Automatically Estimate Clusters in Autoencoder-based Clustering Model for Anomaly Detection

Van Quan Nguyen
Le Quy Don Technical University, Viet Nam
quannv@lqdtu.edu.vn

Viet Hung Nguyen
Le Quy Don Technical University, Viet Nam
hungnv@lqdtu.edu.vn

Nhien - An Le Khac
University College Dublin, Dublin, Ireland
an.lekhac@ucd.ie

Van Loi Cao
Le Quy Don Technical University, Viet Nam
loi.cao@lqdtu.edu.vn

***Abstract*—In a previous work, a clustering-based method had been incorporated with the latent feature space of an autoencoder to discover sub-classes of normal data for anomaly detection. However, the work has the limitation in manually setting up the numbers of clusters in the normal training data. Finding a proper number of clusters in datasets is often ambiguous and highly depends on the characteristics of datasets. This paper proposes a novel data-driven empirical approach for automatically identifying the number of normal sub-classes (clusters) without human intervention. This clustering-based method, afterward, is co-trained with an autoencoder to automatically discover the appreciated number of clusters of normal training data in the middle hidden layer of the autoencoder. The resulting clustering centers are then used to identify anomalies in querying data. Our approach is tested on four scenarios from the CTU13 datasets, and the experimental results show that the proposed model often perform better than those of the model in the previous work on almost scenarios.**

***Index Terms*—Deep Learning, Autoencoders, Clustering Techniques, Anomaly Detection, Latent Representation**

I. Introduction

Anomaly detection is a data analysis task distinguishing patterns deviating so much from normal data [3]. This task is critical for automatically identifying malicious activities and other forms of network abuses from the normal behaviour of network usages. For the last three decades, many machine learning techniques, especially deep learning, have been employed for solving challenging issues in anomaly detection domains [2]. Generally, these learning methods can be classified into three main categories depending on the availability of anomaly data such as supervised learning, unsupervised learning and semi-supervised learning [3], [11]. Though supervised learning approach has been demonstrated attractive performance on a wide range of problems it shows critical drawbacks in anomaly detection problems. This is because supervised learning techniques require the labels of normal data and anomalies, and yield poor performance on detecting novelties. Therefore, it tends to be not commonly used in anomaly detection as the two other approaches, semi-supervised and unsupervised learning.

978-1-6654-0435-8/21/$31.00 ©2021 IEEE

Recently, semi-supervised techniques, such as one-class classification (OCC), have demonstrated many advantages in network anomaly detection. They may require only normal data for constructing anomaly detectors, which can eliminates the tone of work in collecting and labelling anomaly examples. More importantly, the models capturing the characteristics of normal data can have the ability in detecting new/unknown anomalies [10], [11]. Autoencoders (AEs) are typical deep learning methods for anomaly detection in variety of domains [4], [7], [15]. AEs learn to attempt reconstructing the original data at its outputs. A trained AE on normal data will represent normal data well resulting in small reconstruction error (RE) when evaluating normal data points. Thus, the RE can be used as anomaly score in identifying anomalies regularized AEs are alternative approaches of using AEs for identifying anomalies. In this approach, regularizers are designed to learn a "good feature representation" to benefit following classifiers. Dirac Delta Variational AE (DVAE) and Shrink AE (SAE) and [11] can be known as typical examples of learning latent feature representations. The latent representation of AEs can be employed to facilitate clustering algorithms [6], [9], [12], [16]. In this combination, AEs learn to represent data in a more meaningful feature space while clustering techniques aim to reveal appropriate clusters in the feature space. For example, a hybrid of Autoencoders and Self Organizing Maps (SOM) was introduced for identifying smart phone users by DC Le et al. [12].

When using one-class classification for network anomaly detection, normal examples are often assumed to be similar to each others, and belong to only one class (one cluster). Thus, OCC often learns to represent normal data in a specific region in feature space, such as One-class Support Vector Machine (OCSVM), SAE and DVAE. In some scenarios, however, normal observations can be collected from variety of network services and applications. Thus, normal instances may

belong to several different sub-clusters, which shares some common characteristics amongst these sub-clusters and normal data. Our previous work [9] introduced a co-training strategy between an ordinary AE and a variation of K-mean technique to discover sub-clusters of normal data. In other words, the AE learn to compress the normal input data into a more meaningful feature space of its bottleneck layer, whereas the k-mean algorithm reveals a number of normal sub-classes in the new feature space. This co-training is an iterative process until the two methods convergence. The limitation of the study is that the number of normal clusters is manually selected. These clusters (sub-classes) of normal data should depend on the characteristics of normal network traffic, e.g the number of network services, applications, protocols, and differs from dataset to dataset. Thus, fixing this number may result in a decrease in the performance of the proposed model on some anomaly detection problems. However, determining the proper number of clusters in a given dataset is challenging and often requires human intervention.

In this paper, we introduce a novel method that employs a density measurement and a hierarchical clustering method for automatically estimating the number of clusters in datasets. This attempts to overcome the weakness of the co-training hybrid of K-means and an AE, CAE [9]. The resulting number of clusters is used for CAE to reveal clusters at the middle hidden layer of the AE. In other words, the AE can project normal data in a lower dimensional space with more meaningful features, while clustering methods like K-means can help the AE force normal data into sub-clusters and discover them. This is operated in an iterative co-training process. Our proposed model is evaluated on four scenarios in the CTU13 datasets, and the results show that the proposed model often out-performs the CAE model [9] and also SAE and DVAE [11].

The rest of this paper is organized as follows. We briefly introduce Autoencoders and Hierarchical clustering technique in section II. In section III, we review some related works that employed the latent representation of Autoencoders and clustering-based techniques for anomaly detection. Our proposed method is presented in Section III. Experiments, results and discussion are shown in Section V and VI, respectively. Finally, we conclude our paper and give future directions.

II. Background

A. Autoencoders

An conventional autoencoder is a neural network, which is aimed to reconstruct the input at the output layer [5]. It contains two components, which are called *encoder* and *decoder* as shown in Figure 1. Where f_θ is the encoder, and $X = \{x_1, x_2, ...x_n\}$ be a dataset. The *encoder* f_θ aims to map the input $x_i \in X$ into a latent representation $z_i = f_\theta(x_i)$. The *decoder* g_ϕ aims to map the *latent representation* z_i back into the input space, so the reconstruction in calculated by $\hat{x}_i = g_\phi(z_i)$. The encoder and decoder are represented in the

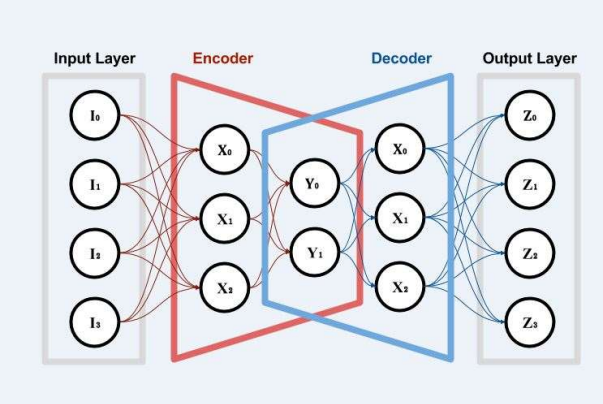


Fig. 1. Autoencoder.

form of activation functions of linear functions with respect to weights and bias as follow:

$$f_\theta(x) = \psi_f(Wx + b) \tag{1}$$

$$g_\phi(z) = \psi_g\left(W^{'}z + b^{'}\right) \tag{2}$$

where θ=(W,b) and ϕ =(W',b') are (weights and biases) for training encoder and decoder, respectively. ψ_f and ψ_g are the activation functions of the encoder and decoder. The most popular activation functions are *logistic sigmoid* or *hyperbolic tangent* non-linear function, or a linear *identity* function. For a single observation x_i, the objective function of an Autoencoder is the dissimilarity between x_i and $\hat{x}_i$. The objective function over all training samples as in (3).

$$\mathcal{L}_{\mathrm{AE}}(\theta;\phi;x) = \frac{1}{n}\sum_{i=0}^{n} l(x_i, \hat{x}_i) = \frac{1}{n}\sum_{i=0}^{n} l(x_i, g_\phi(f_\theta(x_i))) \tag{3}$$

Choosing a loss function for autoencoder model is highly depend on the assumption about given datasets. In the case, the values of datapoints in dataset are real, the objective function is determined as the mean squared error (MSE) over all data points as in (4).

$$\mathcal{L}_{\mathrm{AE}}(\theta;x) = \frac{1}{n}\sum_{i=1}^{n}\left(\parallel x_i - \hat{x}_i \parallel^2\right) \tag{4}$$

For binary data, a cross-entropy loss is commonly used, which is shown in (5).

$$\mathcal{L}_{\mathrm{AE}}(\theta;x) = -\frac{1}{n}\sum_{i=1}^{n}\left(x_i \log(\hat{x}_i) + (1 - x_i)\log(1 - \hat{x}_i)\right) \tag{5}$$

B. Hierarchical Clustering

Hierarchical clustering is an algorithm that attempt to classify similar objects into appropriate groups called clusters [8]. The difficulty is how to find optimal set of clusters. Each cluster is clearly separate from each other cluster as much as possible, and the objects within each cluster are as much as similar to each other. Hierarchical clustering might be categorized into two main types :

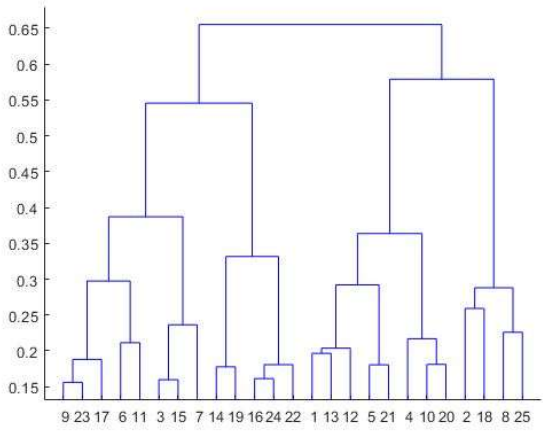


Fig. 2. Example of Dendrogram

- Agglomerative: Each object starts in its own cluster, at each following step of algorithm two of clusters are merged as one moves up the hierarchy.
- Divisive: All objects start in one cluster, at each following step of algorithm, splits are performed recursively as one moves down the hierarchy.

Dendrogram is an effective tool to illustrate the results of hierarchical clustering. A dendrogram contains lot of important information about the distances between clusters and the number of objects in each clusters. The instance of such dendrogram is illustrated in Figure (2). Where horizontal lines represents the distance between clusters, and the vertical lines denote the objects and clusters. We must to find a apporach to stop the clustering process and figure out the optimal number of clusters, especially in a manner that highly depends on nature of the datasets.

III. Related Works

In this section we will analysis various recent accomplishments in this area. Many existing anomaly detection algorithms faced difficulty when dealing with high-volume characteristics as well as high-velocity of datasets. Therefore these algorithms do not retain sufficient accuracy in many situations [14]. Such techniques rely on concepts of proximity to detect anomalies that are based on relationships among data observations. The proximity can be calculated by using plenty of techniques, which are classified as cluster-based, distance-based or density based. Autoencoders (AE) have been becoming one of the most effective tools for anomaly detection [9] [1] [10] [11] [13]. The study in [10] is one-class learning solution, which is a combining autoencoder and density estimation. Two cases for modelling density as a single Gaussian and full kernel density estimation are investigated. Cao at [11] introduced two types of regularized AEs, called Shrink AE (SAE) and Dirac Delta VAE (DVAE) capturing the characteristics of normal data.Afterward, the latent representation of SAE and DVAE were used to assist simple one-class classifiers. The researchers at [1] introduced hybrid model between clustering technique K-means and Shrink Autoencoder (SAE) that is called KSAE. By doing in such way, KSAE tries to force normal training data into several clusters and then applying SAE to explorer latent representation of data in each cluster. The author in [12] introduced the method of using autoencoder - Self Organizing Map (SOM) to explorer user behaviour characterization based smart phone usage information. Such solution consists of two steps. The Autoencoder aims to explorer latent representation at bottleneck and after that used SOM without AE. More recently, Nguyen in [9] proposed an effective combination between clustering methods and AE called CAE and training in a semi-supervised way. This work assumes that normal sample data might have a set of common characteristics and their own private characteristics. Therefore, it might cause a number of sub-clusters in the normal data. During training process an AE learns latent representation of data. While a clustering method tries to explorer clusters in the latent normal data and classify them into appropriate clusters. The solution is tested with four scenarios of CTU13 dataset, the results out-performs other methods on three of four cases. The limitation of CAE is that model is built based on the random of clusters at hidden layer. In this work, we attempt to find a effective approach based on nature of data to estimate the appropriate number of clusters of datasets and afterward use this one for latter training clustering-based autoencoder in one-class manner.

IV. Proposed Method

In this section, we present our proposed method for automatically estimating and picking up the number of clusters based on the hierarchical clustering technique. This method is then employed for enhancing the productivity of the clustering-based deep Autoencoder (CAE) introduced in [9]. Our previous clustering-based model, CAE, is faced the limitation of randomly choosing the number of clusters for K-means working in the latent feature space of Autoencoders. The proposed method tries to overcome such obstacle. In fact, it will attempt to estimate a proper number of clusters in the normal training dataset beforehand the co-training process of CAE operating.

Our approach consists of two stages: (1) we propose a density-based measurement combining with the hierarchical clustering technique to automatically picking up the number of clusters in the normal training datasets; (2) the results from the first stage is used to guide the co-training process of CAE on the only normal training samples. The complete procedure is illustrated in Figure 3. In this section, we will present in details how to automatically pick up a optimal number of sub-clusters in normal training datasets. For the details of the CAE method, please find in our previous work [9]. By applying the hierarchical clustering technique on a given normal dataset, we will receive a dendrogram that split the dataset into as many sub-clusters as possible. We must to find an optimal point on the vertical lines of the dendrogram to prevent the hierarchical clustering algorithm further splitting the dataset

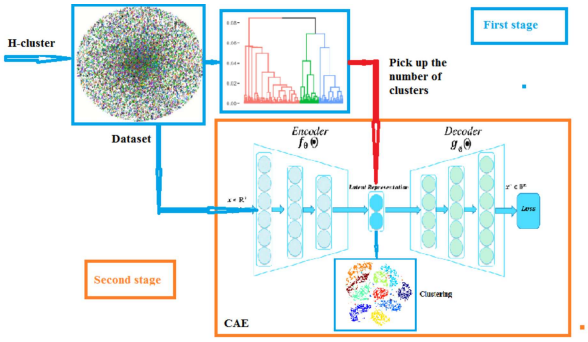


Fig. 3. Proposed Method: automatically estimating the number of clusters and the co-training process of CAE.

into smaller sub-clusters. The approach is to determine a horizontal line crossing the dendrogram. This can result in a number of intersections (clusters) on the dendrogram. We assume that the number of clusters in a given dataset is related to the density of the dataset. Thus, we use the density as a parameter to estimate the number of clusters in datasets. Suppose we have a collection of m observations in dataset $\{x^1,x^2,...x^m\} \in \mathrm{R}^n$. The density parameter of the dataset can be determined as the ratio of the number of data samples to the supervolume of n-dimensional parallelepiped. The supervolume of n-dimensional parallelepiped can be calculated by multiplying n maximal distances over n dimensions. The proposed calculation is presented as follows,

$$\mathcal{D} = \frac{m}{\prod_{j=1}^{n}(Max_{i=1}^{m}(x_j^i) - Min_{i=1}^{m}(x_j^i))} \tag{6}$$

where $\mathcal{D}$ is density of the dataset, m is the number of datapoints in the dataset, and Max and Min are the functions calculating the maximum and minimum values of j-th feature over all data points relatively. Our goal is to determine the position of horizontal line crossing the dendrogram. The number of vertical lines are crossed by the horizontal line is equal to the number of clusters. Therefore, we propose a formula to estimate the location of horizontal cutting the vertical lines of the dendrogram:

$$h = h_0.\log_{10}^{\mathcal{D}} \tag{7}$$

where h is the distance from the origin to intersections between the horizontal line and the vertical lines of the dendrogram, h_0 is coefficient selected for every dataset. Once the number of clusters has estimated as in Eq. 7, a novel variation of K-means is applied to iterative explorer sub-clusters in the latent feature space of a conventional Autoencoder. In other words, the model is force to learn to reconstruct the normal training data at the output layer of the AE. In the meantime K-means supports the AE represents the normal data into appropriate clusters at the bottleneck layer of the AE. This is happened in co-training manner.

V. Experiments

This section describers the anomaly detection datasets chosen for evaluating our proposed model, parameter settings and our experiments.

TABLE I
FOUR DATASETS FOR EVALUATING THE PROPOSED MODELS

No	Dataset	Dimension	Training set	Normal Test	Anomaly Test
1	Rbot (CTU13-10)	38	6338	9509	63812
2	Murlo (CTU13-8)	40	29128	43694	3677
3	Neris (CTU13-9)	41	11986	17981	110993
4	Virut (CTU13-13)	40	12775	19164	24002

A. Datasets

For evaluation purpose the performance of our proposed method, we have conducted the experiments on four scenarios in the CTU13 dataset as shown in the Table I. The CTU13 dataset consists of botnet traffic, which was captured in the CTU University, Czech Republic,in 2011. This dataset is a collection of a large number of real botnet traffic together with normal and background traffic. It contains thirteen scenarios of many botnet samples. In this paper, four scenarios (CTU13-8, CTU13-9, CTU13-10 and CTU13-13) are employed. Each of these datasets was splited into 40% for training (normal samples) and 60% for evaluating purposes (both normal and botnet traffic). In terms of categorical features, including dTos, sTos and protocol are encoded by using the one-hot encoding technique.

B. Experimental Settings

In this work, we conducted experiments consisting of two steps. In the first step, the hyper-parameter h_0 in Eq. (7) is set by a common value, 20, for all datasets. In the stage 2, We have chosen the number of hidden layers is 5, and the size of bottleneck layer by using the equation $h = [1+\sqrt{n}]$, where n is the number of input features [10].We have applied Xavier initialization method to initialize the weights of CAE for speeding up the convergence process. The activation function is $TANH$, and the batch size is set as 100. The optimization algorithm is $Adadelta$ with a learning rate of 10^{-1}. The early stopping method is employed with evaluation step at every 5 epochs. With the purpose of getting insights into the latent feature spaces, we have visualized the latent representations of normal training data, normal testing data and anomaly testing data.

VI. Results and Discussion

We have conducted two core experiments. Firstly, the hierarchical clustering algorithm is applied on the four datasets to receive dendrograms and afterward pick up the number of clusters as shown in Figures. 4 and 5. The height of the deprograms indicates the order in which the clusters are joined, and also represents the distance between the clusters. Samples joined together below the line are located in clusters. Afterward, our proposed empirical formulas presented in Eq. 6 and 7 are used to calculate the height of horizontal line for the four datasets, and to estimate the number of clusters in each of

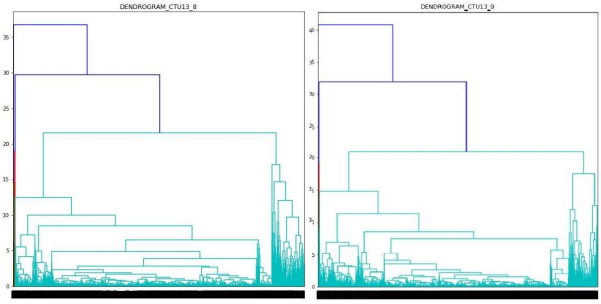


Fig. 4. Dendrogram of CTU13-8 and CTU13-9

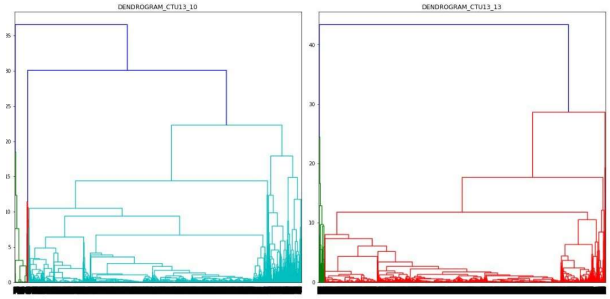


Fig. 5. Dendrogram of CTU13-10 and CTU13-13

TABLE II
THE ESTIMATING NUMBER OF CLUSTERS

	CTU13-08	CTU13-09	CTU13-10	CTU13-13
h_0	20	20	20	20
K	2	5	1	3

these datasets. The results of this step are shown in Table II. Obviously, the number of clusters in all datasets are different from dataste to dataset. It tends to reflect the characteristics of these datasets. Normal data points might share some common characteristics representing corresponding normal traffic, but they can have some their own private features. Thus, normal dataset may contain some sub-clusters

The second step is the co-training process of the CAE model with the guidance from the first step. The most popular metric, the Area Under the ROC Curve (AUC), is employed to evaluate the performance of our model in comparison to the original CAE (manually setting the number of subclusters, $K = 2$). The latent representations of the normal training sets and the testing sets of CTU13-8, CTU13-9, CTU13-10 and CTU13-13 are visualized in Figures 7, 8, 9 and 10 respectively. Performance of our proposed method is also compared with those of SAE and DVAE in [1]. In this case, two classifiers, such as CEN and MDIS, are selected because these classifiers were reported as the best classifier and the worse classifier respectively. All the resulting records are shown in Table III. We also present visualizations of the ROC curves when evaluating our proposed method on the four scenarios as shown in Figure 6.

It can be seen that, with the appropriate number of clusters shown in Table II, our proposed model shows performance improvements in comparison to the original version of CAE on three out of the four scenarios. Furthermore, the automatically process of estimating the number of clusters has shown that CTU13-10 contain only a single cluster in comparison with two clusters in [9], and the AUC has been sharply improved from 0.996 (CAE) to 0.999. Meanwhile, the increase in the number of detected clusters for CTU13-09 from $K = 2$ to $K = 5$ has greatly impact on AUC, increasing from 0.959 to 0.962. For CTU13-08, the AUC is unchanged due to the fact that the number of clusters in data remaining $K = 2$. Interestingly, with the changing number of subclusters from $K = 2$ to $K = 3$, the AUC on CTU13-13 seems to be a constant value at 0.9790. This may reveal that these clusters are much close at each other.

Overall, the results of these experiments clearly confirm that our proposed method contribute to enhance the performance of anomaly detection issues throughout one-class training process.

VII. Conclusion and Future Work

A new data-driven method is proposed with purpose of getting the optimal number of clusters in datasets, and applying to make the performance of the Clustering-based Autoencoder method (CAE) better. This work tries to overcome the limitation of previous study in [9]. Our method is divided into two steps, the first one is to building dendrograms for datasets and using our proposed empirical formula for choosing a proper number of clusters. The second step is to combine K-mean and Autoencoder in co-training manner with the guidance from the first step. This work has revealed that the proper number of clusters in dataset is very important parameter for unsupervised-based anomaly detection problems, particularly one-class learning-based methods. We have evaluated our proposed model on four scenarios in the CTU13 dataset, and the outcomes have illustrated that our model often performs better than the previous ones such as CAE, SAE and DVAE. Our future work is an extension on investigating other methods in automatically estimating the number of clusters and incorporating the process into a co-training process with Autoencoders.

References

[1] Bui, T.C., Hoang, M., Nguyen, Q.U., et al.: A clustering-based shrink autoencoder for detecting anomalies in intrusion detection systems. In: 2019 11th International Conference on Knowledge and Systems Engineering (KSE). pp. 1–5. IEEE (2019)

[2] Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407 (2019)

[3] Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 1–58 (2009)

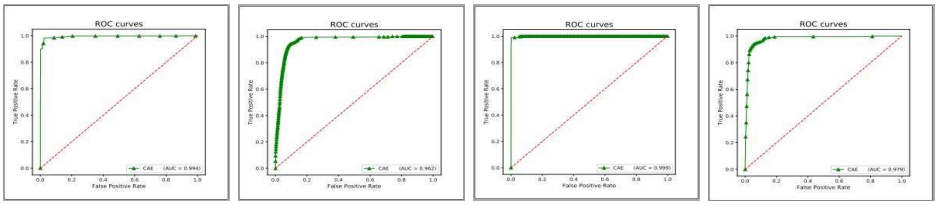
[4] Fiore, U., Palmieri, F., Castiglione, A., De Santis, A.: Network anomaly detection with the Restricted Boltzmann Machine. Neurocomputing **122**, 13–23 (2013)

[5] Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT press Cambridge (2016)

[6] Guo, X., Liu, X., Zhu, E., Yin, J.: Deep clustering with convolutional autoencoders. In: International conference on neural information processing. pp. 373–382. Springer (2017)

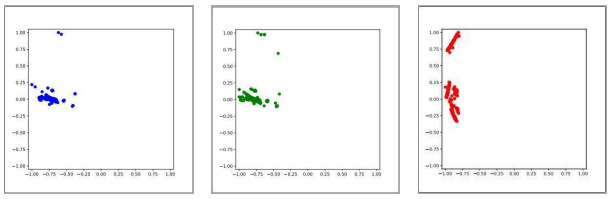
TABLE III
AUCs OF SAE-OCCs, DVAE-OCCs,CAE AND H-CLUSTER + CAE MODELS.

Represen- -tation	One-class Classifiers	Datasets			
		CTU13-08	CTU13-09	CTU13-10	CTU13-13
SAE $\lambda = 10$	CEN MDIS	0.991 0.990	0.950 0.950	0.999 0.999	0.969 0.968
DVAE $\lambda = 0.05$, $\alpha = 10^{-8}$	CEN MDIS	0.982 0.984	0.956 0.957	0.999 0.999	0.963 0.964
CAE	K=2	0.994	0.959	0.996	0.979
H-cluster + CAE		0.994	**0.962**	**0.999**	**0.979**



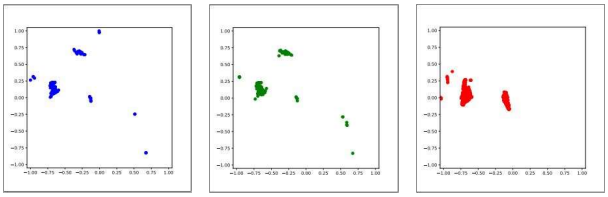
(a) CTU13-08 (b) CTU13-09 (c) CTU13-10 (d) CTU13-13

Fig. 6. The ROC curves of our proposed model.



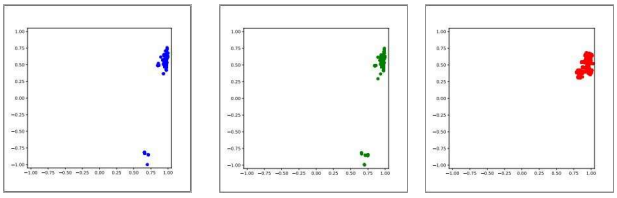
(a) Normal training (b) Normal testing (c) Anomaly testing

Fig. 7. Visualize the latent data of the CTU13-8 dataset



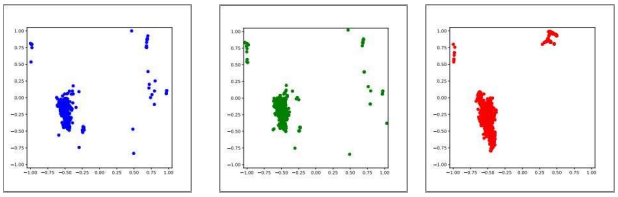
(a) Normal training (b) Normal testing (c) Anomaly testing

Fig. 8. Visualize the latent data of the CTU13-9 dataset



(a) Normal training (b) Normal testing (c) Anomaly testing

Fig. 9. Visualize the latent data of the CTU13-10 dataset



(a) Normal training (b) Normal testing (c) Anomaly testing

Fig. 10. Visualize the latent data of the CTU13-13 dataset

[7] Japkowicz, N., Myers, C., Gluck, M., et al.: A novelty detection approach to classification. In: IJCAI. pp. 518–523 (1995)

[8] Johnson, S.C.: Hierarchical clustering schemes. Psychometrika **32**(3), 241–254 (1967)

[9] Nguyen, V.Q., Nguyen, V.H., Le-Khac, N.A., et al.: Clustering-based deep autoencoders for network anomaly detection. In: International Conference on Future Data and Security Engineering. pp. 290–303. Springer (2020)

[10] Nicolau, M., McDermott, J., et al.: A hybrid autoencoder and density estimation model for anomaly detection. In: International Conference on Parallel Problem Solving from Nature. pp. 717–726. Springer (2016)

[11] Nicolau, M., McDermott, J., et al.: Learning neural representations for network anomaly detection. IEEE transactions on cybernetics **49**(8), 3074–3087 (2018)

[12] Rajashekar, D., Zincir-Heywood, A.N., Heywood, M.I.: Smart phone user behaviour characterization based on autoencoders and self organizing maps. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). pp. 319–326 (2016). https://doi.org/10.1109/ICDMW.2016.0052

[13] Sakurada, M., Yairi, T.: Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis. pp. 4–11 (2014)

[14] Thudumu, S., Branch, P., Jin, J., Singh, J.J.: A comprehensive survey of anomaly detection techniques for high dimensional big data. Journal of Big Data **7**(1), 1–30 (2020)

[15] Vu, L., Nguyen, Q.U., Nguyen, D.N., Hoang, D.T., Dutkiewicz, E., et al.: Learning latent representation for iot anomaly detection. IEEE Transactions on Cybernetics (2020)

[16] Zhang, D., Sun, Y., Eriksson, B., Balzano, L.: Deep unsupervised clustering using mixture of autoencoders. arXiv preprint arXiv:1712.07788 (2017)