Deep Nested Clustering Auto-Encoder for Anomaly-Based Network Intrusion Detection

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

Thanh Long Ngo
Le Quy Don Technical University
ngottlong@mta.edu.vn

Le Minh Nguyen
Japan Advanced Institute of Science and Technology
nguyenml@jaist.ac.jp

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

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

***Abstract*—Anomaly-based intrusion detection system (AIDS) plays an increasingly important role in detecting complex, multi-stage network attacks, especially zero-day attacks. Although there have been improvements both in practical applications and the research environment, there are still many unresolved accuracy-related concerns. The two fundamental limitations that contribute to these concerns are: i) the succinct, concise, latent representation learning of the normal network data, and ii) the optimization volume of normal regions in latent space. Recent studies have suggested many ways to learn the latent representation of normal network data in a semi-supervised manner to construct AIDS. However, these approaches are still affected by the above limitations, mainly due to the inability to process high data dimensionality or ineffectively explore the underlying architecture of the data. In this paper, we propose a novel Deep Nested Clustering Auto-Encoder (DNCAE) model to thoroughly overcome the aforementioned difficulties and improve the performance of network attack detection. The proposed model consists of two nested Deep Auto-Encoders (DAE) to learn the informative and tighter data representation space. In addition, the DNCAE model integrates the clustering technique into the latent layer of the outer DAE to learn the optimal arrangement of data points in the latent space. This harmonious combination allows us to effectively deal with the limitations outlined. The performance of the proposed model is evaluated using standard datasets including NSL-KDD, UNSW-NB15, and six scenarios of CIC-IDS2017 (Tuesday, Wednesday, Thursday-Morning, Friday-Morning, Friday-Afternoon-PortScan, Friday-Afternoon DDoS). The experimental results strongly confirm that the proposed model clearly outperforms the baselines and the existing methods for network anomaly detection.**

***Index Terms*—Latent Representation, Deep Auto-Encoder, Clustering, Anomaly Detection, Intrusion Detection System**

I. Introduction

Today, the world is entering the digital transformation progress of industrial revolution 4.0. The advent of innovative technologies has created unprecedented rapid growth in history [1]. For instance, the explosive increase of internet connectivity and global communication technologies have brought great benefits to mankind. As a result, people are experiencing new services such as smart homes, intelligent transportation, smart education and online transactions [2]. However, besides the positive aspects of these developments, the challenges in cyberspace are increasingly complex and unpredictable. Specifically, cyber-attack campaigns have sharply increased in volume and span around the globe [3]. In addition, attackers are always evolving, using well-designed, sophisticated attack techniques to disrupt the availability, confidentiality, and integrity of information systems. Especially, attacks using vulnerable compromised Internet-of-Things (IoTs) devices, Advanced Persistent Threat (APT) attack campaigns, and zero-day attacks are fierce challenges for network administrators [4]. Detecting and preventing these attacks are difficult tasks because of the large amounts of high-dimensional data, heterogeneity, and the diversity of attack techniques.

Network intrusion detection systems (NIDS) have been widely accepted as reliable, efficient, and potential solutions for dealing with the aforementioned difficulties [5]. In general, based on detection techniques, NIDS can be divided into two categories including signature-based IDS (SIDS) and Anomaly-based IDS (AIDS) [6]. SIDS detects network attacks by matching network traffic against a predefined database of known attack signatures. Although these solutions give high accuracy for known attacks, they are not capable of detecting unknown attacks or variants of known attacks. In addition, the need to regularly update the attack signatures and the matching time required for large attack signature databases are also limitations of SIDS. On the other hand, AIDS builds a profile of common, expected behaviors in the network environment. Then, any deviation higher than a predefined threshold is labeled as anomalous behavior. Various solutions have been proposed to build AIDS and it has been shown that these approaches are capable of detecting unknown attacks. However, the accuracy and false positive rates of these methods are still high and need to be further improved [7].

In recent years, Deep Learning (DL) has proven its effectiveness in fields such as big data processing, image recognition, natural language processing and video processing [8]. Therefore, DL has occupied the first choice for new smart solutions. Specifically, in cyber security, researchers are using various neural network architectures such as Deep Belief Networks (DBNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs)and Autoencoders (AEs) for building AIDS [9].

Among these architectures, with the ability to learn a latent representation of network traffic, the AE model is emerging as the widely applied model for network anomaly detection [10]. In this work, we are concerned with two core limitations of the above solutions, which are barriers to developing an efficient network anomaly detector. Firstly, the proposed models have not yet demonstrated an efficient, expressive latent representation space to build a profile of network normal data. Therefore, it is difficult to detect anomalies in such a learned space. This can result from a variety of reasons such as the dynamic nature of network normal behavior or the emergence of new multiple protocols, modern types of equipment, etc. However, we argue that the underlying cause of this issue is that the proposed models have not yet produced a general representation space suitable to fully describe the normal network data. In other words, the proposed methods have not yet learned the best and most outstanding features of the normal network data to strongly support anomaly detection on such representation space. Secondly, in learned latent representation spaces, it is not efficient to determine the distribution and boundary of the normal data region. There are also many factors contributing to this limitation such as the inherent diversity of normal network data, poor data sampling, etc. However, we argue that the previously proposed methods have not used suitable regularizers to push the normal data points closer together in the latent space, which minimizes the normal data region. In other words, the arrangement of the normal data points in the latent space is not optimal. Therefore, to improve the performance of network anomaly detectors, it is necessary to learn a tighter and more compact latent representation space from the normal network data.

In this work, we introduce a new DL-based solution to overcome the aforementioned limitations. Particularly, our proposed model consists of two nested Deep Auto-Encoders (DAE), which are called outer DAE and inner DAE. In addition, in the latent layer of the outer DAE, the K-means algorithm is integrated to push the normal data points in the same sub-clusters closer together, as well as the centers of the sub-clusters to move closer together. As a result, a better arrangement of the normal network data points in the latent representation space of the outer DAE will be established. Then, the inner DAE squeezes this normal network data space to create a tighter data representation area. This model is called Deep Nested Clustering Auto-Encoder (DNCAE), which aims to overcome the above limitations. We will estimate the performance of the proposed model using standard data sets including NSL-KDD, UNSW-NB15, and six scenarios of CIC-IDS2017. A detailed description of DNAE will be presented in Section IV. In summary, our major contributions in this work are as follows:

1) We propose a novel DL-based approach that consists of two nested DAEs and integrates the K-means clustering algorithm into the latent layer of the outer DAE. Our proposed model will produce a tighter data representation space thus improving anomaly detection performance significantly.
2) We conduct extensive experiments on benchmark datasets (NSL-KDD, UNSW-NB15, and six scenarios of CIC-IDS2017) to evaluate the performance of the proposed model. The experimental results have demonstrated that DNCAE works better in comparison with baselines and existing models.
3) We study the influence of the number of clusters in K-means on the model's performance. Furthermore, we conduct a comprehensive analysis of the experimental results on all selected benchmark datasets.

The remainder of this work is organized as follows. We review several prominent and latest research on network anomaly detection in Section II. Then, Section III will briefly present the mathematical background of the DAE model. Our proposed DNCAE model is described in Section IV. Experiments, results, and discussion are presented in Sections V and VI, respectively. Finally, we summarize our paper in Section VII and will highlight future research directions.

II. EXISTING WORKS

In this section, we will review some recent prominent works for network anomaly detection.

The authors in the work [11] proposed a method using 1D CNN architecture to detect network anomalies. Specifically, they divide network data based on connection protocols including TCP, UDP, and Other. Then, each protocol is investigated independently of the others. Before conducting the model training process, some feature selection techniques are used to improve the performance of the proposed model. The experiments performed on the UNSW-NB15 produced F-score results of 0.85, 0.97, and 0.86 for TCP, UDP, and Other protocols respectively. However, this method has not been implemented on other standard datasets to estimate performance.

Researchers in [12] presented a DL framework to build an AIDS. Specifically, this solution consists of three different stages, which are a combination of unsupervised K-means clustering, semi-supervised GANomaly, and supervised learning CNN architecture. They argue that the experimental results on the datasets including NSL-KDD, CIC-IDS2018, and TON IoT are better than other methods. Particularly, this solution produced a lower false positive rate with a comparable true positive rate. However, in this paper, they have not shown how to choose the number of clusters for the K-means algorithm and have not compared it with the latest methods.

In the paper [13], authors have proposed a novel DL solution for network anomaly detection. Specifically, the K-means clustering algorithm is integrated into a hidden layer of the Encoder part in the Variational Autoencoder (VAE) model. They train and estimate the proposed model using standard data sets including NSL-KDD, UNSW-NB15, CIC-IDS2017, and five scenarios from CTU13. The experimental results show that the model produces better performance than the previous existing models in terms of Area Under The Curve (AUC). However, the limitation of this paper is that they only use a Centroid-based one-class classifier for testing.

Sultan Zavrak et al. in [14] introduced an AE-based method to build anomaly-based IDS. In this paper, the authors use a VAE model to detect network anomalies. They used the benchmark CIC-IDS2017 to train and test the proposed model. They argue that VAE gives better performance than AE and One-class Support Vector Machine (OCSVM) in terms of AUC.

In general, there are a number of DL-based solutions for network anomaly detection. However, these solutions have not yet learned expressive, compact latent representation space from normal network data supporting to improve network anomaly detection performance. In this paper, we will propose a new approach to overcome the discussed limitations. In the next section, background knowledge related to our proposed model will be presented.

III. BACKGROUND

In this section, the background knowledge needed to build the proposed model will be presented in detail. Deep Auto-Encoder (DAE) is a neural network architecture widely used for applications such as data dimensionality reduction, denoising, clustering, and anomaly detection [8], [15]. The primary goal of training a DAE is to discover rich, expressive, and informative representation spaces from data that can be used for a variety of applications. The internal structure of a DAE consists of two main components called Encoder and Decoder parts [8]. Generally, in modern DAE models, Encoder and Decoder parts are both deep neural architectures with non-linear activation functions. The main task of the Encoder part is to map data from the original input space to the latent space using the mapping function $\mathbf{F}(.)$. In contrast, the responsibility of the Decoder part is to reconstruct from the latent space to the input space using the mapping function $\mathbf{G}(.)$. Given the training set $\mathbf{X} = \{\mathbf{x}_i | i = 1, 2, 3, ...n\}$; $\mathbf{H} = \{\mathbf{h}_i | i = 1, 2, 3, ...n\}$ latent representation space; $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_i | i = 1, 2, 3, ...n\}$ is the reconstructed version of $\mathbf{X}$. The general functions of the Encoder and Decoder are shown as follows:

$$\mathbf{H} = \mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X}) \tag{1}$$

$$\hat{\mathbf{X}} = \mathbf{G}_{\boldsymbol{\Phi}}(\mathbf{H}) = \mathbf{G}_{\boldsymbol{\Phi}}\big(\mathbf{F}_{\boldsymbol{\Theta}}(\mathbf{X})\big) \tag{2}$$

Where n is the number of observations in the training dataset; $\mathbf{x}_i, \hat{\mathbf{x}}_\mathbf{i} \in \mathbf{R}^D$; $D \in \mathbf{N}$ is the dimension of the input data; $\mathbf{h}_i \in \mathbf{R}^d$; $d \in \mathbf{N}$ is the dimension of the latent space; $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ are weights and biases matrices of the Encoder and Decoder, respectively. Through the training process, DAE aims to discover a meaningful latent representation space that can have many positive effects on tasks such as feature extraction, or clustering. The training process of a DAE focuses on minimizing the reconstruction error between the input and output data. In other words, the learning process is the search for functions $\mathbf{F}_{\mathbf{\Theta}}(.)$ and $\mathbf{G}_{\mathbf{\Phi}}(.)$ that satisfy the condition:

$$\underset{\mathbf{F},\mathbf{G}}{\operatorname{argmin}} \mathbf{E}\Big[\mathcal{L}\Big(\mathbf{x}_i, \mathbf{G}\Big[\mathbf{F}(\mathbf{x}_i)\Big]\Big)\Big] \tag{3}$$

Where, $\mathcal{L}$ is the loss function, which is a measure of how the input and the output of a DAE differ; $\mathbf{E}$ is the expectation over the training set $\mathbf{X}$. Two objective functions are widely used for training a DAE including Mean Squared Error (MSE) and Binary Cross-Entropy (BCE). The mathematical notation of these loss functions are as follows:

$$\mathcal{L}_{\mathrm{MSE}}\Big(\mathbf{X}, \hat{\mathbf{X}}\Big) = \frac{1}{n}\sum_{i=1}^{n}\Big(\mathbf{x}_i - \hat{\mathbf{x}}_i\Big)^2 \tag{4}$$

$$\mathcal{L}_{\mathrm{BCE}}\Big(\mathbf{X}, \hat{\mathbf{X}}\Big) = -\frac{1}{n}\sum_{i=1}^{n}\Big(\mathbf{x}_i \log(\hat{\mathbf{x}}_i) + (1 - \mathbf{x}_i)\log(1 - \hat{\mathbf{x}}_i)\Big) \tag{5}$$

IV. Proposed Methodology

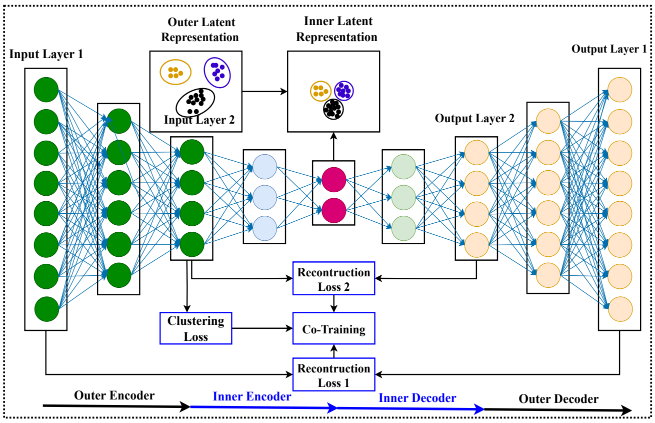


Fig. 1. Deep Nested Clustering Auto-Encoder (DNCAE)

Our goal is to build a powerful, informative latent representation space from only normal network data, which facilitates the performance of conventional anomaly detection techniques. The proposed model aims to overcome the limitations pointed out in recent previous works [16], [14], [17]. In order to do that, we designed a model capable of learning the compact, tight representation from normal network data, in which the arrangement of data points is as optimal as possible. Based on the observation that the data is naturally clustered in nature [18], even though the training data is normal network data, there is an underlying clustered architecture within it. This can be explained because normal network data is generated from many different devices, services, protocols, and common user behavior.

Therefore, we propose a novel DL-based model, which is called Deep Nested Clustering Auto-Encoder (DNCAE). DNCAE has the ability to capture the underlying clustering architecture as well as minimize the "normal region" on the latent representation to stronger support anomaly detection. Our proposed model takes advantage of DAE's rich latent representation learning capability in combination with the efficient clustering ability of K-means. Specifically, our proposed model consists of two nested DAEs, which are called Outer DAE and Inner DAE. In addition, the K-means clustering technique is integrated into the latent layer of the outer DAE, before entering the Inner DAE. The task of the Outer DAE is to explore the meaningful latent representation space from the normal network data and also push the data points in this space to sub-clusters as optimally as possible. The responsibility of the Inner DAE is to distill the most core, prominent features and simultaneously minimize the volume of the "normal region" in latent space. The overall architecture of DNCAE is depicted in Figure 1. The objective function of DNCAE includes 3 components, which are the reconstruction error of the Outer DAE, the reconstruction error of the Inner DAE, and the clustering error as follows:

$$\mathcal{L}_{\mathrm{DNCAE}} = \alpha_1\mathcal{L}_{\mathrm{Outer}}\Big(\mathbf{X}, \hat{\mathbf{X}}\Big) + \alpha_2\mathcal{L}_{\mathrm{Inner}}\Big(\mathbf{H}, \hat{\mathbf{H}}\Big) + \alpha_3\Omega(\mathbf{H}) \tag{6}$$

Where $\mathcal{L}_{\mathrm{Outer}}\Big(\mathbf{X}, \hat{\mathbf{X}}\Big)$ and $\mathcal{L}_{\mathrm{Inner}}\Big(\mathbf{H}, \hat{\mathbf{H}}\Big)$ are reconstruction losses of Outer DAE and Inner DAE, respectively; $\Omega(\mathbf{H})$ is clustering loss at thr latent representation of the Outer DAE. The method of calculating this clustering error is presented very specifically in the work [16]. The only difference used in this work, is that we use a variation of K-means. Specifically, this version of K-means minimizes the distance between data points in the same cluster and pulls the cluster centers closer together; $\mathbf{X} = \{\mathbf{x}_i | i = 1, 2, 3, ...n\}$ is training dataset of n observations. $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_i | i = 1, 2, 3, ...n\}$ is the reconstructed version of $\mathbf{X}$ through the Outer DAE; $\mathbf{H} = \{\mathbf{h}_i | i = 1, 2, 3, ...n\}$ is a latent representation of the Outer DAE; $\hat{\mathbf{H}} = \{\hat{\mathbf{h}}_i | i = 1, 2, 3, ...n\}$ is a reconstructed version of $\mathbf{H}$ through the Inner DAE; The coefficients α_1, α_2, and α_3 are used to balance these loss components in the objective function. In our work, the reproduced error components have the following specific forms:

$$\mathcal{L}_{\mathrm{Outer}}\Big(\mathbf{X}, \hat{\mathbf{X}}\Big) = \mathcal{L}_{\mathrm{MSE}}\Big(\mathbf{X}, \hat{\mathbf{X}}\Big) = \frac{1}{n}\sum_{i=1}^{n}\Big(\mathbf{x}_i - \hat{\mathbf{x}}_i\Big)^2 \tag{7}$$

$$\mathcal{L}_{\mathrm{Inner}}\Big(\mathbf{H}, \hat{\mathbf{H}}\Big) = \mathcal{L}_{\mathrm{MSE}}\Big(\mathbf{H}, \hat{\mathbf{H}}\Big) = \frac{1}{n}\sum_{i=1}^{n}\Big(\mathbf{h}_i - \hat{\mathbf{h}}_i\Big)^2 \tag{8}$$

We train the DNCAE model in a co-training manner using only normal network data. The weights and biases of deep neural architecture are updated using the Stochastic Gradient Descent (SGD) algorithm [8]. In general, our proposed methodology consists of two sequential phases. Firstly, we train the DNCAE model to learn informative, meaningful latent representation space from the normal network data. Secondly, this learned representation is used to train simple One-Class Classifiers (OCC) including OCSVM with four different kernels (Linear, Poly, RBF and Sigmoid), Isolation Forest (IF), Elliptic Envelope (EP), Local Outlier Factor (LOF), Kernel Density Estimation (KDE) and Centroid (CEN) [19], [20].

V. EXPERIMENTS

In this section, we will briefly present the benchmark data sets, which are used to evaluate the proposed model and compare it with the baseline models as well as the state-of-the-art models. In addition, we will show the configuration settings used for the experimental implementation.

A. Datasets

In order to evaluate the performance of the proposed model, we use standard data sets including NSL-KDD, UNSW-NB15, and six scenarios in CIC-IDS2017 (Tuesday, Wednesday, Thursday Morning, Friday-Bot, Friday-DDoS, Friday Port Scan). Details of these data sets are given in Table I.

TABLE I
DATASETS FOR EVALUATING THE PROPOSED MODELS

No	Dataset	Dimension	Normal Training	Normal Test	Anomaly Test
1	NSL-KDD	122	67343	9711	12833
2	UNSW-NB15	196	55999	36999	45332
3	Tuesday	78	260830	171244	13835
4	Wednesday	78	264016	176015	252,672
5	Thursday	78	100910	67276	2180
6	Friday-BOT	78	113500	75567	1966
7	Friday-PortScan	78	48859	78678	158930
8	Friday-DDoS	78	58630	39088	128027

1) NSL-KDD: NSL-KDD is an improved version of the KDD99 dataset, designed to overcome some of the inherent limitations identified by the research community [21]. Although this new version has its own limitations, it is still used to compare IDS solutions. NSL-KDD consists of normal network data and 22 different attack types, in which each data point has 41 features.

2) UNSW-NB15: UNSW-NB15 was published at the Cyber Range Lab in New South Wales in 2015 [22]. This is a dataset published with the desire to overcome the limitations of the previous data sets including KDD99 and NSL-KDD. UNSW-NB15 consists of normal network data and 9 different attack data samples.

3) CIC-IDS2017: CIC-IDS2017 is a dataset published in 2017 at the Canadian Institute for Cybersecurity (CIC) [23]. The data set includes 8 different scenarios, including many modern attack patterns including DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltration, Port Scan, and Botnets. Each data point has 78 features.

B. Experiments Settings

In this work, we conduct experiments using data sets NSL-KDD, UNSW-NB15, and 6 cases of CIC-IDS2017 including Tuesday, Wednesday and Thursday Morning, Friday-Bot, Friday PortSca and Friday-DDoS. In the implementation process, we take 60% of the normal network data for training and the remaining 40% is combined with the attack data sample to make a testing set.

The architecture of the proposed DNCAE model is designed as follows: The Encoder parts of the Outer DAE and Inner DAE consist of 4 hidden layers. More specifically, the latent representation layer of the Outer DAE is used as the Input Layer of Inner DAE. Dimensions of the latent layers of the Outer DAE and Inner DAE are calculated using the formula $d_h = [1 + \sqrt{D}]$ as in [13]. Where D is the number of features at the Input layer of each DAE, and d_h is the dimension of the latent space of each DAE. The Xavier initialization technique is used to initialize the weights of the proposed model DNCAE to speed up the convergence process. The Tanh function is used as an activation function, the batch size is set to 256, the Adadelta optimization algorithm is used, and the learning rate is set to 0.01. The coefficients α_1, α_2, α_3 in the objective function (Equation 6) are selected based on the grid search method to determine the best performance of the model.

To evaluate the performance of the proposed model, we conduct three different sets of experiments. Firstly, we conduct experiments using Stand-alone OCC classifiers (OCSVM with four different kernels (Linear, Poly, RBF and Sigmoid Kernels), Isolation Forest (IF), Elliptic Envelope (EP), Local Outlier Factor (LOF), Kernel Density Estimation (KDE) and Centroid (CEN), on each data set to record the effectiveness of these classifiers. Then, we use the DAE model to learn the latent representation space of the normal network data before training the aforementioned OCC classifiers. We use these results as a baseline. Secondly, we reconstruct the experiments using Clustering-based DAE (DCAE) [16] and stacked PCA and DCAE model (PCADCAE) [17] as state-of-the-art methods for learning the latent space of the normal network data before training the OCC classifiers. Finally, we train the proposed model DNCAE to learn the expressive, compact latent space and then fit it into the aforementioned OCC classifiers in the same experimental conditions. For measuring the performance of baseline, previous, and proposed models, we evaluate the AUC. In addition, we conduct experiments to study the influence of the number of clusters in the latent spaces of Outer DAE applied to the K-means algorithm on the performance of the proposed model. In practice, all experiments are implemented in Python 3.10 using the Pytorch framework and run on a machine with an Intel Core 2 Duo i5-825 CPU at 2.8 GHz, 16 GB RAM with a frequency of 1600 MHz.

VI. RESULTS AND DISCUSSION

In this section, we will present the experimental results of the proposed model on the benchmark data sets NSL-KDD, UNSW-NB15, and 6 scenarios of the CIC-IDS2017 (Tuesday, Wednesday and Thursday Morning, Friday-Bot, Friday-DDoS and Friday Port Scan). The performance of the proposed model is compared with the baseline and most recent works in terms of the AUC score, which is a reliable measure to compare the performance of anomaly detectors. The experimental results are shown in Table II.

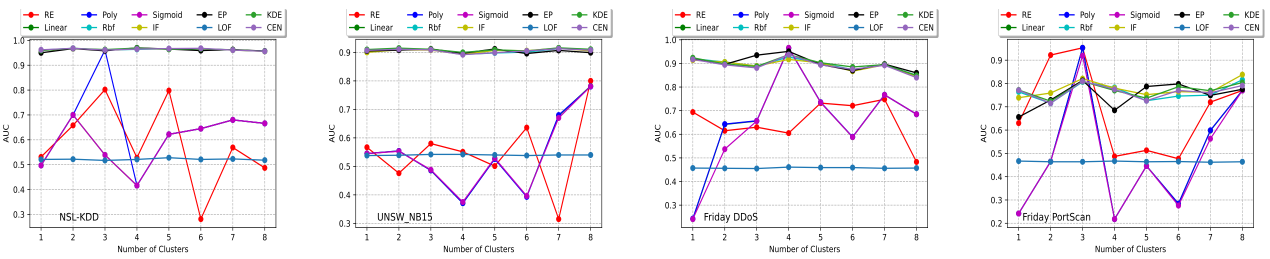
In general, the proposed model DNCAE has demonstrated powerful and efficient latent representation learning ability from normal network data. As a result, this learned representation strongly supports simple anomaly detectors including OCSVM with four different kernels (Linear, Poly, RBF, Sigmoid Kernels), IF, EP, LOF, KDE, and CEN. For the NSL-KDD data set, the latent representation space generated from DNCAE helped the OCCs (OCSVM Poly, OCSVM Rbf, IF, EP, CEN) generate AUC scores that outperformed the stand-alone OCCs, DAE+OCCs, DCAE+OCCs, and PCA-DCAE+OCCs models. Specifically, the AUCs of these anomaly detectors are 0.961, 0.968, 0.967, 0.970, and 0.968, respectively. Among the anomaly detectors, DNCAE+EP achieved the highest AUC score of 0.970 when compared with all other methods. In addition, based on Reconstruction Error (RE) to identify anomalies, the proposed method also gives better results than other methods. Promising experimental results on the set UNSW-NB15 have proved that the proposed model has the effective ability to learn data representation of normal network data. Particularly, DNCAE+ all OCCs classifiers outperform the baseline methods and the latest methods including DCAE+OCCs and PCA-DCAE +OCCs. Among them, the best-performing anomaly detectors are OCSVM Rbf, IF, KDE, and CEN. Their AUCs scores are 0.913, 0.915, 0.916, and 0.913 respectively.

Experimental results on 6 scenarios of the CIC-IDS2017 set are strong evidence for the performance of the proposed model. Especially with Friday Port Scan and Friday DDoS scenarios, DNCAE's data representation space has supported 7 anomaly detectors with the highest results when compared to other methods. Specifically, the AUC score reached the highest for Friday Port Scan and Friday DDoS at 0.953 and 0.966 respectively. For the remaining scenarios, 5 out of 9 anomaly detectors produce promising results, which are superior in comparison with other approaches.

Experimental results on all selected data sets have shown that combining three loss components in the objective function gives better results than only one or two of the three components in the DAE and DCAE models. In addition, in the presence of the reconstruction loss of the Inner DAE, the proposed model not only learns the dominant representative features of the normal network

TABLE II
AUCS OF OCCS, DAE+OCCS, DCAE+OCCS, PCADCAE+OCCS, DNCAE+OCCS MODELS

Latent Representation	**One-Class Classifiers**	**Datasets**							
		NSL-KDD	UNSW-NB15	Tuesday	Wednesday	Thurday Morning	Friday BOT	Friday PortScan	Friday DDoS
None	OCSVM Linear	0.785	0.646	0.252	0.164	0.266	0.448	0.270	0.381
	OCSVM Poly	0.780	0.622	0.370	0.211	0.265	0.435	0.264	0.398
	OCSVM Rbf	0.812	0.655	0.560	0.789	0.726	0.739	0.740	0.588
	OCSVM Sigmoid	0.787	0.668	0.260	0.175	0.265	0.455	0.288	0.374
	IF	0.817	0.672	0.493	0.803	0.483	0.473	0.457	0.531
	EP	0.816	0.736	0.494	0.544	0.459	0.453	0.448	0.737
	LOF	0.530	0.541	0.453	0.464	0.448	0.476	0.455	0.448
	KDE	0.954	0.881	0.701	0.912	0.681	0.698	0.757	0.843
	CEN	0.955	0.743	0.602	0.893	0.650	0.545	0.497	0.632
DAE	RE	0.623	0.622	0.711	0.710	0.534	0.523	0.612	0.637
	OCSVM Linear	0.869	0.615	0.490	0.082	0.584	0.619	0.195	0.181
	OCSVM Poly	0.871	0.618	0.311	0.156	0.461	0.356	0.451	0.264
	OCSVM Rbf	0.937	0.823	0.435	0.751	0.465	0.517	0.246	0.610
	OCSVM Sigmoid	0.871	0.619	0.493	0.213	0.463	0.256	0.082	0.343
	IF	0.951	0.834	0.747	0.873	0.588	0.676	0.571	0.793
	EP	0.936	0.789	0.692	0.889	0.797	0.671	0.729	0.815
	LOF	0.513	0.536	0.449	0.469	0.450	0.471	0.458	0.452
	KDE	0.946	0.857	0.740	0.879	0.643	0.632	0.651	0.773
	CEN	0.943	0.794	0.452	0.869	0.621	0.704	0.276	0.598
DCAE	RE	0.647	0.630	0.580	0.694	0.320	0.741	0.670	0.808
	OCSVM Linear	0.455	0.380	0.641	0.901	0.235	0.319	0.173	0.483
	OCSVM Poly	0.455	0.380	0.641	0.901	0.235	0.320	0.173	0.483
	OCSVM Rbf	0.967	0.888	0.508	0.897	0.677	0.677	0.761	0.871
	OCSVM Sigmoid	0.455	0.380	0.641	0.901	0.235	0.319	0.173	0.483
	IF	0.966	0.892	0.674	0.905	0.696	0.725	0.755	0.893
	EP	0.957	0.908	0.361	0.890	0.802	0.752	0.824	0.851
	LOF	0.521	0.537	0.468	0.466	0.459	0.476	0.465	0.455
	KDE	0.970	0.893	0.657	0.897	0.708	0.706	0.766	0.884
	CEN	0.967	0.878	0.639	0.877	0.669	0.707	0.755	0.880
PCADCAE	RE	0.756	0.589	0.619	0.579	0.436	0.523	0.401	0.657
	OCSVM Linear	0.768	0.746	0.727	0.444	0.072	0.605	0.406	0.270
	OCSVM Poly	0.748	0.735	0.727	0.444	0.072	0.605	0.406	0.270
	OCSVM Rbf	0.965	0.846	0.750	0.904	0.829	0.667	0.650	0.848
	OCSVM Sigmoid	0.768	0.746	0.428	0.444	0.072	0.605	0.406	0.300
	IF	0.963	0.843	0.732	0.904	0.813	0.639	0.657	0.849
	EP	0.962	0.852	0.793	0.807	0.783	0.711	0.728	0.885
	LOF	0.514	0.531	0.493	0.483	0.473	0.497	0.455	0.467
	KDE	0.962	0.845	0.743	0.907	0.819	0.700	0.674	0.860
	CEN	0.964	0.846	0.746	0.905	0.830	0.702	0.655	0.846
DNCAE	RE	**0.802**	**0.800**	0.680	0.436	**0.920**	0.644	**0.922**	0.748
	OCSVM Linear	0.700	**0.782**	**0.925**	0.398	**0.885**	**0.759**	**0.953**	**0.966**
	OCSVM Poly	**0.961**	**0.782**	**0.925**	0.398	**0.885**	**0.759**	**0.953**	**0.966**
	OCSVM Rbf	**0.968**	**0.913**	0.711	**0.938**	**0.742**	**0.746**	**0.816**	**0.927**
	OCSVM Sigmoid	0.700	**0.782**	**0.925**	0.398	**0.885**	**0.759**	**0.953**	**0.966**
	IF	**0.967**	**0.915**	**0.753**	**0.936**	0.803	0.712	**0.838**	**0.917**
	EP	**0.970**	**0.912**	0.708	**0.917**	0.774	0.649	0.798	**0.951**
	LOF	0.528	**0.542**	0.476	0.471	0.466	0.491	0.467	0.461
	KDE	0.969	**0.916**	0.735	**0.929**	0.795	0.700	**0.804**	**0.937**
	CEN	**0.968**	**0.913**	**0.788**	**0.933**	0.799	0.722	**0.812**	**0.923**



(a) NSL-KDD (b) UNSW-NB15 (c) Friday-DDoS (d) Friday-PortScan

Fig. 2. Investigation number of Clusters

data but also pushes the data points to more suitable clusters. As a result, it will minimize the normal region in the latent space. Consequently. when an anomalous data point occurs, it is easily separated from this normal region.

In Figure 2 we show the results of studying the influence of the number of clusters of the K-means algorithm in Outer DAE on the model's performance. In this experiment, NSL-KDD, UNSW-NB15 dataset, and 2 scenarios of the CIC-IDS2017 set (Friday PortScan, Friday DDoS) are used. The results have shown that most of the anomaly detectors (OCSVM Linear, OCSVM Linear, EP, KDE, LOF, CEN) are relatively stable with the number of clusters ranging from 1 to 8 on the selected data sets. In contrast, the OCSVM Poly, OCSVM Sigmoid, as well as RE-based anomaly detectors are very sensitive to the number of clusters in the K-means algorithm. Generally, the optimal number of clusters for the NSL-KDD, UNSW-NB15, Friday-DDoS, and Friday-PortScan are 3, 8, 4, and 4, respectively. In summary, the experimental results strongly confirmed that the proposed DNCAE model has a powerful representation learning capability from normal network data. As a result, simple anomaly detectors are effectively improved in terms of AUC scores from learned concise features.

VII. CONCLUSION AND FUTURE WORK

A novel DL approach is introduced to build anomaly-based IDSs in a semi-supervised manner. The proposed model aims to overcome the limitations of recently proposed methods [16], [17], [14] that effectively learn profiles of normal network data. In addition, the proposed model will provide a more optimal arrangement for normal network data points in the feature space to increase the efficiency of anomaly detection. The proposed model is a combination of two nested DAEs, in which the latent layer of the outer DAE is the input to the Inner DAE. At the latent layer of the Outer DAE, a variant of the K-means clustering algorithm is integrated to learn the optimal arrangement of normal network data points. As a result, the proposed model aims to achieve two parallel goals, which are learning significant, prominent features and minimizing the normal data regions so that anomalies will be easier to identify. We evaluate the proposed model on benchmark data sets including NSL-KDD, UNSW-NB15, and 6 cases of CIC-IDS2017 (Tuesday, Wednesday, Thursday Morning, Friday-Bot, Friday-PortScan, Friday-DDoS). Experimental results have clearly demonstrated that the proposed model supports anomaly detectors much better than the baselines and the latest methods in terms of the AUC score produced. In addition, we also study the effect of the number of clusters in the data sets on the performance of the model.

Our future work will focus upon extending the research toward learning the latent probability distribution of normal network data to explore even more compact and meaningful representation. Furthermore, we will build more experiments on many other cutting-edge datasets.

REFERENCES

[1] Weiyu Wang and Keng Siau. Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: A review and research agenda. *Journal of Database Management (JDM)*, 30(1):61–79, 2019.

[2] Naomi Haefner, Joakim Wincent, Vinit Parida, and Oliver Gassmann. Artificial intelligence and innovation management: A review, framework, and research agenda. *Technological Forecasting and Social Change*, 162:120392, 2021.

[3] Khushnaseeb Roshan and Aasim Zafar. Deep learning approaches for anomaly and intrusion detection in computer network: A review. *Cyber Security and Digital Forensics: Proceedings of ICCSDF 2021*, pages 551–563, 2022.

[4] Amit Sharma, Brij B Gupta, Awadhesh Kumar Singh, and VK Saraswat. Advanced persistent threats (apt): evolution, anatomy, attribution and countermeasures. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–27, 2023.

[5] Zhen Yang, Xiaodong Liu, Tong Li, Di Wu, Jinjiang Wang, Yunwei Zhao, and Han Han. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, 116:102675, 2022.

[6] Ziadoon Kamil Maseer, Robiah Yusof, Nazrulazhar Bahaman, Salama A. Mostafa, and Cik Feresa Mohd Foozy. Benchmarking of machine learning for anomaly based intrusion detection systems in the cicids2017 dataset. *IEEE Access*, 9:22351–22370, 2021.

[7] Oluwadamilare Harazeem Abdulganiyu, Taha Ait Tchakoucht, and Yakub Kayode Saheed. A systematic literature review for network intrusion detection system (ids). *International Journal of Information Security*, pages 1–38, 2023.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[9] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38, 2021.

[10] Youngrok Song, Sangwon Hyun, and Yun-Gyung Cheong. Analysis of autoencoders for network intrusion detection. *Sensors*, 21(13):4294, 2021.

[11] Mohammad Kazim Hooshmand and Doreswamy Hosahalli. Network anomaly detection using deep learning techniques. *CAAI Transactions on Intelligence Technology*, 7(2):228–243, 2022.

[12] Rahul Kale, Zhi Lu, Kar Wai Fok, and Vrizlynn LL Thing. A hybrid deep learning anomaly detection framework for intrusion detection. In *2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 137–142. IEEE, 2022.

[13] Van Quan Nguyen, Viet Hung Nguyen, Tuan Hao Hoang, and Nathan Shone. A novel deep clustering variational auto-encoder for anomaly-based network intrusion detection. In *2022 14th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–7. IEEE, 2022.

[14] Sultan Zavrak and Murat İskefiyeli. Anomaly-based intrusion detection from network flow features using variational autoencoder. *IEEE Access*, 8:108346–108358, 2020.

[15] Umberto Michelucci. An introduction to autoencoders. *arXiv preprint arXiv:2201.03898*, 2022.

[16] Van Quan Nguyen, Viet Hung Nguyen, Nhien-An Le-Khac, and Van Loi Cao. Clustering-based deep autoencoders for network anomaly detection. In *Future Data and Security Engineering: 7th International Conference, FDSE 2020, Quy Nhon, Vietnam, November 25–27, 2020, Proceedings 7*, pages 290–303. Springer, 2020.

[17] Van Quan Nguyen, Viet Hung Nguyen, Nhien-An Le Khac, Nathan Shone, et al. A robust pca feature selection to assist deep clustering autoencoder-based network anomaly detection. In *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 335–341. IEEE, 2021.

[18] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[19] Pramuditha Perera, Poojan Oza, and Vishal M Patel. One-class classification: A survey. *arXiv preprint arXiv:2101.03064*, 2021.

[20] Shehroz S Khan and Michael G Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

[21] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*, pages 1–6. Ieee, 2009.

[22] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE, 2015.

[23] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116, 2018.