efficiency between the FL framework with GCM and the traditional FL framework without GCM. We apply the same model (i.e., AMCNN-LSTM, CNN-LSTM, LSTM, GRU, SAEs, and SVM) in the proposed framework and the traditional FL framework. Note that we fix the communication overhead of each round, so we can compare the running time of the model to compare communication efficiency. In Fig. 7, we show the running time of FL with GCM and FL without GCM using different models. As shown in Fig. 7, we observe that the running time of the FL framework with GCM is about 50% that of the framework without GCM. The reason is that GCM can reduce the number of gradients exchanged between the edge devices and the cloud aggregator. In Section V-B, we show that GCM can compress the gradient by 300 times without compromising accuracy. Therefore, the proposed communication efficient framework is practical and effective in real-world applications.

E. Discussion

Due to the tradeoff between privacy and model performance, we will discuss the privacy analysis of the proposed framework in terms of data access and model performance.

1) *Data Access:* An FL framework allows edge devices to keep the data set locally and collaboratively learn deep learning models, which means that any third party cannot access the user's raw data. Therefore, the FL-based model can achieve anomaly detection without compromising privacy.
2) *Model Performance:* Although the FL-based model can protect privacy, the model performance is still an important metric to measure the quality of the model. It can be seen from the experimental results that the performance of the proposed model is comparable to many advanced centralized machine learning models, such as CNN-LSTM, LSTM, GRU, and SVM model. In other words, the proposed model makes a good compromise between privacy and model performance.

VII. Conclusion

In this article, we propose a novel communication-efficient on-device FL-based DAD framework for sensing time-series data in IIoT. First, we introduce an FL framework to enable decentralized edge devices to collaboratively train an anomaly detection model, which can solve the problem of data islands. Second, we propose an AMCNN-LSTM model to accurately detect anomalies. An AMCNN-LSTM model uses attention mechanism-based CNN units to capture important fine-grained features and prevent memory loss and gradient dispersion problems. Furthermore, this model retains the advantages of the LSTM unit in predicting time-series data. We evaluate the performance of the proposed model on four real-world data sets and compare it with CNN-LSTM, LSTM, GRU, SAEs, and SVM methods. The experimental results show that the AMCNN-LSTM model can achieve the highest accuracy on all four data sets. Third, we propose a GCM based on Top-k selection to improve communication efficiency. The experimental results validate that this mechanism can compress the gradient by 300 times without losing accuracy. To the best of our knowledge, this is one of the pioneering works for DAD by using on-device FL.

In the future, we will focus on researching privacy-enhanced FL frameworks and more robust anomaly detection models. The reason is that the FL framework is vulnerable to malicious attacks by malicious participants and a more robust model can be applied to a wider range of application scenarios.

References

[1] H. Peng and X. Shen, "Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks," *IEEE Trans. Netw. Sci. Eng.*, early access, Mar. 6, 2020, doi: 10.1109/TNSE.2020.2978856.
[2] Y. Wu *et al.*, "Dominant data set selection algorithms for electricity consumption time-series data analysis based on affine transformation," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4347–4360, May 2020.
[3] Y. Peng, A. Tan, J. Wu, and Y. Bi, "Hierarchical edge computing: A novel multi-source multi-dimensional data anomaly detection scheme for industrial Internet of Things," *IEEE Access*, vol. 7, pp. 111257–111270, 2019.
[4] H. Peng, S. Si, M. K. Awad, N. Zhang, H. Zhao, and X. S. Shen, "Toward energy-efficient and robust large-scale WSNs: A scale-free network approach," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4035–4047, Dec. 2016.
[5] P. Malhotra *et al.*, "LSTM-based encoder–decoder for multi-sensor anomaly detection," 2016. [Online]. Available: arXiv:1607.00148.
[6] T. Luo and S. G. Nagarajan, "Distributed anomaly detection using autoencoder neural networks in WSN for IoT," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.

[7] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 3, pp. 924–935, Sep. 2019.
[8] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 756–767.
[9] S. Garg, K. Kaur, S. Batra, G. Kaddoum, N. Kumar, and A. Boukerche, "A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications," *Future Gener. Comput. Syst.*, vol. 104, pp. 105–118, Mar. 2020.
[10] S. Garg *et al.*, "EN-ABC: An ensemble artificial bee colony based anomaly detection scheme for cloud environment," *J. Parallel Distrib. Comput.*, vol. 135, pp. 219–233, Jan. 2020.
[11] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019. [Online]. Available: arXiv:1901.03407.
[12] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. ESANN*, vol. 89, 2015, pp. 89–94.
[13] W. Lu, X. Zhang, H. Lu, and F. Li, "Deep hierarchical encoding model for sentence semantic matching," *J. Vis. Commun. Image Represent.*, vol. 71, Aug. 2020, Art. no. 102794.
[14] H. Lu, M. Zhang, X. Xu, Y. Li, and H. T. Shen, "Deep fuzzy hashing network for efficient image retrieval," *IEEE Trans. Fuzzy Syst.*, early access, Apr. 3, 2020, doi: 10.1109/TFUZZ.2020.2984991.
[15] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2018.
[16] E. Lundin and E. Jonsson, "Anomaly based intrusion detection: Privacy concerns and other problems," *Comput. Netw.*, vol. 34, no. 4, pp. 623–640, 2000.
[17] I. Butun, B. Kantarci, and M. Erol-Kantarci, "Anomaly detection and privacy preservation in cloud-centric Internet of Things," in *Proc. IEEE Int. Conf. Commun. Workshop*, 2015, pp. 2610–2615.
[18] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet Things J.*, early access, Apr. 30, 2020, doi: 10.1109/JIOT.2020.2991401.
[19] R. Ito *et al.*, "An on-device federated learning approach for cooperative anomaly detection," 2020. [Online]. Available: arXiv:2002.12301.
[20] M. Tsukada *et al.*, "A neural network based on-device learning anomaly detector for edge devices," 2019. [Online]. Available: arXiv:1907.10147.
[21] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, Oct. 2016.
[22] T. S. Buda, B. Caglayan, and H. Assem, "DeepAD: A generic framework based on deep learning for time series anomaly detection," in *Proc. Pac.–Asia Conf. Knowl. Disc. Data Min.*, 2018, pp. 577–588.
[23] D. Wulsin *et al.*, "Modeling electroencephalography waveforms with semi-supervised deep belief nets: Fast classification and anomaly measurement," *J. Neural Eng.*, vol. 8, no. 3, 2011, Art. no. 036015.
[24] B. Zong *et al.*, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. ICLR*, 2018, pp. 1–6.
[25] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 146–157.
[26] J. Konečnỳ *et al.*, "Federated learning: Strategies for improving communication efficiency," 2016. [Online]. Available: arXiv:1610.05492.
[27] N. Agarwal, A. T. Suresh, F. X. Yu, S. Kumar, and B. McMahan, "CPSGD: Communication-efficient and differentially private distributed SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7564–7575.
[28] A. Reisizadeh *et al.*, "FEDPAQ: A communication-efficient federated learning method with periodic averaging and quantization," 2019. [Online]. Available: arXiv:1909.13014.
[29] E. Jeong *et al.*, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data," 2018. [Online]. Available: arXiv:1811.11479.
[30] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1299–1309.
[31] L. Zhao *et al.*, "Shielding collaborative learning: Mitigating poisoning attacks through client-side detection," 2019. [Online]. Available: arXiv:1910.13111.
[32] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[33] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.
[34] Y. Lin *et al.*, "Deep gradient compression: Reducing the communication bandwidth for distributed training," 2017. [Online]. Available: arXiv:1712.01887.
[35] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1709–1720.
[36] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014. [Online]. Available: arXiv:1409.0473.
[37] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.
[38] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[39] J. Kang *et al.*, "Scalable and communication-efficient decentralized federated edge learning with multi-blockchain framework," in *Proc. Int. Conf. Blockchain Trustworthy Syst. (BlockSys)*, Aug. 2020.
[40] T. Ryffel *et al.*, "A generic framework for privacy preserving deep learning," 2018. [Online]. Available: arXiv:1811.04017.
[41] T.-Y. Kim and S.-B. Cho, "Web traffic anomaly detection using C-LSTM neural networks," *Expert Syst. Appl.*, vol. 106, pp. 66–76, Sep. 2018.
[42] Y. Guo, W. Liao, Q. Wang, L. Yu, T. Ji, and P. Li, "Multidimensional time series anomaly detection: A GRU-based Gaussian mixture variational autoencoder approach," in *Proc. Asian Conf. Mach. Learn.*, 2018, pp. 97–112.
[43] N. Chouhan, A. Khan, and H.-U.-R. Khan, "Network anomaly detection using channel boosted and residual learning based deep convolutional neural network," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105612.

Yi Liu (Student Member, IEEE) received the B.Eng. degree in network engineering from Heilongjiang University, Harbin, China, in 2019. He is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Monash University, Melbourne, VIC, Australia.

His research interests include security and privacy, federated learning, edge computing, and blockchain.

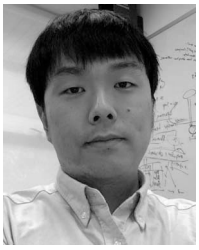
Sahil Garg (Member, IEEE) received the Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, India, in 2018.

He is a Postdoctoral Research Fellow with the École de technologie supérieure, Université du Québec, Montreal, QC, Canada. He has over 50 publications in high ranked journals and conferences, including 25+ IEEE transactions/journal papers. He has many research contributions in the area of machine learning, big data analytics, security and privacy, the Internet of Things, and cloud computing.

Dr. Garg received the IEEE ICC Best Paper Award in 2018 in Kansas City, Missouri. He serves as the Managing Editor of Springers Human-Centric Computing and Information Sciences journal. He is also an Associate Editor of IEEE NETWORK, IEEE SYSTEM JOURNAL, *Applied Soft Computing* (Elsevier), *Future Generation Computer Systems*, and the *International Journal of Communication Systems* (Wiley). In addition, he also serves as a Workshops and Symposia Officer of the IEEE ComSoc Emerging Technology Initiative on Aerial Communications. He has guest edited a number of Special Issues in top-cited journals, including the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *Future Generation Computer Systems*. He serves/served as the Workshop Chair/Publicity Co-Chair for several IEEE/ACM conferences, including IEEE INFOCOM, IEEE GLOBECOM, IEEE ICC, and ACM MobiCom. He is a member of ACM.

Jiangtian Nie (Graduate Student Member, IEEE) received the B.Eng. degree (Hons.) in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2016. She is currently pursuing the Ph.D. degree with ERI@N in the Interdisciplinary Graduate School, Nanyang Technological University, Singapore.

Her research interests include incentive mechanism design in crowdsensing and game theory.



Yang Zhang (Member, IEEE) received the B.Eng. and M.Eng. degrees from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2008 and 2011, respectively, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2015.

He is currently an Associate Professor with the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China. His current research interests include: market-oriented modeling for network resource allocation, multiple agent machine learning, and deep reinforcement learning in network systems.

Dr. Zhang is an Associate Editor of the *EURASIP Journal on Wireless Communications and Networking*, and a technical committee member of *Computer Communications* (Elsevier). He is also an Advisory Expert Member of Shenzhen FinTech Laboratory, China.

Zehui Xiong (Member, IEEE) received the B.Eng. degree (Highest Hons.) in telecommunications engineering from Huazhong University of Science and Technology, Wuhan, China, in July 2016, and the Ph.D. degree in computer science and engineering from the Nanyang Technological University, Singapore, in April 2020.

He is currently a Researcher with Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University. He was a Visiting Scholar with Princeton University, Princeton, NJ, USA, and the University of Waterloo, Waterloo, ON, Canada. His research interests include resource allocation in wireless communications, network games and economics, blockchain, and edge intelligence.

Dr. Xiong is a recipient of the Chinese Government Award for Outstanding Students Abroad in 2019, and NTU SCSE Outstanding PhD Thesis Runner-Up Award in 2020. He has won several Best Paper Awards, including IEEE WCNC 2020 and the IEEE Vehicular Technology Society Singapore Best Paper Award in 2019. He is an Editor of *Computer Networks* (Elsevier) and *Physical Communication* (Elsevier), and an Associate Editor of *IET Communications*. He serves as a Guest Editor for the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, the IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY, and the *EURASIP Journal on Wireless Communications and Networking*.

Jiawen Kang received the M.S. and Ph.D. degrees from Guangdong University of Technology, Guangzhou, China, in 2015 and 2018, respectively.

He is currently a Postdoctoral Fellow with Nanyang Technological University, Singapore. His research interests mainly focus on blockchain and security and privacy protection in wireless communications and networking.

M. Shamim Hossain (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Ottawa, Ottawa, ON, Canada, in 2009.

He is a Professor with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He is also an Adjunct Professor with the School of Electrical Engineering and Computer Science, University of Ottawa. He has authored and coauthored more than 250 publications, including refereed journals, conference papers, books, and book chapters. Recently, he co-edited a book on *Connected Health in Smart Cities* (Springer). His research interests include cloud networking, smart environment (smart city, smart health), AI, deep learning, edge computing, Internet of Things, multimedia for health care, and multimedia big data.

Prof. Hossain is a recipient of a number of awards, including the Best Conference Paper Award and the 2016 *ACM Transactions on Multimedia Computing, Communications and Applications* Nicolas D. Georganas Best Paper Award. He has served as the co-chair, the general chair, the workshop chair, the publication chair, and a TPC for over 12 IEEE and ACM conferences and workshops. He is currently the Co-Chair of the 3rd IEEE ICME workshop on Multimedia Services and Tools for smart-health (MUST-SH 2020). He is on the editorial board of the IEEE TRANSACTIONS ON MULTIMEDIA, IEEE MULTIMEDIA, IEEE NETWORK, IEEE WIRELESS COMMUNICATIONS, IEEE ACCESS, the *Journal of Network and Computer Applications* (Elsevier), and the *International Journal of Multimedia Tools and Applications* (Springer). He also currently serves as a Lead Guest Editor for *Multimedia Systems*. He serves/served as a Guest Editor for *IEEE Communications Magazine*, IEEE NETWORK, the *ACM Transactions on Internet Technology*, the *ACM Transactions on Multimedia Computing, Communications, and Applications*, the IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE (currently, JBHI), the IEEE TRANSACTIONS ON CLOUD COMPUTING, MULTIMEDIA SYSTEMS, the *International Journal of Multimedia Tools and Applications* (Springer), *Cluster Computing* (Springer), and *Future Generation Computer Systems* (Elsevier). He is a Senior Member of ACM.