TMANomaly: Time-Series Mutual Adversarial Networks for Industrial Anomaly Detection

Lianming Zhang, Wenji Bai, Xiaowei Xie, Liying Chen, and Pingping Dong

***Abstract*—Large-scale sewage treatment plants are one of the typical Industrial Internet of Things systems, where the presence of a large number of sensors generates massive dynamic time series data, and such multivariate time series data are usually time-dependent and random. Therefore, there is a certain risk when fitting the potential anomalies of real-world data, which will bring great challenges to anomaly detection. In this article, we propose a time-series mutual adversarial network (TMAN), a novel reconstruction model for anomaly detection on multivariate time series. It is based on the idea of adversarial learning and consists of two identical subnetworks. During the training process, two subnetworks can independently complete the learning of the time distribution of normal samples of industrial time series data for mutual adversarial. In the process of detecting, we obtain the residual values of TMAN reconstructed for different time series samples to discriminate anomalies. We combine TMAN and anomaly determination mechanisms to build a new industrial time series anomaly detection framework named TMANomaly. In addition, we select the dataset features with a grey correlation algorithm to achieve very high performance with a small number of features. Experimental results show that our proposed TMANomaly outperforms five popular anomaly detection methods and effectively improves the accuracy of industrial multivariate time series anomaly detection.**

***Index Terms*—Anomaly detection, generative adversarial networks (GANs), mutlivariate time series, mutual adversarial networks.**

I. Introduction

With the vigorous development of the Industrial Internet of Things (IIoT), many industrial systems and data centers have installed a large number of sensors to get a large amount of monitoring data. For such systems with a large number of interconnected sensors, as in water treatment plants, power grids, etc., not only are they under pressure to handle huge amounts of high-dimensional and complex data on a daily basis, but they also have to bear some risk [1]. It has become particularly important to secure the critical infrastructure of such IIoT systems against attacks. However, with the large increase in data dimensions, it is difficult for monitors to immediately detect anomalies, which leads to an increased need for automated anomaly detection methods.

Therefore, anomaly detection of industrial time series data is gradually becoming a hot issue in the field of IIoT. Anomalies generally refer to values or behaviors that do not conform to the expected pattern. Anomaly detection is the identification of anomalies, that is, the detection of anomalous samples that differ significantly from other observations [2]. Anomaly detection is usually an unsupervised machine learning task due to the lack of inherently labeled data. Most existing unsupervised methods are mostly based on linear approaches [3], and there is less modeling of nonlinear time series data.

In recent years, with the development of deep learning technology, time series anomaly detection methods have made improvements. In general, existing methods can be roughly divided into two categories: forecasting-based methods [4], [5], where researchers usually predict future normal conditions based on historical data, then calculate the residuals between the actual and predicted data, and determine anomalies based on the relative magnitude of the residuals. Reconstruction-based methods [6], [7], [8], as [7], are classical reconstruction-based anomaly detection methods, which determine the presence of anomalies by autoencoder (AE) reconstruction errors. Such methods aim to learn the low-dimensional feature representation space of a sample and be able to better reconstruct a particular data sample on that space. Reconstruction methods in anomaly detection usually minimize reconstruction errors by forcing the feature representation to learn the significant laws of the data. Anomalous samples are difficult to reconstruct anomalies from the resulting laws and therefore have large reconstruction errors.

With the further optimization and iteration of the deep learning model framework, Goodfellow et al. [9] proposed the generative adversarial network (GAN), which was originally used to generate image data by gaming the generator and discriminator against each other to learn the features of the training image to generate samples similar to the training image. GAN is one of the most promising deep learning models today and is widely used in different fields. The application of GAN in the direction of anomaly detection [10], [11]. These works have proved that GAN has good performance in the detection of point anomalies, but there are still few works on GAN applied to

Manuscript received 2 January 2023; revised 20 May 2023; accepted 13 June 2023. Date of publication 27 June 2023; date of current version 19 January 2024. This work was supported in part by the Hunan Provincial Natural Science Foundation of China under Grant 2022JJ30398, Grant 2022JJ40277, and Grant 2021JJ30455 and in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 22A0056 and Grant 22B0102. Paper no. TII-23-0015. *(Corresponding author: Pingping Dong.)*

The authors are with the College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China (e-mail: zlm@hunnu.edu.cn; oreoB1024@hunnu.edu.cn; xiexiaowei@hunnu.edu.cn; lychen@hunnu.edu.cn; ppdong@hunnu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3288226.

Digital Object Identifier 10.1109/TII.2023.3288226

multidimensional and dynamic time series. Moreover, these methods do not explain clearly the highly nonlinear time dependence and complex interactions between variables, nor do they efficiently extract temporal information from industrial time series data, and they still have room for improvement. Also, as labeling data in an automated industrial production environment requires a degree of labor cost, it is significant how a small amount of data are used to improve model performance.

To address the abovementioned challenges, we propose a new multivariate time series anomaly detection strategy framework named TMANomaly, and model the complex multivariate correlations between large-scale sensor data streams to achieve effective extraction of industrial time series data information and subsequently complete the anomaly detection of IIoT data. In TMANomaly, we use a time series mutual adversarial network (TMAN) to detect anomalies based on reconstructing industrial time series. TMAN extracts industrial time series information by constructing two identical subnetworks composed of long short-term memory (LSTM). Furthermore, unlike the traditional GAN method with fixed subnetwork role, the roles of the two subnetworks of TMAN can be swapped with each other as each other's reconstructors or discriminators, which allows the subnetworks to obtain better learning capability and thus obtain better anomaly detection performance than that of the traditional GAN-based methods.

In this article, we use TMAN to learn the distribution of industrial high-dimensional dynamic time series data while detecting anomalies based on reconstruction and recognition losses. The main contributions of this article are as follows.

1) We propose a novel mutual adversarial multivariate time series anomaly detection strategy framework named TMANomaly, which can model complex multivariate correlations between multidimensional industrial time series data streams.
2) In TMANomaly, we propose a TMAN, which consists of two identical subnetworks that can exchange roles as reconstructors or discriminators, performing mutual adversarial training to obtain better reconstructive capabilities than traditional methods.
3) Through performance evaluations of the proposed TMANomaly, this work shows that the scheme can successfully perform reliable and robust learning to process IIoT time series data and perform anomaly detection.

The rest of this article is organized as follows. In Section II, we give the related review on anomaly detection. Section III describes our proposed TMANomaly framework and anomaly evaluation function. Section IV describes SWaT, WADI, and experimental setup, and evaluates the detection performance of TMANomaly. Finally, Section V concludes this article.

II. Related Work

A. Time Series Anomaly Detection

Time series anomaly detection is generally divided into univariate time series anomaly detection and multivariate time series anomaly detection. Univariate detection methods consider just a single time dependent variable, while multivariate detection methods are able to deal with multiple time dependent variables simultaneously. Nowadays, most research involves univariate models, and in the traditional anomaly detection framework, anomaly detection is divided into an estimation phase, where a specific method is used to predict the value of a time-stamped variable, and then this value is compared to a threshold value to detect anomalies. Such as the autoregressive integrated moving average (ARIMA) model [12] was to capture the linear relationship between future and historical values to model the time series for its anomaly detection. However, ARIMA was only for the processing of univariate and relatively smooth time series data and cannot handle complex time series data.

As deep learning has developed, various neural networks have been applied to time series anomaly detection. DeepAnT [13] used a convolutional neural network to model time series prediction to improve prediction performance for periodic and seasonal time series, and defines anomalies for their detection using the square root of the error between the predicted and true values. LSTM networks are also widely used for time-series-based anomaly detection, such as the LSTM-VAE [14] used LSTM networks to reconstruct industrial time series data in a reconstructed manner by fusing signals and reconstructing their expected distributions for anomaly detection. However, these methods often fail to perform well when faced with the processing of high-dimensional and complex dynamic industrial time series data, which indicates that their work in IIoT has certain defects.

In the IIoT environment, most of the inputs are multivariate time series with some correlation, and these IIoT data need to be continuously processed for statistical analysis and anomaly detection. For multivariate time series, many studies begin to learn the correlation between multiple variables to enhance the performance of anomaly detection methods. Hundman et al. [5] used a dynamic unsupervised method to detect anomalies in telemetry data returned by spacecraft, the method predicts only one step backwards and only one dimension, while collecting the errors at each step to form an error vector and doing exponential smoothing of the error vector to obtain a threshold value based on the smoothed data. Wu et al. [4] used LSTM networks to detect anomalies in industrial data by getting prediction errors from Gaussian Bayesian models. These abovementioned methods are able to accomplish the detection of anomalies in industrial data, but there is still need for improvement in terms of performance, such as [5] has failed to solve the correlation, dependency problems inherent in telemetry.

B. Anomaly Detection in GAN

As an unsupervised learning algorithm, GAN has a good performance in anomaly detection. Schlegl et al. [15] proposed AnoGAN, which was the first attempt to use GAN for anomaly detection. AnoGAN used DCGAN to unsupervisedly learn to train distributions from normal samples. The test samples was then feed in and the loss function is defined for multiple backpropagation iterations to obtain mapping results from the image to the potential space using residual losses.

However, it also suffers from a time efficiency assessment. Therefore, f-AnoGAN [16] was proposed based on AnoGAN, which was optimized to improve the performance of anomaly detection by learning the representation of training data through WGAN [17]. Zenati et al. [18] proposed a bidirectional GAN-based adversarial learning anomaly detection method, also used reconstruction loss error to determine whether data samples are anomalous, and the method adds potential space to the image space based on AnoGAN, allowing the network to update faster. GANomaly [19] modified the GAN network structure and loss function to constrain the potential space. These methods are all anomaly detection for point anomalies, but are deficient for temporal anomaly detection and cannot effectively learn the distribution of temporal data.

C. Time Series Anomaly Detection in GAN

In recent years, researchers have been trying to combine GAN with multivariate time series and anomaly detection to achieve a crossover of multistream. At the earliest, Li et al. [8] proposed MAD-GAN, which is the first attempt of GAN for time series anomaly detection. In the training module, detection module, the underlying framework of GAN consists of LSTM-RNN to achieve capturing the temporal correlation of time series distribution for the determination of anomalies.

After this, many studies have been done based on [8], such as FID-GAN [20], TAnoGAN [21], TadGAN [22], and they were all unsupervised anomaly detection methods based on GAN. The data are modeled as multivariate time series, and LSTM-RNN is used as a generator and discriminator to capture the correlation of the time series for anomaly score determination in a reconstructed manner. The difference is that FID-GAN was combined with fog computing to train encoders that accelerate the reconstruction loss computation to make it better, TAnoGAN used a different structure to achieve processing of smaller datasets, and TadGAN does not use the original loss function. Yoon et al. [23] proposed TimeGAN, combines unsupervised GAN algorithms with regression principles for joint training to generate time series that retain time dynamics. Kong et al. [11] proposed an AMBi-GAN based on bidirectional LSTM and attention mechanism, which uses a bidirectional LSTM network with attention mechanism as its generator and discriminator for anomaly detection by reconstruction loss and generation loss reconstruction error.

All these methods have good performance in time series anomaly detection. However, the performance of the discriminator and generator of traditional GAN is difficult to balance during training, which leads to the GAN remaining difficult to train. In training reconstructed neural networks using traditional adversarial training methods, the imbalance of the subnetworks may cause the model to enter a local optimum solution, which results in the combination of GAN and LSTM subnetworks not working optimally. Especially, when the input is high-dimensional, dynamic and complex IIoT data, these methods are more difficult to handle the extraction of temporal information, limiting the performance of the model. Although existing works have utilized adversarial training to enhance the reconstruction capability of the decoder in GAN-based frameworks for anomaly detection, these methods typically do not impose any constraints on the distribution of the encoded latent variables used to generate reconstructed samples. Consequently, the resulting models lack the ability to generate new samples that conform to the distribution of the training data, and the reconstruction loss often dominates the overall loss function. Furthermore, the end-to-end adversarial training strategy employed in these models may not be optimal for fitting the underlying data distribution, as it does not explicitly account for the distributional properties of the data.

Therefore, we propose a TMAN with two identical subnetworks that can balance the performance of the subnetworks. Each subnetwork can act as a reconstruction subnetwork and a discriminator subnetwork. The subnetworks can swap their roles during training, which can alleviate the problem of underperformance due to imbalanced subnetworks in traditional adversarial learning. We apply TMAN to the proposed time series anomaly detection framework TMANonaly.

III. Proposed TMANomaly

A. Problem Definition

In this work, we focus on multivariate time series anomaly detection. Due to the effect of the high imbalance in industrial time series data, our training data are derived from true normal samples (without anomalies) for S sensors with brakes and anomaly detection is performed by reconstructing the error on test data with anomalous data. The time granularity of these data is T_{train}. We define the values of these datasets as $D_{\text{train}} = [D_{\text{train}}^{(1)}, \ldots, D_{\text{train}}^{(T_{\text{train}})}]$, where $D_{\text{train}}^{(t)} \in R^s$ denotes the timing data at a timestamp of T_{train}, and S represents the quantity value of different sensors, i.e., the feature number. The test data come from same S sensors with a time scale of T_{test}. The test data are represented as $D_{\text{test}} = [D_{\text{test}}^{(1)}, \ldots, D_{\text{test}}^{(T_{\text{test}})}]$, where $D_{\text{test}}^{(t)} \in R^s$, and it has a specific window size w.

Our goal is to detect anomalies in industrial time series data, so given a multivariate sequence of consecutive w time steps, our output is based on a reconstructed output vector, i.e., $\hat{y} \in R^n$, where $\widehat{y^{(t)}} \in \{0, 1\}$, which indicates the presence or absence of anomalies at timestamp t. That is, we obtain the final anomaly result by selecting different thresholds for the anomaly scores at each timestamp returned.

For better understanding, we also provide some basic related concepts as follows.

1) GAN: The original GAN is based on the ideas of game theory and is an unsupervised approach. It uses a generator G and a discriminator D for adversarial training. The generator G is used to generate fake data, i.e., it receives an input of a random vector z, which follows a particular distribution $P_z(z)$, such as a Gaussian distribution, and subsequently outputs a generated sample to the discriminator, which classifies it in this way and ends when it reaches convergence, and its optimization objective function can be expressed as

$$\min_G \max_D V(DG) = E_{x \sim p_{dt}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \quad (1)$$

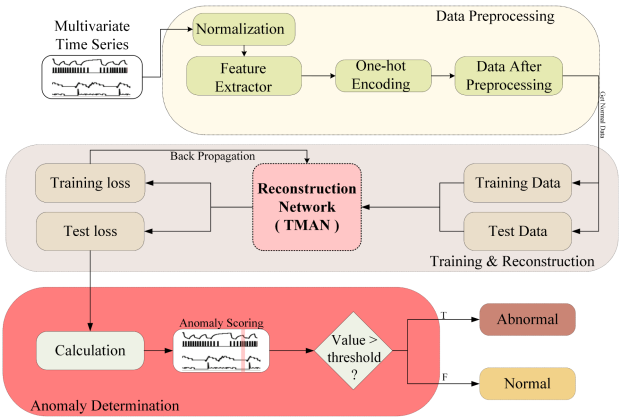

Fig. 1. Overview of the framework of TMANomaly.

2) LSTM: LSTM is a special form of RNN that can solve the gradient vanishing problem that RNNs may face, and is a widely used method in various fields. It consists of input gate F_t, forgetting gate I_t and output gate O_t as well as the storage unit $\bar{S}_t$ to protect and control the information. In a traditional LSTM network, the input gate is used to update the information and additionally to pass the hidden information from the previous layer to the tanh function, the output gate is used to determine the value of the next hidden state, and the forgetting gate is used to control how many input values the cell state has after moment t.

B. TMANomaly Framework

The proposed TMANomaly framework consists of data preprocessing, reconstruction network, and anomaly determination mechanism, as shown in Fig. 1. First, the original dataset is preprocessed and normal samples are extracted as training data, after which the training data are modeled using the reconstruction network. Then, the test data are input into the reconstruction network and the reconstruction loss of the samples is obtained after reconstruction. Finally, anomalies are determined from reconstruction losses by the anomaly determination mechanism.

1) TMAN: In IIoT environments, data from sensors and brakes often interact with each other, and each sensor can be treated as a specific feature whose anomalies are then predicted by training on the features. We design TMAN to enable two identical subnetworks capable of automatically learning distributions of multivariate time-series data to fight each other, thus allowing both to reach equilibrium.

The proposed TMAN is an anomaly detection method based on reconstruction, i.e., the industrial multivariate time series data are captured by a subnetwork TRecAE consisting of an LSTM, and TMAN then outputs a new reconstruction sequence based on the variability of the sequence from the input sequence at each step length to obtain the reconstruction error, after which the anomaly or normality of the step length data in the sequence is determined. Fig. 2 further details the overall model architecture.

As shown in Fig. 2, first, TRecAE is given the input data $x^{(t)} \in D^s_{\text{train}}$. In order to obtain the distribution of the multivariate time series data, our subnetwork consists of an LSTM, so at timestamp t we set the active window in such a way that it is neighbor-fetching and does not overlap, so we define the input model as follows:

$$x^{(t)} = \left[x^{(t-w)}, x^{(t-w+1)}, \ldots, x^{(t-1)}\right] \tag{2}$$

where, w is the active window size.

The LSTM network is used to allow both subnetworks to learn the distribution of multivariate time series. From the abovementioned explanatory note on the LSTM model, it is clear that after the LSTM receives the input $x^{(t)}$, the output ω at timestamp t is obtained through the processing of the input gate F_t, the forgetting gate I_t and the output gate O_t and the storage unit $\bar{S}_t$. The process is formulated as follows:

$$\omega = h\left(x^{(t)}; \kappa_e\right) \tag{3}$$

where, $h(*)$ represents the encoding function of the LSTM network $(F_t, I_t, O_{t,}, \bar{S}_t)$ and κ_e represents the different sets of parameters.

To allow TRecAE to perform adversarial training while obtaining our reconstruction error, we plugged in a fully connected layer after the LSTM, which received the output value ω from the LSTM processing and fed it to two other fully connected (FC) layers, which is mapped and then fed into the other two FC layers to obtain our reconstruction error $x_i^{'(t)}$. The process is expressed as follows:

$$x_i^{'(t)} = T\text{rec}_1(\omega). \tag{4}$$

Likewise, the other two FC layers receive w and are able to output our discriminant score $c^{'(t)}$, the process is formulated as follows:

$$c^{'(t)} = T\text{rec}_2(\omega). \tag{5}$$

Equations (2)–(5) previously form the entire TRecAE subnetwork. Similar to (1), our reconstruction network is constructed with TRecAE_1 to obtain the reconstructed input. In the classic GAN, there is also a discriminant network to discriminate the data input by the generator, so we set TRecAE_2 as the discriminant network.

Throughout the training process, the reconstruction network and the discriminator network play a max-minimization game, i.e., the G_{R1} reconstruction network and the D_{R2} discriminator network fight each other's game, one discriminates and the other generates reconstruction data, and stops after reaching maximum balance. First, we train the discriminative network D_{R2}, set the loss function to cross-entropy loss function, and finally obtain the loss L_D^t as shown in the following:

$$L_D^t = \frac{1}{m}\sum_i^m \left[\log D_{R2}\left(x_i^{(t)}\right) + \log\left(1 - D_{R2}\left(G_{R1}\left(x_i^{(t)}\right)\right)\right)\right] \tag{6}$$

where, m is the sample size of minibatch training data $x_i^{(t)}$, and $G_{R1}(x_i^{(t)})$ denotes the reconstructed data obtained.

After updating the parameters, we keep the network parameter values fixed and train the reconstructed network G_{R1}. At this

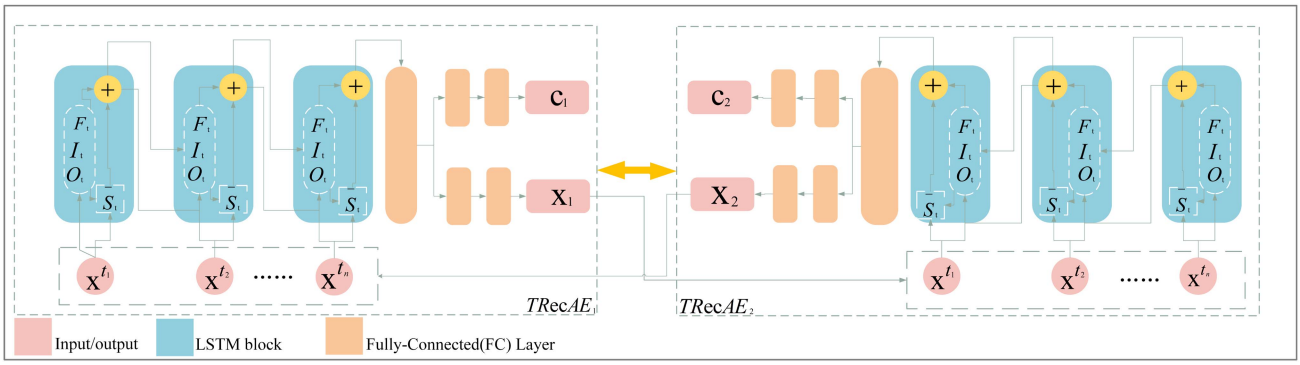

Fig. 2. Outline of TMAN.

stage, we still use the cross-entropy loss function, which also gives us the loss value L_g^t for this part of the training

$$L_G^t = \frac{1}{m}\sum_{i}^{m}\left[\log\left(1 - D_{R2}\left(G_{R1}\left(x_i^{(t)}\right)\right)\right)\right]. \quad (7)$$

After training these two subnetworks, the reconstructed network can obtain reconstruction capability, but it is still some distance away from the goal we aim to achieve. Our proposed TMAN, which uses the TRecAE as the reconstruction model, takes the input real multivariate time series data $x^{(t)}$ as input instead of the random vector z. Usually, we discriminate the reconstruction data $x^{(t)}$ based on the input data $x^{(t)}$. However, in the abovementioned training process, we only considered the aspect that the output is real/fake, and did not consider other details, which made our model not achieve the best results in anomaly detection, so we use $x'^{(t)}$ and $x^{(t)}$ feature matching loss L_{rec}^t and L_G^t for joint training G_{R1} reconstructs the network. L_{rec}^t is defined as follows:

$$L_{\text{rec}}^t = \frac{1}{m}\sum_{i}^{m}\left[x_i^{(t)} - G_{R1}\left(x_i^{(t)}\right)\right]^2. \quad (8)$$

Finally, we use the weights shown in (9) and L_{total}^t to set the total loss for backpropagation to update G_{R1}

$$L_{\text{total}}^t = \beta_1 L_{\text{rec}}^t + \beta_2 L_G^t \quad (9)$$

where, β_1, β_2 are the two loss weighting parameters, respectively.

The abovementioned process of performing one iteration for two subnetworks is only a part of the whole adversarial learning process. In the whole adversarial learning process, there may be an imbalance situation where the discriminator is weak and the generator is strong, and how to balance the two has been the direction of many scholars' efforts. We choose the mutual adversarial approach to alleviate this imbalance. As shown in Fig. 2, TRecAE is designed to have a reconstruction output, and a discriminative category output. In the training process, two TRecAEs can form a mutual adversarial model, and both subnetworks can act as each other's reconstruction network or discriminative network, as shown in Fig. 3. If there is unbalance, the roles of our two subnetworks are switched so that the whole

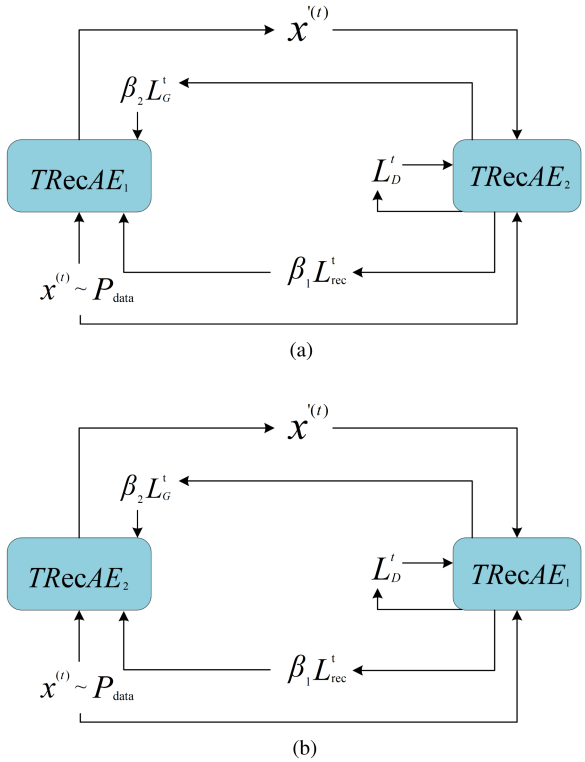

Fig. 3. Training process of TMAN. (a) Model architecture: TRecAE$_1$ as the reconstruction network, TRecAE$_2$ as the discriminant network. (b) Model architecture: TRecAE$_2$ as the reconstruction network, TRecAE$_1$ as the discriminant network.

model gets not just a local optimum, but a better reconstruction capability.

2) Mechanism for Determining Abnormal Scores: During the testing process, in order to obtain the anomaly scores for each time scale, we have obtained two subnetworks that can reconstruct the data independently through time series mutual adversarial training. During the testing process, we use the reconstructed output of TRecAE for anomaly detection. Therefore, we compose the reconstructed outputs of the two subnetworks as the final output x'_j, as shown in the following:

$$x'_j = \alpha\text{TRecAE}_1(x_j) + (1-\alpha)\text{TRecAE}_2(x_j) \quad (10)$$

Algorithm 1: TMANomaly.

Input: training set, test set.
Output: The anomaly sample x'_j.

for each epoch **do**
  Initialization: Set TRecAE$_1$ as G, TRecAE$_2$ as D.
  for minibatch data from training set **do**
    Sampling $\{x_i^{(t)}\}_{i=1}^m$ from minibatch data.
    Get the reconstruction sample from G: $x_i'^{(t)} = G(x_i^{(t)})$.
    Use (6) to update discriminator D.
    Use (9) to update reconstructor G.
    Swap the roles of TRecAE$_1$and TRecAE$_2$.
    Execute Lines 5 − 7 again.
  end for
end for
Sampling $\{x_j^{(t)}\}_{j=1}^n$from test set.
Get the final output x'_j of TMAN by (10).
Calculate the anomaly scores $\mathcal{A}(x_j)$ by (11).
for $j = 1, 2, \ldots, n$ **do**
  if $s'_j >$ threshold **then**
    x'_j is abnormal sample.
  end if
end for
return x'_j

where, $\alpha \subseteq [0, 1]$, is the weight parameter that regulates the contribution of the outputs of the two reconstructed self-encoders to the final output data, and x_j is the data at a certain timestamp t in the D_{test}.

Then, similar to the loss objective, we calculate the residual value between x_j and x'_j using the loss function $\mathcal{L}_1$ and calculate the deviation level score as the anomaly score for the data at timestamp t. $\mathcal{A}(x_j)$ is defined as follows:

$$\mathcal{A}(x_j) = |x_j - x'_j|. \tag{11}$$

Finally, we use the obtained anomaly scores normalized to get the final output vector $\hat{y}$, and if $\hat{y}$ exceeds a fixed threshold, we mark the timestamp t as an anomaly.

The entire training and detection process of TMANomaly is shown in Algorithm 1.

IV. Experimental Results and Analysis

A. Datasets and Preprocessing

To evaluate our proposed TMANomaly, we chose two different datasets to train and detect industrial time series anomalies, including SWaT, WADI. Prior to the experiments, we preprocessed the datasets.

1) SWaT: SWaT [24], from the Water Treatment Testbed coordinated by the Singapore Public Utilities Board, is a real-world sensor dataset from a water treatment physical testbed system that records real anomalies, which integrates digital, physical in order to serve as a control and monitoring system, and many large-scale plants have applications for such systems. The dataset was run for 11 days (seven days in normal operation and four consecutive days in attack) and collected all values obtained from 51 sensors and actuators, labeling the data according to normal and abnormal behavior.

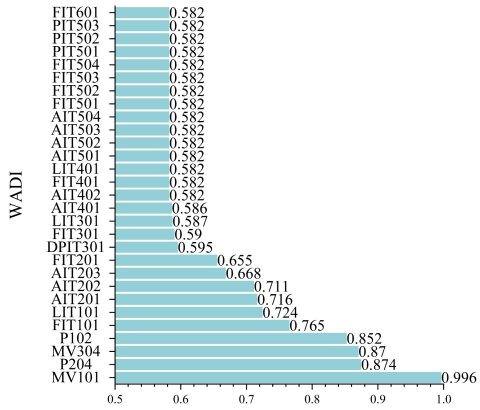

Fig. 4. Grey correlation degree of sensors on WADI.

2) WADI: WADI [25] is also a safe water treatment dataset, but it is an extension of the SWAT dataset, which consists of three processes—water treatment, storage, and distribution network—to form a complete, realistic water treatment process. The dataset was run continuously for 16 days (14 days in normal operation and two days under attack) and data from 123 sensors and actuators were collected.

3) Data Preprocessing: During data preprocessing, we transform each record's label by replacing it with a [0,1] numeric label, while having some features encoded using one-hot and mapping other numeric features between [0,1] by using normalization.

B. Feature Selection

Feature selection is a key part of the machine learning task. Feature selection allows selecting features that are important to the model and can reduce the dimensionality of the training data and speed up the training of the model, while also reducing model complexity and avoiding overfitting. Grey relation analysis (GRA) [26] is a multifactor statistical analysis method for feature extraction of interacting factors in a system. In this article, features are selected for SWaT and WADI, which are derived from real-world water treatment plants and feature correlated and interacting sensor and brake values. Therefore, the GRA algorithm is chosen for feature selection of our industrial time series dataset to select the features with the highest impact on the whole system, in order to achieve optimal experimental results with a small set of features. For SWaT, we apply the GRA algorithm to obtain its 30 features with the highest correlation, as shown in Fig. 4. These 30 features were chosen as the new features for the dataset. For WADI, after the same process, we obtain 35 features with the highest correlation, as shown in Fig. 5. We chose 36 features as the new features for the dataset. A summary table of statistical information from our processing of the two datasets in Table I.

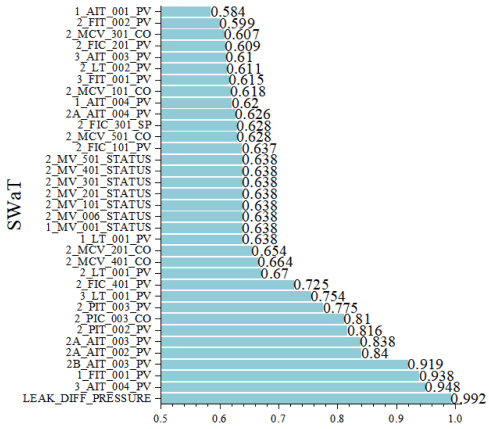

Fig. 5. Grey correlation degree of sensors on SWaT.

TABLE I
STATISTICAL SUMMARY OF DATASETS: SWAT AND WADI

Datasets	SWaT	WADI
#Features	30	35
#Attacks	41	15
Attacks duration (mins)	2˜25	1.5˜30
Training size (normal data)	49 619	120 899
Testing size (data with attacks)	44 931	17 219
Anomaly rate (%)	12.14	5.75

C. Baseline

To evaluate the performance of TMANomaly, we compare the performance of it with that of five different anomaly detection methods as follows.

1) *LSTM-VAE [14]:* It replaces feedforward networks with self-encoders using LSTM, while using anomaly scores to obtain reconstruction errors.
2) *MAD-GAN [8]:* Anomaly scores are calculated based on a reconstruction approach using GAN networks and the temporal correlation of the time series distribution is captured using LSTM-RNN.
3) *THOC [27]:* It is a model for anomaly detection with one-class model of time series, using an expansive recurrent neural network to capture temporal dynamics.
4) *GDN [28]:* It is proposed that aims to learn relationships between sensors in the form of graphs, and then recognize and interpret the deviations of the learned patterns.
5) *GTA [29]:* It is a combinable microconnectivity learning strategy combined with information propagation convolution that automatically learns the graph structure of relationships between sensors.

D. Indicators and Experimental Settings

1) Evaluation Indicators: We use precision (Prec), recall (Rec), and F1-scores (F1) to evaluate the performance of TMANomaly and baseline model. $\text{Prec} = \frac{\text{TP}}{\text{TP+FP}}$, $\text{Rec} = \frac{\text{TP}}{\text{TP+FN}}$, $F1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}$, and TP, TN, FP, FN are the numbers of true positives, true negatives, false positives, and false negatives. Given that datasets are extremely unbalanced, we chose these unbalanced indicators so that we could rationalize their selection.

TABLE II
EXPERIMENTAL RESULTS ON SWAT AND WADI

Datasets	Methods	Precision(%)	Recall(%)	F1
SWaT	LSTM-VAE [14]	96.24	59.91	0.74
	MAD-GAN [8]	98.97	63.74	0.77
	THOC [27]	98.08	79.94	0.88
	GDN [28]	99.35	68.12	0.81
	GTA [29]	94.83	88.10	0.91
	TMAN*(ours)	**99.75**	80.50	0.89
	TMAN**	95.68	**92.30**	0.86
	TMAN***	98.59	82.96	**0.91**
WADI	LSTM-VAE [14]	87.79	14.45	0.25
	MAD-GAN [8]	41.44	99.99	0.37
	THOC [27]	–	–	–
	GDN [28]	97.50	40.19	0.57
	GTA [29]	83.91	83.61	0.84
	TMAN*(ours)	**95.60**	91.24	0.94
	TMAN**/***	93.85	**99.00**	**0.96**

Best performance in bold.
* Represents the results chosen by best precision.
** Represents the results chosen by best FI-score.
*** Represents the results chosen by best recall.

2) Experimental Settings: We perform all experiments on a computer equipped with an 11th Gen Intel (R) Core (TM) i7-11800H and 16 G RAM. The development framework was Pytorch 1.7 with CUDA 11.6, and an NVIDIA GTX3060 GPU was used for training acceleration, where the learning rate was set to 0.000005, the batch size was set to 64, and the loss value of the network, such as $\beta_1 = 50$, $\beta_2 = 1$. The neural network parameters were initialized in training and testing using a random seed 777. We trained the model for up to 1000 periods, while setting the sliding window size w to 30 for both WADI and SWaT.

E. Performance Evaluation

We obtain the best Prec scores (labeled*), F-1 (labeled**), and Rec scores (labeled***), and show this and compare all baselines separately across the board. The results are shown in Table II.

In Figs. 6 and 7, TMANomaly largely outperforms the baseline on both SWaT and WADI, and both have high Prec of 99.75% and 95.60%, respectively. In terms of F-1 scores, TMANomaly outperforms the best baseline by 14.28% on WADI and is on par with the best baseline (GTA) on SWaT. For recall, TMAN is on par with the best baseline on WADI, while on SWaT the recall is only 0.05% worse than the best baseline. WADI is more unbalanced than SWaT. For Prec, TMANomaly also performs well on SWaT, while TMANomaly performs well for recall, F-1 scores and achieves a balanced effect while

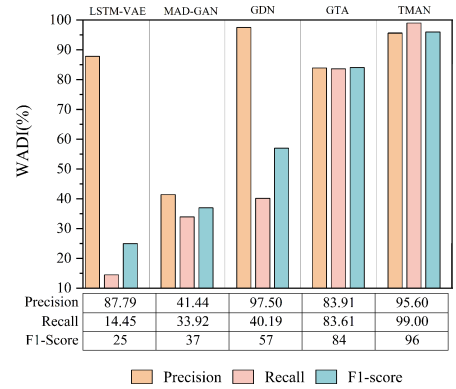

Fig. 6. Performance of the models on WADI.

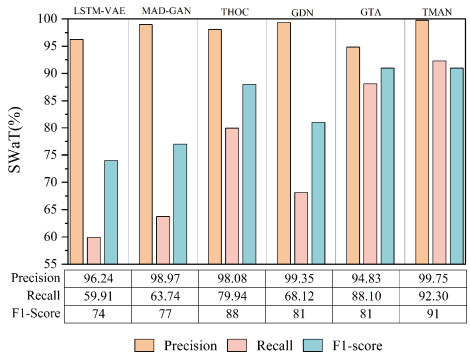

Fig. 7. Performance of the models on SWaT.

TABLE III
RESULTS OF ABLATION MUTUAL ADVERSARIAL EXPERIMENTS ON WADI AND SWAT

	SWaT			WADI		
Methods	Prec	Recall	F1	Prec	Recall	F1
Single TRecAE	0.981	0.841	0.906	0.923	0.953	0.906
TMAN	0.998	0.923	0.910	0.956	0.990	0.960

ensuring very high accuracy, which is important in practical applications and can represent a strong effectiveness even in unbalanced and high-dimensional attack scenarios.

F. Ablation Experiment

In order to evaluate the necessity of our proposed TMANomaly, we will perform ablation experiments on TMANomaly.

1) Ablation of Mutual Adversarial Training: In TMAN, we use two TRecAEs to perform mutual adversaries to reach the final balance. To evaluate the necessity of these components, we will exclude them. We remove the adversarial component of the mutual adversarial and change the original two TRecAEs into a single TRecAE. Also, we compare our proposed TMAN (i.e., the case of using two TRecAEs) as a baseline model. The results are shown in Table III.

2) Ablation of Selecting Features: In TMANomaly, we perform feature selection. To evaluate it, we perform ablation experiments on SWaT and WADI for the feature selection part as well. We remove the feature selection part, leaving all the original features of the dataset intact, and then perform anomaly detection, Also, we compare our proposed TMAN as a baseline model. The results are shown in Table IV.

TABLE IV
RESULTS OF ABLATION FEATURE SELECTING EXPERIMENTS ON WADI AND SWAT

	SWaT			WADI		
Methods	Prec	Recall	F1	Prec	Recall	F1
Original features	0.981	0.841	0.906	0.923	0.953	0.906
TMAN	0.998	0.923	0.910	0.956	0.990	0.960

From the abovementioned ablation experiments, we note the following observations.

1) There is a gap between TMAN and time-series methods that do not use mutual adversarial networks, which plays an important role in our mutual adversarial network time-series modeling in dealing with high-dimensional time-series anomaly detection.
2) The feature selection method allows us to optimize the performance of TMANomaly, and with fewer features than other baseline methods, we can effectively reduce the cost of manually labeling data.

These findings show that each component of TMANomaly is important and makes the framework play an important role in time series anomaly detection.

V. CONCLUSION

In this article, we propose TMANomaly, a mutual adversarial network-based anomaly detection framework for industrial time series, which uses LSTM as a subnetwork in the mutual adversarial network to learn the distribution correlation of time series data, and the subnetwork can independently obtain reconstruction errors to compute reconstruction losses for anomaly determination of industrial time series. We also apply the GRA algorithm to the feature selection of SWaT and WADI, which can capture the anomalies of industrial time series with fewer features. Experiments on two IIoT datasets, such as WADI, SWaT, show that TMANomaly outperforms other advanced studies in terms of accuracy, F-1 score, and recall rate. Next, we plan to optimize the model structure of the proposed TMANomaly to further improve the performance of industrial anomaly detection.

REFERENCES

[1] Y. Zhang, Z. Dong, W. Kong, and K. Meng, "A composite anomaly detection system for data-driven power plant condition monitoring," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4390–4402, Jul. 2020.
[2] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques infinancial domain," *Future Gener. Comput. Syst.*, vol. 55, pp. 278–288, 2016, doi: 10.1016/j.future.2015.01.001.
[3] S. Li and J. Wen, "A model-based fault detection and diagnostic methodology based on PCA method and wavelet transform," *Energy Buildings*, vol. 68, pp. 63–71, Jan. 2014.
[4] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, "LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5244–5253, Aug. 2020.
[5] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. Int. Conf. Knowl. Discov. Data Mining*, London, U.K., 2018, pp. 387–395.

[6] L. Zhang, X. Xie, K. Xiao, W. Bai, K. Liu, and P. Dong, "MANomaly: Mutual adversarial networks for semi-supervised anomaly detection," *Inf. Sci.*, vol. 611, pp. 65–80, Sep. 2022.

[7] C. C. Aggarwal, "An introduction to outlier analysis," in *Outlier Analysis*. New York, NY, USA: Springer, 2013, pp. 1–34.

[8] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S. K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw.*, Munich, Germany, 2019, pp. 703–716.

[9] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM.*, vol. 63, no. 11, pp. 139–144, Oct. 2020.

[10] Z. Xiong, Z. Cai, Q. Han, A. Alrawais, and W. Li, "ADGAN: Protect your location privacy in camera data of auto-driving vehicles," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6200–6210, Sep. 2021.

[11] F. Kong, J. Li, B. Jiang, H. Wang, and H. Song, "Integrated generative model for industrial anomaly detection via bidirectional LSTM and attention mechanism," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 541–550, Jan. 2023.

[12] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50. pp. 159–175, Jan. 2003.

[13] 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, 2019.

[14] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.

[15] T. Schlegl, P. Seebock, and S. M. Waldstein, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, Boone, NC, USA, 2017, pp. 146–157.

[16] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks," *Med. Image Anal.*, vol. 54, pp. 30–44, May 2019.

[17] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[18] H. Zenati, M. Romain, C. S. Foo, B. Lecouat, and V. R. Chandrasekhar, "Adversarially learned anomaly detection," in *Proc. IEEE Int. Conf. Data Mining,* Singapore, 2018, pp. 727–736.

[19] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervise anomaly detection via adversarial training," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 622–637.

[20] P. Freitas de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. G. Santos, D. Macêdo, and C. Zanchettin, "Intrusion detection for cyber-physical systems using generative adversarial networks in fog environment," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6247–6256, Apr. 2021.

[21] M. A. Bashar and R. Nayak, "TAnoGAN: Time series anomaly detection with generative adversarial networks," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2020, pp. 1778–1785.

[22] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "TadGAN: Time series anomaly detection using generative adversarial networks," in *Proc. IEEE Int. Conf. BigData*, 2020, pp. 33–43.

[23] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5508–5518.

[24] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *Proc. IEEE Int. Workshop Cyber-Phys. Syst. Smart Water Netw.*, 2016, pp. 31–36.

[25] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "WADI: A water distribution testbed for research in the design of secure cyber physical systems," in *Proc. Int. Workshop Cyber-Phys. Syst. Smart Water Netw.*, 2017, pp. 25–28.

[26] L. Y. Zhai, L. P. Khoo, and Z. W. Zhong, "Design concept evaluation in product development using rough sets and grey relation analysis," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 7072–7079, Apr. 2009.

[27] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 13016–13026.

[28] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4027–4035.

[29] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, "Learning graph structures with transformer for multivariate time series anomaly detection in IoT," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9179–9189, Jun. 2022.

Lianming Zhang received the B.S. in physical education and M.S. degree in curriculum and instruction in physics from Hunan Normal University, Changsha, China, in 1997 and 2000, respectively, and the Ph.D. degree in computer applied technology from Central South University, Changsha, China, in 2006.

He is currently a Professor with Hunan Normal University. His research interests include Internet of Things, intelligent network, edge computing, and machine learning.

Wenji Bai received the B.S. degree in computer science and technology from the College of Big Data and Intelligent Engineering, Yangtze Normal University, Chongqing, China, in 2020. She is currently working toward the M.S. degree in computer science and technology with the College of Information Science and Engineering, Hunan Normal University, Changsha, China.

Her research interests include anomaly detection and deep learning.

Xiaowei Xie received the B.S. degree in resource surveying engineering from the College of Information and Computer, Taiyuan University of Technology, Taiyuan, China, in 2020 and the M.S. degree in computer science and technology from Hunan Normal University, Changsha, China, in 2023. He is currently working toward the Ph.D. degree in computer science and technology with the College of Information Science and Engineering, Central South University, Changsha, China.

His research interests include anomaly detection and deep learning.

Liying Chen received the B.S. degree in information and computing sciences from the College of Information, Shanghai Ocean University, Shanghai, China, in 2022. She is currently working toward the M.S. degree in computer science and technology with the College of Information Science and Engineering, Hunan Normal University, Changsha, China.

Her research interests include datacenter transport, network performance analysis, and protocol optimization and protocol design in RDMA.

Pingping Dong received the B.S. in communication engineering, M.S. in information and communication engineering, and Ph.D. degree in computer technology from the School of Information Science and Engineering, Central South University, Changsha, China, in 2008, 2011, and 2015, respectively.

She is currently an Associate Professor with the College of Information Science and Engineering, Hunan Normal University, Changsha, China. Her research interests include protocol optimization and protocol design in wide area networks and wireless local area networks.