Applied Intelligence (2023) 53:18125–18137
https://doi.org/10.1007/s10489-023-04449-z

HaarAE: an unsupervised anomaly detection model for IOT devices based on Haar wavelet transform

Xin Xie[1] · Xinlei Li[1] · Lei Xu[1] · Weiye Ning[1] · Yuhui Huang[1]

Accepted: 1 January 2023 / Published online: 24 January 2023

Abstract

Given the shortcomings of the existing anomaly detection methods based on IoT devices, including insufficient feature extraction, poor model fitting effect and low accuracy, this paper proposes an unsupervised IoT device traffic anomaly detection model called HaarAE, which introduces Haar wavelet transform to enhance the feature expression of original data and improve the model's ability to identify anomalies. The convolutional autoencoder was used to construct the network structure, the memory module is introduced to increase the reconstruction error, and the ConvLSTM layer was added to the encoder to extract the temporal characteristics of the data. The output of each layer of decoder is cascaded with the output of the corresponding ConvLSTM layer, so that the decoder can obtain more coding information of each layer to reconstruct the original data and enhance the fitting ability of the model. Experiments on public datasets and real traffic datasets indicate that compared to the mainstream unsupervised models, HaarAE improves the anomaly detection effect.

Keywords Internet of things · Unsupervised learning · Haar wavelet transform · HaarAE

1 Introduction

At present, the IoT (Internet of Things) has penetrated into all levels of social life and has been deeply applied in intelligent environment, personal and social fields [1]. While the IoT brings convenience to social production, it also easily leads to security risks that cannot be ignored. The existing method is to found abnormal behaviors or network attacks of devices by detecting network traffic of the IoT, so as to protect the security of the IoT to the maximum extent.

✉ Xinlei Li
lixinyang002@vip.qq.com

Xin Xie
xiexin@ecjtu.edu.cn

Lei Xu
isleixu@sina.com

Weiye Ning
niniye1998@163.com

Yuhui Huang
huiyuhyh@163.com

[1] School of Information Engineering,
East China Jiaotong University, Nanchang, China

With the development of intelligent devices, IoT traffic detection technology based on machine learning has been widely studied [2]. Traditional traffic detection methods, such as Bayesian, support vector machine [3], are mostly based on statistics, requiring experts to mark traffic data and extract statistically significant features, including transmission rate, byte change and time interval of network traffic. Kong et al. [4] proposed an abnormal traffic identification system based on multi-classification support vector machine, which can classify and identify various attack traffic and has good performance through the experiment of KDDCUP99 dataset. Shafiq et al. [5] used NetMate to extract features of data packets from HIT and MIMS datasets. Vu et al. [6] proposed a feature engineering technique to extract important attributes of network traffic by analyzing data packets and mine the correlation of data packets. Experiment results show that this method can significantly improve the identification accuracy of abnormal traffic and the calculation efficiency of the model. With the exponential growth of IoT device traffic data, the above methods will face huge challenges that are difficult to overcome to extract statistical features from massive data.

In recent studies, some researcher try to apply deep learning method to network traffic anomaly detection. Radford et al. [7] used the improved recursive neural network (RNN) to learn the computer network traffic

sequence, and the experiment proved that the model could detect malicious traffic patterns in the computer system. Zou [8] proposed a new method to identify network traffic by deep neural network, and improved the accuracy of classification results by combining convolutional neural network (CNN) and RNN. Experimental results show that the model has a great improvement in efficiency and reliability. However, these solutions are based on supervised learning and and require a large amount of labeled data. But, in actual scenarios, it is extremely difficult and time-consuming to obtain abnormal sample labels. Especially, with the rapid development of network attacks, the model needs to be retrained to detect new attacks, resulting in low detection efficiency of the model for unknown attacks.

In order to break through the bottleneck and difficulties of supervised learning, researchers turn their attention to semi-supervised/unsupervised learning methods. In recent years, unsupervised algorithms based on Auto-Encoder (AE) [9] have attracted extensive attention and in-depth research. Nguyen et al. [10] proposed a network framework based on variational AE, which can effectively detect and interpret various network traffic anomalies. Yisroel et al. [11] proposed a plug and play Network Intrusion Detection system (NIDS) named Kitsune, which extracted characteristic data from data stream using damping increment statistical method, and then used the core algorithm KitNet to detect abnormal traffic in real time. Realize online efficient detection of network attack under unsupervised learning. Akcay et al. [12] used GAN (Generative Adversarial Networks) to train an encoder model with jump connections, which could learn the normal distribution of sample. Experiments proved that this model had a high detection rate for images in different fields. Zenati et al. [13] achieved effective results by jointly training two sub-networks to capture normal data distribution. In general, anomaly detection based on AE is judged by the reconstruction loss of original sample and decoder output sample. Under normal circumstances, the reconstruction of the abnormal loss is bigger, the reconstruction of the normal loss is smaller, but the decoder refactoring effect is limited by a fixed length of latent vector, less available information in the decoding process. Therefore, in the IoT scenario with stronger data diversity and more data, the reconstruction loss of some normal samples is larger, while that of some abnormal samples is smaller, which will reduce the anomaly detection accuracy.

However, complex feature selection combined with well-designed neural network structure can help anomaly detection. In this paper, an unsupervised anomaly detection model named Haar-AE is proposed. Specifically, HaarAE firstly uses Haar wavelet transform to enhance the input features of the original data, and the wavelet transform can retain the characteristic information of the original data in time domain and frequency domain. Then the data is input into the convolution encoder, and the convolution results of each layer in the encoding stage are not only input into the next layer of convolution, but also into the ConvLSTM layer. Cascade the results of ConvLSTM and deconvolution of each layer in the decoding stage, and input the cascade results into the next deconvolution layer, so as to increase the information that can be captured by the decoder and strengthen the reconstruction effect of normal samples. At the same time, a memory module is added to the AE, and a small number of limited normal sample latent vectors are stored in the memory module.

The contributions and innovations are as follows:

1) An unsupervised IoT traffic anomaly detection model named HaarAE is proposed, through this method, the reconstruction error of the model to normal samples can be kept at a low level, and the reconstruction error of the model to abnormal samples can be kept at a high level.
2) The combination of Haar wavelet transform and AE is applied to traffic anomaly detection of IoT devices. Besides analyzing the timing characteristics of traffic data, the characteristics of frequency domain are also analyzed to obtain the periodicity and fluctuation information of traffic data. The remainder of this paper as follows. Section 2 describes the related work. Section 3 proposes the HaarAE model and introduces its structural design. Section 4 verifies the effectiveness of this method by experiments. Conclusions and prospective research directions are described in Section 5.

2 Related work and theories

2.1 Spectrum analysis

With the development of deep learning, although RNN [14], Convolutional Neural Network (CNN) [15] and other methods can make use of their network structure to autonomously learn the feature of samples. However, feature analysis still plays an significant role in obtaining desirable results from network models. Among them, spectrum analysis is one of the typical representative technologies, which analyzes the frequency properties of time series to find the hidden periodicity. It has been widely used in time series processing, acoustics, computer vision, biomedicine and other fields [16, 17]. According to Fourier theory [18], each time domain signal has the corresponding frequency domain signal, the traditional spectrum analysis is to transform time series signal from time domain to frequency domain, and reveals the time domain can be difficult to find the information. Livera et al. [19] proposed a

state space modeling framework based on Fourier transform for predicting complex seasonal time series, such as time series with high-frequency seasonality and dual calendar effect. Experimental results show that this framework can reduce the computational burden of maximum likelihood estimation and can effectively identify and extract seasonal features in time domain. Wang [20] proposed a face recognition algorithm based on fractional Fourier transform, which improved the robustness to illumination, noise and other factors and effectively improved the face recognition rate. Due to the natural defects of Fourier transform in the processing of discrete signals, it can no longer meet the actual needs. However, wavelet transform can retain information in time domain and frequency domain and process local similarity of signals and data, showing great advantages in feature extraction and data mining [21]. Zhao et al. [22] used wavelet transform to reveal the frequency domain information of univariate time series, and used different neural networks to simultaneously capture time-frequency feature and long-term trends. Furthermore, attention mechanism is used to blend local and global features, which improves the prediction accuracy of time series effectively. Yuan et al. [23] proposed a full-convolutional neural network based on wavelet transform to capture frequency domain information of multivariate time series through wavelet transform. Experimental results show that adding wavelet transform can effectively improve the detection effect of convolutional neural network. Ma et al. [24] proposed an anomaly detection method named WAGAN for industrial sensor, which removes noise and enhances data features by means of decomposition and reorganization of multi-level discrete wavelet transform. Attention mechanism is introduced into WAGAN model, thus improving the accuracy of anomaly detection. Zhang et al. [25] roposed a new unsupervised learning framework P^2GAN, which can map the input samples to Gaussian distribution factors through discriminators, so as to fully extract the true distribution information. The author verified the effectiveness of P^2GAN from both theoretical and experimental aspects. Hou et al. [26] proposed a framework called Divide and Assembly Anomaly Detection (DAAD), which interprets image reconstruction as a process of divide and assembly. And add memory module to adjust the reconstruction ability of the model. It is difficult for common network models to capture the frequency domain information of data, which leads to the failure of models to learn more data features. However, a full understanding of data flow feature of the IoT is a great significance for analyzing equipment anomalies and judging network attacks. Therefore, in the real environment of device traffic of the IoT, more advantages can be brought by using wavelet transform to expand data features.

2.2 Autoencoder

AE is an unsupervised neural network model, which is widely used in anomaly detection, data dryness, data dimensionality reduction, image repair, information retrieval and other fields. The AE can learn the hidden features of the input data, this process called encoding, and hidden features learned can reconstruct the original input data, this process called decoding. The basic AE consists of two parts, the first part is the encoder and the second part is the decoder, as shown in Fig. 1

In training phase, the encoder learns the nonlinear mapping between the original data space and latent vector H, and maps the high-dimensional data to the low-dimensional data to reduce the amount of data. The process can be expressed by the following formula:

$$H = f(x) = s(Wx + p)) \tag{1}$$

Among them, S represents nonlinear mapping function, W represents weight matrix. After the latent vector is obtained, the decoder learns the nonlinear mapping of the potential vector to the original input space to realize data

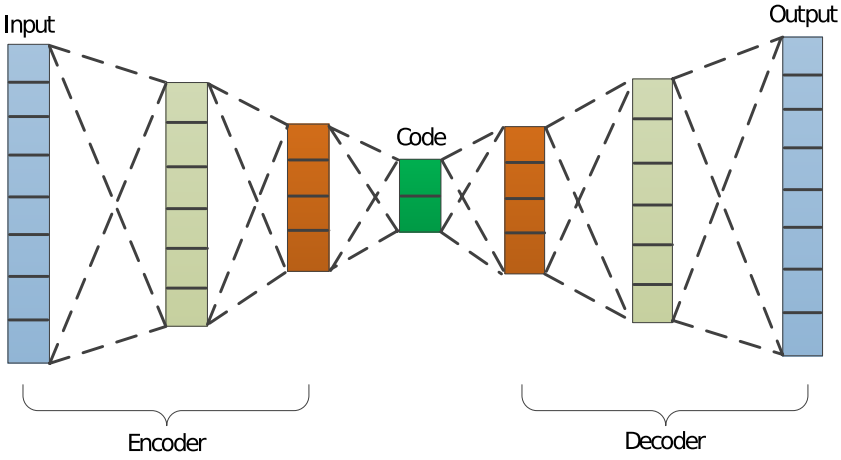


Fig. 1 Structure of AE

reconstruction. The formula used in the decoder phase is defined:

$$x' = g(x) = \alpha(WH + p) \tag{2}$$

The main work of the AE is to accurately reconstruct the input data, and its objective functions mainly include Mean Square Error (MSE) and Kullback-Leible(KL) divergence. MSE can be defined as:

$$MSE = \frac{1}{N}\sum_{i}^{N}(y_i - y_i')^2 \tag{3}$$

KL divergence describes the relative entropy between two probability distributions p and q. Its formula can be defined as:

$$D_{KL}(p||q) = -\sum_{x} p(x)\log(\frac{q(x)}{p(x)}) \tag{4}$$

Finally, the back-propagation algorithm is used to propagate the error back to the hidden layer, so as to optimize the loss function and model parameters.

2.3 Conclusion

The traffic of IoT devices belongs to time series data, so it has the same frequency domain information as other time series data. In our work, Haar wavelet transform is introduced to bring more diverse features to the device traffic data of the IoT, which effectively makes up for the deficiency of neural network in feature engineering. Build unsupervised AE network structures to avoid tagging large data sets, At the same time, aiming at the poor of model fitting effect, and the insufficiency of access to information, a memory module is added in AE to increase the reconstruction error of abnormal samples. In addition, a ConvLSTM layer is added, and the output of each layer of the decoder is cascated with the output of the corresponding ConvLSTM layer, which can not only capture the timing features of the data, but also enable the decoder to make full use of the coding information of each layer to reconstruct the original data, enhancing the fitting ability and detection accuracy of the model.

3 Unsupervised anomaly detection model based on Haar wavelet transform

3.1 Wavelet transform of original data

Wavelet transform can analyze signals in different frequency bands with different resolutions, so it can find the complex feature of signals. The idea is to use scale function to represent the original data and obtain the low frequency part of the data, which contains the global features of the original data. As the scale becomes larger, the scale function becomes more ambiguous to the original signal, and the difference from the original signal becomes larger and larger, so the wavelet function needs to be introduced to represent the difference. The wavelet function can obtain the high frequency part of the data, which contains the details of the data. Specifically, the wavelet transform decomposed the original data into approximate coefficient and detail coefficient under the action of wavelet function and scale function [27]. Figure 2 describes the process of wavelet transform. Given a set of raw traffic data $X = (x_1, x_2, ..., x_{N-1})$, N is the length of the data, given the scale function $\beta = \{\beta_1, \beta_2, \ldots, \beta_{N-1}\}$ and the wavelet function $\varepsilon = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{N-1}\}$, Next, for the given sequence of traffic data X, projection on the two functions β and ε, the approximate coefficients and detailed coefficients are obtained respectively, as shown in formula (5) and formula (6).

$$W_\beta(f_0, c) = (x, \beta_{f_0,k}) = \frac{1}{\sqrt{N}}\sum_{m=0}^{N-1} x[m]\beta_{f_0,k}[m] \tag{5}$$

$$W_\varepsilon(f, c) = (x, \varepsilon_{f,k}) = \frac{1}{\sqrt{N}}\sum_{m=0}^{N-1} x[m]\varepsilon_{f,k}[m] \tag{6}$$

W_β is the approximate coefficient, W_ε is the detail coefficient, c is the wavelet transform translation, f_0 and f is wavelet transform scale level. The detail coefficient reveals the variance of the data on different scales, and the approximate coefficient gets the smoothed average on that scale. Furthermore, the decomposition result of each

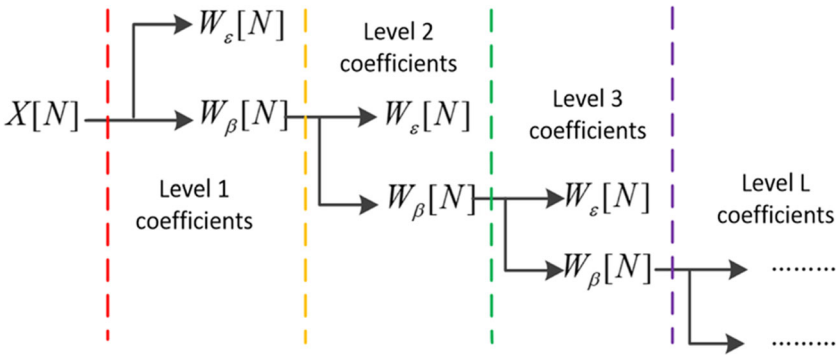


Fig. 2 Process of wavelet transform

level of wavelet transform is that the low frequency part obtained from the previous decomposition is decomposed into two parts, low frequency and high frequency. After l-level decomposition, the source signal X can be expressed by the following formula:

$$X = W_{\varepsilon_1} + W_{\varepsilon_2} + W_{\varepsilon_3} + \ldots + W_{\varepsilon_l} + W_{\beta_l} \tag{7}$$

Among them, $W_{\varepsilon_1} W_{\varepsilon_2} W_{\varepsilon_3} \ldots W_{\varepsilon_l}$ are the high-frequency signal decomposed from the first, second layer to the l th level, W_{β_l} is the low-frequency signal obtained by decomposition of the l th level. The detail coefficients and approximate coefficients obtained through wavelet transform can provide multi-scale data features for the model, which is difficult to be achieved by ordinary neural networks.

Yuan et al. [23] pointed out that as the smooth average of the original data, the approximate coefficient is easy to be learned by the CNN, and due to the non-orthogonality of the approximate coefficient, the additional input will bring redundant parameters to the convolution. calculate. In fact, for IoT device traffic data, the low-frequency information in the normal sample is still important, because it represents the normal pattern of the sample. Although the convolutional neural network can learn part of it, experiments show that good feature data cooperates with the neural network can bring better results.

3.2 Convolutional AE based on memory module

Figure 3 includes a wavelet transform module, a memory module and a ConvLSTM cascade module. First, for each input original data x_d, $d \in \{0, 1, 2 \ldots\}$, apply wavelet transform to decompose it to a specific level L to obtain feature information on different scales, where d is the data dimension, Formally, in scale level $l \in (1, \ldots, L)$, the detail coefficient obtained by wavelet transform is $W_\varepsilon = W_{\varepsilon_1}(l) + W_{\varepsilon_2}(l) + \ldots + W_{\varepsilon_d}(l)$, the approximate coefficient is $W_{\beta_1} = W_{\beta_1}(l) + W_{\beta_2}(l) + \ldots + W_{\beta_d}(l)$. Next, the original data, detail coefficients and approximation coefficients are put into an independent convolution decoder, in order to fully extract the feature in time domain and frequency domain. The input of each convolutional decoder is $x \in R^{D*N}$ or $W(l) \in R^{D*\frac{N}{2^c}}$, setting the kernel size of the first convolution layer to D forces the convolution encoder to merge all dimensions to capture the global correlation

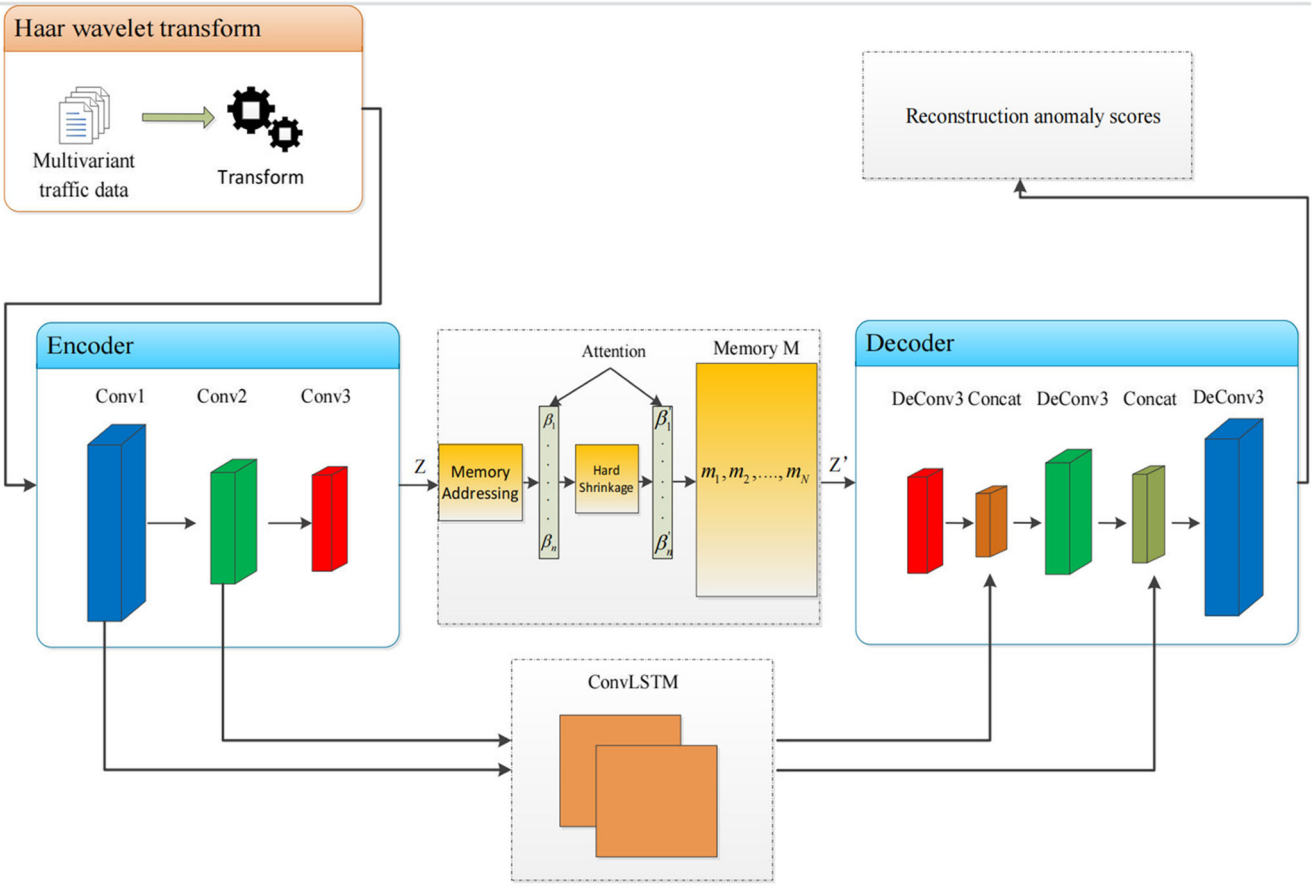


Fig. 3 The overall structure of the model

between dimensions. Unified definition X_{in} is input data, X_{out} is the output data, so the output of the convolution decoder can be expressed as:

$$X_{out} = f(A * X_{in} + b) \tag{8}$$

Where f represents nonlinear mapping function, $*$ is the convolution operation, A and b are the parameters learned in the convolution encoder. Generally, the encoder and decoder form a symmetric structure through the neural network layer of reverse stacking. For the input sample X, The encoder $f_{encoder}(\cdot) : x \rightarrow z$ encodes data to obtain hidden vector z, then the original data is restored through the decoder, and the reconstruction error between the input data and the restored data is used to judge whether there is an anomaly. However, in the actual environment, the data imbalance and difference lead to the strong AE generalization ability, and the reconstruction error of some abnormal samples is very small. In this paper, the memory module [28] structure is introduced after the encoder. and stores a small number of latent vectors that can represent normal sample distribution into the memory module. In the test stage, The coding result of the test sample is used to retrieve the most similar latent vector Z for reconstruction. Therefore, when a test sample is an abnormal sample, its reconstruction error will increase. In essence, the memory module stores a matrix with the size of N times m, where n represents the number of n samples and M represents the feature dimension of samples. The memory addressing module is used to calculate the attention weight of the latent vector β, as shown in formula (9):

$$\beta_i = \frac{e^{\cos(z,m_i)}}{\sum_{j=1}^{N} e^{\cos(z,m_j)}} \tag{9}$$

Where $\cos(\cdot,\cdot)$ is the cos distance similarity calculation function between any two vectors. m_i represents a storage item in memory. The specific formula of $\cos(\cdot,\cdot)$ is shown in (10).

$$cos(z, m_i) = \frac{z \cdot m_i^T}{\|z\|_2 \|m_i\|_2} \tag{10}$$

In practice, some exceptions are still well reconstructed, so hard shrinkage method is used to constrain β, as shown in formula (11).Where λ represents the set threshold, which is valid only when the attention weight is greater than this value, otherwise it is 0.

$$\beta_i' = \left\{ \begin{array}{c} \frac{\max(\beta_i - \lambda, 0)}{|\beta_i - \lambda| + \varepsilon} \cdot \beta_i, \beta_i > \lambda \\ 0, \beta_i \leq \lambda \end{array} \lambda \in \left[\frac{1}{N}, \frac{3}{N}\right] \right\} \tag{11}$$

Finally, the constrained attention weight and the basic implicit feature vector in the memory module are used to calculate z', as shown in formula (12).

$$z' = \beta \cdot M = \sum_{i=1}^{N} \beta_i' m_i \tag{12}$$

3.3 ConvLSTM cascaded structure

The main advantage of memory module is to increase the reconstruction error, since the hidden vector generated by encoder is fixed, the decoding effect of decoder is limited by the expression of hidden vector, and its performance may decrease as the sequence length increases. A cascade structure of ConvLSTM [29] is constructed to improve the decoder performance. As shown in Fig. 3, the output of each layer in the encoding stage is not only input to the next layer of convolution, but also input to a ConvLSTM layer. The result of each layer of deconvolution in the decoding stage is cascaded with the result generated by ConvLSTM and input to the next layer of deconvolution. Given the output $X_{T,l}$ of the l-th convolutional layer, and the previous hidden state $Z_{T-1,l}$ So the current hidden state is updated to $Z_{T,l} = ConvLSTM(X_{T,l}, Z_{T-1,l})$ in the ConvLSTM layer. The specific formula for the ConvLSTM layer is as follows:

$$i_{T,l} = \sigma(b_{i,l} + W_{xi,l} * X_{T,l} + W_{zi,l} * Z_{T-1,l} + W_{ci,l} \circ C_{T-1,l}) \tag{13}$$

$$f_{T,l} = \sigma(b_{f,l} + W_{xf,l} * X_{T,l} + W_{zf,l} * Z_{T-1,l} + W_{cf,l} \circ C_{T-1,l}) \tag{14}$$

$$C_{T,l} = f_{T,l} \circ C_{T-1} + i_{t,l} \circ \tanh(b_{c,l} + W_{xc,l} * X_{T,l} + W_{hc,l} * Z_{T-1,l} \tag{15}$$

$$o_{T,l} = \sigma(b_{o,l} + W_{xo,l} * X_{T,l} + W_{zo,l} * Z_{T-1,l} + W_{co,l} \circ C_{T-1}) \tag{16}$$

$$Z_{T,l} = o_{t,l} \circ \tanh(C_{T,l}) \tag{17}$$

In this paper, some changes are made to the symbols of formulas (9), (10), (11), (12) and (13) to make them consistent in the whole paper. "$\circ$" is known as Hadamard product. $W_{xi,l}$, $W_{zi,l}$, $W_{ci,l}$, $W_{xf,l}$, $W_{zf,l}$, $W_{cf,l}$, $W_{xc,l}$, $W_{hc,l}$, $W_{xo,l}$, $W_{zo,l}$, $W_{co,l}$ are the parameters of ConvLSTM, all $X_{T,l}$, $C_{T,l}$, $Z_{T-1,l}$, $i_{T,l}$, $o_{T,l}$, $f_{T,l}$ are tensors in three dimensions. Finally, in order to reconstruct the original data, detail coefficients and approximation coefficients, the latent

vector output by the encoder needs to be decoded. We design the deconvolution operation as follows:

$$\bar{x}_{T,l-1} = f(W_{T,l} \otimes Z_{T,l} + b_{T,l}), l = 3 \tag{18}$$

Where f is the same activation function as the convolution encoder, "$\otimes$" is the deconvolution operation, and W, b are the learning parameters of the convolution decoder. In order to be able to cascade with ConvLSTM layer, we updated the deconvolution operation:

$$\bar{x}_{T,l-1} = f(W_{T,l} \otimes [Z_{T,l} \oplus \bar{x}_{T,l}] + b_{T,l}), l = 1, 2 \tag{19}$$

"$\oplus$" is a connection operation. Specifically, when "l" is equal to 3, the last layer of deconvolution only accepts information from hidden vectors, when l =1,2, The deconvolution layer will accept not only the deconvolution output $\bar{x}_{T,l-1}$ from the previous layer, but also the output $Z_{T,l}$ from the ConvLSTM, and connect these two parts, further input to the next deconvolution layer, so the decoder is able to combine the output at different deconvolution layers and ConvLSTM layers. The ConvLSTM layer captures data features at different time scales. In this way, the decoder relies not only on hidden vector features, but also on features provided by ConvLSTM layer, which enables the model to comprehensively utilize information of different scales to reconstruct data, effectively utilizing the feature extraction capability of ConvLSTM and improving anomaly detection performance. The reconstruction loss of the HaarAE is defined as formula (20). Finally, anomaly detection is carried out based on the reconstruction error of original data, detail coefficient and approximate coefficient. The specific results will be described in detail in the next section.

$$loss = \left\| x - \bar{x} \right\|_2 + \sum_{l=1}^{L} \left\| w_\beta(l) - \bar{w_\beta}(l) \right\| + \sum_{l=1}^{L} \left\| w_\varepsilon(l) - \bar{w_\varepsilon}(l) \right\| \tag{20}$$

4 Experiment

This section describes the experimental evaluation of the HaarAE model, this paper uses a common benchmark dataset and a collection of real IoT traffic dataset to evaluate the HaarAE. The aim is to answer the following questions.

Question 1: Is the anomaly detection performance of IoT device traffic superior to mainstream unsupervised methods?

Question 2: How does each component of HaarAE affect its performance (ablation study)?

Question 3: Is HaarAE more robust to input noise? HaarAE uses the Keras [30] framework, and the Adam [31] with a learning rate of 0.001 is used to optimize the model.

4.1 Indicators

In the experimental part, the anomaly detection performance of the proposed model was evaluated based on three evaluation indexes. The details are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{21}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{22}$$

$$F1 = 2 * \frac{\text{Pre*Rec}}{\text{Pre} + \text{Rec}} \tag{23}$$

In this paper, the traffic sample with attack as positive sample and the normal sample as negative sample. In formula (21), (22) and (23), TP is the number of samples that are actually attack samples and are predicted to be attack samples, FN is the number of samples that are actually attack samples but predicted to be normal samples, FP is the number of normal samples but predicted to be attack samples, and TN is the number of normal samples and predicted to be normal samples. Precision indicates the predicted correct value in the number of attack samples predicted, and recall is used to evaluate whether all attack samples are predicted by the percentage of coverage. However, it is difficult to objectively describe the performance of the model with precision and recall alone, therefore, F1 value is added, which is the harmonic mean of the two values.

4.2 Dataset

This paper uses the following two datasets: KDDCUP99, Comprehensive data.

1. KDDCUP99: KDDCUP99 can be obtained in the UCI repository. In KDDCUP99, each sample in the training dataset contains 41 fixed feature attributes and one label.
2. Comprehensive data: the comprehensive data is composed of the real IoT device traffic data collected by Ayyoob [32]. They collected 16 days of data packets from the test platform, including benign and attack traffic. This paper collates these real traffic data and makes a balance processing.

Details about the dataset are shown in Table 1.

Table 1 Statistics of the datasets

Dataset	Instances	Anomaly ratio
KDDCUP99	494021	0.20
Comprehensive data	732746	0.20

4.3 Baseline methods

This paper uses several mainstream unsupervised learning methods as the baseline method

1. OC-SVM: OC-SVM is a popular kernel based anomaly detection method.
2. AE: AE is an unsupervised learning algorithm. It compresses the input into a latent spatial representation, then reconstructs the output through the representation. It is mainly used for data dimensionality reduction or feature extraction.
3. DCN: Deep clustering network (DCN) is a most advanced clustering algorithm, which adjusts the performance of automatic encoder through k-means.
4. DAGMM [33]: A depth autoencoder Gaussian mixture model for unsupervised anomaly detection, which organically combines the dimension reduction process and density estimation process for end-to-end joint training, and avoids the local optimization of the model due to the independence of two steps.
5. MemAE [28]: The MemAE adds a memory module to the AE. In the training stage, the content in the memory module is updated to construct the prototype elements of normal samples.
6. Kitsune [11]: A network intrusion detection system uses the damping increment statistical method to extract the characteristic data from the data flow, and then uses the core algorithm kitnet to detect the abnormal traffic in real time, so as to realize the online and efficient detection of network attacks under unsupervised learning.
7. InterFusion [34]: A multi-dimensional time series unsupervised detection method, which can simultaneously model the dependence between different indicators of multi-dimensional time series and the dependence on time sequence. In addition, in order to answer question 2, the following variants of HaarAE are used as a baseline to demonstrate the impact of a single component in HaarAE on the accuracy of the model.
8. HaarAE-HN: HaarAE-HN (HaarAE-HaarNone) model does not process the original data by wavelet transform, but retains the cascade structure and memory module.
9. HaarAE-CN: HaarAE-CN (HaarAE-CascadeNone) adopts the traditional autoencoder structure to reconstruct the data, without the cascade structure of ConvLSTM, and retains the memory module and processing of original data by wavelet transform .

4.4 Experimental evaluation

1) *Unsupervised anomaly detection performance experiment*

The ultimate goal of well-designed Feature Engineering and appropriate network model is to achieve higher accuracy in anomaly detection of IoT devices. In this experiment, the training dataset containing only normal samples is used to construct HaarAE model that can capture the normal mode. 70% of the normal samples are used as the training set and 5% of the normal samples are used as the verification set. 25% of the normal samples are used as the test set.

The level of wavelet transform is the key factor to expand data features. Therefore, firstly, the relationship between model performance and wavelet transform level is studied. In the face of huge data, high-level wavelet transform will produce high time overhead. Therefore, in the case of performance permitting, this paper adopts four levels of wavelet transform, represented by HaarAE-1, HaarAE-2, Haa-rAE-3 and HaarAE-4 respectively. HaarAE-1 means that only one layer of wavelet transform is performed on the original data, i.e. $X = W_{\varepsilon_1} + W_{\beta_1}$, HaarAE-2 represents the two-layer wavelet transform of the original data, i.e. $X = W_{\varepsilon_1} + W_{\varepsilon_2} + W_{\beta_2}$, The same goes for HaarAE-3 and HaarAE-4. The effect of wavelet transform level on model performance is shown in Table 2. It can be seen from the table that when the wavelet transform level is Four, the best anomaly detection effect is obtained on KDDCUP99 dataset and comprehensive dataset, which are recall = 0.9975/0.9445, precision = 0.9758/0.9599 and F1 = 0.9863/0.9522 respectively. Therefore, the wavelet transform level used in the experimental part of this paper is 4, which can meet most anomaly detection requirements.

Next, for question 1 and question 2, we evaluate the performance of the model on two datasets. In the training phase, only normal samples are used as the training set and verification set. In the test phase, the reconstruction MSE of training set samples and test set samples are compared to determine the anomaly. In the experiment, 1D-CNN (One dimensional convolutional neural network) is used as the basic structure of encoder and decoder. Three groups of symmetrical 1D-CNN layers are set, and the ReLu layer [35] is used as the activation function. At the end of the encoder, we further set the global average pooling layer [36]. Compared with the full connection layer, the global average pooling layer can prevent over fitting by

Table 2 Wavelet transform level comparison

Level	Precision	Recall	F1
	KDDCUP99		
HaarAE-4	0.9975	0.9758	0.9863
HaarAE-3	0.9818	0.9569	0.9727
HaarAE-2	0.9775	0.9684	0.9727
HaarAE-1	0.9821	0.9488	0.9652
	Comprehensive data		
HaarAE-4	0.9445	0.9599	0.9522
HaarAE-3	0.9429	0.9033	0.9227
HaarAE-2	0.9388	0.8685	0.9023
HaarAE-1	0.9437	0.7351	0.8264

reducing the total number of parameters in the model. The specific results are shown in Table 3.

Table 3 reports recall, precision and F1 of HaarAE and other baseline models on two datasets. Generally speaking, the model proposed in this paper achieves the highest recall, precision, and F1 on two datasets, but the effect of the two variant models of HaarAE is not as effective as the latest unsupervised model, which shows that it is very important to combine effective feature extraction with appropriate network model. It is worth noting that the KDDCUP is not a IoT traffic dataset, but the model also worked pretty well, Therefore, the proposed model has certain universality in the field of traffic anomaly detection.

2) *Effect of cascade structure on model accuracy*

The purpose of cascade structure is to provide more information for the decoder, so that the decoder can reconstruct the normal sample better. Therefore, This paper studied the performance of the model under four different cascade structures. The schematic diagram of the network is shown in Fig. 4.

Cascade-Full refers the complete cascade structure in the HaarAE model. Cascade-None refers the AE model of none cascade structure. Cascade -None is consistent with HaarAE-CN. Cascade-1 (Conv2-Deconv1) refers that it is connected to ConvLSTM layer after conv2 layer and cascaded with the output of deconv1 layer; Cascade-2 (Conv1-Deconv2) refers that it is connected to ConvLSTM layer after conv1 layer and cascaded with the output of deconv2 layer. For these four network structures, this paper conducted experiments, as shown in Table 4.

Table 4 shows the recall, precision and F1 of four cascade structures under two datasets. From the table, it can be seen that Cascade-Full has the best result, The result of Cascade-1 (Conv2-Deconv1) and Cascade-1 (Conv1-Deconv2) is between Cascade-Full and Cascade-None, which is consistent with the result that we expect. This shows that the position of the cascade layer has little effect on the model performance, but increasing the number of ConvLSTM layers can improve the effect of anomaly detection. In order to further observe the effect of cascade structure and memory module on the reconstruction of normal samples and abnormal samples, the reconstruction errors of different models are counted on comprehensive data, as shown in Figs. 5 and 6.

It can be seen that adding ConvLSTM layer and memory module can greatly enlarge the reconstruction error between abnormal samples and normal samples, and the fluctuation range of normal sample reconstruction error is also very small. It shows that HaaAE model can reconstruct most data in normal samples.

3) *HaarAE robustness experiment*

In the actual environment, the traffic data of IoT devices usually contain noise. Therefore, whether the anomaly detection algorithm has strong robustness to the input noise is of great significance. For problem 3,

Table 3 Precision, recall, and F1 from HaarAE and the baseline methods

Method	Precision	Recall	F1
	KDDCUP99		
OC-SVM	0.7457	0.8523	0.7954
AE	0.9355	0.9327	0.9341
DCN	0.7831	0.7697	0.7763
DAGMM	0.9441	0.9296	0.9368
MemAE	0.9655	0.9627	0.9641
Kitsune	—	—	—
InterFusion	0.9668	0.9538	0.9604
HaarAE-HN	0.9543	0.9143	0.9328
HaarAE-CN	0.9361	0.9280	0.9321
HaarAE	0.9975	0.9758	0.9863
	Comprehensive data		
OC-SVM	0.6526	0.4047	0.4496
AE	0.6683	0.7801	0.7431
DCN	0.7119	0.6944	0.7011
DAGMM	0.6322	0.6871	0.7089
MemAE	0.8293	0.8353	0.8343
Kitsune	0.8113	0.9261	0.8674
InterFusion	0.9011	0.9133	0.9277
HaarAE-HN	0.7921	0.7283	0.7943
HaarAE-CN	0.8940	0.8747	0.8843
HaarAE	0.9445	0.9599	0.9522

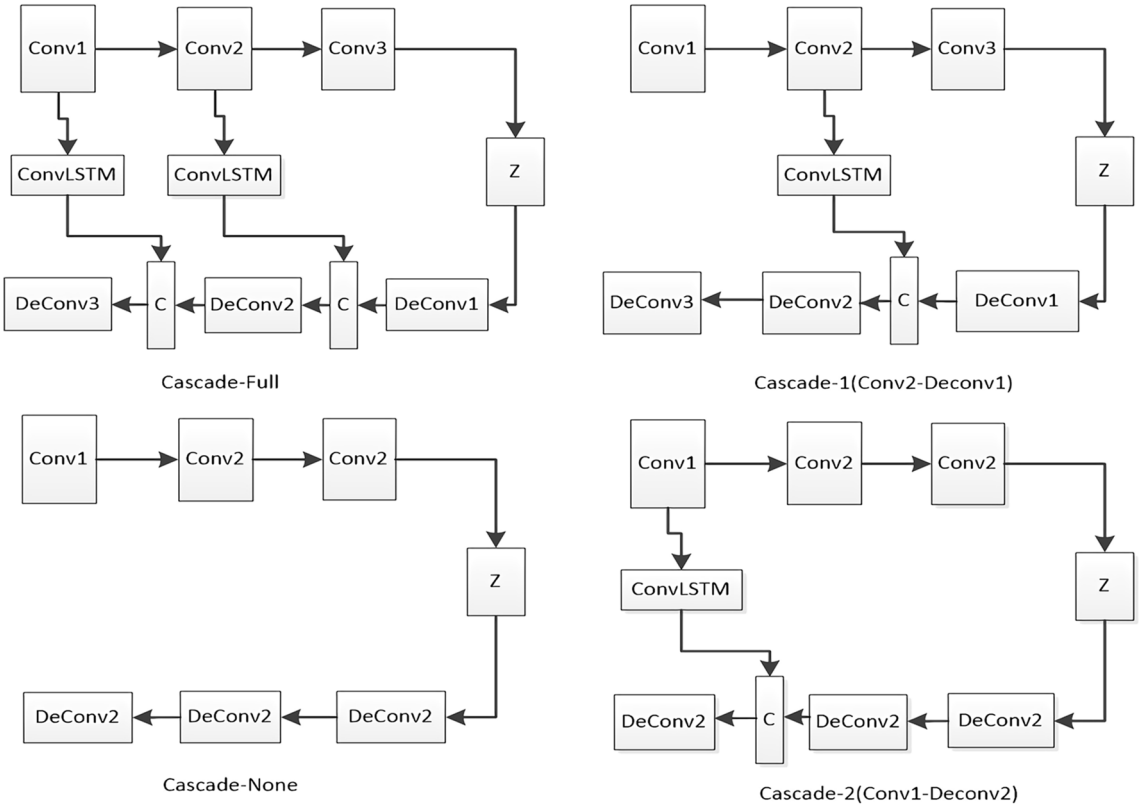


Fig. 4 Model sketch of different cascade structures

in order to study the robustness of HaarAE in anomaly detection, in training stage, we add different proportions of abnormal samples to the training data containing only normal samples to simulate noise, so that it can be mixed with normal samples for training. The detailed results of robustness experiment are shown in Table 5.

Table 5 reports the precision, recall and F1 values of HaarAE, HaarAE-HN, OC-SVM and DCN on KDDCUP99 respectively. It can be observed that the precision, recall and F1 of the four models decrease with the increase of anomaly ratio, which means that the noise data will have an impact on the performance of anomaly detection, and the impact is negatively correlated. It is worth noting that although the performance of HaarAE model decreases when facing noise data, but it still maintains a high anomaly

Table 4 Effects of different cascade structures on Model

Method	Precision	Recall	F1
	KDDCUP99		
Cascade-Full	0.9975	0.9758	0.9863
Cascade-None	0.9361	0.9280	0.9321
Cascade-1(Conv2-Deconv1)	0.9721	0.9649	0.9685
Cascade-2(Conv1-Deconv2)	0.9762	0.9649	0.9705
	Comprehensive data		
Cascade-Full	0.9445	0.9599	0.9522
Cascade-None	0.8940	0.8747	0.8843
Cascade-1(Conv2-Deconv1)	0.9266	0.9109	0.9224
Cascade-2(Conv1-Deconv2)	0.9260	0.9099	0.9182

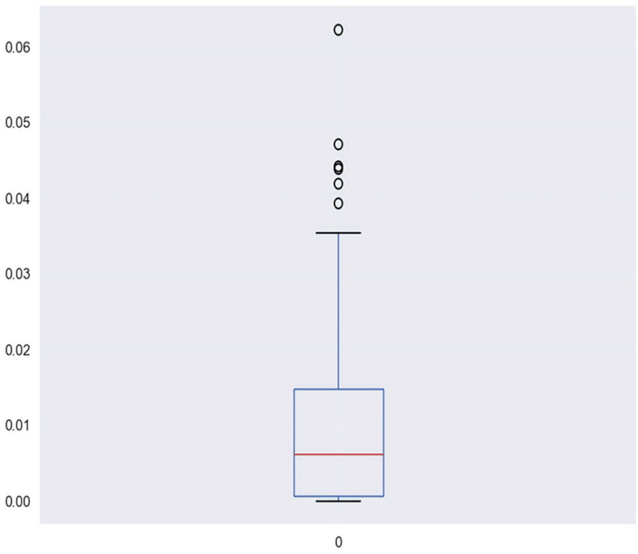


Fig. 5 Reconstruction error of Ordinary AE

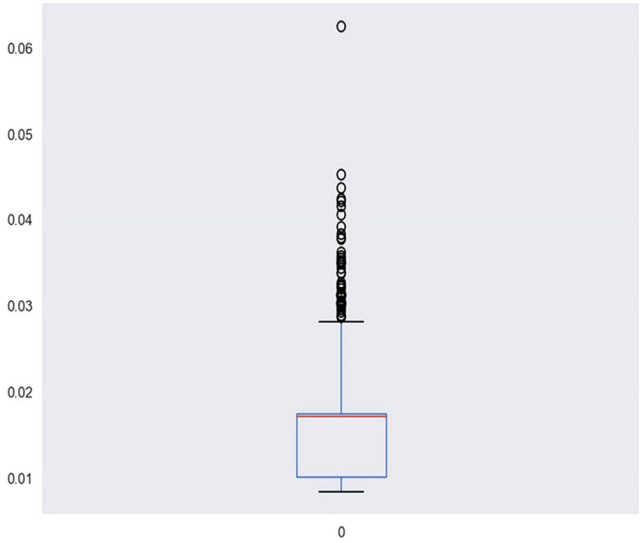


Fig. 6 Reconstruction error of HaarAE

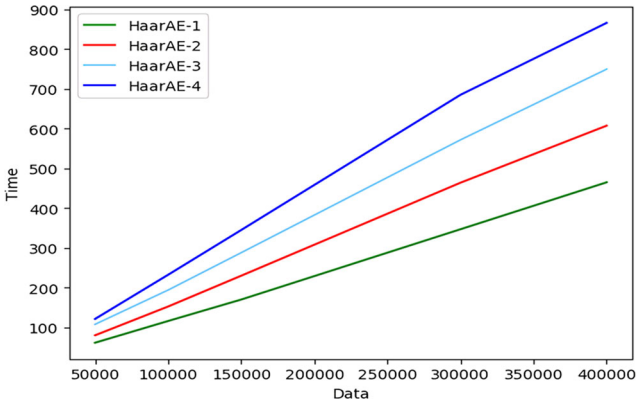


Fig. 7 Time overhead at different wavelet transform levels

detection performance compared with other models. Especially the results of Haar-HN and HaarAE show that the data features obtained by wavelet transform can enhance the robustness of the model.

4) *Some problems of HaarAE*

However, HaarAE performs wavelet transform when facing a large amount of data, which will bring a certain amount of time overhead. Figure 7 shows a comparison of time overhead at different wavelet transform level. It can be clearly found that the level of wavelet transform is positively correlated with time overhead, which inevitably increases the time cost of HaarAE. Therefore, in future work. it is not necessary to perform wavelet transform on all data, and also the same anomaly detection performance can be achieved, further solve the problem of time overhead.

Table 5 Anomaly detection results on contaminated training data from KDDCUP99

Ratio	Precision	Recall	F1	Precision	Recall	F1
	HaarAE			OC-SVM		
1%	0.9890	0.9541	0.9686	0.7129	0.6785	0.6953
2%	0.9881	0.9484	0.9708	0.6668	0.5207	0.5847
3%	0.9415	0.9318	0.9304	0.6393	0.4470	0.5261
4%	0.9029	0.8835	0.8931	0.5991	0.6785	0.4589
5%	0.8673	0.8424	0.8547	0.1155	0.3369	0.1720
	HaarAE-HN			DCN		
1%	0.9399	0.9047	0.9239	0.7611	0.7585	0.7598
2%	0.9260	0.8719	0.8981	0.7424	0.7380	0.7402
3%	0.9240	0.8316	0.8742	0.7293	0.7163	0.7228
4%	0.8997	0.8178	0.8582	0.7106	0.6971	0.7037
5%	0.8375	0.8090	0.8230	0.6893	0.6763	0.6827

5 Conclusion

In order to solve the problems existing in traffic anomaly detection of IoT equipment, an unsupervised anomaly detection model called HaarAE is proposed in this paper. First, feature extraction of original data is carried out by using Haar wavelet transform to capture the features in time and frequency domain and enhance the feature expression of original data. Next, the original data is input into the model together with the data from the wavelet transform. In the encoding phase, the ConvLSTM layer is added to capture the time feature of data. In the decoding phase, the ConvLSTM layer results are cascaded with each layer of the decoder to provide more effective feature information for the decoder and improve the fitting ability of the model. Finally, a memory module is added between the encoder and decoder to increase the reconstruction error of abnormal samples.The model has been validated on different types of datasets for its effectiveness and versatility. Experimental shows that HaarAE has a certain versatility while improving the effect of anomaly detection.

Acknowledgements This work is supported by the National Natural Science Foundation of China, under Grant No. 62162026, the Science and Technology Key Research and Development Program of Jiangxi Province, under Grant No. 20202BBEL53004 and Science and Technology Project supported by education department of Jiangxi Province, under Grant No. GJJ210611.

Declarations

Conflict of Interests All the authors do not have any possible conflicts of interest.

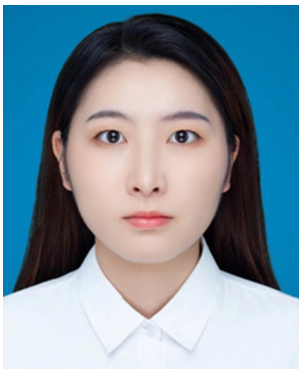
References

1. Kaur H, Singh G, Minhas J (2013) A review of machine learning based anomaly detection techniques. Int J Comput Appl Technol Res 2:185–187
2. Shon T, Moon J (2007) A hybrid machine learning approach to network anomaly detection. Inform Sci 177:3799–3821
3. Shon T, Kim Y, Lee C, Moon J (2005) A machine learning framework for network anomaly detection using svm and ga. In: Information assurance workshop, IAW05. Proceedings from the sixth annual IEEE SMC, pp 176–183
4. Kong L, Huang G, Wu K (2017) Identification of Abnormal Network Traffic Using Support Vector Machine. In: 2017 18th International conference on parallel and distributed computing, applications and technologies(PDCAT). IEEE Computer Society
5. Shafiq M, Yu X, Wang D (2018) Network Traffic Classification Using Machine Learning Algorithms[J]. Adv Intell Syst Comput 686:621–627
6. Vu L, Hoang VT, Quang UN et al (2018) Time series analysis for encrypted traffic classification: a deep learning approach. In: 18th International symposium on communications and information technologies(ISCIT), pp 121—126
7. Radford BJ et al (2018) Network traffic anomaly detection using recurrent neural networks
8. Zou Z, Ge J, Zheng H et al (2018) Encrypted traffic classification with a convolutional lonG Shorterm memory neural network. IEEE 20th International Conference on High Performance Computing and Communications
9. Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks. In: Advances in neural information processing systems, pp 153–160
10. Nguyen QP, Lim KW, Divakaran DM, Low KH, Chan MC (2019) Gee: a gradient-based explainable variational autoencoder for network anomaly detection. IEEE
11. Mirsky Y, Doitshman T, Elovici Y, Shabtai A (2018) Kitsune: an ensemble of autoencoders for online network intrusion detection
12. Akcay S, Atapour-Abarghouei A, Breckon TP (2018) Ganomaly: Semisupervised anomaly detection via adversarial training. arXiv:1805.06725
13. Zenati H, Foo CS, Lecouat B, Manek G, Chandrasekhar VR (2018) Efficient gan-based anomaly detection, arXiv:1802.06222
14. Mikolov T, Karafiát M, Burget L, Černock'y J, Khudanpur S (2010) Recurrent neural network based language model. In: Eleventh annual conference of the international speech communication association
15. He Y, Zhao J (2019) Temporal convolutional networks for anomaly detection in time series. Journal of Physics: Conference Series
16. Chen Y, Jiang H, Li C, Jia X, Ghamisi P (2016) Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Trans Geosci Remote Sens 54:6232–6251
17. Fujieda S, Takayama K, Hachisuka T (2018) Wavelet Convolutional Neural Networks. arXiv:1805.08620
18. Glafcos (2006) Fourier analysis. English. China Machine Press
19. De Livera AM, Hyndman RJ, Snyder RD (2011) Snyder Forecasting time series with complex seasonal patterns using exponential smoothing. J Am Stat Assoc 106:1513–1527
20. Wang Y (2015) Face recognition based on fractional Fourier transform. Doctoral discrimination, Zhengzhou University
21. Li T, Li Q, Zhu S, Ogihara M (2002) A survey on wavelet applications in data mining. ACM SIGKDD Explorations Newsl 4:49–68
22. Zhao Y, Shen Y, Zhu Y, Yao J (2018) Forecasting wavelet transformed time series with attentive neural networks. In: 2018 IEEE international conference on data mining (ICDM). IEEE pp 1452–1457
23. Yuan B, Chen W, Fei J, Long M, Yuan L (2019) Waveletfcnn: a deep time series classification model for wind turbine blade icing detection. Machine Learning
24. Ma B, Jia J, Dong G, Hong Z, Lu G (2021) Wagan: industrial control sensor data anomaly detection method based on wavelet transform and attention mechanism. Journal of Chinese Computer Systems
25. Zhang X, Cheng Z, Zhang X, Liu H (2021) Posterior promoted GAN with Distribution discriminator for unsupervised image synthesis. IEEE Conference on Computer Vision and Pattern Recognition(CVPR)
26. Hou J, Zhang Y, Zhong Q et al (2021) Divide-and-assemble: learning block-wise memory for unsupervised anomaly detection. IEEE International Conference on Computer Vision(ICCV)
27. Liu C-L (2010) A tutorial of the wavelet transform. Taiwan, National Taiwan University(NTUEE) Press
28. Gong D, Liu L, Le V et al (2020) Memorizing normality to detect anomaly: memory-augmented deep Autoencoder for unsupervised anomaly detection. In: 2019 IEEE/CVF international conference on computer vision (ICCV). IEEE
29. Zhang C, Song D, Chen Y, Feng X, Lumezanu C (2018) A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. 10.48550/arXiv:1811.08055
30. Chollet, Francois et al (2015) Keras, GitHub. https://github.com/keras-team/keras
31. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
32. Hamza A, Gharakheili HH, Benson TA, Sivaraman V (2019) Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity, the 2019. ACM Symposium, pp 36–48
33. Song Q (2018) Deep Autoencoding gaussian mixture model for unsupervised anomaly detection. ICLR
34. Li Z, Zhao Y, Han J, et al (2021) Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. KDD'21: proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining
35. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 807–814
36. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929

Xin Xie received his Master's degree from Nanchang University of Control Theory and Control Engineering in Nanchang. He is a Professor in the School of Information Engineering, East China Jiaotong University. His research interests are computer networks, information security and machine vision.

Weiye Ning received a B.S. degree from Jiangxi Agricultural University in 2019, now she is pursuing an MA.Eng degree in East China Jiaotong University. Her research interests include information security and anomaly detection.

Xinlei Li received a B.E. degree from Qingdao University in 2019. He is currently pursuing an MA.Eng degree in computer technology with the School of Information Engineering, East China Jiaotong University, His research interests include information security and machine learning.



Yuhui Huang received a B.E. degree from Jiangxi Science and Technology Normal University in 2020, where she is currently pursuing an MA.Eng degree East China Jiaotong University. Her research interests include machine learning.

Lei Xu received a B.E. degree from Changsha University in 2019. He is currently pursuing an MA.Eng degree in computer technology with the School of Information Engineering, East China Jiaotong University. His research interests include computer vision, deep learning and machine learning.