Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Attack detection model for BCoT based on contrastive variational autoencoder and metric learning

Chunwang Wu[1,2], Xiaolei Liu[3], Kangyi Ding[3], Bangzhou Xin[3], Jiazhong Lu[2], Jiayong Liu[1*] and Cheng Huang[1]

Abstract

With development of blockchain technology, clouding computing and Internet of Things (IoT), blockchain and cloud of things (BCoT) has become development tendency. But the security has become the most development hinder of BCoT. Attack detection model is a crucial part of attack revelation mechanism for BCoT. As a consequence, attack detection model has received more concerned. Due to the great diversity and variation of network attacks aiming to BCoT, tradition attack detection models are not suitable for BCoT. In this paper, we propose a novel attack detection model for BCoT, denoted as cVAE-DML. The novel model is based on contrastive variational autoencoder (cVAE) and deep metric learning (DML). By training the cVAE, the proposed model generates private features for attack traffic information as well as shared features between attack traffic information and normal traffic information. Based on those generated features, the proposed model can generate representative new samples to balance the training dataset. At last, the decoder of cVAE is connected to the deep metric learning network to detect attack aiming to BCoT. The efficiency of cVAE-DML is verified using the CIC-IDS 2017 dataset and CSE-CIC-IDS 2018 dataset. The results show that cVAE-DML can improve attack detection efficiency even under the condition of unbalanced samples.

Keywords Attack detection, BCoT, Metric learning, Contrastive variational autoencoder, Oversample

Introduction

In recent years, the technology of blockchain, cloud computing and IoT has been applied in many aspects of our lives, such as finance, government services and so on [1, 2]. As a consequence, the blockchain and cloud of things (BCoT), which is the integration of blockchain technology with cloud computing assisted Internet of Things (IoT), has received more attention [3].

The BCoT make used of advantages of blockchain technology, cloud computing and IoT to cover the shortage of those three technologies. Internet of Things (IoT) utilize internet technology to interconnection among devices to achieve collection, analysis and interaction of data. How to efficiently manage the resource is key for IoT. So, the BCoT make source shared pool provided by cloud computing to achieve efficient management. The data security is most hinder for IoT. The BCoT used the blockchain technology, such as distributed ledger and smart contract, to ensure security of data in IoT [4]. So, the BCoT has become development tendency.

The security has become the most development hinder of the BCoT. Although the BCoT used blockchain technology to security of data, the vulnerabilities of IoT and blockchain is still threaten the security of BCoT [5]. For example, the interconnection of some devices in BCoT applies wireless

*Correspondence:
Jiayong Liu
ljy@scu.edu.cn
[1] School of Cyber Science and Engineering, Sichuan University, Chengdu, China
[2] School of Cybersecurity, Chengdu University of Information Technology, Chengdu, China
[3] Institute of Computer Application, China Academy of Engineering Physics, Mianyang, China

technology. As a consequence, the security hole of wireless technology also threatens security of BCoT. BCoT utilizes blockchain technology to ensure the security of data. In recent years, The number of BCoT attack incidents has grown exponentially in recent years. As one of the most critical components of BCoT security, attack detection model for BCoT has been paid more and more attention recently [6]. Existing attack detection model make used of the attack features containing in attack signature database to detect attack. Due to diversities and variations of attack against BCoT, it is difficult to extract attack features to construct attack signature database. As a consequence, existing attack detection model cannot be directly applied in the security field for BCoT [7].

Meanwhile, machine learning technology has made significant progress in many fields. It can learn knowledge hiding in training datasets. Therefore, the machine learning-based attack detection model is widely applied in BCoT security [8, 9]. The attack detection system is essentially classification model that can distinguish attacks from regular visits [10]. The training dataset for the attack detection system consists of the data that records the information of ordinary and attacking visits. For the training dataset, three problems result in inefficiency for the machine learning-based attack detection system. The three problems are shown as following:

Firstly, the problem of distribution imbalance between normal and attack traffic data exists in the training dataset. In practice, compared with the frequency of normal traffic devices in BCoT, the frequency of attack traffic is relatively low. For training dataset, the amount of data recording normal visiting is much more than that recording attack traffic. As a result the distribution imbalance leads to the attack detection results being biased toward the frequent visits and inefficiency in detecting new intrusion attack [11].

Secondly, attack traffic data contains features both relevant and irrelevant attack information. The attack traffic data consists of several attributes. The information recorded in some attributer is relevant with attack. Those information can be applied to distinguish attack traffic from normal visiting. The information recorded in the other attributes is irrelevant attack. Traditional attack detection model based on machine learning technology can extract the information relevant with attack to distinguish attack traffic from normal traffic. But the information irrelevant with attack can make the model inefficient [12].

Thirdly, the distribution of normal traffic data is more concentrated than that of attack traffic data, which is scattered in a wide area. In essence, because traditional machine learning-based attack detection systems are classification models based on clustering technology, they cannot accurately distinguish attack visiting from normal traffic. As a result, attack detection systems based on nonlinear models have better efficiency than traditional attack detection systems [13].

In recent years, there has been much significant research on the above three problems regarding attack detection based on machine learning. Some researchers designed an attack detection system based on setting different weights for a single or a set of complex classified samples. Because of the high diversity and variation of the network intrusion attack, it is difficult to set different weights for all attack samples [3, 7]. Some researchers apply oversampling technology to balance training datasets to generate new attack samples. Then, the balanced training dataset is used to train a machine learning-based attack detection system for BCoT. However, generation of new attack samples cannot fully exploit the information hidden in the known samples. As a result, those generated attack samples provide low-level improvement in the efficiency of the attack detection system [14]. Some researchers combine generative models, such as VAE and GAN, with oversample technology to generate new attack samples to improve the efficiency of the attack detection system. However, the generated new attack samples ignore the hidden information in the traffic data. In conclusion, how to efficiently extract features from attack traffic is key of construction attack detection model.

The cVAE combines contrastive learning with VAE to identify to enhance its salient features. The cVAE is trained in two related but unpaired datasets. And the cVAE explicitly models latent features shared between the datasets and the rich potential features of one dataset relative to another, enabling the algorithm to separate and enhance significant possible features. Therefore, the cVAE can be used as a generative model to generate new samples with various levels of salient and irrelevant latent variables [15].

The triplet network as one of the most popular metric learning technologies is successfully applied in many fields [16]. The triplet network is created based on a triplet of samples. A triplet consists of three samples selected from the training dataset. The first one is to denote an anchor. The second one belongs to the same class with the anchor and is denoted as the positive sample. The third one belongs to the opposite class with the anchor, denoted as the negative sample. The triplet network takes triples as input, and it learns an embedding space where the distance between samples labeled with opposite classes is more significant than between the samples labeled with the same class [17]. Finally, the triplet network can distinguish attack traffic from normal based on the learned embedding space. Traditional triplet networks suffer from poor convergence because of the random selection of samples from the training dataset to construct a triplet.

In this paper, we propose a novel attack detection model to distinguish attack traffic from normal traffic based on

the binary classification model, denoted as cVAE-DML. The novel make used of the cVAE to generated new sample to solve imbalance problem. The cVAE model two types of features. The first type is features shared between normal traffic and attack traffic. The second type is private features for attack traffic. The two type features are not only used to generate samples to solve imbalance problem, but also improve efficiency of distinguish attack traffic from normal traffic. Based these two types of features, the cVAE-DML utilizes oversample technology to generate new attack samples to solve the imbalance problem in the training dataset. Based on the trained cVAE and generated samples, cVAE-DML utilizes triplet network, one of the most popular metric learning technologies to achieve attack detection. Existing attacking detection model extract attack features by analysis attack traffic. But attack traffic contains relevant and irrelevant attack information. Existing attack detection model cannot distinguish private information of attack from attack traffic. As a consequence, compared with existing attacking detection model, cVAE-DML can efficiently extract attack features for detecting attack. In short, the main contributions are listed below.

Firstly, a novel attack detection system for IoT, called cVAE-DML, is proposed based on the cVAE and deep metric learning. To solve the imbalance problem in the training dataset, the new attack detection systems combine oversample technology with the cVAE to generate new attack samples to balance the training dataset. The cVAE introduce contrastive learning to VAE to enhance latent features of attack traffic. As a consequence, the generated samples have more diverse. And then, the decoder of the cVAE is fully connected with the triplet network to embedding space, where the distance between samples labeled with opposite classes is more significant than between samples labeled with the same class. Finally, the cVAE-DML can apply the distance between samples to the detection attack.

Secondly, two public datasets, the CIC-IDS 2017 dataset, and the CSE-CIC-IDS 2018 dataset, are utilized to verify the efficiency of the cVAE-DML. The experiment results show that the cVAE-DML can improve the efficiency of attack detection under the condition of unbalanced samples compared with traditional attack detection.

The remaining of this paper is organized as follows: the related works are presented in Sect. 2. The details of the cVAE-DML are described in Sect. 3. The details of the experiment are introduced in Sect. 4.

Related works

Recently, the attack detection model despite its significant progress, have still meet challenges including distribution imbalance problem in training datasets and the high diversity and variation of network attack [14]. According to the detection mechanism, the attack detection system can be classified into four categories: signature-based, anomaly-based, specification-based, and hybrid detection systems, the first being the most widely used approaches [14, 18, 19].

A signature-based attack detection system maintains the attack signature database. For a network visit, if the feature of this network visiting matches the pattern stored in the database, the network visit is detected as an intrusion attack. A lightweight signature-based attack detection system was proposed for IoT [20]. There are four-layered architectures for the attack detection system: signature generator, pattern generator, attack detection engine, and output engine. Liu et al. [21] proposed a novel attack detection system based on the artificial immune system. This attack detection system applies immune cells to stored attack features. Rebbah et al. [22] combined the attack detection system with Cloud technology. Such a detection system is based on the temporal and spatial profiles calculated for each client according to the data of its request. The attack without any documented analysis and studies the client provides.

An anomaly-based attack detection system can identify an unknown activity by comparing it with a normal behavior profile and then classifying it as normal or abnormal. Unlike signature-based attack detection systems, anomaly-based attack detection systems effectively identify new intrusion attack [18]. Larijani et al. proposed an attack detection system based on a random neural network for IoT [23]. Roy et al. proposed a machine learning-based two-layer hierarchical attack detection mechanism to detect intrusion attack while satisfying the IoT resource constraint [24]. This mechanism applies a fog layer to offload networking and computation overhead from the cloud, and it provides the first-line defense closer to the physical IoT devices. Yin et al. proposed an attack detection system based on recurrent neural networks and studied the performance of this system in binary and multiclass classification [25]. For the attacks with lower detection accuracy in the attack detection systems, Vinayakumar et al. proposed a bidirectional long- and short-term memory-based attack detection system [26]. Sun et al. applied the combined CNN and LSTM to extract spatiotemporal features of traffic data to detect intrusion attack [27]. Zhang et al. applied DBN to predict the kind of traffic data. And Zhang applies a genetic algorithm to optimize DBN [28]. Ge et al. used forward neural networks based on multi-classification to construct the attack detection system and transferred learning technology to encoder classification features [29]. Based on GANS and fog architecture, the attack detection system was proposed to detect unknown attacks and conquer the acquisition

challenge [30]. The authors used the AE to extract attack features. The CNN and MLP are applied to detect the intrusion attack [31–33]. Meanwhile, the authors applied DNN to learn attack features. Then, learned features were applied to detect the features [26, 34]. Some research on the attack detection system has focused on model generation through intensive analysis of feature engineering instead of considering the real environment. They have limitations in applying the previous methods for a real network environment to detect real-time attacks.

Traditional machine learning technologies are applied in attack detection systems for IoT. Li et al. combined an attack detection system with a K-Nearest Neighbor (KNN) classification algorithm to detect intrusion attack in wireless sensor networks [35]. Shapoorifard et al. proposed an attack detection system based on the KNN classification algorithm and the K-MEANS algorithm [36].

Through the above analysis, we can obtain conclusions as following. The efficiency of signature-based attack detection model depends on signature database. The attack traffic data is diversity. As a consequence, it is difficult to efficiently extract attack signature for traffic data. Compared with signature-based attack detection model, anomaly-based and specification-based attack detection model use machine learning to learn attack feature. As a consequence, those attack detection model effectively identify new intrusion attack. But the three problems mentioned in Introduction is hinder for the efficiency.

Method

In this section, the details of the cVAE-DML will be described. The cVAE-DML essentially construct binary classification model to distinguish attack traffic from normal traffic. As a result, all the attack classes are assigned to the same label regardless of the attack type. There are four modules in the cVAE-DML: the data preprocessing module, the oversampling module, the training module, and the predictive module. The original data consists of the values of multiple attributes. Firstly, the values of all the attributes in the original data are encoded in the data preprocessing module. The training dataset is composed of all the encoded data. And then, the training dataset is fed to the oversampling module to generate new attack samples. In the training dataset, the amount of normal traffic data far exceeds the amount of the attack traffic data. So, all the generated new attack samples are added to the attack traffic data to alleviate the imbalance problem in the training dataset. Thirdly, all the training datasets are fed to the training module to train a binary classification model that can distinguish the attack traffic from the normal visiting. Finally, new visiting data is put into a trained classification model to judge whether it is an intrusion attack.The meaning of all the abbreviation is shown in Table 1.

Data preprocessing

There are two kinds of attributes in the original data: the category attribute and the value attribute. The strategy of one-hot encoding is used for the category attribute. For instance, the encoders for the category attribute 'protocol', including 'TCP', 'UDP', and 'ICMP', are [1,0,0], [0,1,0], and [0,0,1]. For the value attributes, the value of each attribute is a continuous variable. For the value attributes, Eq. (1) is applied to map the value of each attribute to the range of [0,1], where it is denoted as the value of the attribute, and *Max* and *Min* are represented as the max value and min value of this attribute, respectively. Finally, all the preprocessed data are

Table 1 The meaning of all the abbreviation

ID	Abbreviation	Meaning
1	x_i	The value of value attributes before preprocessing
2	Min	The min value of value attribute
3	Max	The max value of value attribute
4	x^n	The set consist of all the samples belonging to normal traffic
5	x^a	The set consist of all the samples belonging to attack traffic
6	s	private latent variables for attack traffic
7	z	latent variable irrelevant with detection attack
8	q_{Φ_s}	the encoder of cVAE
9	q_{Φ_z}	the encoder of cVAE
10	f_θ	the decoder of cVAE
11	s_{X^a}	the output of q_{Φ_s}, whose input is x^a
12	z_{X^a}	the output of q_{Φ_z}, whose input is x^a
13	z_{X^n}	the output of q_{Φ_z}, whose input is x^n
14	x^+	triplet positive counterpart
15	x^-	triplet negnitive counterpart

collected into a training dataset. Data preprocessing is beneficial for improving the speed and stability of training. To describe clearly the details of the cVAE-DML, we call the preprocessed data the training dataset.

$$x_i = \frac{x_i - Min}{Max - Min} \tag{1}$$

Oversampling

As mentioned in the introduction, the seriously distribution imbalance in the training dataset reduce efficiency of traditional attack detection model. The cVAE-DML applies the cVAE to generate new attack samples to balance the training dataset. Traditional oversampling technology based on network traffic data extracts attack features from attack data. Based on those features, the traditional technology generates new samples. For total attack data, the main information, which can efficiently improve the effectiveness of attack detection, is only portion of attack data. As a consequence, attack features extract by the traditional technology contain relevant and irrelevant attack information. That irrelevant information can reduce the effectiveness of attack detection. The cVAE model two types of features. The first type is features shared between normal traffic and attack traffic. The second type is private features for attack traffic. belonging to normal traffic. The second part, denoted as x^a, consists of all the samples belonging to attack traffic. And then all the divided training datasets are fed to the cVAE. There are two latent variables in the cVAE. The first latent variable, denoted as s, is salient for detecting attack against the BCoT. This latent variable is private latent variables for x^a. The second latent variable, denoted as z, is irrelevant variable with detection intrusion attack and shared between x^n and x^a. Two conditional distributions, which are followed by x^a and x^n, are shown as Eqs. (2) and (3), respectively. When the set x^n is fed to the encoder q_{Φ_z}, the output of the encoder q_{Φ_z} is z_{x^n}. And then the decoder of f_θ can use concatenating z_{x^n} with 0 to reconstruct set x^n. When the set x^a is fed to two encoders q_{Φ_s} and q_{Φ_z}, the output of those two encoders q_{Φ_s} and q_{Φ_z} are z_{x^a} and s_{x^a}, respectively. Finally, the decoder of f_θ can use concatenating z_{x^a} with s_{x^a} to reconstruct set x^a. Therefore, we can get the conclusion that concatenating z_{x^a} with 0 is latent variable for x^n. And concatenating z_{x^a} with s_{x^a} is latent variable for x^a. The lower bounds of likelihood of x^n and x^a are shown as Eqs. (4) and (5), respectively. The cVAE is trained by the maximizing sum of Eqs. (4) and (5). The detail of training the cVAE is shown in Dai et al. [15].

$$x^a \sim f_\theta(x|z,s) \tag{2}$$

$$x^n \sim f_\theta(x|0,z) \tag{3}$$

$$L\left(x_i^a\right) \geq E_{q_{\Phi_z}(z),q_{\Phi_s}(s)}\left[f_\theta\left(x_i^a|s,z\right)\right] - KL\left(q_{\Phi_s}\left(s|x_i^a\right)||p(s)\right) - KL\left(q_{\Phi_z}\left(z|x_i^a\right)||p(z)\right) \tag{4}$$

And the cVAE introduce contrastive learning to VAE to enhance latent features of attack traffic. As a consequence, the new sample generated by the cVAE-DML can efficiently improve effectiveness of attack detection. Here are the two steps for oversampling.

The first step is to train the cVAE based on the attack detection dataset. The structure of cVAE is shown in Fig. 1. Firstly, the training dataset is divided into two parts. The first part, denoted as x^n, consists of all the samples

$$L\left(x_i^n\right) \geq E_{q_{\Phi_z}(z)}\left[f_\theta\left(x_i^n|0,z\right)\right] - KL\left(q_{\Phi_z}\left(z|x_i^n\right)||p(z)\right) \tag{5}$$

The second step is to generate new attack sampling. As is mentioned in the part of Introduction, the amount of data recorded for the normal visiting information is much greater than that of attack traffic. Consequently, there are imbalance problems in the training dataset.

All the samples in the training dataset are divided into three kinds, namely, safe samples, dangerous

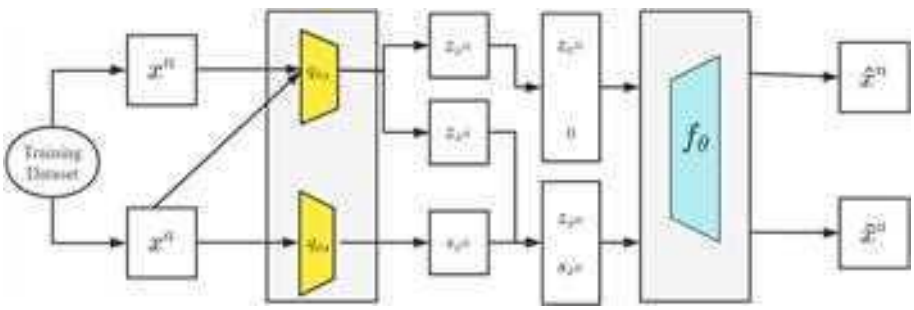

Fig. 1 Structure of cVAE

samples and noisy samples [11]. Oversampling with dangerous samples is helpful to improve the efficiency of attack detection, because dangerous samples belong to attack samples. Based on dangerous samples, synthetic Minority Oversampling Technique (SMOTE)is applied to generate new attack samples. The pseudo code of this specific algorithm is shown in Algorithm 1.

The detail of oversampling is described as follows: Firstly, the training dataset is applied to train the cVAE. Secondly, the training data is fed to the encoder of the cVAE, as it shown in Line 5 of Algorithm 1. At this time, for $(l_i, y_i) \in Z$, if y_i equals to attack, l_i equals to the concatenating $q_{\Phi_s}(x_i)$ with $q_{\Phi_z}(x_i)$. If y_i equals to normal, l_i equals to the concatenating $q_{\Phi_s}(x_i)$ with 0. According to the value l_i, Euclidean distance is used to obtain all the dangerous samples in Z^a shown from Lines 8 to 11 of Algorithm 1. Finally, SMOTE is used to generate new attack samples. And the generated samples are used to balance the training dataset

Algorithm 1. Oversampling

Input: T: Training dataset $\{(x_i, y_i)\}_{i=1}^N, y_i \in \{attack, normal\}$
K: number of the nearest neighbors
s: number of generating samples
Output: Synthetic attack samples set
1. $B' = \emptyset$, B $= \emptyset, T' = \emptyset$
2. Initialize the cVAE network framework
3. for each minibatch d in T:
4. Training cVAE
5. $Z = \{(l_i, y_i)\}_{i=1}^N \leftarrow$ feed T to the encoder q_{Φ_s} and $q_{\Phi_z}$of cVAE
6. $Z^a \leftarrow \{(l_i, y_i) | (l_i, y_i) \in Z, y_i = attack\}$
7. $Z^n \leftarrow \{(l_i, y_i) | (l_i, y_i) \in Z, y_i = normal\}$
8. for each element (l_i, y_i) in Z^a:
9. m ←the number of the samples belonging to Z^n and the K nearest neighbors of (l_i, y_i).
10. if $\frac{K}{2} \leq m \leq K$
11. Add (l_i, y_i) to B.
12. for each element b in B do:
13. add K the nearest neighbors of b in Z^a to B'
14. for i from 1 to s do:
15. Choose a random sample b' from B'
16. $b'' \longleftarrow b + random(0,1) * (b - b')$
17. Add $f_\theta(b'')$ to T'
18. Return T $\leftarrow T' \cup T$

Training

The cVAE-DML combined cVAE with triplet networks, which is one of the main types of deep metric learning, to construct the attack detection kmodel. Triplet use triplets of samples to learn an embedding space, where distances between samples labelled with opposite classes are greater than the distance labelled with the same class. Each triple (x, x^+, x^-) consists of anchor sample x (a training sample choose from the training dataset), positive counterpart of x and negative counterpart of x. The positive counterpart of x, denoted as x^+, is the sample which belongs to the same class with x. The negative counterpart of x, denoted as x^-, is the sample which belongs to different class with x.

The cVAE-DML uses the cVAE to construct triplets. And then the triplets are used to train triplet networks. The pseudo code of the training stage is shown in Algorithm 2. The detail is shown in Fig. 2. For each sample $x = (x_i, y_i)$ in training dataset, the triplet anchor sample

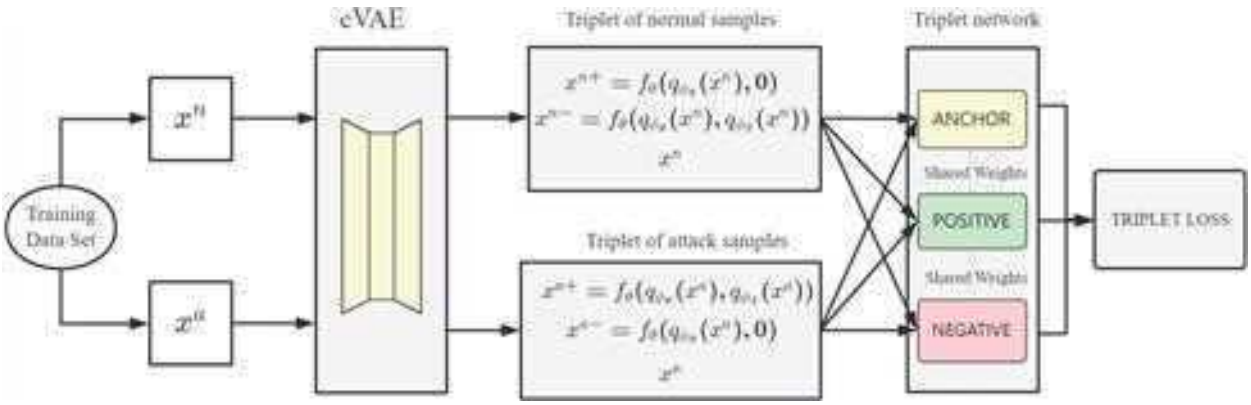

Fig. 2 The cVAE-DML training stage

is x. The negative and positive counterpart are obtained as follows.

If sample x belongs to attack sample, that is, $y_i = attack$, the decoder f_θ of the cVAE can use concatenating $q_{\Phi_s}(x_i)$ with $q_{\Phi_z}(x_i)$ to reconstruct x_i. As a consequence, for sample x, the triplet positive counterpart x_i^+ is $f_\theta\left(q_{\Phi_s}(x_i), q_{\Phi_z}(x_i)\right)$, and the negative counterpart x_i^- is $f_\theta\left(q_{\Phi_s}(x_i), 0\right)$.

If sample x belongs to normal sample, that is, $y_i = normal$, the decoder f_θ of the cVAE can use concatenating $q_{\Phi_s}(x_i)$ and 0 to reconstruct x_i. As a consequence, for sample x, the triplet positive counterpart x_i^+ is $f_\theta\left(q_{\Phi_s}(x_i), 0\right)$, and the negative counterpart x_i^- is $f_\theta\left(q_{\Phi_s}(x_i), q_{\Phi_z}(x_i)\right)$.

When all the triplets are constructed, those triplets are used to train the triplet networks. The triplets consist of three networks. And those networks share common weights and structure. The soft-margin triplet loss [17] is used as the loss function and the formula is as follows:

$$L_{soft} = \sum_{x \in X} \ln\left(1 + exp\left(\|\varphi(x) - \varphi(x^+)\|^2 - \|\varphi(x) - \varphi(x^-)\|^2\right)\right) \quad (6)$$

where x is anchor, x^+ is the positive counterpart of x, x^- is the negative counterpart.

When training the triplet has finished, distances between the samples and the positive counterpart are greater than distance between the samples and the negative counterpart.

Algorithm 2. Training stage

Input: cVAE,
T: Training dataset $\{(x_i, y_i)\}_{i=1}^N, y_i \in \{attack, normal\}$
Output: φ the learned attack detection model
1. D = ϕ
2. For each $(x_i, y_i) \in T$ do:
3. if $y_i = attack$ then
4. $x_i^+ = f_\theta\left(q_{\Phi_s}(x_i), q_{\Phi_z}(x_i)\right)$
5. $x_i^- = f_\theta\left(q_{\Phi_s}(x_i), 0\right)$
6. if $y_i = normal$ then
7. $x_i^+ = f_\theta\left(q_{\Phi_s}(x_i), 0\right)$
8. $x_i^- = f_\theta\left(q_{\Phi_s}(x_i), q_{\Phi_z}(x_i)\right)$
9. d = $[x_i, x_i^+, x_i^-]$
10. D = D ∪ d
11. φ =trainTripletNetork(D)
12. return φ

Predictive stage

The predictive stage is described in Algorithm 3. The cVAE-DML uses the distance from $\varphi(x)$ to $\varphi(x_1)$ and $\varphi(x_2)$ to predict the classification for x. If $\mathrm{d}(x_1) < \mathrm{d}(x_2)$, then x is classified as an attack traffic. Otherwise, x is classified as a normal visiting.

Algorithm 3. Predictive stage

Input:
x: a new sample
φ: the attack detection system
output: the predicted class y

1. $x_1 = f_\theta\left(q_{\Phi_s}(x), q_{\Phi_z}(x)\right)$
2. $x_2 = f_\theta\left(q_{\Phi_s}(x), 0\right)$
3. $\mathrm{d}(x_1) = \|\varphi(x) - \varphi(x_1)\|^2$
4. $\mathrm{d}(x_2) = \|\varphi(x) - \varphi(x_2)\|^2$
5. if $\mathrm{d}(x_1) < \mathrm{d}(x_2)$:
6. y is attack
7. else:
8. y is normal

Experiment

This paper proposed a novel attack detection system for IoT, denoted as cVAE-DML. And two public datasets were applied, including CIC-IDS 2017 [37] and CSE-CIC-IDS 2018 [38], to verify the efficiency of the cVAE-DML. The experiment results showed that the efficiency of the cVAE-DML is better than traditional attack detection. In this section, the detail of experiments will be introduced.

Dataset description

CIC-IDS 2017 was released as a public dataset for attack detection in 2017 and was collected by the Canadian Institute of Cybersecurity for a total of 5 days. The CIC-IDS 2017 has more species diversities than KDDCUP 99 and NSLKDD. For CIC-IDS 2017, the researchers apply CICFlowMeter to construct real network environment. And based on networking protocols such as HTTP and FTP, researchers created abstract behaviors of 25 users and collected traffic data from Monday to Friday. The traffic data on Monday belongs to normal traffic data. The traffic data from Tuesday to Friday belongs to anomaly traffic data. The dataset contains 2,273,097 normal samples and 557,646 attack samples. The attack includes Brute Force FTP, Brute Force SSH, Dos, Heartbleed, Web Attacks, Infiltration, Botnets and DDos. Because Monday is a normal day, all training samples randomly are chosen from the other four days. CSE-CIC-IDS 2018 dataset is heterogeneous detection data in the real world. Compared with the CIC-IDS 2017, it is more complex. The CSE-CIC-IDS 2018 dataset was collected in a 10-day IoT environment and there are a lot of missing values and redundant features. Those datasets are filtered to a certain extent. The sampling rate of Benign is 50%. And 80% of Benign and 20% of attacks are selected. The rate between training data and testing data is 4:1. That sampling method is applied in [11–13]. After finishing sampling, the cVAE is applied to generated new attack samples.

Evaluation metric

The following indicators are appliedto verify the efficiency of the cVAE-DML and those indicators are shown from Eqs. (7) to (10). And the cVAE-DML network structure parameters of the cVAE-DML are shown in Table 2.

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

$$\mathrm{F1-score} = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

$$\mathrm{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

Balance training dataset

As is mentioned in Introdction, the serious imbalanced class distribution in the training dataset causess

Table 2 cVAE-DML network structure

Dataset	Structure of Encoder q_{Φ_s} for cVAE	Structure of Encoder q_{Φ_z} for cVAE	Structure of Decoder f_θ for cVAE	The Structure of the Triplet networks
CIC-IDS 2017	85–128-128–64-16	85–128-128–64-16	32–64-128–128-85	85–512-256–128
CSE-CIC-IDS 2018	79–128-128–64-16	79–128-128–64-16	32–64-128–128-79	79–512-256–128
		Predicted as Normal	**Predicted as Intrusion attack**	
	Actual Normal	TN	FP	
	Actual Intrusion attack	FN	TP	

traditional attack detection system not to work properly. So, the cVAE-DML combines oversampling technology with the cVAE to generate new attack samples to balance the training dataset. For each class attack, we search the hidden variables of dangerous samples to synthesize new hidden variables for specific attack. And then it feeds those new hidden variables to the decoder of the cVAE to generate specific attack samples. For the CIC-IDS2017 dataset, 50% Benign data selected for experiment reaches 696,300. But the number of samples in some categories such as infiltration is only 29. If the number of generated samples for those categories is equal to the number of samples in the Benign data, the generated data will be redundant and lose its diversity, even affecting the judgment of the Benign class [11]. The detail of those two public datasets is shown in [37, 38]. Therefore, for each attack class, the number of samples of this class is gradually increased by an integer multiple of 10, and is finally determined. This method is applied in [11, 13]. In the experiment, the number K of the nearest neighbors is set to 6. The number of generated samples for each class attack is shown in Table 3.

Performance of the cVAE-DML before and after the balanced training dataset

In this paper, two experiments were made to verify efficiency improvement of new generated attack samples for the cVAE-DML. For the first experiment, only samples in the original dataset were used to train the cVAE-DML without generating new attack samples. For the second experiment, both samples in original dataset and in new generated attack samples were used to balance the training dataset. And then the balanced training dataset is used to train the cVAE-DML. The experiment results are shown in Table 4. The experiment results show that the efficiency of the cVAE-DML with balanced training dataset is better than that without balanced training dataset. For the CIC-IDS2017 dataset, the accuracy and F1-score is improved from 0.896 and 0.921 to 0.993 and 0.964, respectively. For the CSE-CIC-IDS 2018 dataset, the accuracy and F1-score are improved from 0.852 and 0.905 to 0.981 and 0.957, respectively. It means that the generated attack samples alleviate problem of imbalanced class distribution in the training dataset. As a consequence, the efficiency of the attack detection has been improved.

Comparative experiments

In this paper, five algorithms were used, that is, THEODORA [31], AIDA [32], MINDFUL [33], DNN-3 [19] and DNN4Layers [26], as the comparative algorithms to verify the efficiency of the cVAE-DML. The details of those comparative algorithms are shown as follows.

The results of comparative experiments are shown in Table 5. For the CIC-IDS2017 dataset and the CSE-CIC-IDS 2018 dataset, the efficiency of the cVAE-DML is better than five comparative models,which means the cVAE-DML can distinguish attack traffic from normal visiting than others algorithms more accurately. The drawback of those comparative models is that they cannot accurately extract private information from intrusion attack. In addition, they cannot alleviate imbalance problem in the training dataset. As a result, as for the accuracy and F1-scores measurements, those comparative models are lower than the cVAE-DML.

Table 3 The number of original samples and generated sample

Dataset	Category	Number of original samples	Number of generated samples	Total samples in training dataset
CIC-IDS 2017	Begnign	696,300	0	696,300
	Bot	603	14,820	15,423
	DoS	118,879	570	119,449
	Infiltration	29	2940	2969
	PortScan	49,605	410	50,015
	Web Attack	680	9020	9700
	Brute Force	4300	440	4740
CSE-CIC-IDS 2018	Begnign	1,078,776	0	1,078,776
	Bot	23,000	249,130	264,930
	DoS	54,005	840	54,845
	Infiltration	13,087	117,680	130,767
	DDoS	111,083	520	111,603
	Web Attack	74	8720	8794
	Brute Force	31,005	520	31,525

Table 4 The performance of the cVAE-DML on CIC-IDS 2017 and CSE-CIC-IDS 2018 with oversampling and without oversampling

Dataset	Samples	Accuracy	F1-score
CIC-IDS2017	Original Samples	0.896	0.921
	Hybrid Samples	0.993	0.964
CSE-CIC-IDS 2018	Original Samples	0.852	0.905
	Hybrid Samples	0.981	0.957

Table 5 Comparative result

Dataset	Algorithm	Accuracy	F1-score
CIC-IDS2017	cVAE-DML	0.993	0.964
	THEODORA	0.952	0.935
	AIDA	0.911	0.839
	MINDFUL	0.959	0.902
	DNN-3	0.872	0.891
	DNN4Layers	0.863	0.887
CSE-CIC-IDS 2018	cVAE-DML	0.981	0.957
	THEODORA	0.947	0.925
	AIDA	0.921	0.813
	MINDFUL	0.924	0.911
	DNN-3	0.867	0.883
	DNN4Layers	0.859	0.887

Conclusion

This paper proposes a novel attack detection system for IoT, denoted as cVAE-DML. The cVAE-DML combines oversample technology with the cVAE to generate new attack samples. Then, it generates attack samples, which are added to the original data in the balanced training dataset. Finally, the cVAE-DML combines the cVAE with the triplet networks to achieve attack detection. In the end, two public datasets and five comparative models were used to verify the efficiency of the cVAE-DML. The results of experiments show that the accuracy and F1-scores of the cVAE-DML are better than five other comparative models.

Authors' contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Xiaolei Liu, Kangyi Ding, Bangzhou Xin, Jiazhong Lu and Cheng Huang. The first draft of the manuscript was written by Chunwang Wu, Jiayong Liu and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported in part by National Natural Science Foundation of China (U20B2045).

Availability of data and materials

No datasets were generated or analysed during the current study.

Declarations

Competing interests

The authors declare no competing interests.

Received: 19 March 2024 Accepted: 29 May 2024
Published online: 02 August 2024

References

1. Namasudra S et al (2021) The revolution of blockchain: State-of-the-art and research challenges. Arch Comput Methods Engine 28:1497–1515
2. Liu Y, Wangyuan Yu, Ai Z, Guangxia Xu, Zhao L, Tian Z (2023) A blockchain-empowered federated learning in healthcare-based cyber physical systems. IEEE Trans Netw Sci Eng 10(5):2685–2696
3. Mishra R (2023) Cloud of Things and Blockchain Integration: Architecture, Applications, and Challenges. 2023 International Conference on Artificial Intelligence and Smart Communication (AISC). IEEE 1150–1154
4. Liu Y, Lan Y, Li B, Miao C, Tian Z (2021) Proof of Learning (PoLe): Empowering neural network training with consensus building on blockchains. Comput Netw 2021:201:108594
5. Chaudhari S, Dhumal R, Maktum T (2022) Blockchain of Things: Benefits and Research Challenges[J]. Real-Life Applications of the Internet of Things 313–347
6. Nguyen DC, Pathirana PN, Ding M et al (2020) Integration of blockchain and cloud of things: Architecture, applications and challenges[J]. IEEE Communications surveys & tutorials 22(4):2521–2549
7. Balogh S et al (2021) IoT security challenges: cloud and blockchain, post-quantum cryptography, and evolutionary techniques. Electronics 10:2647
8. Alkadi O et al (2020) A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. IEEE Int Things J 8:9463–9472
9. Albulayhi K et al (2022) IoT intrusion detection using machine learning with a novel high performing feature selection method. Appl Sci 12:5015
10. Zhang J et al (2021) Deep learning based attack detection for cyber-physical system cybersecurity: a survey. IEEE/CAA J Automatica Sinica 9:377–391
11. Zhang Y, Liu Q (2022) On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples. Futur Gener Comput Syst 133:213–227
12. Andresini G, Appice A, Malerba D (2021) Autoencoder-based deep metric learning for network intrusion detection. Inf Sci 569:706–727
13. Jin F, Chen M, Zhang W et al (2021) attack detection on internet of vehicles via combining log-ratio oversampling, outlier detection and metric learning [J]. Inf Sci 579:814–831
14. Zarpelão BB et al (2017) A survey of intrusion detection in Internet of Things. J Netw Comp Appl 84:25–37
15. Dai W, Ng K, Severson K et al (2019) Generative oversampling with a contrastive variational autoencoder[C]//2019 IEEE International Conference on Data Mining (ICDM). IEEE 101–109
16. Abid A, Zou J. Contrastive variational autoencoder enhances salient features[J]. arXiv preprint arXiv:1902.04601, 2019.
17. Hoffer E, Ailon N (2015) Deep metric learning using triplet network[C]// Similarity-based pattern recognition: third international workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3, Springer International Publishing, p 84–92
18. Roy S, Li J, Choi BJ et al (2022) A lightweight supervised attack detection mechanism for IoT networks[J]. Futur Gener Comput Syst 127:276–285
19. Vigneswaran RK, et al (2018) Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security. 2018 9th International conference on computing, communication and networking technologies (ICCCNT). IEEE 1–6
20. Sheikh N U, Rahman H, Vikram S, et al. A lightweight signature-based IDS for IoT environment[J]. arXiv preprint arXiv:1811.04582, 2018.
21. Liu RT, Huang NF, Chen CH et al (2004) A fast string-matching algorithm for network processor-based attack detection system[J]. ACM Trans Embed Comput Syst (TECS) 3(3):614–633

22. Rebbah M, Rebbah DEH, Smail O (2017) Intrusion detection in Cloud Internet of Things environment[C]//2017 International Conference on Mathematics and Information Technology (ICMIT). IEEE 65–70
23. Larijani H, Ahmad J, Mtetwa NA, novel random neural network-based approach for attack detection systems[C]. (2018) 10th Computer Science and Electronic Engineering (CEEC). IEEE 2018:50–55
24. Roy S, Li J, Bai Y (2022) A two-layer fog-cloud attack detection model for IoT networks [J]. Internet of Things 2022;19:100557
25. Yin C, Zhu Y, Fei J et al (2017) A deep learning approach for attack detection using recurrent neural networks[J]. Ieee Access 5:21954–21961
26. Vinayakumar R, Alazab M, Soman KP et al (2019) Deep learning approach for intelligent attack detection system[J]. Ieee Access 7:41525–41550
27. Sun P et al (2020) DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. Sec Commun Netw 2020:1–11
28. Zhang Y, Peisong L et al (2019) Intrusion detection for IoT based on improved genetic algorithm and deep belief network. IEEE Access 7:31711–31722
29. Ge M et al (2021) Towards a deep learning-driven intrusion detection approach for Internet of Things. Comp Netw 2021:186:107784
30. de Araujo-Filho PF, Kaddoum G, Campelo DR, Santos AG, Macêdo D, Zanchettin C (2020) attack detection for cyber-physical systems using generative adversarial networks in fog environment. IEEE Internet Things J. 18:6247
31. Andresini G, Appice A, Paolo Caforio F et al (2021) Improving cyber-threat detection by moving the boundary around the normal samples[J]. Machine intelligence and big data analytics for cybersecurity applications 105–127
32. Andresini G et al (2019) Exploiting the auto-encoder residual error for intrusion detection. 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS & PW). IEEE 281–290
33. Andresini G et al (2020) Multi-channel deep feature learning for intrusion detection. IEEE Access 8:53346–53359
34. Vigneswaran RK, Vinayakumar R, Soman KP et al (2018) Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security[C]//2018 9th International conference on computing, communication and networking technologies (ICCCNT). IEEE 1–6
35. Li W et al (2014) A new intrusion detection system based on KNN classification algorithm in wireless sensor network. J Electr Comp Engine 2014:2014:240217
36. Shapoorifard H, Shamsinejad P (2017) attack detection using a novel hybrid method incorporating an improved KNN[J]. Int J Comput Appl 173(1):5–9
37. Boukhamla A, Gaviro JC (2021) CICIDS2017 dataset: performance improvements and validation as a robust intrusion detection system testbed[J]. International Journal of Information and Computer Security 16(1-2):20–32
38. License:http://www.unb.ca/cic/datasets/ids-2018.html [Acessed:14-SEP-2018]

Publisher's Note