Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

Research on unsupervised anomaly data detection method based on improved automatic encoder and Gaussian mixture model

Xiangyu Liu[1], Shibing Zhu[1*], Fan Yang[1] and Shengjun Liang[2]

Abstract

With the development of cloud computing, more and more security problems like "fuzzy boundary" are exposed. To solve such problems, unsupervised anomaly detection is increasingly used in cloud security, where density estimation is commonly used in anomaly detection clustering tasks. However, in practical use, the excessive amount of data and high dimensionality of data features can lead to difficulties in data calibration, data redundancy, and reduced effectiveness of density estimation algorithms. Although auto-encoders have made fruitful progress in data dimensionality reduction, using auto-encoders alone may still cause the model to be too generalized and unable to detect specific anomalies. In this paper, a new unsupervised anomaly detection method, MemAe-gmm-ma, is proposed. MemAe-gmm-ma generates a low-dimensional representation and reconstruction error for each input sample by a deep auto-encoder. It adds a memory module inside the auto-encoder to better learn the inner meaning of the training samples, and finally puts the low-dimensional information of the samples into a Gaussian mixture model (GMM) for density estimation. MemAe-gmm-ma demonstrates better performance on the public benchmark dataset, with a 4.47% improvement over the MemAe model standard F1 score on the NSL-KDD dataset, and a 9.77% improvement over the CAE-GMM model standard F1 score on the CIC-IDS-2017 dataset.

Keywords: Cloud security, Unsupervised machine learning, Anomalous data detection, Memory module, Deep autoencoder, Gaussian mixture model

Introduction

With the development of computing power, cloud computing has affected the way we store and manage data. And the concept of building IT infrastructure has also changed dramatically, with a consequent reduction in start-up costs [1] and operational costs of new businesses. In addition, cloud computing enables reduced system complexity, fast access to information, rapid scaling and a lower threshold for innovation. However, a new security issue arises: the disappearing boundary.

In traditional information security, data is stored in the enterprise or organization and can be effectively secured internally using firewalls, anti-virus gateways, watermark detection [2] and even physical isolation. However, with the large-scale application of cloud technology, an organization's data will eventually leave the user premises and be uploaded into the cloud platform. It can be argued that data is the most important commodity in all aspects of cloud computing [3] that must be defended. In general, data protection in a cloud computing environment can be divided into two categories: data security away from organizational boundaries and data security within organizational boundaries [4]. However, frequent data interactions mean that the network boundaries of organizations and cloud platforms are gradually weakening. And the traditional boundary protection model is no longer effective in preventing attack [5, 6] patterns based on "supply chain pre-implantation + social engineering

*Correspondence: ZSBpaper@163.com

[1] University of Space Engineering, Beijing 101416, China
Full list of author information is available at the end of the article

attacks (account hijacking and insider threats)". Therefore, this paper proposes a new data security protection method. The method is about anomaly detection technology based on network traffic to ensure data security during the interaction between organizations and the cloud platform. By monitoring the interaction traffic in the cloud platform network, business features in the traffic are extracted. On this basis, the impact (anomaly) of advanced attacks on business features when moving laterally in the intranet is identified. After that, hidden internal attacks can be better detected. In general this paper hopes to design an unsupervised anomaly detection model to address the following issues that may arise in the cloud security domain.

(1) Data security hazards caused by abnormal insider behavior.
(2) Advanced attacks that are extremely stealthy but have the potential to cause fluctuations in traffic characteristics.
(3) Hidden risks involving social engineering attacks such as account theft and hardware implantation.

In the process of anomaly detection, the proposed model only analyzes network traffic features, including ip address, login location, number of packet interactions, and data flow duration, and does not detect data content.

In recent years, machine learning has been widely used in unsupervised anomaly detection, especially in the field of high-dimensional big data anomaly detection represented by cloud security [7]. It has been extensively studied by many researchers [8–12], such as deep auto-encoder (Deep auto-encoder), improved K-mean algorithms, etc. In anomaly detection tasks, sensitivity to anomalous data is usually improved by training the model so that it could learn the internal relationships of normal data as much as possible. For example, deep auto-encoders make it difficult for anomalous data to be reconstructed through the encoder by training on normal data. And it is also difficult to produce a higher re-construction error than normal data, which in turn serves as a criterion for identifying anomalies. However, the above is not always effective in practice. Sometimes the auto-encoder can be so "overgeneralized" that it can still re-construct anomalies well when faced with partial anomalies, resulting in missed or false positives. As shown in the figure below, most of the samples have enough low-dimensional information to support the anomaly detection task. However, there are still some anomalies that are difficult to distinguish from normal samples, such as the red and blue overlapping regions.

Figures 1 and 2: Low-dimensional information of samples from public cyber-security datasets: (1) Each original data sample contains 49 features, which can be expanded to 122 dimensions after one-hot encoding; (2) The red dots represent abnormal samples and blue dots represent normal samples. Each image contains 1000 samples from public datasets; (3) The low-dimensional information represented by the horizontal and vertical axes is generated by a structure of 119-60- 30-10-1-10-30-60-119 generated by a deep auto-encoder. The horizontal axis indicates the reconstruction error caused during the

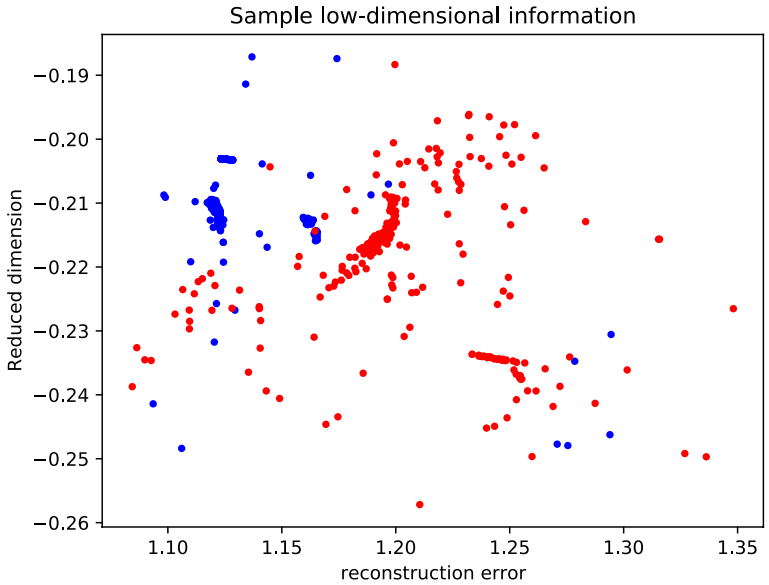

Fig. 1 Sample low-dimensional information

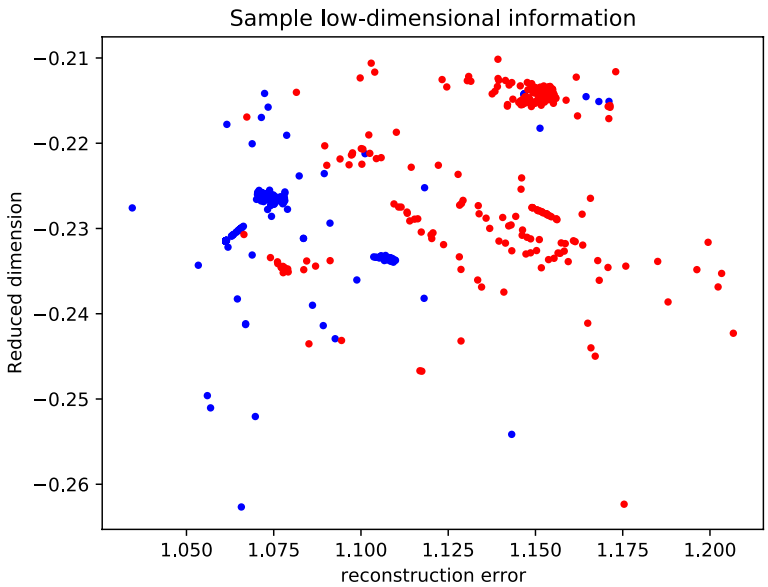

Fig. 2 Sample low-dimensional information

encoding and decoding of the depth auto-encoder. And the vertical axis indicates the one-dimensional features of the samples after compression.

In this paper, we propose a new unsupervised anomaly detection method, MemAe-gmm-ma. The model uses a deep auto-encoder to generate a low-dimensional representation and reconstruction error for each input sample. Meanwhile a memory module is added inside the auto-encoder to better learn the inner meaning of the training sample. Then the low-dimensional information of the sample is fed into a Gaussian mixture model [13–15] (GMM) for density estimation. The Gaussian model affiliation of the output is used to calculate the martingale distance of the samples, and finally the anomaly index is obtained.

This paper makes the following contributions to unsupervised anomaly detection for traffic data in cloud security by.

(1) Jointly optimizing the parameters of the deep self-encoder and Gaussian mixture model simultaneously in an end-to-end manner. The joint optimization well balances the auto-coding reconstruction and density estimation. It helps the auto-coder to get rid of the local optimum problem.
(2) The incorporation of a memory module to sparse the low-dimensional data space effectively solves the problem that the model may be too generalized.
(3) The calculation of loss function and sample energy is optimized and innovated, which achieves excellent results on two public datasets and also demonstrates better robustness of the model.

The rest of the paper is structured as follows: Related jobs section provides an overview of existing unsupervised anomaly detection that may be applicable to cloud security. A hybrid threshold anomaly detection model based on improved autoencoder and Gaussian mixture model section provides a detailed description of the model structure proposed in this paper. Experiment and analysis section shows the experimental results of this paper's model under two public datasets and evaluates the robustness of the model. Conclusion section concludes the paper and the direction of future work.

Related jobs

Cloud environments face many challenges. And in this paper, we mainly consider the hidden risks that exist in the cloud platform during the interaction of various nodes, as well as the hidden attacks from within the platform. Since the cloud environment contains a variety of complex device access points and runs a large number of virtual and physical nodes, some network attacks in the cloud environment come from outside and some from inside. And the attack traces are distributed in multiple nodes, and the system continuously generates a large amount of business data, security logs and alarm information. Traditional analysis means are difficult to cope with the rapid analysis of massive security data, which must rely on machine learning technology. For the cloud platform arithmetic power, the characteristics of the huge scale of data, it is necessary to give full play to the advantages of artificial intelligence neural network with strong adaptive capacity. And providing end-to-end intelligent

scanning for all kinds of application vulnerabilities and DDOS and other network attacks abnormal traffic.

In recent years, domestic mainstream cloud service providers and network security companies have gradually applied artificial intelligence technology to cloud security. Machine learning algorithms are used for feature extraction and modeling of normal and abnormal traffic to detect traces of attacks disguised as normal traffic. The model parameters are adjusted to optimize the protection model in response to the continuously generated traffic data to achieve continuous iteration and update of the protection strategy.

The existing unsupervised anomaly detection [16–18] broadly includes:

Methods based on sample reconstruction such as principal component analysis [19, 20] (PCA), kernel PCA [21], robust PCA, sparse representation, and self-encoder. Among them, the PCA class of methods is divided into two ideas [22–24]. One is to map the data to a low-dimensional feature space and then check the deviation of each data point from other data in different dimensions of the feature space. The other is to map the data to a low-dimensional feature space and then remap the data back to the original space by the low-dimensional feature space. The second idea tries to reconstruct the original data with the low-dimensional features, and determine the anomaly according to the magnitude of the reconstruction error. The autoencoder [25, 26] is similar to this, generating a low-dimensional representation of the data and reconstruction error by the neural network structure under the constraint of the loss function. The sparse representation-based approach detects anomalies by jointly learning a dictionary and a sparse representation of normal data [27, 28].

Methods based on probability density estimation such as k-means, multidimensional Gaussian model, and hybrid Gaussian model. Clustering algorithms divide data points into relatively dense "clusters", and those points that cannot be classified as a certain cluster are regarded as outliers. This type of algorithm is highly sensitive to the choice of the number of clusters. In the case of k-means, the number of clusters is not chosen properly, which may result in more normal values being classified as outliers, or small clusters of outliers being classified as normal. Therefore, specific parameters need to be set for each data set to ensure the effect of clustering, which is less generalizable among data sets. Xie Bin et al. [12] proposed an intrusion detection algorithm based on three-branch dynamic threshold K-means clustering by improving on the basis of fixed threshold to discriminate anomalies. And the team used the algorithmic idea of dynamic threshold to successfully optimize the final number of K-means clusters and reduce the impact of fixed initial number of clusters on the model detection efficiency.

Methods based on support domain such as one class support vector machine [29–31] (One Class SVM) and support vector data description (SVDD). Its assumption is that normal and abnormal samples can be distinguished accordingly by boundaries. However, as the dimensionality of the data increases, the support domain-based methods are limited in performance and are very sensitive to outliers. Therefore, when there are outliers (dirty data) in the training data, the detection effectiveness of the method is greatly affected.

Since the performance of reconstruction-based and support domain-based methods is affected when dealing with high-dimensional data, jointly trained model building is gradually gaining attention [32–34]. In 2018, Bo Zong et al. [35] trained auto-encoder and Gaussian mixture model jointly, which not only solved the local optimum problem in the detection process, but also significantly improved the performance of the model. Ning Hu et al. [36] proposed the RF-DAGMM method, based on DAGMM, not only improved the model training efficiency, but also improved several metrics such as accuracy, precision and recall.

The reconstruction-based approach relies on the model's comprehensive learning of the normal sample connotation. Thus it can accurately establish the mapping relationship between sample-low-dimensional information-reconstructed sample. But in practice, sometimes the model can accurately reconstruct the normal sample while reconstructing part of the abnormal sample in the meantime, which is the main reason for the reduced accuracy of the model. To avoid the problems above, many researchers have put great efforts in the field of memory-enhanced networks [37–39]. Since the memories generated by models such as RNN and LSTM compress information and weights into a low-dimensional space. And the memories generated by the models are relatively scarce, Jason Weston et al. proposed the model of memory networks. This model jointly trains a read-write external memory module and an interface component to produce long-term (large amount) and easy-to-read memories. In 2019, Dong Gong et al. [40] proposed a memory-enhanced self-encoder (MemAE) which tightens the low-dimensional information space of the samples by fixing the memory module to the inner information of the training set (normal data). It effectively improved the anomaly detection performance of the self-encoder on picture and video and provided an improved direction for reconstruction-based anomaly detection algorithms.

A hybrid threshold anomaly detection model based on improved autoencoder and Gaussian mixture model

As shown in Fig. 3, MemAe-gmm-ma consists of two main components, namely the low-dimensional information network and the anomaly estimation network. The low-dimensional information network uses the feature of the auto-encoder to compress the samples into the low-dimensional space and introduces a memory module to allow the model to better learn the intrinsic relationships of the training samples. The anomaly estimation network uses a Gaussian mixture model, in which the sample anomaly indices in the low-dimensional space are further evaluated based on the martingale distance of the samples in this framework.

Low-dimensional information network

As shown in the Fig. 4, the low-dimensional information network consists of a deep auto-encoder, which contains a memory module. The sample x is downscaled by a multi-layer neural network encoder with θ_e as the parameter to obtain the low-dimensional sample z_c. z_c is weighted and matched by a memory module to obtain z_c', and z_c' is reconstructed by a multi-layer neural network decoder with θ_d as the parameter to obtain the reconstructed sample x'.

$$z_c = h(\mathrm{x}; \theta_e) \tag{1}$$

$$x' = g\left(z_c'; \theta_d\right) \tag{2}$$

The memory module structure is shown in Fig. 5.

The memory module $M \in R^{N_\mathrm{m} \times C}$ contains N_m memory messages. The dimensions of the messages in memory C are aligned with those of z_c and each memory message is represented as $m_i (i \leq N_\mathrm{m})$.

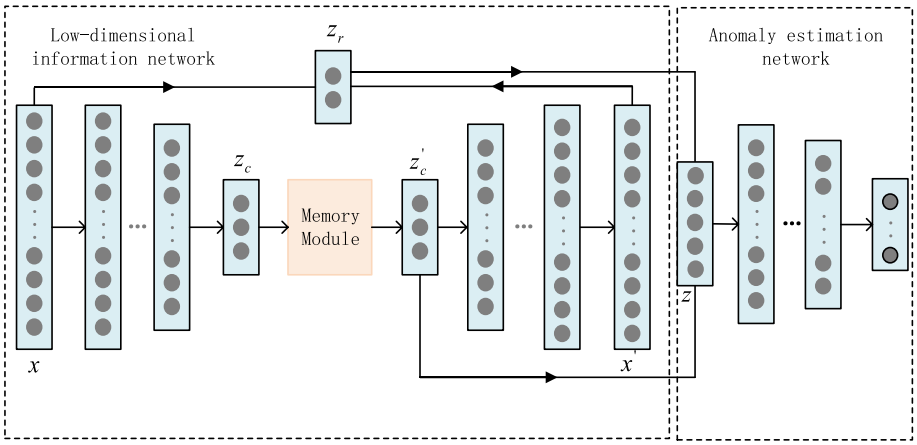

Fig. 3 MemAe-gmm-ma structure diagram

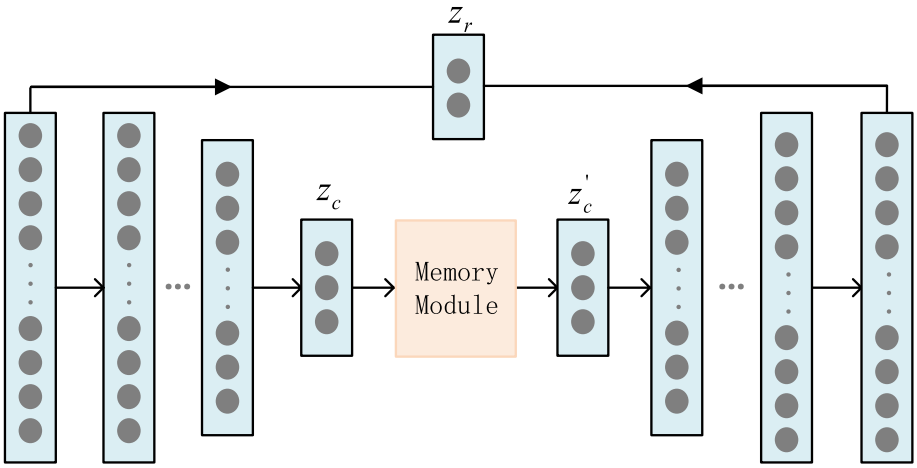

Fig. 4 Low-dimensional information network structure diagram

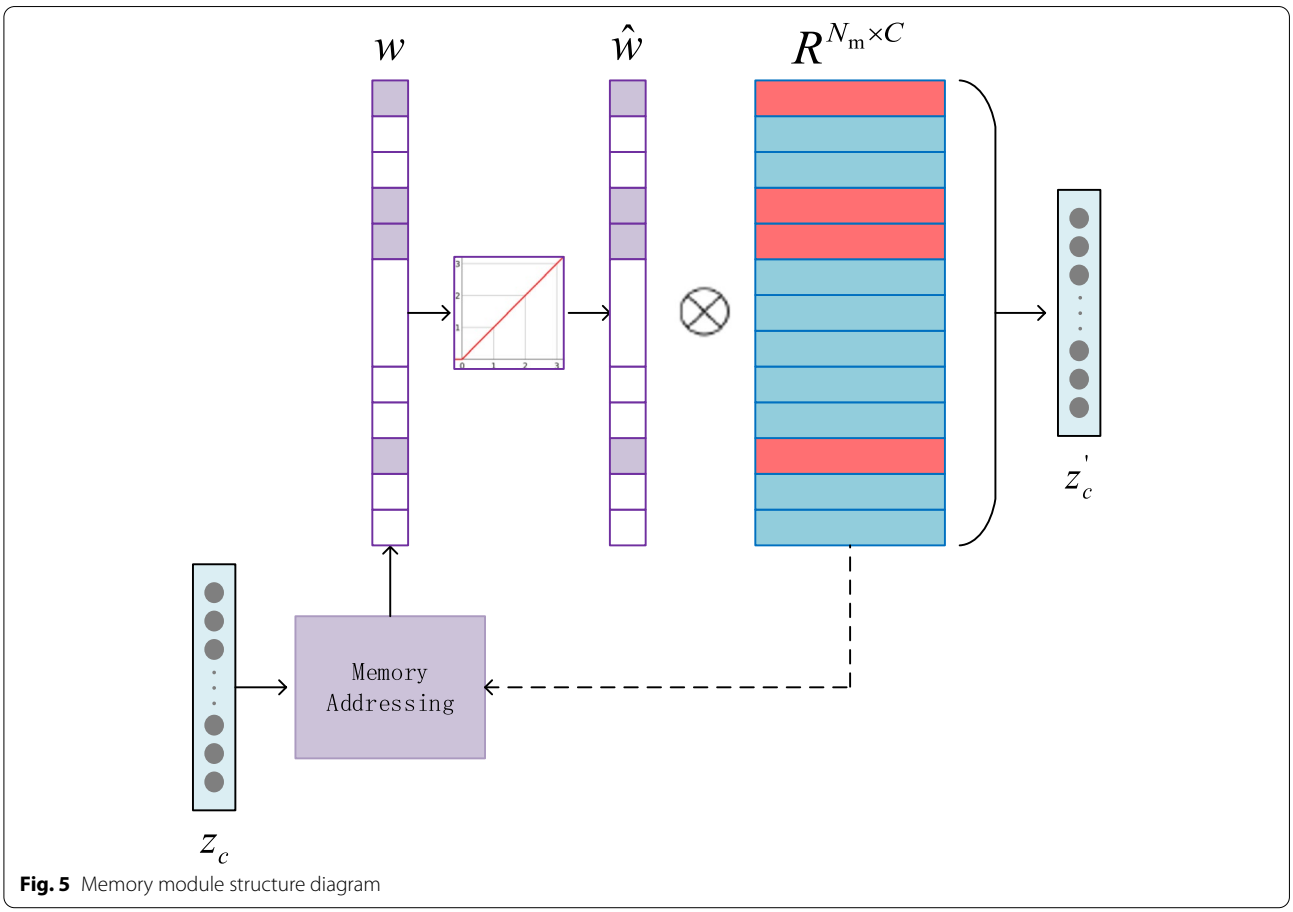

Fig. 5 Memory module structure diagram

The memory module first uses a softmax function in non-exponential form to calculate the weights

$$w_i = \frac{\exp\left(d(z_c, m_i)\right)}{\sum_{j=1}^{N_m} \exp\left(d\left(z_c, m_j\right)\right)} \tag{3}$$

where $d(\cdot)$ is the cosine similarity.

$$d(z_c, m_i) = \cos < z_c, m_i > \tag{4}$$

However, some anomalous samples may have the opportunity to combine with the information in memory through a w set containing many low weights, which in turn can be well re-constructed. To alleviate this problem, this paper uses a hard shrink operation for the set w

$$\hat{w}_i = \frac{\max\left(w_i - \lambda, 0\right) \cdot w_i}{\mid w_i - \lambda \mid + \varepsilon_1} \tag{5}$$

$\varepsilon_1$is a minimal value and the threshold λ is usually set to a value in the interval [1/N,3/N]. After the shrinkage process, the weights are normalized and then the output of the memory module is obtained at z_c'.

$$\hat{w}_i = \frac{\hat{w}_i}{\left\|\hat{w}\right\|_1} \tag{6}$$

$$z_c^{'} = \sum_{i=1}^{N_m} \hat{w}_i m_i \tag{7}$$

The output of the compressed network z contains two sources of features: (1) the low-dimensional information z_c' and (2) the reconstruction error between x and $x^{'}z_r$.

$$z_r = f\left(\mathrm{x}^{'}; \mathrm{x}^{''}\right) \tag{8}$$

$$z = \left[z_c^{'}; z_r\right] \tag{9}$$

among which, z_r are the 2-dimensional features, cosine similarity and Euclidean distance, respectively.

$$L_1\left(\mathrm{x}; \mathrm{x}^{'}\right) = \sqrt{\sum_{i=1}^{N}\left(\mathrm{x}_i - \mathrm{x}_i^{'}\right)^2} \tag{10}$$

The cosine similarity is expressed as

$$L_2\left(\mathrm{x}^{'};\mathrm{x}^{''}\right)=\cos<\mathrm{x}^{'},\mathrm{x}^{''}>=\frac{\sum_{i=1}^{N}\mathrm{x}_i^{'}\bullet\mathrm{x}_i^{''}}{\sqrt{\sum_{i=1}^{N}\mathrm{x}_i^{'2}}\bullet\sqrt{\sum_{i=1}^{N}\mathrm{x}_i^{''2}}} \tag{11}$$

$$z_r=\left[L_1\left(\mathrm{x}^{'};\mathrm{x}^{''}\right);L_2\left(\mathrm{x}^{'};\mathrm{x}^{''}\right)\right] \tag{12}$$

Anomaly estimation network

The energy estimation network is a Gaussian mixture model (GMM) It performs density estimation by predicting the mixed affiliation of each sample using a multilayer neural network, which is a clustering algorithm [14]. $P=MLN(z;\theta_m)$ is the output of a multi-layer neural network parameterized by θ_m, and $\widehat{\gamma}=\mathrm{softmax(p)}$ is a K-dimensional vector.

Given N data samples, $\forall 1\le k\le K$, the parameters in the GMM are shown below.

$$\hat{\varphi}_k=\sum\nolimits_{i=1}^{N}\frac{\hat{\gamma}_{ik}}{N} \tag{13}$$

$$\hat{\mu}_k=\frac{\sum_{i=1}^{N}\hat{\gamma}_{ik}z_i}{\sum_{i=1}^{N}\hat{\gamma}_{ik}} \tag{14}$$

$$\hat{\sum}_k=\frac{\sum_{i=1}^{N}\hat{\gamma}_{ik}\left(z-\hat{\mu}_k\right)\left(z-\hat{\mu}_k\right)^T}{\sum_{i=1}^{N}\hat{\gamma}_{ik}} \tag{15}$$

$\hat{\varphi}_k$ is the mixture probability of component K in GMM. $\hat{\mu}_k$ is the mean. $\hat{\sum}_k$ is the covariance, and $\hat{\gamma}_{ik}$ is the density estimate of the i th input sample z_i under the k th Gaussian mixture model component.

Suppose there is a data set $X=(X_1,X_2,\cdots X_n)$ with mean $u=(u_1,u_2,\cdots u_j)^{\mathrm{T}}$ and covariance matrix $\sum$. The number of samples is n and the dimension of the data is j. Then its martingale distance is expressed as

$$D_M(X)=\left|\left(\sqrt{(X-\mu)^{\mathrm{T}}\sum^{-1}(X-\mu)}\right|\right. \tag{16}$$

Then the martingale distance of the low-dimensional sample z is given by

$$D_M(z)=\left|\left(\sqrt{\left(z-\hat{\mu}_k\right)^{\mathrm{T}}\hat{\sum}^{-1}\left(z-\hat{\mu}_k\right)}\right|\right. \tag{17}$$

Using the above parameters, the sample abnormality index can be calculated by the following formula. Lower sample abnormality index represents a higher normality of the sample, while the high-energy sample can be judged as abnormal by a pre-selected threshold.

$$E(z)=-\log\left(\sum_{k=1}^{K}\hat{\varphi}_k\exp\left(-\lambda_z D_M(z)\right)\right) \tag{18}$$

Objective function

Given N data samples, according to the model described in the previous section, the objective function guiding the training of the model in this paper is constructed as follows.

$$J(\theta_e,\theta_d,\theta_e)=\frac{1}{N}\sum_{i=1}^{N}L_1\left(\mathrm{x}_i,\mathrm{x}_i^{'}\right)+\frac{\lambda_1}{N}\sum_{i=1}^{N}E(\mathrm{z}_i)+\frac{\lambda_2}{N}\sum_{i=1}^{N}\left(-\cdot\widehat{\mathrm{w}}_i\cdot\log\left(\widehat{\mathrm{w}}_i\right)\right)+\lambda_3 P\left(\hat{\sum}\right) \tag{19}$$

The objective function consists of four components: $L_1\left(\mathrm{x}_i^{'},\mathrm{x}_i^{''}\right)$ is the re-construction error (Euclidean distance) caused by the deep auto-encoder during encoding and decoding. $E(z_i)$ is the sample anomaly index of the Gaussian mixture model output. $\sum_{i=1}^{N}\left(-\cdot\widehat{\mathrm{w}}_i\cdot\log\left(\widehat{\mathrm{w}}_i\right)\right)$ is the negative log-likelihood from the sparsely processed weights. $P\left(\hat{\sum}\right)$ is a minimal value, which is mainly used to prevent the values on the diagonal of the covariance matrix from becoming zero and eventually leading to matrix integrability in the Gaussian mixture model.

Mixing thresholds

In this model species, the abnormality of samples is determined by the abnormality index threshold. The number of the samples is supposed as N and the percentage of abnormal samples among all samples is ρ, meanwhile the energy value of each sample is calculated by the model of this paper. Then all samples are sorted in descending order according to the energy value and the martingale distance. The threshold value T used for abnormality detection will be the abnormality index of the sample at $\rho\times N$ from the highest to the lowest among all samples.

Table 1 NSL-KDD data distribution

Data Category	Training set (percentage)	Test set (percentage)
Normal	67,343 (53%)	9711 (43%)
DoS	45,927 (37%)	7458 (33%)
Probe	11,656 (9.11%)	2421 (11%)
R2L	995 (0.85%)	2654 (12.1%)
U2R	52 (0.04%)	200 (0.9%)
Total	125,973	22,544

Table 2 CIC-IDS-2017 data distribution

Data Category	Training set (percentage)	Test set (percentage)
BENIGN	396,454 (79.29%)	158,556 (79.28%)
DoS Hulk	40,078 (8.02%)	15,929 (7.97%)
PortScan	27,959 (5.5918%)	11,387 (5.6935%)
DDoS	22,606 (4.5212%)	8956 (4.478%)
DoS GoldenEye	1795 (0.359%)	668 (0.334%)
Infiltration	1469 (0.2938%)	572 (0.286%)
FTP-Patator	1390 (0.278%)	550 (0.275%)
Bot	1367 (0.2734%)	577 (0.2885%)
Web Attack	2933 (0.5866%)	1183 (0.5915%)
SSH-Patator	1018 (0.2036%)	444 (0.222%)
DoS slowloris	1008 (0.2016%)	392 (0.196%)
DoS Slowhttptest	995 (0.199%)	377 (0.1885%)
Heartbleed	928 (0.1856%)	409 (0.2045%)
Total	500,000	200,000

$$\text{result} = \begin{cases} \text{abnormal}, & E(i) > T \\ \text{normal}, & \text{others} \end{cases} \tag{20}$$

Experiment and analysis

This section experiments and analyzes the anomalous data detection methods mentioned in this paper on the KDD99 dataset and the CIC-IDS-2017 dataset respectively.

Introduction to the data set

In this section, the two network traffic datasets NSL-KDD and CIC-IDS-2017 used in this paper respectively will be introduced [16]. The NSL-KDD dataset solves these inherent problems. The CIC-IDS-2017 dataset contains normal data and the latest common attacks, including DoS, DDoS, Web attacks and penetration attacks, etc. [18], which can better simulate real-world data.

Data set distribution

Table 1 shows the distribution of different types of data after re-organization of NSL-KDD dataset. Table 2 shows the distribution of different types of data after re-organization of CIC-IDS-2017 dataset. Figures 6 and 7 show the heat map of feature correlation between NSL-KDD and CIC-IDS-2017 respectively. Warmer color tune (yellow) indicates higher correlation and vice versa.

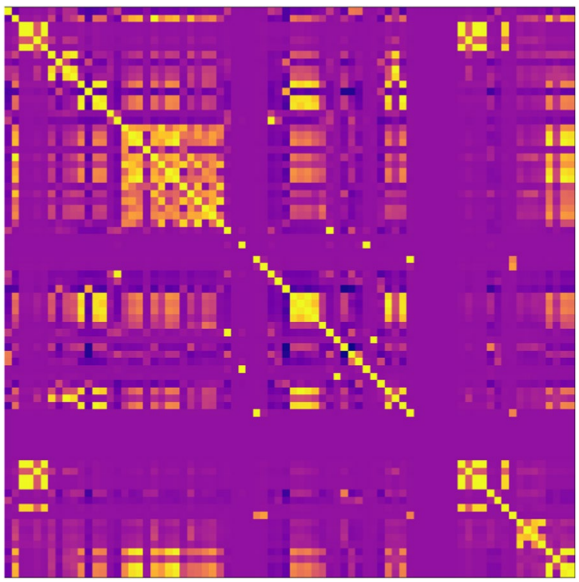

Fig. 6 Heat map of NSL-KDD feature inter-correlation

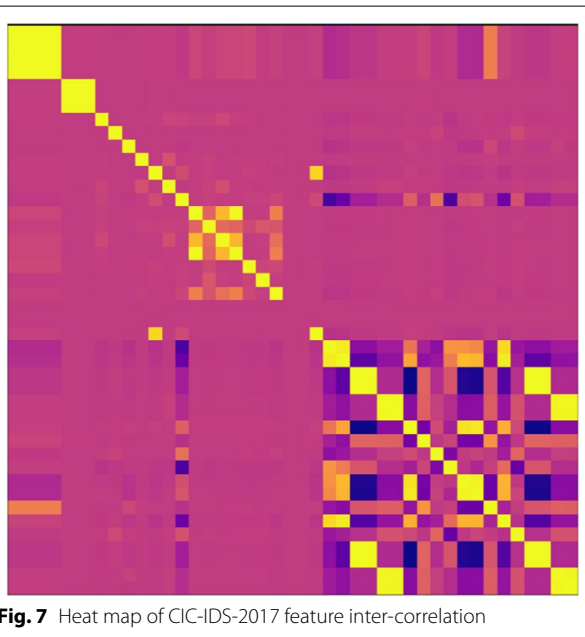

Fig. 7 Heat map of CIC-IDS-2017 feature inter-correlation

The percentage of feature relations with correlation in the NSL-KDD dataset is 10.89%, and the percentage of feature relations with correlation in the CIC-IDS-2017 dataset is 11.79%.

Symbolic feature one-hot encoding

One-hot coding can quantify the symbolic features in the dataset into numeric features, while each feature is independent and of equal distance from each other. Kddcup99 dataset contains three symbolic features: service, flag and protocol type. According to one-hot coding theory, the number of N option degrees of freedom the symbolic features have is equal to the number of dimensional features they can be expanded to. For example, if the service feature has 70 options, it can be expanded to 70 dimensions. Since there are no symbolic features in the CIC-IDS-2017 dataset, one-hot coding is not required.

Numerical feature normalization process

In the dataset, some features take values in the range of 0 to 1 billion, and some take values in the range of 0 to 1. There is a large order of magnitude difference between the features. In order to eliminate this difference, the Min-Max algorithm is used to normalize the numerical features in this paper. The formula of the Min-Max algorithm is shown in (21).

$$x^{'} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{21}$$

x is the value of the input sample. $x_{\min}$ is the minimum value of the sample range. $x_{\max}$ is the maximum value of the sample range.

Model configuration

In this section, the model is configured according to the number of features screened by the feature selection algorithm. Table 3 shows the structural configuration of the encoder in the compression network. The decoder structure is symmetric with the encoder, where the memory capacity N_{m} is set to 50, as shown in Fig. 8, and the whole model is not sensitive to N_{m}.λ_z is the distance coefficients for calculating the anomaly index. λ_1, λ_2, and λ_3 are the coefficients of the anomaly index, shrinkage weight, and minimum value in the objective function respectively.

Figures 9 and 10 show the 3D images of the sample low-dimensional information z_c' without and pretending the memory module, respectively. It can be seen that the memory module has a strong shrinkage constraint effect.

The self-encoder part of the activation function is tanh. The structure of the estimated network is FC(5,10,tanh)-Dropout(0.5)-FC(10,2,softmax). The minimal value used to prevent matrix integrability in the Gaussian mixture model is taken as 1×10^{-12}.

Baseline algorithm

In this paper, some traditional and latest anomaly detection algorithms are considered as baseline.

– Multi-level Support Vector Machine [41] (Multi-level SVM): Wathiq Laftah Al-Yaseen et al. used modified K-means to reduce the 10% KDD99 training dataset by 99.8% and construct a new set of high quality training dataset for training SVM and ELM. They also proposed multi-level model to improve the detection accuracy. The overall accuracy of the calibrated KDD99 dataset reached 95.75%.

Table 3 Encoder structure configuration

Dataset	Number of features	Mem Capacity	Compression Network Encoder structure	λ_z	λ_1	λ_2	λ_3
NSL-KDD	122	50	122-60-30-10-3-mem	−0.5	0.1	0.0025	0.005
CIC-IDS 2017	78	50	78-39-20-10-3-mem				

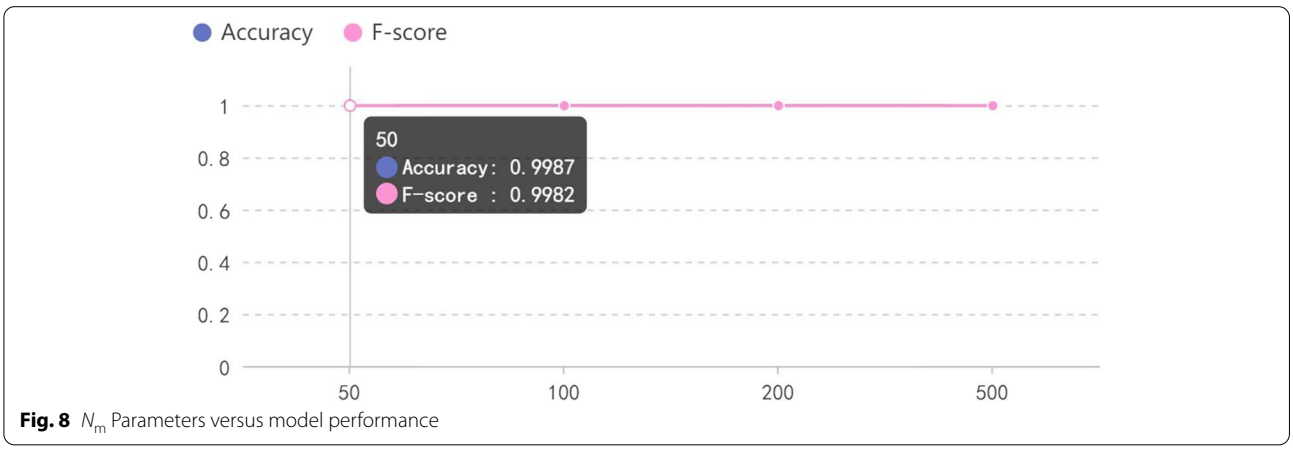

Fig. 8 N_{m} Parameters versus model performance

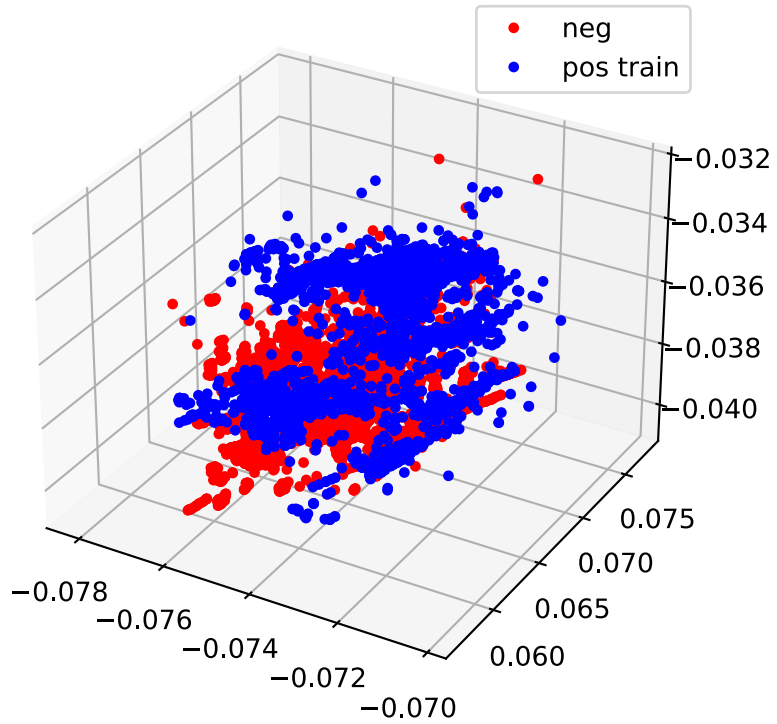

Fig. 9 Sample low-dimensional information of the model without memory module

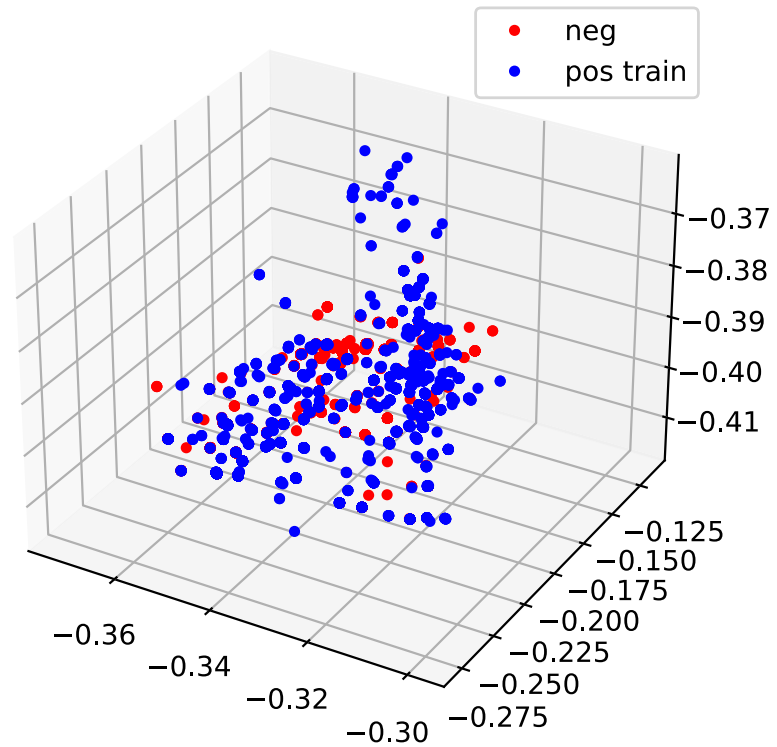

Fig. 10 Sample low-dimensional information of the model with memory module

- Isolation Forest [42]: This algorithm was proposed by Zhihua Zhou's team in 2008 and is widely used in industry for anomaly detection of structured data with its linear time complexity and excellent accuracy.
- Auto-coders [43] (Auto-encoders): H. Choi et al. designed a network intrusion detection system based on auto-encoders and achieved an accuracy of 91.70%.
- Deep auto-coding Gaussian mixture model [35] (DAGMM): in 2018, Bo Zong et al. trained the auto-encoder and Gaussian mixture model jointly to solve the local optimum problem in the detection process. The model jointly optimizes the parameters of the deep auto-encoder and hybrid model in an end-to-end mode and performs excellently on a public benchmark dataset, providing a new idea in the field of anomaly detection.
- Memory Enhanced Deep Auto-encoder [40] (MemAE): Dong Gong et al. use memory modules to enhance auto-encoders. Experiments on various datasets demonstrate the excellent generalization and efficiency of the proposed MemAE.
- Shrinkage Self-Coding Gaussian Mixture Model [44] (CAE-GMM): The authors designed an unsupervised anomaly detection algorithm for CAE-GMM by improving the DAGMM algorithm, which combines the dimensionality reduction of CAE and the density estimation of GMM. The proposed algorithm also reduces the overfitting problem and improves the model generalization ability compared to DAGMM.

Experimental results

This section contains two sets of experiments, in which we use Accuracy, Precision, Recall, and F1-score as the criteria for judging whether performance of the model is good or bad.

In the first set of experiments, this paper uses completely clean data for training and testing, and uses data samples from the normal class as training samples. In each run, using random sampling, we take 50% of the data for training, and the remaining 50% is reserved for testing.

Tables 4 and 5 show the accuracy, precision, recall, and F1 scores of MemAe-gmm-ma and other baseline algorithms for different datasets. In general, MemAe-gmm-ma outperforms the baseline algorithms on all datasets in terms of F1 scores. On NSL-KDD and CIC-IDS-2017, MemAe-gmm-ma achieves 4.47% and 9.77% improvement in F1 scores comparing to the existing methods. Figures 9 and 10 show the low-dimensional distributions of 20,000 test samples. It can be seen that the normal and abnormal samples have high differentiation of the abnormality index after being output by the model in this paper.

Table 4 Comparison of the results of each anomaly detection algorithm under the NSL-KDD dataset

Dataset	Models	Accuracy	Precision	Recall	F-score
NSL-KDD	Multi-level SVM	0.9575	0.9311	0.9517	0.9413
	K-means	0.8944	0.8008	0.7515	0.7754
	Autoencoder	0.9170	0.8745	0.8468	0.8605
	DAGMM	0.8985	0.9214	0.7560	0.8305
	MemAE	0.9636	0.9627	0.9655	0.9641
	CAE-GMM	0.9682	0.9532	0.9578	0.9555
	Model of this paper	0.9987	0.9964	1.0000	0.9982

Table 5 Comparison of the results of each anomaly detection algorithm under the CIC-IDS 2017 dataset

Dataset	Models	Accuracy	Precision	Recall	F-score
CIC-IDS 2017	Isolation Forest	0.89	0.83	0.81	0.82
	K-means	0.87	0.81	0.79	0.80
	OC-SVM	0.90	0.84	0.81	0.82
	DAGMM	0.91	0.84	0.87	0.85
	CAE-GMM	0.95	0.92	0.91	0.91
	Model of this paper	0.9986	0.9978	1.0000	0.9989

Figures 11 and 12: Sample low-dimensional information of the NSL-KDD dataset and CIC-IDS-2017 dataset after model processing in this paper: (1) The horizontal axis indicates the re-construction error (Euclidean distance) caused during the encoding and decoding of the auto-encoder, and the vertical axis indicates the anomaly index of the samples; (2) The red/blue dots are the anomaly/normal samples respectively, and the green solid line indicates the threshold. Each image contains 20,000 samples from the public dataset.

Figures 13 and 14: Sample low-dimensional information of the NSL-KDD dataset and CIC-IDS-2017 dataset after DAGMM model processing.

In the second set of experiments, the main study is how MemAe-gmm-ma responds to contaminated training data. In each run, we mix a certain number of anomalous samples into the normal samples used for model training in advance, with the mixed anomalous samples accounting for c% of the normal samples. Then we retain 50% of the data for model training by random sampling, and the remaining 50% for testing.

Table 6 shows the accuracy, precision, recall, and F1 scores of the training tests on the NSL-KDD dataset containing dirty data. As expected, contaminated

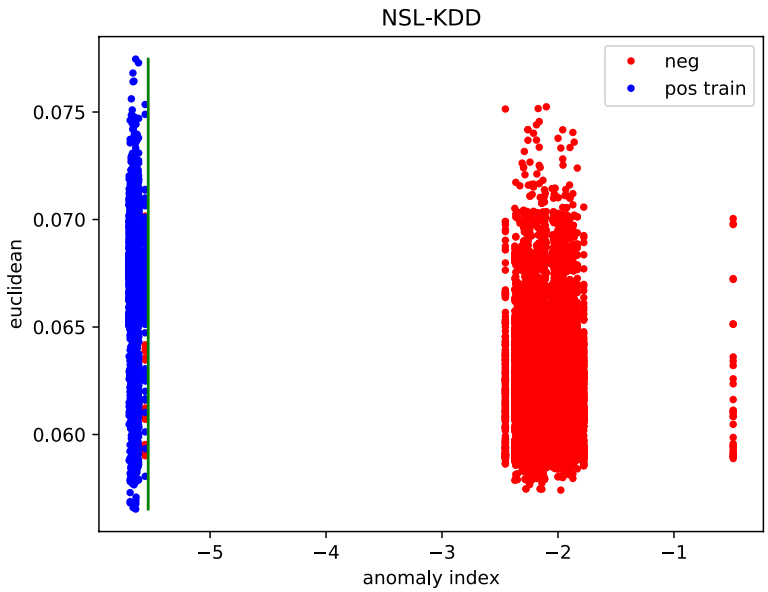

Fig. 11 Low-dimensional information of NSL-KDD sample after the model processing in this paper

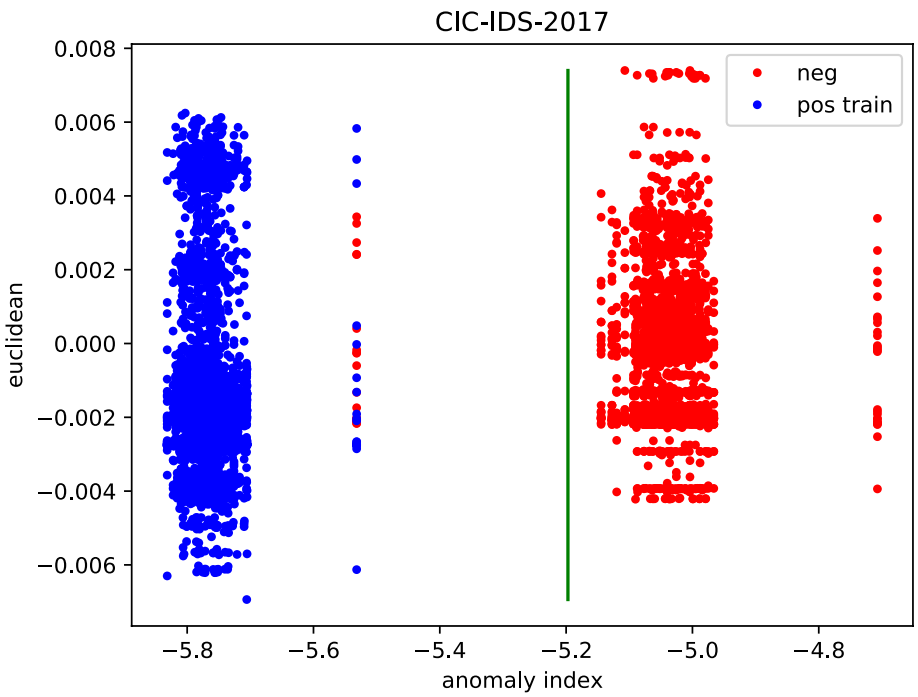

Fig. 12 Low-dimensional information of CIC-IDS-2017 sample after the model processing in this paper

training data negatively affects detection accuracy. As the contamination rate increases from 1% to 5%, each performance metric decreases. The good side is that even with 5% contaminated data, MemAe-gmm-ma still maintains good detection accuracy, reflecting the good robustness of the model.

Figure 15 shows the low-dimensional distribution of the samples tested by the model generated from the completely clean training data. Figure 16 shows the low-dimensional distribution of the samples tested by the model generated from the training data with 5%

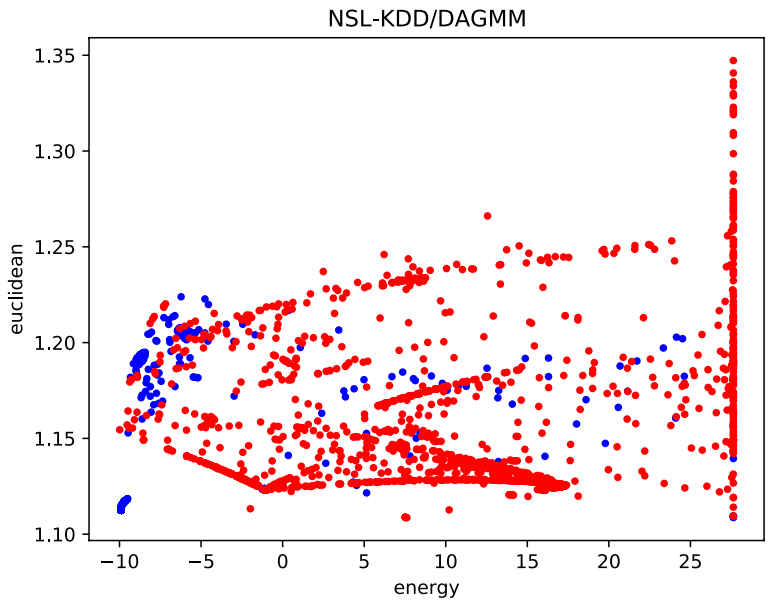

Fig. 13 Low-dimensional information of NSL-KDD sample after DAGMM processing

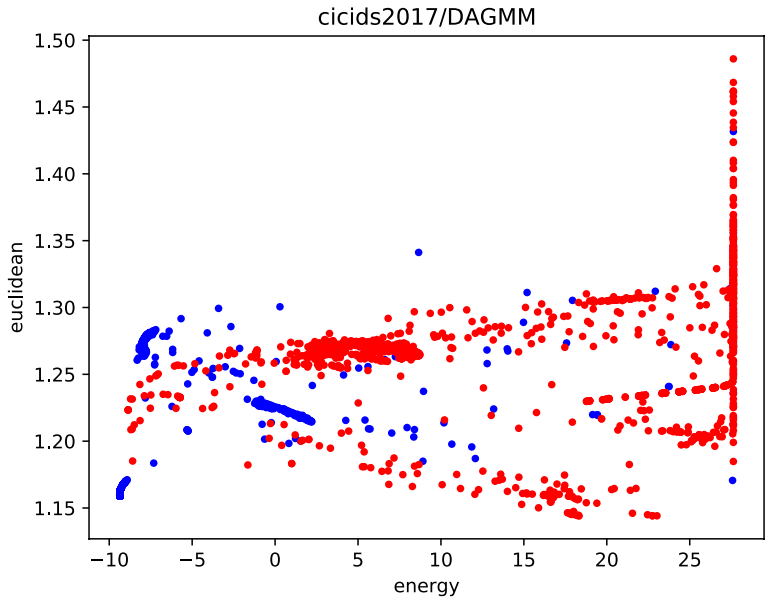

Fig. 14 Low-dimensional information of CIC-IDS-2017 sample after DAGMM processing

dirty data. Both figures are with fixed random seeds during training and testing.

Conclusion

In this paper, we propose an improved auto-coded Gaussian mixture model (MemAe-gmm-ma) for unsupervised anomaly detection. MemAe-gmm-ma consists of two main components: a low-dimensional information network and an anomaly estimation network. The low-dimensional information network uses the features of the auto-encoder to compress samples into a low-dimensional space. And it introduces a memory module to enable the model to better learn the intrinsic relationships of the training samples.

Table 6 Response of the model in this paper with dirty data

Ratio c	Accuracy	Precision	Recall	F-score
0	0.9986	0.9978	1.0000	0.9989
1%	0.9985	0.9964	1.0000	0.9982
2%	0.9983	0.9960	1.0000	0.9980
3%	0.9962	0.9896	1.0000	0.9948
4%	0.9944	0.9877	1.0000	0.9938
5%	0.9896	0.9863	1.0000	0.9931

The anomaly estimation network uses a Gaussian mixture model, in which the sample anomaly indices in the low-dimensional space are further evaluated based on the martingale distance of the samples in this framework.

In the experimental study, MemAe-gmm-ma demonstrates better performance on the public benchmark dataset, with a 4.47% improvement over the MemAe model standard F1 score on the NSL-KDD dataset, and

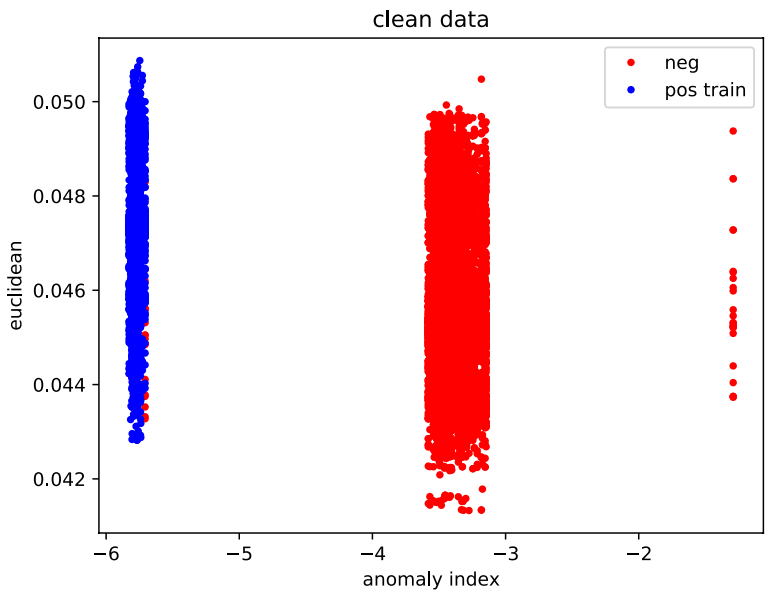

Fig. 15 Low-dimensional distribution with clean training data

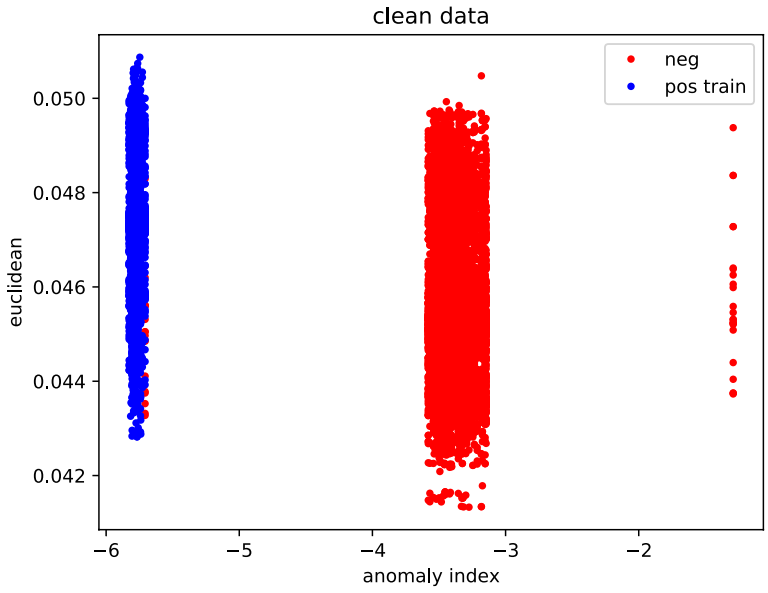

Fig. 16 Low-dimensional distribution with dirty data

a 9.77% improvement over the CAE-GMM model standard F1 score on the CIC-IDS-2017 dataset. It is able to maintain better detection accuracy at 5% of contaminated data, reflecting better redundancy performance of the whole model. A promising direction is proposed for unsupervised anomaly detection of high-dimensional data in cloud security.

Abbreviations

MemAE: Memory Enhanced Deep Auto-encoder; SVM: Support Vector Machine; RF: Random forest; DAGMM: Deep auto-coding Gaussian mixture model; CAE-GMM: Shrinkage Self-Coding Gaussian Mixture Model.

Authors' contributions

Xiangyu Liu was the experimental designer and the executor of the experimental study, completed the data analysis, and wrote the first draft of the paper; Shibing Zhu was the conceptualizer and leader of the project, directed the experimental design and data analysis; Fan Yang was involved in writing and revising the paper; Shengjun Liang was involved in the experimental design and analysis of the experimental results. All authors read and agreed on the final text.

Funding

The research was financially supported by Ministry Key Project (Project No. 1900).

Availability of data and materials

Data are available on the websites: NSL-KDD: https://www.unb.ca/cic/datasets/nsl.html; CIC-IDS-2017: https://www.unb.ca/cic/datasets/ids-2017.html.

Declarations

Competing interests

The authors declare no conflict of interest.

Author details

[1]University of Space Engineering, Beijing 101416, China. [2]Beijing Information Science and Technology University, Beijing 100026, China.

Received: 15 August 2022 Accepted: 13 September 2022
Published online: 29 September 2022

References

1. Sengupta S, Kaulgud V, Sharma VS (2011) Cloud computing security--trends and research directions. In: 2011 IEEE World Congress on Services. IEEE, Washington, DC
2. Iwendi C et al (2020) KeySplitWatermark: zero watermarking algorithm for software protection against cyber-attacks. IEEE Access 8:72650–72660. https://doi.org/10.1109/ACCESS.2020.2988160
3. Rubóczki ES, Rajnai Z (2015) Moving towards cloud security. Interdiscip Description Complex Syst 13(1):9–14
4. Eltaeib T, Islam N (2021) Taxonomy of challenges in cloud security. In: 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), pp 42–46. https://doi.org/10.1109/CSCloud-EdgeCom52276.2021.00018
5. Zheng L, Zhang J (2021) Threats and future development trends to the cloud security. Netinfo Secur 21(10):17–24
6. Peng Z, Xing G, Chen X (2022) A review of the applications and technologies of artificial intelligence in the field of cyber security. Inf Secur Res 8(2):110–116
7. Zimek A, Schubert E, Kriegel H-P (2012) A survey on unsupervised outlier detection in high-dimensional numerical data. Stat Anal Data Min 5(5):363–387
8. Yang B, Fu X, Sidiropoulos ND et al (2017) Towards K-means-friendly spaces: simultaneous deep learning and clustering. In: Proceedings of the 34th international conference on machine learning, pp 3861–3870
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. Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: International conference on learning representations (ICLR)
11. Liu M, Chen W, Liu G (2019) A research on network traffic anomaly detection model based on K-means algorithm. Wirel Interconnect Technol 16(18):25–27
12. Xie B, Dong X, Liang H (2020) Intrusion detection algorithm based on three-branch dynamic threshold K-means clustering. J Zhengzhou Univ Sci Ed 52(02):64–70
13. Xiong L, Póczos B, Schneider J (2011) Group anomaly detection using flexible genre models. In: Advances in neural information processing systems, pp 1071–1079
14. Wang J, Jiang J (2021) Unsupervised deep clustering via adaptive GMM modeling and optimization,Neurocomputing 433:199-211. https://doi.org/10.1016/j.neucom.2020.12.082.
15. Yang X, Huang K, Zhang R (2014) Unsupervised dimensionality reduction for gaussian mixture model. In: Loo CK, Yap KS, Wong KW, Teoh A, Huang K (eds) Neural information processing. ICONIP 2014. Springer, Cham, pp 84-92.
16. Zou C-M, Chen D (2021) Unsupervised anomaly detection method for high-dimensional big data analysis. Comput Sci 48(02):121–127
17. Chen Z, Huang Y, Zou H (2014) Anomaly detection of industrial control system based on outlier mining. Comput Sci 41(5):178–181
18. Wu JF, Jin YD, Tang P (2017) Survey on monitoring techniques for data abnormalities. Comput Sci 44(z11):24–28
19. Jolliffe I (2011) Principal component analysis. In: Lovric M (ed) International encyclopedia of statistical science, vol 28243034. Springer, Cham, pp 1094–1096.
20. Kim J, Grauman K (2009) Observing locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In: The IEEE conference on computer vision and pattern recognition (CVPR)
21. Zeng JH (2018) A kernel PCA-based algorithm for network traffic anomaly detection. Comput Appl Softw 35(03):140–144
22. Veeramachaneni K, Arnaldo I, Korrapati V, Bassias C, Li K (2016) AI^2 : training a big data machine to defend. In: 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), pp 49–54
23. Shyu ML, Chen SC, Sarinnapakorn K, Chang L (2003) A novel anomaly detection scheme based on principal component classifier. In: IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03), IEEE, Melbourne, FL.
24. Hoang DH, Nguyen HD (2018) A PCA-based method for IoT network traffic anomaly detection. In: 2018 20th international conference on advanced communication technology (ICACT), pp 381–386. https://doi.org/10.23919/ICACT.2018.8323766
25. Zhai S, Cheng Y, Lu W, Zhang Z (2016) Deep structured energy based models for anomaly detection. In: International conference on machine learning (ICML), pp 1100–1109
26. Zhou C, Paffenroth RC (2017) Anomaly Detection with Robust Deep Autoencoders. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, Halifax, NS.
27. Zhao Y, Deng B, Shen C, Liu Y, Lu H, Hua X-S (2017) Spatio-temporal autoencoder for video anomaly detection. In: Proceedings of the 25th ACM international conference on Multimedia, Association for Computing Machinery, Mountain View, CA.
28. Lu C, Shi J, Jia J (2013) Abnormal event detection at 150 fps in matlab. In: The IEEE international conference on computer vision (ICCV), pp 2720–2727
29. Chen Y, Zhou XS, Huang TS (2001) One-class svm for learning in image retrieval. In: International conference on image processing, vol 1, pp 34–37

30. Williams G, Baxter R, He H, Hawkins S (2002) A comparative study of RNN for outlier detection in data mining. In: Proceedings ofICDM02, pp 709–712
31. Song Q, Hu WJ, Xie WF (2002) Robust support vector machine with bullet hole image classification. IEEE Trans Syst Man Cybern 32:440–448
32. Paulik M (2013) Lattice-based training of bottleneck feature extraction neural networks. In: Interspeech, pp 89–93
33. Variani E, McDermott E, Heigold G (2015) A gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture. In: ICASSP, pp 4270–4274
34. Zhang C, Woodland PC (2017) Joint optimisation of tandem systems using gaussian mixture density neural network discriminative sequence training. In: ICASSP, pp 5015–5019
35. Zong B, Song Q, Min MR et al (2018) Deep autoencoding Gaussian mixture model for un-supervised anomaly detection. In: International conference on learning representations
36. Hu N, Fang LT, Qin CY (2020) An unsupervised intrusion detection method based on random forest and deep self-coding Gaussian mixture model. Cyberspace Secur 11(08):40–44+50
37. Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T (2016) One-shot learning with memory-augmented neural networks. In: International conference on machine learning (ICML)
38. Graves A, Wayne G, Danihelka I (2014) Neural turing machines. arXiv preprint arXiv:1410.5401
39. Weston J, Chopra S, Bordes A (2015) Memory networks. In: International conference on learning representations (ICLR)
40. Gong D, Liu L, Le V et al (2019) Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE international conference on computer vision, pp 1705–1714
41. Al-Yaseen WL, Othman ZA, Nazri MZA (2017) Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system. Expert Syst Appl 67:296–303
42. Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, IEEE, Pisa.
43. Choi, H, Kim, M, Lee, G et al (2019) Unsupervised learning approach for network intrusion detection system using autoencoders. J Supercomput 75, 5597–5621. https://doi.org/10.1007/s11227-019-02805-w.
44. Tang C (2021) Research on network traffic anomaly detection based on unsupervised learning. Dissertation, Southwest University of Science and Technology, Chongqing.

Publisher's Note