A Robust PCA Feature Selection To Assist Deep Clustering Autoencoder-Based Network 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

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

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

Nathan Shone
Liverpool John Moores University, UK
n.shone@ljmu.ac.uk

***Abstract*—This paper presents a novel method to enhance the performance of Clustering-based Autoencoder models for network anomaly detection. Previous studies have developed regularized variants of Autoencoders to learn the latent representation of normal data in a semi-supervised manner, including Shrink Autoencoder, Dirac Delta Variational Autoencoder and Clustering-based Autoencoder. However, there are concerns regarding the feature selection of the original data, which stronger support Autoencoders models exploring more intrinsic, meaningful and latent features at bottleneck. The method proposed involves combining Principal Component Analysis and Clustering-based Autoencoder. Specifically, PCA is used for the selection of new data representation space, aiming to better assist CAE in learning the latent, prominent features of normal data, which addresses the aforementioned concerns. The proposed method is evaluated using the standard benchmark NSL-KDD data set and four scenarios of the CTU13 datasets. The promising experimental results confirm the improvements offered by the proposed approach, in comparison to existing methods. Therefore, it suggests a strong potential application within modern network anomaly detection systems.**

***Index Terms*—Anomaly Detection, Clustering-based Autoencoders (CAEs), Principal Component Analysis (PCA), Latent Representation, Deep Learning**

I. Introduction

Nowadays, with the explosive development of the Internet, the number of networked devices and network services is increasing at an exponential rate. Especially, the ubiquitous presence of Internet of Things (IoT) devices have been bringing many essential benefits to our lives such as healthcare, transportation, energy and industry. IoTs devices have the ability to automatically connect, process and transfer data with each other without human intervention [12]. However, the widespread use of network devices and IoTs also faces many security risks [17]. Attackers use diverse and increasingly complex techniques to break the integrity, confidentiality and availability of information systems. Zero-day exploits are the most concerning form of attack, which has the most potential to cause serious consequences for network infrastructure and sensitive data [3] [1] [20]. These attacks can be also be referred to as anomalies or outliers [7] [28]. Anomalies or outliers are substantial variations, which show significant differences from behavioural norms [19]. Identifying these anomalies in large network data streams is always a challenging task, due to the nature of these anomalies including their rarity, heterogeneity and low frequency of occurrence [24]. Many anomaly detection techniques have been researched, deployed and applied in a variety of domains. These techniques include statistical techniques, spectral analysis techniques and non-machine learning techniques [8]. Specifically, in the scope of network security these techniques face many challenges with the large amount of data generated by network devices and the increasing emergence of novel attack techniques.

Many machine learning methods have also been implemented to improve the efficiency of network anomaly detection systems [22] [15] [25] [27]. However, these methods still have inherent limitations, such as human intervention in building feature extractors, using expert knowledge in data labeling etc. These techniques are not very effective in the era of big data, with data volume and data dimensions increasing rapidly. Furthermore, classical machine learning algorithms fail to unearth and capture the complex structures of big data.

Recent years have seen a proliferation of applications of deep learning algorithms and unprecedented results in many different fields. Deep learning techniques have shown superior results when compared to other classical machine learning methods, especially when the data volume increases dramatically [7]. Anomaly detection systems based on deep learning algorithms are increasingly popular and widely applied in both academic and industrial environments. The selection of a deep learning neural network architecture for anomaly detection is basically based on the nature and availability of the collected data in the training set [7]. In general, the deep learning algorithms being used for anomaly detection can belong to one of three main categories: (1). Supervised learning algorithms; (2). Semi-supervised learning algorithms;

978-1-6654-1001-4/21/$31.00 ©2021 IEEE

(3). Unsupervised learning algorithms. The labels are used to train the deep learning model will indicate which samples are normal and which observations are outliers. Although there have been improvements in the performance of supervised learning models, these solutions still face many obstacles due to the difficulty of data labeling, notably anomalous data and training dataset imbalances. In fact, it is much easier to collect and label normal data than anomalous data, therefore semi-supervised learning algorithms are becoming increasingly relied upon. These algorithms depend on the assumption that normal data and outlier data are generated from different probability distributions. Subsequent learning models are trained in a semi-supervised manner with the aim of capturing the essential characteristics of normal data, so that it is easier to distinguish from outliers.

One of the widely deployed solutions is to use deep neural network autoencoders, which are trained using only normal data in a one-class training manner [23] [6] [5]. Deep learning neural network autoencoders (AE) have shown to be a very effective and efficient method in building anomaly detection models in many different domains such as network intrusion detection and IoTs Anomaly Detection [7] [28]. The latent features discovered and explored in the feature representation space of AE have improved the efficiency of the network anomaly detectors. Specifically in the semi-supervised learning scenarios, these latent representations are a reliable foundation for clearly distinguishing between normal and abnormal data. The common limitation of the above approaches is that the data used to train deep learning autoencoders has not been properly preprocessed, which greatly affects the model's ability to learn the latent representation space at bottleneck.

To overcome such limitations, we propose a novel technique that combines the use of Principal Component Analysis (PCA) for preprocessing data, and deep neural network Clustering-based Autoencoders (CAE) to build semi-supervised anomaly detector. By utilizing PCA's power to define new coordinate axes, the data representation capabilities will improve significantly. This will enhance the CAE's ability to discover many hidden, yet meaningful architectures that are difficult to explore in the original space. We will implement a prototype of our technique and evaluate it using popular benchmark datasets including NSL-KDD, CTU13-08, CTU13-09, CTU13-10 and CTU13-13.

The rest of the paper is organized as follows: We will briefly present the background knowledge of the PCA algorithm and the deep neural network autoencoder in Section II. Section III reviews prominent and current studies related to the using of AE and clustering-based AE for cyber anomaly detection. Our proposed method is detailed in Section IV. Experiments, results and discussion are presented in Sections V and VI, respectively. Finally, we conclude our paper in Section VII and propose future research directions.

II. Background

In this section, we provide the necessary background knowledge to understand concepts related to our proposed models.

A. Principal Component Analysis

PCA is a technique renowned for dimensionality reduction, data compression and feature extraction in plenty of research domains [4] [16]. In general, PCA is defined as an orthogonal projection of data into a lower dimensional linear space, in which the variance of the projected data is maximized [14]. We will shortly introduce the mathematical formulation and the outline the overall procedure of PCA. Let $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_N}\}$ be a collection of observations, where $\mathbf{x_i}$, $i = 1, 2...N$ is a sample in Euclidean space with dimensionality *D*, meaning that $\mathbf{x_i} \in R^D$. Our goal is to project these data points into the new space with the least loss of information, notably this new space has a significantly lower intrinsic dimensionality $M \leq D$. In other words, we have to find a new space with dimensionality *M* that maximizes the variance of the projected data points. Without loss of generality, we firstly consider the situation in which we aim to project data points into one-dimensional space with $M = 1$. We use a *D*-dimensional vector $\mathbf{e_1}$ to define the direction of this new space. Notice that if vector $\mathbf{e_1}$ determines the direction of space, then vector $k * \mathbf{e_1}$ also determines the direction of that space, where $\forall k \neq 0$ and $k \in R$. We are only interested in the direction of the vectors, not the magnitude, so we will choose the unit vector so that $\mathbf{e_1}^T\mathbf{e_1} = \mathbf{1}$. The mean of the dataset is given in equation 1.

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x_i} \tag{1}$$

The covariance matrix $\mathbf{C}$ of the data samples is defined in equation 2.

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^T. \tag{2}$$

The coordinates of the data point $\mathbf{x_i}$ and the mean $\bar{\mathbf{x}}$ of samples in the new space are $\mathbf{e_1}^T\mathbf{x_i}$ and $\mathbf{e_1}^T\bar{\mathbf{x}}$, respectively. The variance of the projected data points in the new space is calculated by equation 3.

$$\bar{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{e_1}^T\mathbf{x}_i - \mathbf{e_1}^T\bar{\mathbf{x}})^2 = \mathbf{e_1^T C e_1} \tag{3}$$

Our goal is to maximize the variance of the dataset on the new space. This means we are going to maximize the value $\bar{\sigma}^2$ with respect to $\mathbf{e_1}$. This is a constrained maximization problem, where the constraint is derived from the normalization of the basic vector $\mathbf{e_1}^T\mathbf{e_1} = \mathbf{1}$. We use the Lagrange multiplier method to establish the objective function as given in the equation 4.

$$\zeta(\lambda_1, \mathbf{e_1}) = \mathbf{e_1}^T\mathbf{C e_1} + \lambda_1(1 - \mathbf{e_1}^T\mathbf{e_1}) \tag{4}$$

By setting the partial derivative of objective function with respect to $\mathbf{e_1}$ equal to zero, we get the equation (5).

$$\mathbf{C e_1} = \lambda_1\mathbf{e_1} \tag{5}$$

This shows us that vector $\mathbf{e_1}$ must be an eigenvector of the covariance matrix $\mathbf{C}$ and λ_1 is the eigenvalue corresponding to the eigenvector $\mathbf{e_1}$. We left-multiply by $\mathbf{e_1}$ on the both sides of the equation 5 and combine with the constraint $\mathbf{e_1}^T\mathbf{e_1} = \mathbf{1}$ to get equation 6.

$$\mathbf{e_1}^T\mathbf{C}\mathbf{e_1} = \lambda_1 \tag{6}$$

By combining equation 3 and equation 6 we realize that the variance of the projected data reaches its maximum value when we set the vector $\mathbf{e_1}$ to be the eigenvector with the largest corresponding eigenvalue λ_1. We call this eigenvector $\mathbf{e_1}$ the first principal component. Similarly, we find the next principal components by selecting new directions that maximize the value of projected variance amongst all possible directions, which are orthogonal to the selected principal components. Using the induction method, we give a solution for the general case of *M*-dimensional projection as follows: The best solution for a linear projection where the variance of the projected data reaches its maximum value is to determine the *M* eigenvectors $(\mathbf{e_1}, \mathbf{e_2}...\mathbf{e_M})$ of the covariance matrix $\mathbf{C}$ of dataset corresponding to the *M* largest eigenvalues $(\lambda_1, \lambda_2, ...\lambda_M)$. In general, we can summarize the PCA algorithm implementation procedure as shown in Algorithm 1 and illustrated in Fig.1.

Algorithm 1 Principal Component Analysis

1: **Input: Given the dataset $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_N}\}$ where $\mathbf{x_i} \in R^D$, $i = 1, 2...N$; M and D are dimensions.**
2: **Calculate the mean of the dataset by equation (1)**
3: **Subtracting the mean from each data point: $\hat{\mathbf{x_i}} = \mathbf{x_i} - \bar{\mathbf{x}}$**
4: **Compute Covariance Matrix C by equation (2)**
5: **Compute eigenvalues and eigenvectors of C $(\lambda_1, \mathbf{e_1}), ...(\lambda_D, \mathbf{e_D})$.**
6: **Pick up M eigenvectors $(\mathbf{e_1}, \mathbf{e_2}...\mathbf{e_M})$ with M highest eigenvalues $(\lambda_1, \lambda_2, ...\lambda_M)$.**
7: **Project data to selected eigenvectors $(\mathbf{e_1}, \mathbf{e_2}...\mathbf{e_M})$.**
8: **Output: Projected points in lower dimensions.**

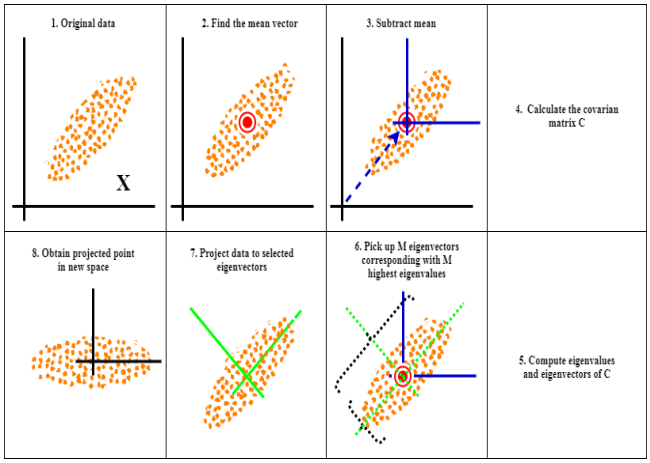


Fig. 1. PCA Procedure

B. Autoencoder

Deep AEs are a type of neural network, which are designed with purpose of encoding input data into latent and meaningful representations, then decoding them so that they are as similar to the input data as possible [2] [10] [13]. In this subsection, we will present the structure and the loss functions of AE [13] and CAE [23]. They are the important components of our proposed model.

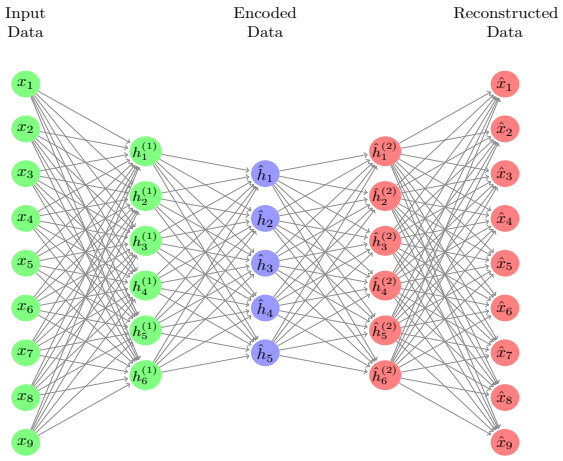


Fig. 2. Example Autoencoder Structure

An AE is a neural network used to learn a lower representation of high dimensional data in an unsupervised manner. It consists of two parts: an encoder and a decoder, as shown in Figure 2. Internally, an AE has a hidden layer $\hat{\mathbf{h}}$, which denotes a latent representation of the input. The task of the encoder is to learn the function $\mathbf{f}$, which maps input $\mathbf{x}$ to that latent representation $\hat{\mathbf{h}}$. The job of the decoder is to learn the function $\mathbf{g}$, which maps the latent variable $\hat{\mathbf{h}}$ to an output (called reconstruction) $\hat{\mathbf{x}}$. An AE is trained for the purpose of copying its input to its output. However, usually they are designed so that copying is not perfect. They are often forced to approximately copy the input data, which in turn helps them learn many potentially meaningful properties of the data. Giving constraints $\mathbf{h}$ to have smaller dimension than $\mathbf{x}$ is an effective way to acquire useful features from an AE. Such AE are called under-complete. Learning an under-complete latent representation forces the AE to capture the most important and prominent features of data. The learning process is presented as a reconstruction error minimization, which is shown in equation 7.

$$\mathcal{L}_{\mathrm{AE}}(\mathbf{x}, \hat{\mathbf{x}}) = \mathcal{L}_{\mathrm{AE}}(\mathbf{x}, \mathbf{g}(\mathbf{f}(\mathbf{x}))) \tag{7}$$

Where $\mathcal{L}_{\mathbf{AE}}$ is a loss function penalizing $\hat{\mathbf{x}}$ for being not similar to $\mathbf{x}$. $\mathbf{f}$ and $\mathbf{g}$ are the encoder and decoder functions respectively. The most popular choice for loss function of an AE is the Mean-Squared Error (MSE) over all data observations, as shown in equation 8.

$$\mathcal{L}_{\mathrm{AE}}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{N}\sum_{i=0}^{N}(\mathbf{x_i} - \hat{\mathbf{x}}_\mathbf{i})^2 \quad (8)$$

where $\mathbf{x_i}$ is a sample in the training dataset $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_N}\}$, and N is the number of data samples in the dataset.

III. Existing Work

Deep learning is achieving very promising results in solving anomaly detection problems within a variety of research areas and applied domains. State-of-the-art deep learning techniques are capable of learning hierarchical discriminative features from data. This powerful capacity has gradually reduced human intervention in manual processing of features, especially in the discovery of latent features, thus improving the quality of the trained models. Various deep learning neural network architectures have been proposed for use within network anomaly detection including Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), deep hybrid models and its variants [7]. However, AEs models are showing prominent efficiency in comparison with other architectures in many circumstances. Therefore, they are the core of most deep learning-based unsupervised models applied to the network anomaly detection problem [7] [6] [23] [11] [9] [21]. In this section, we will discuss the most current and prominent autoencoder-based methods.

Cao et al. [6] have proposed two autoencoder-based models called, Shrink AE (SAE) and Dirac Delta VAE (DVAE) to learn the latent representation space at the bottleneck of AE. These models were trained using only normal data in a one-class training manner to overcome the limitations of traditional AE and variational AE when dealing with high-dimensional and sparse network data. Specifically, they introduced regularizers to the objective function during training, to force the normal samples into a very tight region around the origin in the non-saturating area of the bottleneck unit activations. Whereas the anomalous data points that are fundamentally different from the normal observations, will be pushed away from the normal region. Experimental results have shown that their method using latent representation can support anomaly detection algorithms to work effectively with sparse and high-dimensional data, even with relatively few training samples.

The authors in [21] proposed a network intrusion detection system based on stacked AEs and deep neural networks (DNN). In this work, stacked AEs tends to learn the properties of the input network data in a unsupervised way, in order to reduce the feature width. After that, the DNN is trained in a supervised manner to extract the meaningful features for the classifier. They have evaluated their proposed model using standard datasets including KDD Cup 99 and NSL-KDD. The authors claimed that the achieved accuracies on these datasets were 94.2 and 99.7%, respectively for multiclass classification.

Yang et al. [9] proposed the Self-Organizing Map assisted Deep Autoencoding Gaussian Mixture Model (SOM-DAGMM) to better preserve the architecture of the input data topology for more accurate network intrusion detection. They claimed that the Deep Autoencoding Gaussian Mixture Model (DAGMM) faces a dilemma of choosing between the low-dimensional space for Gaussian mixture model, and the input structure preservation. Therefore, they proposed a two-stage approach, in which a pre-trained SOM is plugged into the DAGMM. Experimental results show that this model has improved performance compared to the original DAGMM.

Researchers in [11] introduced a combination model of sparse autoencoder with kernel for network attack detection. Specifically, in this paper they used an iterative method of adaptive genetic algorithm to optimize the objective function of a sparse autoencoder with combined kernel. They argued that this solution will overcome the shortcomings of the previous models when faced with large-dimensional data. The model was trained and evaluated using a dataset based on IoT botnet attacks.

Nguyen et al . [23] introduced a hybrid solution combining clustering methods and AEs for detecting network anomalies in a semi-supervised manner. These combined models were trained using only normal samples. This work is based on the assumption that normal network data might come from different network services or types of devices. Therefore, although they share some common characteristics, they also have their own separated features. Their proposed hybrid model tends to discover clusters in the latent representation of AE. This co-training strategy supports the revealing of true clusters inside normal data and improves the performance of the network anomaly detection model in [6]. The limitation of this method is that there has not been a way to force AE to learn latent features that have good clustering characteristics aiming to stronger support performance of the clustering algorithms at the bottleneck. Therefore, our work aims to develop novel solution to help Autoencoders discover more powerful latent properties, which assists clustering algorithm more quickly, accurately, easily separate data clusters. Furthermore, our solution aims to narrow the normal data region, making the identification of outliers with more stability and high accuracy. Hence, we believe that the proposed model in this paper will make a contribution to overcome the current limitations of one-class training strategy.

IV. Proposed Methodology

A. Clustering-based Autoencoder

Clustering-based Autoencoder (CAE) is a hybrid combination between clustering methods and AE [23], in which the clustering algorithms are applied at the bottleneck of AE. Such combined neural networks are trained to achieve two goals. Particularly, while AEs are expected to learn the potential latent properties of the data, the clustering algorithm will split the data points into appropriate clusters. Both of these goals are optimized in parallel in the co-training manner. Therefore, the jointly objective function consists of two components, including reconstruction loss and clustering loss as shown in equation 9.

$$\mathcal{L}_{\mathrm{CAE}}(\mathbf{X}, \hat{\mathbf{X}}) = \alpha_1 \mathcal{L}_{\mathrm{AE}}(\mathbf{X}, \mathbf{g}(\mathbf{f}(\mathbf{X}))) + \alpha_2 \mathbf{\Omega}(\mathbf{H}) \quad (9)$$

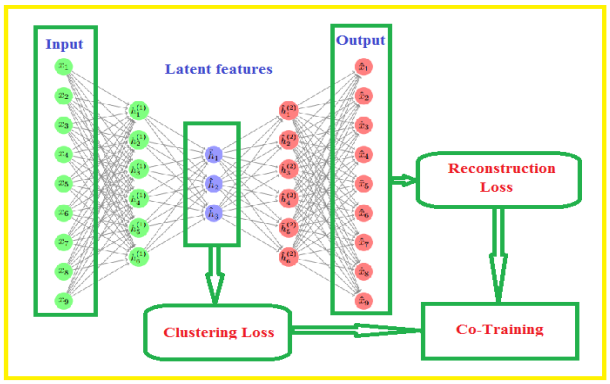


Fig. 3. Clustering-based Autoencoder

Where $\mathcal{L}_{\mathrm{AE}}$, $\mathbf{\Omega}(\mathbf{H})$ are reconstruction loss and clustering loss, respectively and α_1, α_2 are coefficients used to trade-off between these components. The general structure of a clustering-based autoencoder is shown in Fig. 3.

B. Proposed Approach

In this section, we describe our proposed approach, which facilitates CAE in [23] [6] for anomaly identification in a semi-supervised manner. In one-class learning, the model will be trained using only normal data, because outliers are rare and sometimes it is very costly to collect and label them. This method is based on the assumption that normal data points have common characteristics and are different from anomalous data. In the latent representation, the normal observations will be pushed closer to the origin and into a very tight normal region, as in SAE [6]. Conversely, abnormal data will be forced out further from the origin and normal area. Hence, the model's ability to detect anomalies is more accurate when the normal region is as tight as possible. The main limitation in [23] [6] is that AE has not yet captured the most intrinsic latent features of normal data, which enables clustering techniques to separate normal samples into appropriate clusters. The proposed method aims to overcome such shortcomings. In particular, we attempt to implement a preprocessing step for selecting good features of data beforehand, fitting it to the CAE training process.

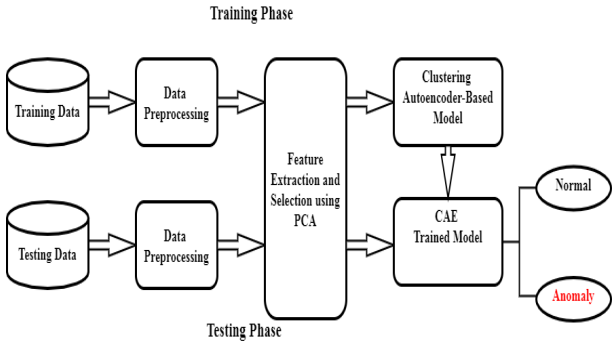


Fig. 4. General Flow of Proposed Approach

Our method consists of two stages: (1). We will use the PCA algorithm for normal data preprocessing. In other words, we will project the original data down to the new space, whose bases are orthogonal. In this way, we will find new representations of the data without much loss of information through simple linear transformations. More specifically, PCA is used for better selection of representation space rather than for dimension reduction purposes. (2). The attributed resulting from the first stage will be used to train the clustering-based autoencoder model (CAE) in a co-training manner. The complete flow of our proposed approach is illustrated in Fig. 4. We hypothesize that given the good features, the AE will better show its ability to discover other latent representations that both characterize the normal data, and separate such observations into appropriate clusters. Then the normal samples will tend to be distributed more suitably according to its underlying clusters, thus arranging the normal region much more tightly. Thanks to this, when an outlier appears, the trained model's detection ability will be significantly improved.

V. Experiments

In this section, we introduce the anomaly detection datasets chosen for evaluating our proposed approach, parameter settings and experiments.

A. Datasets

The experiments will be conducted on 5 datasets, as summarised in Table I.

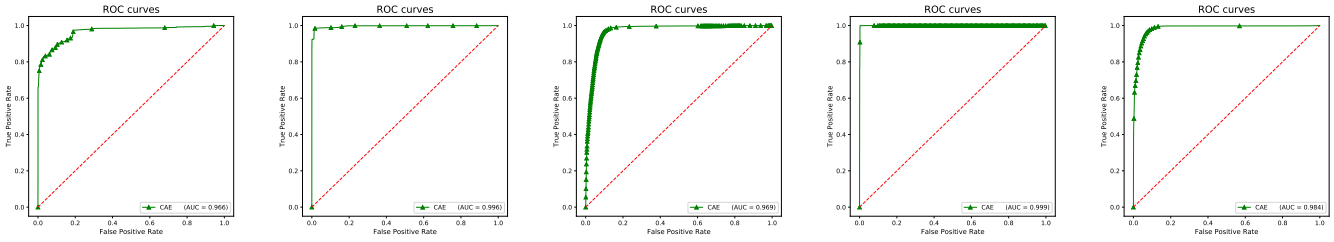
TABLE I
DATASETS FOR EVALUATING THE PROPOSED MODELS

No	Dataset	Dimension	Training set	Normal Test	Anomaly Test
1	NSL-KDD	122	67343	9711	12833
2	Rbot (CTU13-10)	38	6338	9509	63812
3	Murlo (CTU13-8)	40	29128	43694	3677
4	Neris (CTU13-9)	41	11986	17981	110993
5	Virut (CTU13-13)	40	12775	19164	24002

1) **NSL-KDD:** The NSL-KDD dataset is a newer filtered version of KDD99 dataset, which was introduced by Tavallaee et al. to overcome the inherent issues of KDD99

TABLE II
AUCs OF SAE-OCCs, DVAE-OCCs,CAE AND PCA + CAE MODELS.

Represen--tation	One-class Classifiers	Datasets				
		NSL-KDD	CTU13-08	CTU13-09	CTU13-10	CTU13-13
SAE	CEN	0.963	0.991	0.950	0.999	0.969
$\lambda = 10$	MDIS	0.964	0.990	0.950	0.999	0.968
DVAE	CEN	0.960	0.982	0.956	0.999	0.963
$\lambda = 0.05$, $\alpha = 10^{-8}$	MDIS	0.961	0.984	0.957	0.999	0.964
CAE		0.963	0.994	0.959	0.996	0.979
PCA + CAE		**0.966**	**0.996**	**0.969**	**0.999**	**0.984**



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

Fig. 5. The ROC curves of our proposed model on five datasets

[26]. Although this new dataset still has a number of issues that have been discussed in [18], current studies still use this dataset. Therefore, we believe that it is still a effective enough dataset for the research community to conduct experiments and evaluate their methods. In general, NSL-KDD has the same architecture as KDD99, specifically it has 22 attack patterns and normal traffic. Each data record contains 41 features; among these features are three categorical features including protocol type, service, and flag. They are preprocessed using one-hot-encoding which increases the number of features to 122.

2) **CTU13:** The CTU13 is a botnet dataset, which was captured in 2011 at CTU University, Czech Republic. This dataset is a huge collection of real botnet traffic, normal and background traffic. In this work, four scenarios (CTU13- 8, CTU13-9, CTU13-10 and CTU13-13) are chosen. A detailed description of each scenario is provided in Table I. There are three categorical features including dTos, sTos and protocol, which are encoded by using the one-hot encoding technique.

Each of these datasets were split into 40% for training (normal observations) and 60% for evaluation purposes (both normal and anomaly samples).

B. Experiments Settings

In this work, we conducted experiments consisting of two stages. In the first stage, we implement PCA for feature selection. In the second stage, we implement the proposed CAE model, the exact configuration of which is as follows. The number of hidden layers is 5, and the size of latent layer is defined by using the equation $h = [1 + \sqrt{n}]$, where n is the number of input features as introduced in [5]. We used the Xavier initialization method to initialize the weights of CAE to facilitate the convergence process. The chosen activation function is Tanh, the batch size is set as 100, the optimization algorithm is Adadelta and the learning rate is set to 0.1. The early stopping method is also applied, with an evaluation step at every 5 epochs.

We will conduct two experiments for evaluating our proposed approach. Firstly, the performance of our proposed model is compared with SAE, DVAE in [6] and CAE in [23]. Therefore, we reproduce the same experiments as in [6] [23], and report the performance of SAE, DVAE, CAE as shown in Table II. Secondly, we train and evaluate the proposed model under the same conditions as in [6] [23] and also visualize the Area Under the ROC curves when evaluating PCA+CAE models on the five datasets as shown in Fig 5.

VI. RESULTS AND DISCUSSION

In this section, we present the promising results obtained from our experiments. The performance of the trained models was evaluated using the AUC, which is summarized in detail in the Table (II). The ROC curves generated by our proposed model on the datasets are also visualized in Fig 5. It can be seen very clearly in Table II that, in terms of classification accuracy, the proposed PCA+CAE model in this paper has outperformed the results of previous SAE, DVAE and CAE models on all five datasets. Specifically, with the data set NSL-KDD, when using SAE, DVAE models with two classifiers CEN and MDIS, the accuracy obtained is 0.963; 0.964; 0.960; 0.961, respectively. While using the original version of CAE, the accuracy is 0.963, the proposed PCA+CAE model give a better outcome of 0.966. With the dataset CTU13-10, most of the methods give very high accuracy results of

0.999. Experimental results on datasets CTU13-08; CTU13-09; CTU13-13 show that the proposed model PCA+CAE has very effective performance clearly outperforming other methods. The promising results on the above datasets are 0.996; 0.969 and 0.984, respectively. This suggests that the data preprocessing stage using PCA has well supported the CAE model in discovering latent features and properly clustering the data at the bottleneck layer. Then the CAE model tends to balance very well the two components of the objective function including reconstruction loss and clustering loss. Therefore, data points are grouped into more suitable clusters, it results the normal data region tighter and easier to distinguish outliers.

Overall, the results of experiments confirm that the proposed method in this paper is a promising method and has contributed greatly to improving the performance of anomaly detection model based on one-class training strategy.

VII. Conclusion and Future Work

A novel method is proposed to improve the performance of network anomaly detection by combining PCA and CAE in a semi-supervised manner. This method aims to overcome the limitations of the previous methods in [6] [23]. This method consists of two specific stages as follows: The first stage is implementing PCA to find a new representation space of the original data that is more suitable at describing the normal data. The second stage is applying a CAE to learn the latent representation of the normal data and also force the data points into appropriate clusters in the normal data region. We have evaluated the proposed model using five different datasets including NSL-KDD and four scenarios in CTU13. Experimental results have shown that this new method is superior to previous methods on all selected datasets. Our future work will focus on expanding the study to include other data preprocessing methods and also to investigate methods for producing more robust, suitable features before training the CAE models in a one-class training manner.

Acknowledgment

This research is funded by the project “A smart network surveillance system based on artificial intelligence” under Vinh Phuc Province Research Programs (Grant no.20/DTKHVP/2021-2022).

References

[1] Babu, M.R., Veena, K.: A survey on attack detection methods for iot using machine learning and deep learning. In: 2021 3rd International Conference on Signal Processing and Communication (ICPSC). pp. 625–630. IEEE (2021)

[2] Bank, D., Koenigstein, N., Giryes, R.: Autoencoders. arXiv preprint arXiv:2003.05991 (2020)

[3] Bhatt, S., Ragiri, P.R., et al.: Security trends in internet of things: A survey. SN Applied Sciences **3**(1), 1–14 (2021)

[4] Bishop, C.M.: Pattern recognition. Machine learning **128**(9) (2006)

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

[6] Cao, V.L., Nicolau, M., McDermott, J.: Learning neural representations for network anomaly detection. IEEE Transactions on Cybernetics **49**(8), 3074–3087 (2019). https://doi.org/10.1109/TCYB.2018.2838668

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

[8] Chandola, V., Banerjee, A., Kumar, V.: Survey of anomaly detection. ACM Computing Survey (CSUR) **41**(3), 1–72 (2009)

[9] Chen, Y., Ashizawa, N., Yean, S., Yeo, C.K., Yanai, N.: Self-organizing map assisted deep autoencoding gaussian mixture model for intrusion detection. In: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC). pp. 1–6. IEEE (2021)

[10] Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)

[11] Han, X., Liu, Y., Zhang, Z., Lü, X., Li, Y.: Sparse auto-encoder combined with kernel for network attack detection. Computer Communications **173**, 14–20 (2021)

[12] Hassan, R.J., Zeebaree, S.R., Ameen, S.Y., Kak, S.F., Sadeeq, M.A., Ageed, Z.S., Adel, A.Z., Salih, A.A.: State of art survey for iot effects on smart city technology: challenges, opportunities, and solutions. Asian Journal of Research in Computer Science pp. 32–48 (2021)

[13] Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length, and helmholtz free energy. Advances in neural information processing systems **6**, 3–10 (1994)

[14] Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of educational psychology **24**(6), 417 (1933)

[15] Injadat, M., Salo, F., Nassif, A.B., Essex, A., Shami, A.: Bayesian optimization with machine learning algorithms towards anomaly detection. In: 2018 IEEE global communications conference (GLOBECOM). pp. 1–6. IEEE (2018)

[16] Jolliffe, I.T.: Principal component analysis, 2nd, edn (2002)

[17] Liang, X., Kim, Y.: A survey on security attacks and solutions in the iot network. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0853–0859. IEEE (2021)

[18] McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. ACM Transactions on Information and System Security (TISSEC) **3**(4), 262–294 (2000)

[19] Mehrotra, K.G., Mohan, C.K., Huang, H.: Anomaly detection principles and algorithms. Springer (2017)

[20] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y.: N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing **17**(3), 12–22 (2018)

[21] Muhammad, G., Hossain, M.S., Garg, S.: Stacked autoencoder-based intrusion detection system to combat financial fraudulent. IEEE Internet of Things Journal (2020)

[22] Nassif, A.B., Talib, M.A., Nasir, Q., Dakalbab, F.M.: Machine learning for anomaly detection: A systematic review. IEEE Access (2021)

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

[24] Pang, G., Cao, L., Aggarwal, C.: Deep learning for anomaly detection: Challenges, methods, and opportunities. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 1127–1130 (2021)

[25] Salo, F., Injadat, M., Nassif, A.B., Shami, A., Essex, A.: Data mining techniques in intrusion detection systems: A systematic literature review. IEEE Access **6**, 56046–56058 (2018)

[26] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. pp. 1–6 (2009). https://doi.org/10.1109/CISDA.2009.5356528

[27] Tsai, C.F., Hsu, Y.F., Lin, C.Y., Lin, W.Y.: Intrusion detection by machine learning: A review. expert systems with applications **36**(10), 11994–12000 (2009)

[28] Vu, L., Cao, V.L., Nguyen, Q.U., Nguyen, D.N., Hoang, D.T., Dutkiewicz, E.: Learning latent representation for iot anomaly detection. IEEE Transactions on Cybernetics pp. 1–14 (2020). https://doi.org/10.1109/TCYB.2020.3013416