Hindawi
Security and Communication Networks
Volume 2020, Article ID 6656066, 11 pages
https://doi.org/10.1155/2020/6656066

Research Article

An Unsupervised Learning-Based Network Threat Situation Assessment Model for Internet of Things

Hongyu Yang[1], **Renyun Zeng**[1], **Fengyan Wang**[1], **Guangquan Xu**[2] **and Jiyong Zhang**[3]

[1]*School of Computer Science and Technology, Civil Aviation University of China, 300300 Tianjin, China*
[2]*College of Intelligence and Computing, Tianjin University, 300350 Tianjin, China*
[3]*School of Computer and Communication Science, Swiss Federal Institute of Technology in Lausanne, CH-1015 Lausanne, Switzerland*

Correspondence should be addressed to Hongyu Yang; hyyang@cauc.edu.cn

Received 16 October 2020; Revised 28 October 2020; Accepted 16 November 2020; Published 28 November 2020

Academic Editor: Wenjuan Li

With the wide application of network technology, the Internet of Things (IoT) systems are facing the increasingly serious situation of network threats; the network threat situation assessment becomes an important approach to solve these problems. Aiming at the traditional methods based on data category tag that has high modeling cost and low efficiency in the network threat situation assessment, this paper proposes a network threat situation assessment model based on unsupervised learning for IoT. Firstly, we combine the encoder of variational autoencoder (VAE) and the discriminator of generative adversarial networks (GAN) to form the V-G network. Then, we obtain the reconstruction error of each layer network by training the network collection layer of the V-G network with normal network traffic. Besides, we conduct the reconstruction error learning by the 3-layer variational autoencoder of the output layer and calculate the abnormal threshold of the training. Moreover, we carry out the group threat testing with the test dataset containing abnormal network traffic and calculate the threat probability of each test group. Finally, we obtain the threat situation value (TSV) according to the threat probability and the threat impact. The simulation results show that, compared with the other methods, this proposed method can evaluate the overall situation of network security threat more intuitively and has a stronger characterization ability for network threats.

1. Introduction

In recent years, the application of various emerging network technologies such as big data, blockchain, artificial intelligence, and other technologies in the field of Internet of Things (IoT) has brought about more and more convenience to people in many fields. At the same time, because of the connection with the Internet, the IoT devices are also vulnerable to more network threats [1], which will result in malicious attacks on physical devices. Reference [2] indicated that cyberphysical systems (CPSs) are vulnerable to traditional network threats, so the entire IoT system and the security and privacy of users are facing a huge threat. IoT devices and applications play an increasingly important role in critical infrastructure and everyday life; recent security incidents show that any successful attack will seriously hinder economic development and even endanger the safety of human life.

Because the IoT devices and applications are connected to the Internet, they are vulnerable to a variety of network attacks, which leads to important information leakage and even allows attackers to obtain permission to operate these devices. The authors of [3, 4] applied encryption algorithm in oblivious RAM to ensure the information security of storage devices. The IoT devices that are attacked by the network may have the management rights of the database stolen. To ensure the privacy and security of the database, the authors of [5, 6] proposed encryption algorithms to prevent the leakage of important information. However, in the face of a large number of complex network attacks, it is necessary to ensure network information security from a more comprehensive perspective.

To strengthen the construction of the network security defense system and deal with the emerging new threat attacks in the IoT network environment effectively, the stable and efficient network threat situation assessment (NTSA) method has become an important research topic. The NTSA evaluates the whole degree of security threats suffered by the IoT network system to analyze the situation of network attack and master the overall security situation of the network. NTSA can evaluate the current network security situation for IoT from a more comprehensive perspective and provide reliable information for network managers to make decision analysis and to minimize the loss that is caused by network threats [7]. However, in the past several years, the network has faced a large number of multisource threat attacks, which poses a huge threat to individuals and enterprises. The traditional network threat situation assessment method has the shortcomings of high modeling cost, low efficiency, and long cycle, which cannot make real-time and effective network security situation assessment.

To evaluate the network threat situation effectively in a multisource data environment of IoT, this paper proposes an unsupervised learning-based network threat situation assessment model for IoT. The contributions of this paper are as follows:

(1) To reduce the damage of network threats to IoT applications and devices, an unsupervised learning-based network threat situation assessment model was proposed. This model can reflect the current network situation of IoT effectively and provide decision support to network managers.

(2) This paper selects multisource heterogeneous network threat data to simulate the threats that IoT will be confronted with and calculate the threat situation value for the network threat situation assessment of IoT.

(3) The simulation results show that, compared to traditional models, this proposed method can evaluate the overall situation of network threats more intuitively and effectively for IoT.

1.1. Organization. The remainder of this paper is organized as follows. In Section 2, we present related works. Section 3 describes our proposed unsupervised network in detail. In Section 4, we propose our network threat situation assessment framework and the quantitative assessment process of the network threat situation in detail. Section 5 reports the experiments and the comparisons with other methods and, in the end, the conclusion is placed in Section 6.

2. Related Works

Assessment methods based on the mathematical model as applied to one of the earliest methods in network threat situation assessment and on account of its features such as being simple and easy to implement are widely used. Yang et al. [8] proposed a cloud computing risk assessment model that used the Markov chain (MC) model to describe the random risk environment and measured the risk value through information entropy (IE). Wang et al. [9] combined the analytic hierarchy process (AHP) with the hierarchical model of situational assessment and integrated the fuzzy results of multisource equipment with D-S evidence theory to solve the problem of single information source and large deviation of accuracy. Because the evaluation method based on the mathematical model is greatly influenced by subjective factors and there is no objective and unified standard definition variable, it is usually unable to achieve relatively perfect evaluation results.

Assessment methods based on probability and knowledge reasoning usually take advantage of the statistical characteristics of prior knowledge and combine with expert knowledge and experience database to build a model and then evaluate the threat situation by adopting logical reasoning. Sallam [10] identified potential network threats through fuzzy logic technology based on fuzzy reasoning (FR) engine and evaluated network security risks according to the attacker's overall capability, the overall probability of attack success, and the impact of the attack on three subfuzzy reasoning systems. Wen et al. [11] conducted a quantitative assessment of network security situation by fusing information sources with graded Naive Bayes classifier. These methods fuse various security assessment indicators in combination with the characteristics of mathematical statistics. However, the limitations of these methods are that they cannot give timely feedback and cannot meet the needs of task processing which result in a decrease in evaluation efficiency.

Deep-learning-based evaluation methods have been widely used in recent years because of their high efficiency and easy implementation. Feng et al. [12] extracted internal and external information features from the original time series network data and then trained and verified the extracted features in the recursive neural network (RNN) model, which has high predictive accuracy and robustness. He et al. [13] combined the wavelet neural network (WNN) with the maximum overlap discrete wavelet transform (MODWT) and proposed the network security situation prediction model through the data-driven method. Nevertheless, in the face of massive network security data, due to the lack of sufficient prior knowledge and established criteria of data category annotation, the task of manual category annotation is large and the cost is high, so the supervised data modeling method based on data label is gradually unable to apply to specific network scenarios.

Unsupervised learning (UL) provides an idea to solve the shortcomings of the above methods. Its main feature is that there is no need to label data categories manually but to conduct feature learning and modeling on the preprocessed data directly.

To evaluate the network threat situation of IoT effectively in a multisource data environment, this paper proposes a network threat situation assessment model based on unsupervised learning for IoT. It applies variant autoencoder and generative adversarial networks (V-G) model for cluster analysis of the training set; then the error threshold is calculated by the 3-layer variation automatic encoder. Then

it uses the abnormal traffic datasets to conduct threat tests and quantify the network situation assessment according to the calculated results of the threat situation value. The experimental results show that the method presented in this paper has a good evaluation effect on network threats and has a strong characterization ability in the face of network threats. Furthermore, it can evaluate the network threat situation effectively without relying on data labels.

3. Unsupervised Generation Network Model

3.1. Variational Autoencoder (VAE) and Generative Adversarial Network (GAN). Autoencoder (AE) and variational autoencoder (VAE) [14] are both composed of encoder and decoder; the biggest difference between them is that VAE adds the "noise constraint" that compels the encoder to produce a collection of latent variables (LV) that are subject to the unit Gaussian distribution. The network structures of AE and VAE are demonstrated in Figures 1 and 2.

Comparing Figures 1 and 2, VAE compels every sample X_k in the original sample $X = \{X_1, X_2, X_3, ..., X_n\}$ to follow the normal distribution $N(\mu, \sigma^2)$, which means fitting the average μ and the variance σ^2 of any sample X_k by the internal neural network, and then obtains a set of potential variables $Z = \{Z_1, Z_2, Z_3, ..., Z_n\}$, in which the element Z_k is subject to the multivariate standard normal distribution $N(0, I)$. In the decoding process, Z generates the sample set $Y = \{Y_1, Y_2, Y_3, ..., Y_n\}$ through the decoder; then the similarity between the generated sample set Y and the original sample set X is statistically computed by the distance function. The reconstruction error loss of the overall data element can be obtained by calculation.

Generative adversarial network (GAN) [15] is one of the most promising deep generation network models in the field of unsupervised learning, which consists of a generator and a discriminator. The network structure of GAN is shown in Figure 3.

As shown in Figure 3, the generator first learns the probability distribution characteristics of a collection of random noises obtained by direct sampling through a prior distribution. Then it tries to generate the data sample $Y = \{Y_1, Y_2, Y_3, ..., Y_n\}$ which is the same as the original sample $X = \{X_1, X_2, X_3, ..., X_n\}$ to "trick" the discriminator that is responsible for determining the similarity between the generated sample Y and the original sample X. The output of the discriminator is a scalar in the range of [0, 1] for each similarity test. The closer the scalar gets to 0, the less likely the generated sample Y_k will be judged as real data. The closer the scalar gets to 1, the more likely the generated sample Y_k will be judged as real data.

Generator and discriminator compose a dynamic game process, and the generator is gradually acquiring the distribution features of the data after the repeated game; when the discriminator's output reaches the NASH equilibrium (NASH = 0.5), it can generate sample Y that has a high degree of similarity to the original sample X through a random noise Z. The training will finish when the discriminant is unable to distinguish between real data and generated data.

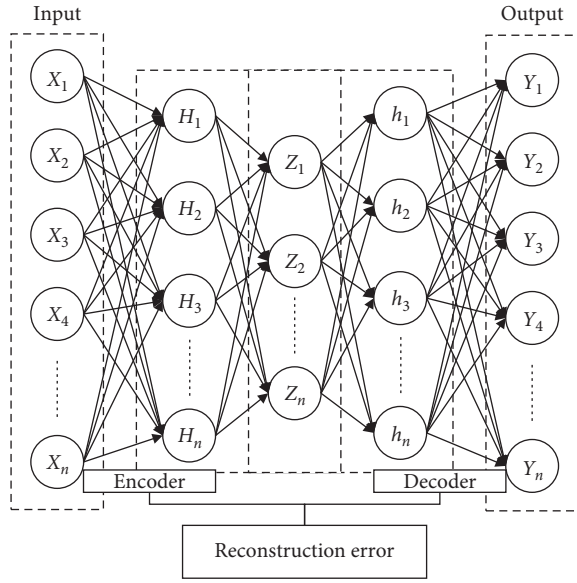

Figure 1: AE's network structure.

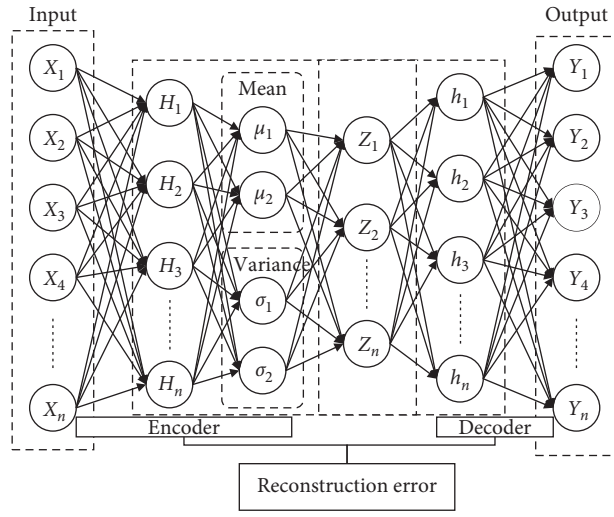

Figure 2: VAE's network structure.

3.2. V-G Network. The design of the V-G network is based on the following analysis:

(1) VAE can learn in the process of encoding data prior distribution and generate samples with good diversity performance while measuring the similarity between generated samples and original samples, can only use the mean square error (MSE) functions to roughly calculate the similarity errors between data elements, and is unable to adopt a more reasonable strategy of the similarity measure, which reduces the accuracy of matching samples.

(2) GAN has a high discriminant standard for generating samples and original samples when it judges the similarity of samples through discriminator. However, it is difficult for the fitting of real sample

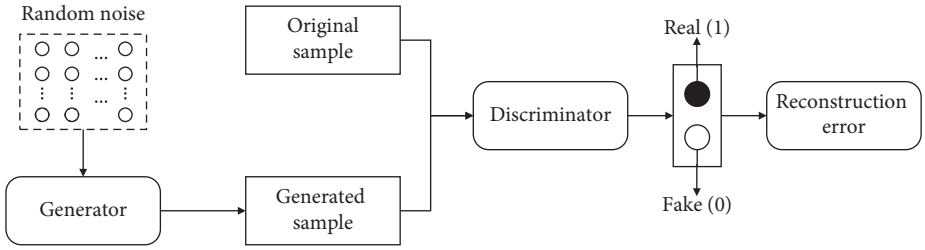

Figure 3: GAN's structure.

distribution to converge to a better result because the generator does not add any condition constraint, which causes a huge solution space when generating samples. Besides, as GAN is prone to input multiple random noise samples corresponding to the same type of sample generation in the process of sample generation, it is easy to reduce the diversity of generated samples and fall into model collapse (MC).

To complement each other's advantages, VAE's encoder and GAN's discriminator are combined to form a V-G network. Besides, when measuring the similarity, the original measurement of element error carried out by VAE is transformed into characteristic error measurement performed by GAN discriminator. For this, the V-G network can capture the data distribution characteristics easier. Therefore, using V-G for the training model not only can ensure that the diversity of sample generation is not restricted and improve its ability of mapping to original samples but also makes the discriminant result of similarity more precise. The V-G network structure is shown in Figure 4.

The V-G network in this paper is mainly used for network threat testing, and its application objects are mainly multisource heterogeneous network traffic data generated by the host, network, and server terminals. Due to the unique structural advantages of the V-G network, it can effectively extract data feature information during model training, so it can improve the accuracy of clustering and ensure higher accuracy of threat testing.

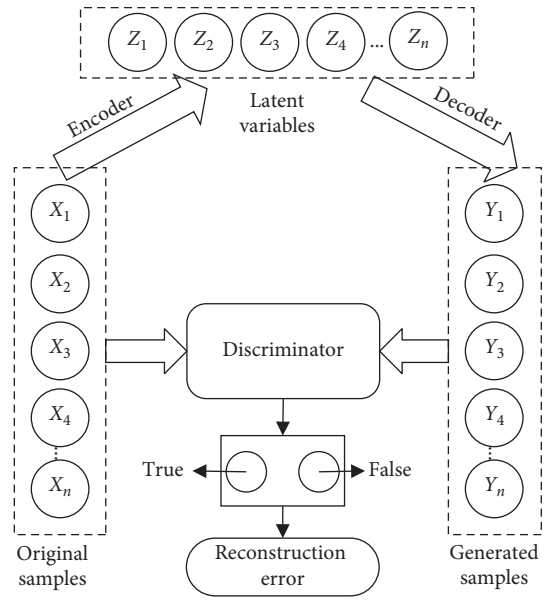

Figure 4: V-G's network structure.

4. Network Security Threat Situation Assessment for IoT Based on the V-G Network

IoT applications and devices are vulnerable to various network threats because of the connection to the Internet. At present, common types of network threats include website information leakage, web attack threat, DDoS attack vulnerability, host commonly used service vulnerability, and system configuration security. Through the threat analysis of host and network traffic data, this paper aims to discover network threats and network vulnerabilities in time and carry out real-time network security situation threat assessment.

The network security threat situation assessment framework for IoT established in this paper is presented in Figure 5.

The architecture includes five parts: assessment data set construction, data preprocessing, multisource data feature selection, network threat testing, and network threat situation assessment.

The steps of network threat quantitative assessment are as follows:

Step 1. Data acquisition: obtain the multisource network security traffic dataset as the evaluation data source.

Step 2. Data preprocessing: the original data is processed by the numerical method and feature specification to meet the requirements of model training and improve the utilization of the data.

Step 3. Feature selection: the characteristics of multisource network security traffic data are selected to reduce data redundancy.

Step 4. Threat testing: the unsupervised threat test model is used to test the threat and obtain the threat probability.

Step 5. Network threat situation assessment: obtain the threat severity and the threat impact according to the threat probability calculated in Step 4; then calculate the threat situation value and evaluate the overall situation of the network.

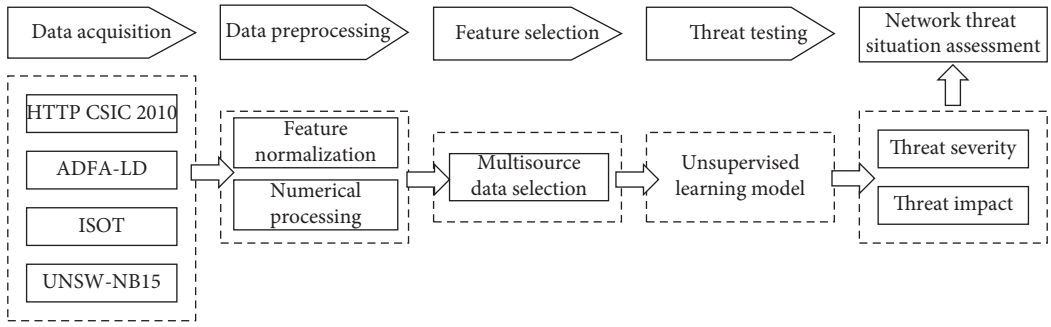

Figure 5: Network threat situation assessment framework.

4.1. Data Acquisition. IoT networks are susceptible to denial-of-service (DDoS) type of network attacks [16, 17]; in reality, however, IoT networks are facing various network attacks. To evaluate the network threat situation comprehensively, this paper selects four different types of network threat traffic datasets in the field of network security as the evaluation data sources; they are, respective, CSIC 2010 HTTP dataset based on web attack, ADFA-LD dataset based on Linux host exception, UNSW-NB15 dataset based on DDoS anonymous traffic attack, and ISOT dataset composed of mixed botnet traffic. Basic information on the four datasets is displayed in Table 1.

TP CSIC 2010 HTTP dataset is a set of normal and abnormal network attack traffic data automatically generated based on Web applications. It contains 36,000 normal requests and more than 25,000 exception requests. There are mainly three types of exception requests, which are divided into 16 attack categories.

ADFA-LD dataset is a network traffic dataset based on Linux host-level intrusion detection system, containing 5925 pieces of traffic data which are mainly divided into six attack categories: Hydra-FTP, Hydra-SSH, Adduser, Java-Meterpreter, Meterpreter, and Webshell.

ISOT dataset is composed of various botnet traffic and normal network data traffic which include 134916 pieces of traffic data divided into 19 characteristic categories: BytesAB, BytesBA, NpacketsAB, NpacketsBA, Duration, and so on.

UNSW-NB15 dataset is mainly composed of DDoS attacks in about an hour of anonymous traffic trace data; it contains 257673 traffic data, mainly divided into 9 types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

Part of the network threat situation indicators contained in the four datasets is shown in Figure 6.

Figure 6 lists some threat situation indicators for V-G network testing. Besides, other types of threat indicators are not present in this paper, but they also are used for effective testing through the V-G network. The premise is to obtain data traffic sets that contain these threatened attacks because the model needs a lot of network traffic data as baseline data for model training.

Table 1: Basic information on four types of datasets.

Dataset	Data size	Category	Data type
HTTP CSIC 2010	61000	16	Web application
ADFA-LD	5925	6	Linux host exception
ISOT	134916	19	Hybrid botnet
UNSW-NB15	257673	10	DDoS anonymous attack

4.2. Data Preprocessing. Data preprocessing mainly includes two operations: numerical processing of character feature and feature normalization. It is necessary to carry out numerical processing for the symbolic data in the evaluation data source and convert all symbolic features into ordered numerical features since the training of the V-G network set requires digital feature vector as input. At the same time, to eliminate the dimension and facilitate the operations, all the numerical characteristics after the numerical treatment are normalized in the same interval.

4.2.1. Numerical Processing of Character Feature. Through the way of one-hot encoding, the 14 HTTP request feature classes of the CSIC 2010 HTTP dataset are transformed into numerical vectors. Specifically, transform 8 kinds of feature data, protocal, userAgent, accept, acceptEncoding, pragma, cacheControl, acceptCharset, and acceptLanguage, into numerical vectors of size between 0 and 1. Convert the 3 types of HTTP request data (GET, POST, and PUT) into binary eigenvectors (1, 0, 1), (1, 0, 0), and (1, 1, 0), respectively; moreover, the three types of URL extensions (JSP, GIF, and PNG) of the web application are converted into binary eigenvectors (1, 1, 1), (0, 1, 1), and (0, 1, 0), respectively; similarly, the 42-dimensional features of the UNSW-NB15 dataset are eventually converted into 196-dimensional binary numeric vectors after numeric processing.

4.2.2. Feature Normalization. There is a significant difference between the minimum and maximum values of some features while evaluating the data source. To suppress the negative impact of these outliers on the model training, the

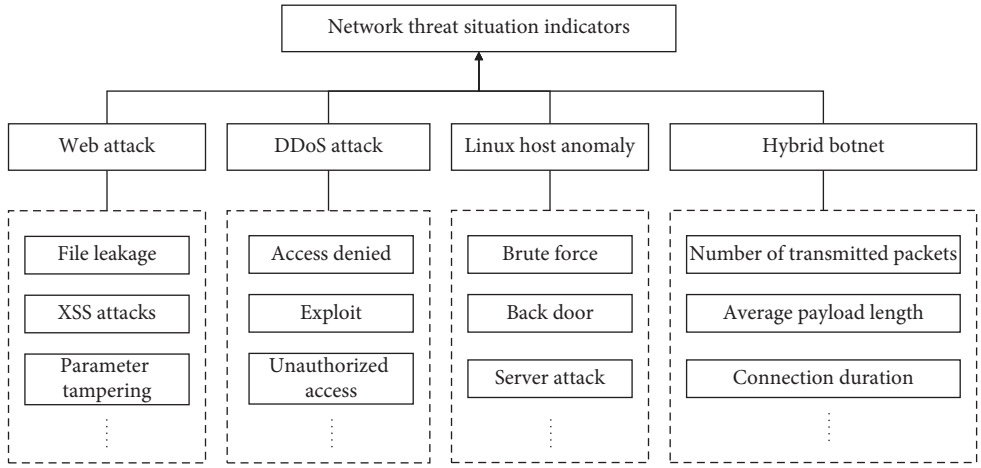

Figure 6: Part of the network threat situation indicators.

Max-Min scaling method is used to unify the feature values in the interval of [0, 1] and the formula is given as

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{1}$$

where $x*$ represents the normalized value of a certain class of features, x represents the initial eigenvalue, $x_{\min}$ is the minimum eigenvalue, and $x_{\max}$ is the maximum eigenvalue.

4.3. Multisource Dataset Feature Selection. To avoid the existing mass of redundant data of evaluating data source which may increase the overfitting risk of the V-G network in the training model and reduce the generalization ability of the model, this paper selects features of the evaluated data source which filter the unrelated features of the data source to ensure the high availability and the redundancy of data, improving the data clustering accuracy of all kinds of features in the V-G network and reducing the time complexity of model training.

In general, the feature selection process does not need to consider the structural characteristics of the data itself, but the flow data in the dataset used in this paper has the characteristics of clustering structure, so the three following factors should be considered before feature selection:

(1) V-G model training is a multifeature clustering process

(2) The data selected by features can keep the clustering structure characteristics of the flow data to the greatest extent

(3) The data selected by features can cover all possible clustering situations in a single dataset

From the above, the multicluster feature selection (MCFS) algorithm is selected for feature selection in this paper. MCFS is an unsupervised feature selection algorithm that does not rely on the data label information in the dataset. The feature selection process is divided into the five following steps.

Step 1. Constructing a k-nearest-neighbor graph. For each data point x_i corresponding to the graph with N vertices, a k-nearest-neighbor graph is constructed by searching for the k-nearest-neighbor points of x_i to obtain the local geometric structure features of the data distribution and the adjacency weight matrix W. In this paper, the Heat Kernel Weighting method is applied to calculate the adjacency weight matrix W among data points and the formula is as follows:

$$W_{ij} = e^{-\left(\left|x_i - x_j\right|^2/\sigma\right)}, \tag{2}$$

where x_i and x_j represent any two data points in the k-nearest-neighbor graph and σ is a fixed parameter.

Step 2. Spectral clustering embedded analysis. Define a diagonal matrix D whose diagonal elements are $D_{ij} = \sum_{j=1} W_{ij}$ and obtain the planar embedding structure of the data stream by calculating the generalized eigenvalue of Laplace matrix L:

$$\mathrm{LH}_k = \lambda \mathrm{Dh}_k, \tag{3}$$

where $L = D - W$ and $H = \{h_1, h_2, ..., h_k\}$ is the set of eigenvectors corresponding to the minimum generalized eigenvalues obtained through equation (3). Each column of H represents the planar embedding of any data point x_i and k represents the inner dimension of the data whose size is usually the number of clusters of the dataset.

Step 3. Sparse coefficient learning. After obtaining the planar embedding H of data points, to evaluate the importance of each feature in its corresponding data dimension (each column of H) and measure the ability of each feature to distinguish data clustering, MCFS takes the embedded h_k given by any column in H as a regression target and the objective function is represented by the following formula:

$$\min_{a_k} \left\|h_k - Q^T a_k\right\|^2 + \beta\left|a_k\right| \min_{a_k} \left\|h_k - Q^T a_k\right\|, \tag{4}$$

where a_k is an m-dimensional vector and Q is a matrix of $N \times M$. For minimizing the objective function, define the $L1$-norm of a_k as

$$\left|a_k\right| = \sum_{j=1}^{M} \left|a_{k,j}\right|, \tag{5}$$

where a_k includes the sparse coefficients used to approximate the different features of h_k. According to the penalty of $L1$-norm, the sparse coefficient of a_k will gradually shrink to zero when β is large enough. At this point, a subset of features that are most relevant to h_k will be selected.

Step 4. Calculate the MCFS score. Calculate k sparse coefficient vectors $\{a_1, a_2, ..., a_k\} \in \mathrm{R}^{\mathrm{M}}$ based on Step 3 for a dataset that contains k clusters, where each nonzero element a_k corresponds to d features. To select d effective features from k sparse coefficient vector, the MCFS score of each feature j is defined as

$$\mathrm{MCFS}(j) = \max_{k} \left|a_{k,j}\right|, \tag{6}$$

where $a_{k,j}$ is the jth element of vector a_k.

Step 5. Feature selection. According to Step 4, calculate the MCFS scores of each class of features in the dataset and sort the MCFS scores of all features in a descending order and the first d important features will be selected.

4.4. Threat Testing. To detect the new attack threats that may appear in the network environment in real time, this paper applies a V-G network to perform network threat testing. The network threat situation test model built in this paper is shown in Figure 7.

The process of threat testing is mainly divided into four processing stages: network collection layer training, network parameters optimization, output layer reconstruction error training, and threat testing.

For the convenience of expression and analysis, let l represent a single V-G network layer, and let L_1 and L_2 represent the network collection layer and network output layer, respectively. L_1 is made out of ml and $L_1 = \{l_1, l_2, ..., l_m\}$. L_2 is a 3-layer variational autoencoder network with k input and output units. The detailed steps of the network threat testing process are designed as follows.

Step 1. Network collection layer training. Normal network traffic data is input to L_1 in batches for training after data preprocessing and multisource data feature selection. The training ends when it reaches a Nash equilibrium.

Step 2. Network parameters optimization. To overcome the parameters' tendency to fall into local optimization which is caused by the parameter tuning process with Gradient Descent (GD) method, Newton method (NM), Gauss Newton (GN) method, and other algorithms, this paper uses Levenberg-Marquardt (LM) optimization algorithm instead of GD and GN algorithm to carry out parameter tuning for the V-G network.

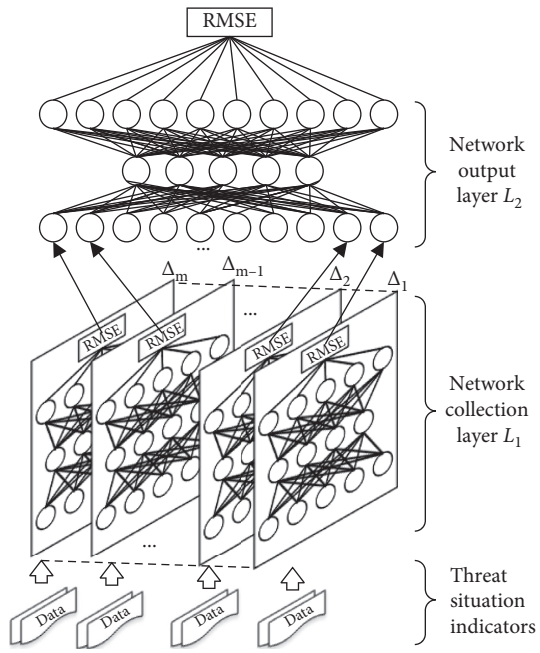

Figure 7: Network threat test based on the V-G network.

In the process of optimizing network parameters, four algorithms, GD, NM, GN, and LM, find the optimal function matching of high-dimensional data by minimizing the error sum of squares, namely, minimizing the objective function $f(x)$:

$$f(x) = \min \sum_{j=1}^{M} \sum_{i=1}^{N} f_{i,j}^{2}(x). \tag{7}$$

The gradient change of the objective function is

$$f'\left(x_{j,k}\right) = \sum_{j=1}^{M} \sum_{i=1}^{N} f_{i,j}(x) \frac{\partial f_{i,j}(x)}{\partial x_{j,k}}. \tag{8}$$

LM algorithm introduces the identity matrix I to avoid the irreversible phenomenon that may occur when the Jacobian matrix J (in GN algorithm) approximately represents the Hessian matrix H (in NM algorithm) and applies the damping factor μ to adjust the operation of the algorithm. LM algorithm combines GD algorithm and GN algorithm to dynamically tune parameters.

When optimizing the parameters, the optimization method is determined according to the gradient descent rate and the damping factor μ. If the gradient descent rate of the function is too slow, the damping factor μ increases. The GD algorithm is used to find the global optimal value:

$$x_{k+1}^{*} = x_k - (H + \mu I)^{-1} f'(x_k). \tag{9}$$

If the gradient descent rate of the function is too high, the damping factor μ decreases. The GN algorithm is used to find the global optimal value:

$$\begin{aligned} x_{k+1}^{*} &= x_k - (V + \mu I)^{-1} J^T f, \\ V &= J^T J. \end{aligned} \tag{10}$$

Step 3. Output layer reconstruction error training. The input item of the output layer network $\mathbf{L}_2$ comes from the 0-1 normalized reconstruction error value of the training output of each corresponding subnetwork in $\mathbf{L}_1$. The reconstructed error value of the output of $\mathbf{L}_1$ and $\mathbf{L}_2$ is calculated by the Root Mean Square Error (RMSE) function:

$$RMSE(\vec{x}, \vec{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}, \tag{11}$$

where $\vec{x}$ and $\vec{y}$ represent the input sample vector and the generated sample vector, respectively, and n is the dimension of the input vector.

The training error set e^* output by L_1 can be expressed as $e^* = \{e_1, e_2, \ldots, e_m\}$. e^* will be the input item of L_2; then calculate training anomaly threshold η through the RMSE function when conducting error training.

Step 4. Threat testing. After the training of the V-G network collection layer and the training of output layer reconstruction error, the test dataset containing abnormal network traffic data is used for threat testing. Select m groups randomly in the same number of test samples v and take them as the input data of L_1. The test error output by L_1 in each test can be expressed as $\beta = \{\beta_1, \beta_2, \ldots, \beta_m\}$.

4.5. Network Threat Situation Quantitative Assessment. In this study, the quantitative assessment results of network threat situation are determined by two key factors that affect network security: threat severity and threat impact.

4.5.1. Threat Severity. In this paper, the unsupervised network model is used to analyze the characteristics of multisource network traffic data. After executing the threat tests, the normalized test error value β obtained according to the threat test results during each test is taken as the probability of threat occurrence:

$$\mathrm{TP}_i = \beta_i. \tag{12}$$

This paper refers to the "Overall Emergency Plans for National Sudden Public Incidents" [18] and develops the classification of network threat situations combined with the attack classification of the Snort Chinese user manual. The threat severity is divided into five levels in this paper: safety, low-risk, middle-risk, high-risk, and super-risk levels, corresponding to the five probability intervals of threat probability: 0.00~0.20, 0.21~0.40, 0.41~0.60, 0.61~0.80, and 0.81~1.00, respectively.

4.5.2. Threat Impact. To classify the degree of impact on the threat probability, the Common Vulnerability Scoring System (CVSS) [19, 20] is used to develop a classification table of threat impact (as shown in Table 2).

The formula for calculating the threat impact (TI) is defined as

$$\mathrm{TI}_i = \log_2\left(\frac{x_1 2^c + x_2 2^I + x_3 2^A}{3}\right). \tag{13}$$

C, I, and A represent the confidentiality, integrity, and availability of three threat impact indicators, respectively, and x_1, x_2, and x_3 represent the weight of quantified value of threat impact in three threat impact indicators, respectively.

Threat situation value (TSV, denoted as T) is determined by the threat probability and the threat impact. The calculation formula is as follows:

$$T = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{TP}_i \times \mathrm{TI}_i). \tag{14}$$

5. Experiments and Results

The training and testing process based on the V-G network is carried out on the Ubuntu system, and the algorithm is implemented by Python programming language. The hardware environment of the experiment includes the Intel Core i7-7700 HQ processor, 8G RAM, and GTX 1050 graphics card, 16 GB.

5.1. Network Threat Test Results Analysis

5.1.1. Network Training. To prove the validity of the model in this paper, four networks, AE, VAE, GAN, and V-G, are, respectively, used to form a network set for model training. Four kinds of models use the same parameters for network training and the training data is the same set of normal network traffic data which ensures the comparability of the results. Model training is carried out when the number of layers of network collection is 5, 10, 15, 20, and 30.

The training anomaly threshold η output from four types of threat test models in the stage of model training under the different network layers is shown in Figure 8.

Figure 8 shows that, compared with the other three models, the V-G network obtains the minimum training error threshold η when the number of the network layers is 15, suggesting that refactoring capability for processing raw data of the V-G model is superior to the other three models.

In the process of model training, four optimization algorithms, GD, NM, GN, and LM, are used to optimize the model parameters of the V-G network, and the convergence of the optimization process of the four algorithms is shown in Table 3.

Table 2: Threat impact classification.

Threat impact	Probability interval	Impact indicators		
		Confidentiality (C)	Integrity (I)	Availability (A)
No-effect	0.00~0.40	0	0	0
Low-effect	0.41~0.80	0.22	0.22	0.22
High-effect	0.81~1.00	0.56	0.56	0.56

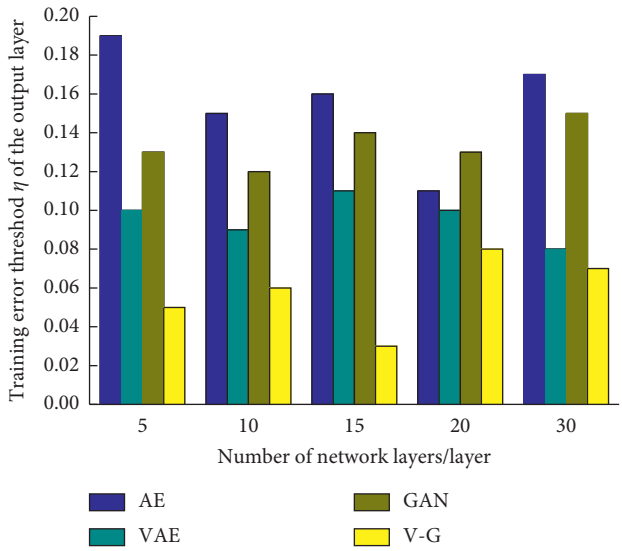

Figure 8: Four kinds of models training error threshold η.

Table 3: The convergence of different optimization algorithms.

Optimization algorithms	Iterations	Time (s)	RMSE
GD	220	350	0.35
NM	210	370	0.37
GN	200	320	0.32
LM	240	340	0.08

As can be seen from Table 3, compared with the other three algorithms, though the LM algorithm has more iterations and consumes more time, the Root Mean Square Error value is the smallest, indicating that the algorithm achieves a better convergence effect for the model which is more helpful for improving the accuracy of threat testing.

5.1.2. Network Testing. We conduct 200 groups of threat tests with random data of the same size, which is selected from the same test dataset. Four models, AE, VAE, GAN, and V-G, are used to carry out threat testing experiments, respectively. The normalized test error β obtained from the 10 groups of threat test experiments is shown in Figure 9.

As can be seen from Figure 9, compared with the other three types of models, the V-G network has the largest test error β when the number of network collection layers reaches 15 with the same test samples which indicate that its ability to detect network threats is more prominent.

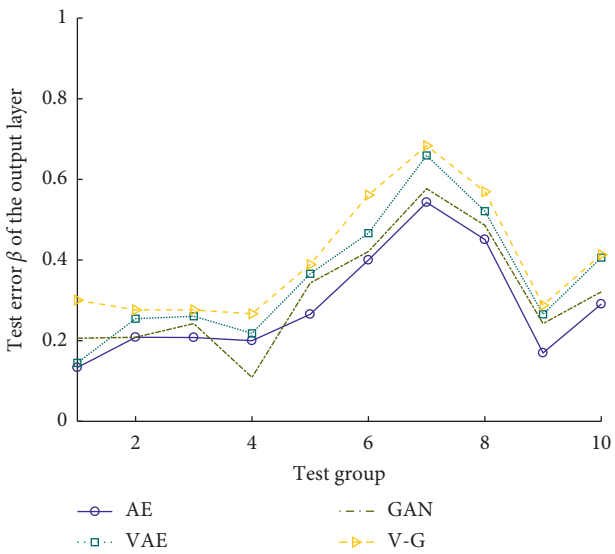

Figure 9: Threat test results of four kinds of models.

5.2. Network Threat Situation Quantitative Assessment Results Analysis. The test error β of each group is normalized to the interval of [0, 1] and is obtained through the process of network threat testing. The evaluation results of the threat severity and the threat impact of 10 groups of network threat situations are shown in Table 4.

Table 4: Evaluation results of the threat severity and the threat impact.

No.	Threat probability	Threat severity	Threat impact
1	0.187	Safety	No-effect
2	0.275	Low-risk	No-effect
3	0.238	Low-risk	No-effect
4	0.426	Middle-risk	Low-effect
5	0.262	Low-risk	No-effect
6	0.557	Mid-risk	Low-effect
7	0.685	High-risk	High-effect
8	0.504	Middle-risk	Low-effect
9	0.358	Low-risk	No-effect
10	0.281	Low-risk	No-effect

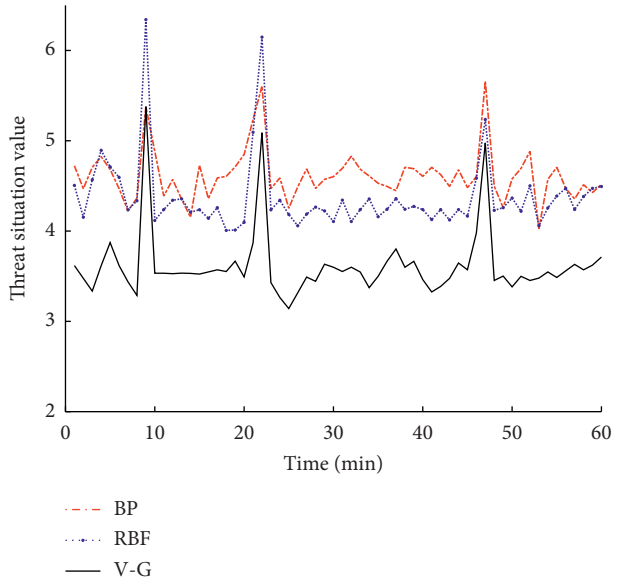

Figure 10: Comparison of threat situation values.

To increase the objectivity and authenticity of the evaluation results, the threat situation value was calculated, respectively, by Back Propagation (BP) [21] and Radial Basis Function (RBF) [22] methods and compared with the calculated results of the V-G network. The calculation results of the threat situation values of three types of methods in a certain period are displayed in Figure 10.

As can be seen from Figure 10, at 9 minutes, 22 minutes, and 47 minutes, the threat situation value shows a large range of changes, which indicates that the threat severity of the network is high at these moments and the network might be subjected to various types of attacks. It is found that, compared with the BP network and the RBF network in the three moments when the network is threatened, the method in this paper has a stronger capability of representing the features of network threats.

Besides, the curve of the V-G network is smoother than the other two networks, which indicates that the threat situation value calculated by the V-G network is more stable.

6. Conclusions

To overcome the limitations that traditional method of network threat situation assessment based on supervised learning needs to rely on data modeling label, this paper proposes a network threat situation assessment model based on unsupervised learning for IoT. This paper selects the multisource and heterogeneous datasets to simulate various network threats to IoT and calculates the threat situation value through quantifying the impact factors of network threat situation and then accomplishes the real-time situation of network threat assessment. The simulation experimental results show that the proposed method can evaluate the overall situation of network threats more intuitively and has a stronger characterization ability for network threats which can analyze the network security situation of IoT more precisely and take effective measures to reduce the risk of network threats. In the future, we will apply more network threat data that IoT will be confronted with on our proposed model, which will verify the general applicability of our proposed method.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Civil Aviation Joint Research Fund Project of the National Natural Science Foundation of China under Grant no. U1833107.

References

[1] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: a survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[2] Y. L. Mo and B. Sinopoli, "On the performance degradation of cyber-physical systems under stealthy integrity attacks," *IEEE Transactions on Automatic Control*, vol. 64, no. 9, pp. 2618–2624, 2015.

[3] Z. L. Liu, B. Li, Y. Y. Huang et al., "New MCOS: Towards a practical multi-cloud oblivious storage scheme," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 4, pp. 714–727, 2019.

[4] Y. Huang, B. Li, Z. Liu et al., "ThinORAM: towards practical oblivious data access in fog computing environment," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 602–612, 2020.

[5] J. Li, Y. Huang, Y. Wei et al., "Searchable symmetric encryption with forward search privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, p. 1, 2019.

[6] Z. Liu, J. Li, S. Lv et al., "EncodeORE: reducing leakage and preserving practicality in order-revealing encryption," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[7] Y. B. Leau, S. Manickam, and Y. W. Chong, "Network security situation assessment: a review and discussion," in *Information Science and Applications*, pp. 407–414, Springer, Berlin, Germany, 2015.

[8] M. Yang, R. Jiang, T. L. Gao et al., "Research on cloud computing security risk assessment based on information entropy and Markov chain," *International Journal of Network Security*, vol. 20, no. 4, pp. 664–673, 2018.

[9] H. Wang, Z. Chen, X. Feng et al., "Research on network security situation assessment and quantification method based on analytic hierarchy process," *Wireless Personal Communications*, vol. 102, no. 2, pp. 1401–1420, 2018.

[10] H. Sallam, "Cyber security risk assessment using multi fuzzy inference system," *International Journal of Engineering and Technology Innovation (IJETI)*, vol. 4, no. 8, pp. 13–19, 2015.

[11] Z. C. Wen, Z. G. Chen, and J. Tang, "Network security situation quantitative evaluation method based on information fusion," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 42, no. 8, pp. 1593–1602, 2016.

[12] W. Feng, Y. Q. Wu, and Y. X. Fan, "A new method for the prediction of network security situations based on recurrent neural network with gated recurrent unit," *International Journal of Intelligent Computing and Cybernetics*, vol. 11, no. 4, pp. 511–525, 2018.

[13] F. N. He, Y. Q. Zhang, D. H. Liu et al., "Mixed wavelet-based neural network model for cyber security situation prediction using MODWT and Hurst exponent analysis," in *Proceedings of the International Conference on Network and System Security*, pp. 99–111, Sapporo, Japan, December 2017.

[14] C. Doersch, "Tutorial on Variational Autoencoders," 2016, http://arxiv.org/abs/1606.05908.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, Cornell University, New York, NY, USA, 2014.

[16] R. Vishwakarma and A. K. Jain, "A survey of DDoS attacking techniques and defence mechanisms in the IoT network," *Telecommunication Systems*, vol. 73, no. 1, pp. 3–25, 2020.

[17] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks," in *Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0452–0457, Las Vegas, NV, USA, January 2019.

[18] The State Council of the People's Republic of China, *Overall Emergency Plans for National Sudden Public Incidents*, The State Council of the People's Republic of China, Beijing, China, 2006.

[19] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security and Privacy Magazine*, vol. 4, no. 6, pp. 85–89, 2006.

[20] Common Vulnerability Scoring System v3.0: Specification Document, 2020, https://www.first.org/cvss/specification-document.

[21] C. H. Tang and S. Z. Yu, "A network security situation prediction method based on likelihood BP," *Computer Science*, vol. 36, no. 11, pp. 97–100, 2009.

[22] Z. Q. Lai, *Network Security Situation Prediction Model Based on Hybrid Optimization RBF Neural Network*, Lanzhou University, Lanzhou, China, 2017.