Information Processing and Management 59 (2022) 102844

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection

Peng An [a,1], Zhiyuan Wang [b,1], Chunjiong Zhang [c,*]

[a] *College of Electronics and Information Engineering, Ningbo University of Technology, Ningbo 315211, Zhejiang, PR China*
[b] *International College of Football, Tongji University, Shanghai 200092, PR China*
[c] *Department of Computer Science, Tongji University, Shanghai 201804, PR China*

ARTICLE INFO

Keywords:
Multidomain data
Cyberattacks
Deep autoencoder
GMM

ABSTRACT

Previous studies have adopted unsupervised machine learning with dimension reduction functions for cyberattack detection, which are limited to performing robust anomaly detection with high-dimensional and sparse data. Most of them usually assume homogeneous parameters with a specific Gaussian distribution for each domain, ignoring the robust testing of data skewness. This paper proposes to use unsupervised ensemble autoencoders connected to the Gaussian mixture model (GMM) to adapt to multiple domains regardless of the skewness of each domain. In the hidden space of the ensemble autoencoder, the attention-based latent representation and reconstructed features of the minimum error are utilized. The expectation maximization (EM) algorithm is used to estimate the sample density in the GMM. When the estimated sample density exceeds the learning threshold obtained in the training phase, the sample is identified as an outlier related to an attack anomaly. Finally, the ensemble autoencoder and the GMM are jointly optimized, which transforms the optimization of objective function into a Lagrangian dual problem. Experiments conducted on three public data sets validate that the performance of the proposed model is significantly competitive with the selected anomaly detection baselines.

1. Introduction

Cyberattacks have become a popular topic in recent years as an increasing number of terminal devices are connected to the Internet. They are highly threatening to the information security of individuals because of the large number of computer viruses spreading in the network, and various types of cyberattacks have emerged in endless streams. Examples of such attacks include disclosure, modification and deletion of specific data, attempts at unauthorized access, manipulation of information, and other malicious behaviors that make a system unreliable and unstable. This damage produces incalculable and disastrous consequences for users of the system (Gong et al., 2019). It is becoming increasingly important to secure users' data through the popularization of networking applications.

However, it has always been difficult to develop an effective and compatible method for complex high-dimensional data (Rezvy, Petridis, Lasebae, & Zebin, 2018). With the rapid development of artificial intelligence, methods based on deep learning are envisioned as competent approaches for cyberattack detection. Some scholars have therefore adopted an unsupervised machine learning method with dimension reduction functions, known as the autoencoder framework (Andresini, Appice, Di Mauro, Loglisci, & Malerba, 2019; Xu, Qian, & Hu, 2019), but this kind of method is limited to performing robust anomaly detection for network

[*] Corresponding author.
E-mail address: chunjiongzhang@tongji.edu.cn (C. Zhang).
[1] These authors contributed equally to this work and should be considered co-first authors.

https://doi.org/10.1016/j.ipm.2021.102844
Received 14 September 2021; Received in revised form 29 November 2021; Accepted 10 December 2021
Available online 10 January 2022

intrusion data with high dimensionality and sparsity without supervision. This is because when the dimension of the input data increases, it is more difficult to estimate the density of multidomain data in the original feature space. Additionally, an input sample may easily be regarded as an abnormal event that has a low probability of being observed (Shone, Ngoc, Phai, & Shi, 2018). To address the performance degradation caused by high dimensions, many works have concentrated on dimension reduction, which searches for a lower-dimensional representation by reducing the number of variables in the data (Kim, Kwon, Chang, & Paik, 2020; Zong et al., 2018). The major drawback is the weak assumption of homogeneous parameters of a specific Gaussian distribution with respect to each domain of data, which significantly deteriorates the original data properties.

Depending on the type of cyberattack, it is usually possible to classify network traffic into different domains. For example, the classic KDDcup'99 network intrusion data set has been found to have four domains: denial of service (DoS), remote to local (R2L), user to root (U2R) and Probe (Chen et al., 2018). The size of each domain is different, and they do not follow a homogeneous distribution. Therefore, treating attack samples indiscriminately will affect the results of anomaly detection and may even mislead the entire machine learning process. Many customized neural network models that are used in a single domain can achieve good anomaly detection performance, but the effect on multidomain data is poor because of the limited ability to acquire the complex information of high-dimensional distributions. One major solution in recent decades to yield enriched features is to decouple the learning process following Dromard, Roudière, and Owezarski (2016) and Zhou and Paffenroth (2017), but this method is still unable to obtain sufficient features of anomalous attack samples and thereby fails to obtain accurate detection results. Cyberattack data are usually collected from a large number of heterogeneous network devices that exhibit uncertainties and severe skewness, which is fatal for conventional machine learning. The prospect is that the multidomain machine learning methods developed in recent years can better address abnormal network intrusion problems of multidomain data by making full use of the features of multidomain data to achieve optimal performance while simultaneously saving training time and model resources. The core of multidomain machine learning is to obtain a model with the smallest average risk in multiple domains and to treat different cyberattacks as independent domains (Hu, Li, Liu, & Li, 2020; Qian et al., 2019).

Therefore, the key problem to be solved in this paper is to use hidden information in multidomain data to improve the performance of the anomaly detection model using multidomain machine learning. For this purpose, a multichannel ensemble autoencoder combined with an attention mechanism is devised. To maintain the different distributions of multidomain data in a low-dimensional hidden space, the minimum reconstructed error of the multichannel autoencoder is fed back to GMM, and the expectation maximization (EM) algorithm is used to estimate the mixing probability of the components. When the estimated sample density exceeds the learning threshold obtained in the training phase, the sample is identified as an attack anomaly. Because the robust optimization algorithm enables multidomain machine learning models to obtain common feature representations and the models can adapt to different domains, this paper further develops an optimized combination of ensemble autoencoders and GMM so that the model used in each domain is optimal.

Based on the above idea, the main contributions of this study lie in the following three areas:

- An ensemble framework of multichannel network anomaly detection called the ensemble multichannel mixed model (EM^3) is proposed that combines deep autoencoders and the GMM model.
- A robust optimization version of EM^3 for multiple domains, namely, EM^{3+}, is proposed, which transforms the optimization problem of the objective function into a Lagrangian dual problem.
- A series of experiments are conducted on two classic data sets and a newly published data set from 2020, which to the best of our knowledge is the first work that performs algorithms on both differentiated data domains and data distributions.

The rest of this paper is organized as follows. Related works are presented in Section 2. Section 3 provides detailed descriptions of the proposed framework, which includes parameter learning and the realization of a robust optimization model. The results of a series of experiments are reported in Section 4, followed by the conclusions drawn in Section 5.

2. Related work

Internet attackers constantly improve their intrusion methods, resulting in the existence of various types of abnormal data in network traffic. It is therefore becoming increasingly urgent to devise effective approaches that are able to contend with increasingly complex cyberattacks (Qian et al., 2019; Zong et al., 2018). The existing library has documented many works with respect to anomaly detection, and the most popular method is the integration of autoencoders in a specific computing framework.

Autoencoders are a type of artificial neural network and are used in both feature engineering and anomaly detection. There are two major types of research on the applications of autoencoders in network intrusion anomaly detection. One type of research mainly uses the autoencoder and a variant of it that incorporates a deep network to recognize distributional heterogeneity by constantly learning and reconstructing normal and abnormal samples. For example, some studies have used automatic variational encoders for network intrusion anomaly detection and have proposed a unified paradigm conversion system based upon unsupervised Gaussian mixture variational autoencoders (Liao, Guo, Chen, & Li, 2018). This method first generates sample distributions through training and extracts reconstructed features and then uses a deep simplified network to estimate the mixing probability of the components through the potential distribution. The authors of Gong et al. (2019) used an autoencoder extended with a storage module. Given an input, the code is first obtained from the encoder and is then used as a query sentence to retrieve the most relevant storage items to reconstruct the features. Rezvy et al. (2018) applied a dense neural network algorithm based on a deep autoencoder in cyberattack detection and evaluated the algorithm with the benchmark data set NSL-KDD. The results show that the deep autoencoder can effectively distinguish the differences in low-dimensional spaces for both normal and abnormal samples in low dimensions so that this algorithm can achieve excellent detection performance. The other type of research focuses on a combination of an autoencoder and functions based on distance metrics or probability distributions. Kim et al. (2020) established an anomaly detection algorithm based on an admissibility attribute, which includes an objective function of integral probability measurement and a type I autoencoder called Lipschitz. The proposed Wasserstein distance metric achieves Lipschitz continuity by minimizing the approximate Wasserstein distance and penalty functions. Other relevant work includes the stacked asymmetric deep autoencoder (Majumdar & Tripathi, 2017; Wang, Xu, Huang, Wang, & Lai, 2018), used for unsupervised feature learning, and the deep autoencoder-based GMM (Zong et al., 2018), which uses a deep autoencoder to generate a low-dimensional representation and inputs the reconstructed error of the input data points into the GMM, which can not only maintain the original deep autoencoder but also discover high-quality and nonlinear features (Zhou & Paffenroth, 2017).

In summary, recent studies based on deep autoencoders indicate that a scheme that combines dimensionality reduction and density estimation is effective in network attack anomaly detection. However, the obvious disadvantage is that the joint optimization of dimensionality reduction and density estimation is usually computationally difficult under a generic framework (Andresini et al., 2019; Liu et al., 2020), and the performance of the model is mainly affected by two limitations. The first is that they cannot handle the heterogeneity of data from different domains and cannot capture the subtle differences in different data distributions. The second is that they cannot obtain the variable information of samples in a low-dimensional space but simply estimate the sample density, which restricts the complex training process of multidomain data (Injadat, Moubayed, Nassif and Shami, 2020).

Although deep autoencoders achieve good performance in single-domain tasks, in many practical situations, a unified model for multidomain data is required for shared tasks. Therefore, multidomain machine learning technology is used to obtain cross-domain adaptability and inherit the advantages of multitask learning at the same time (Hu et al., 2020). Early work (Peng & Dredze, 2016) with regard to multidomain machine learning focused on deep neural network models, which shared training weights in the early layers and used special weights in the later layers. For example, solutions were proposed in Fourure et al. (2017) that shared all core parameters except those in the batch and instance normalization layers, where different domains were modeled separately in individual neural networks. Based on this, Vaca and Niyaz (2018), as an extension, proposed a new parameterization method for the standard residual network architecture, which aimed to increase the number of parameters in a limited way in order to improve the level of parameter sharing between domains. However, these early works did not consider the risk of increased computational complexity caused by the large number of model parameters, which makes such models weak with respect to convergence (Injadat, Moubayed and Shami, 2020). In more recent works inspired by the success of transfer learning, Berriel et al. (2019) and Ren and Lee (2018) proposed a combination of serial and parallel network adapters for multidomain data. They obtained domain-related models with adjustable budgets in terms of the number of parameters and computational complexity. To adjust the computational complexity of the network, they adopted a pretrained architecture to derive a specialized deep model for each domain and then incorporated a budget-aware adapter to select the most relevant feature channel to better process the data from the new domain. Among recent works, there are also some studies that improve domain adaptability through adversarial learning. They decompose a deep network into feature extractor and classifier components and then train each component by tuning its partner (Xu, Chen, Zuo, Yan, & Lin, 2018). They either use a deep cocktail network (DCTN) to deploy multidirectional adversarial learning to minimize the difference between the targets and multidomain data (Ganin et al., 2016) or use domain-specific representation learning (Schoenauer-Sebag et al., 2019), where the data come from similar but heterogeneous distributions.

Although multidomain machine learning solves the problem of domain adaptability to some extent through assumptions about specific data distributions, the performance of most models is poor in practice because they ignore the nature of data skewness and lack a process that effectively determines the robustness of multidomain data. It is claimed that ignoring the inconsistency of the data size and the heterogeneity of the data distribution generated by network traffic will have a serious impact on the detection results and may even mislead the entire machine learning process. In contrast to existing works, the current study focuses on extracting the hidden relationship between data that are not independent and identically distributed (non-IID) and unbalanced data and performing robust optimization on data from different domains so that the proposed scheme can further improve domain adaptability without relying on assumptions about the data distribution.

3. Proposed framework

3.1. System overview

The purpose of this research is to perform effective anomaly detection on network intrusion samples from different domains. The key problem to be solved is the feature learning and reconstruction of high-dimensional multidomain data. However, most of the current studies using single-channel networks cannot capture the hidden spatial information of data in different domains, and the features of the reconstructed errors do not include the differential representation of heterogeneous distributions. Therefore, EM^3 is proposed, as shown in Fig. 1. Overall, EM^3 consists of two parts: the ensemble network and estimation network. The ensemble network uses a deep autoencoder to reduce the dimensionality of the input multidomain samples, and the estimation network feeds back the latent distribution learned from the ensemble autoencoder and the reconstructed error features to the GMM, using the potential distribution and the EM algorithm to estimate the sample density. To avoid the decoupling and suboptimal performance of feature learning for multidomain data, the system performs joint optimization on the ensemble autoencoder and GMM. The use of the GMM protects the ensemble of autoencoders from a local suboptimum, and the use of an autoencoder provides a prior

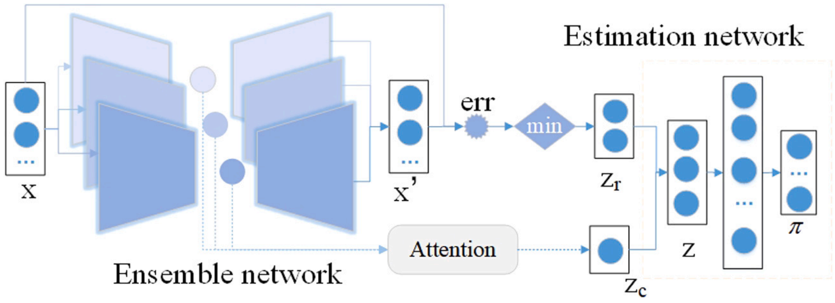


Fig. 1. Algorithm framework.

Table 1
Notation.

Notation	Meaning
$\mathbf{x}$	Sample set
$\mathbf{x}'$	Reconstructed vector of the sample set
n	Dimensions of the samples
$D(\cdot)$	Decoder
$E(\cdot)$	Encoder
ψ	Combination of the activation functions of the hidden layer
$\mathbf{w}$	Weight matrix with respect to the parameters
$\mathbf{b}$	Bias vector
z_c	Hidden space of the autoencoder
$h(\cdot)$	Loss function of the autoencoder
f_a	Fully connected neural network
d_t	State of the encoder at the t'th iteration
$\mathbf{a}$	Attentive weight matrix
$\mathbf{z}$	Hidden state
$\hat{\Gamma}_x$	Mixing probability of the GMM
b_λ	Estimation network
$\xi(\cdot)$	Sample energy
p	Dual management factor

distribution of samples for the GMM. When the estimated sample density exceeds the learning threshold obtained in the training phase, the samples are identified as outliers. The major notation used in this study is listed in Table 1.

In what follows, each part of the system is described in three steps. First, an ensemble of deep autoencoders is utilized to reduce the dimensionality of multidomain data. The deep autoencoder is a popular deep learning model that is constructed with several autoencoders. Each autoencoder minimizes the reconstructed error between the input data and output data. Given an unlabeled n-dimensional sample set $\mathbf{x} = [x^1, x^2, \ldots, x^n] \in \Re^{1\times n}$, the autoencoder uses an activation function such as sigmoid to transform the input data set into a hidden representation and then maps it back into a reconstructed vector $\mathbf{x}'$. That is, the autoencoder attempts to learn the functions $D(E(\mathbf{x})) = \mathbf{x}' \approx \mathbf{x}$, where $D(\cdot)$ and $E(\cdot)$ are the decoder and encoder function, respectively. To measure the reconstructed error, the mean square error is used in this paper. And we utilized cross entropy as the cost function. Due to the difficulty of parameter selection, an individual deep autoencoder will probably show a low generalization ability. An ensemble of multiple deep autoencoders is developed to enhance the generalization performance by combining a series of activation functions (Zhang, Li, Gao, Chen, & Li, 2020). Let $\mathbf{h} = \psi(\ \mathbf{wx} + \mathbf{b})$ denote the hidden representation, where ψ is the combination of the activation functions of the hidden layer, $\mathbf{w}$ is the weight matrix with respect to the parameters and $\mathbf{b}$ is the bias vector. The proposed system inputs high-dimensional multidomain data into each channel and obtains the hidden space distribution of the reduced dimensionality and reconstructed error features for each channel:

$$z_c^l = E_l(\mathbf{x}), \tag{1}$$

where z_c is the hidden space of the autoencoder for channel l.

Assume there are L channels in total. The minimum reconstruction error z_r is represented as:

$$z_r = \min\left\{h_1\left(\mathbf{x}, \mathbf{x}'\right), h_2\left(\mathbf{x}, \mathbf{x}'\right), \ldots, h_L\left(\mathbf{x}, \mathbf{x}'\right)\right\}, \tag{2}$$

where $h(\cdot)$ is the set of reconstructed error function of the autoencoder, which contains one of the relative absolute square error, Euclidean distance and cosine similarity (Pratama & Kang, 2020).

Some existing studies use a single autoencoder as input, which largely ignores the fine discrimination of the reconstruction error features of multidomain data, resulting in suboptimal performance. Reconstructed error features in multiple autoencoders can provide richer information for the estimation network, thereby improving the model's reasoning ability.

Second, to balance the complexity and expressive ability of the model, we use the attention model to optimize the hidden space of each channel so that important information is selected. An attention model based upon the encoder–decoder framework is widely adopted in natural language processing, image classification and speech recognition tasks because of its ability to optimize the memory information of hidden space in multiple channels. The encoder projects $\mathbf{x}$ onto a hidden vector representation $\mathbf{h}$. The decoder uses a recurrent network to maintain an internal hidden state $\mathbf{z}$ and uses the attention model to weight the feature vectors based on a score function:

$$score_t = f_a(\mathbf{z}_{t-1}, \mathbf{h}_t), \tag{3}$$

where t is the number of iterations and f_a is a fully connected neural network with a single hidden layer (Zhang et al., 2020). Clearly, this equation takes both the previous decoder hidden state and the feature vectors at the t'th iteration as input.

The two most commonly used attention functions are additive attention and dot-product attention. Here, we choose additive attention because it computes a compatibility function using a feed-forward network with a single hidden layer. The attention weight is computed using the following function:

$$\mathbf{a}_t^l = \mathrm{softmax}\left(d_t^l \mathbf{w}_t^l z_c^l\right), \tag{4}$$

where d_t is the state of the encoder at the t'th iteration, $l \in L$ is the channel, and $\mathbf{a}$ is the attentive weight after using the softmax function. With the obtained attentive weights, the weighted sum of all the hidden vectors z_c is represented by:

$$z_c = \sum_{l=1}^{L} \mathbf{a}_t^l z_c^l. \tag{5}$$

Finally, the proposed system combines z_c with the minimum error feature of the ensemble network into a new hidden space vector $\mathbf{z} = \left[z_c, z_r\right]$.

3.2. Parameter learning

EM^3 jointly optimizes the parameters of the ensemble autoencoder and GMM in an end-to-end manner and uses a separate estimation network to promote parameter learning. Joint optimization can balance the reconstruction of multiple autoencoders in the ensemble network and contribute to density estimation and regularization, thereby avoiding local optimization and reducing the reconstructed errors. To make the probability model effective, we optimize the model with the addition of objective functions and reparameterization techniques.

The GMM converts high-dimensional data into a mixture of single-mode Gaussian distributions, which can solve mathematical problems in high-dimensional spaces. When training samples, the EM algorithm is used to solve the parameters (Zong et al., 2018). Given the latent space distribution z and the number of mixed components r, the GMM uses the softmax function to generate an r-dimensional vector for each sample, represented as $\hat{\Gamma}_x = \left[\hat{\gamma}_{x1}, \hat{\gamma}_{x2}, \ldots, \hat{\gamma}_{xr}\right]$, where $\hat{\gamma}_{xi}$, $1 \le i \le r$, denotes the probability that the i'th component of the GMM produces $\mathbf{x}$. In the maximization stage of the EM algorithm, the parameters of the GMM are estimated using N samples and the corresponding mixed probability $\hat{\Gamma}_x$ as in Eq. (6):

$$\begin{aligned} \widehat{\phi}_r &= \sum_{i=1}^{N} \frac{\widehat{\gamma}_{ir}}{N}, \\ \widehat{\mu}_r &= \frac{\sum_{i=1}^{N} \frac{\widehat{\gamma}_{ik}\mathbf{z}_i}{N}}{\sum_{i=1}^{N} \widehat{\gamma}_{ir}}, \\ \widehat{\sigma}_k &= \frac{\sum_{i=1}^{N} \widehat{\gamma}_{ir}(\mathbf{z}_i - \widehat{\mu}_r)^2}{\sum_{i=1}^{N} \widehat{\gamma}_{ir}}. \end{aligned} \tag{6}$$

where $\widehat{\mu}_r$ is the average of GMM component probabilities and $\widehat{\sigma}_k$ is the corresponding variance.

In the prediction stage, we utilize the sample energy as the anomaly score. The sample energy characterizes the degree to which a sample deviates the trained distribution. When the anomaly score exceeds a user-defined threshold, it is recognized as an abnormal value. The energy of the samples $\xi(\mathbf{z})$ can be formulated as follows:

$$\xi(\mathbf{z}) = -\log\left(\sum_{k=1}^{r} \widehat{\phi}_k \frac{\exp(-\frac{1}{2}(\mathbf{z} - \widehat{\mu}_r)^{\mathrm{T}})\widehat{\sigma}_r^{-1}(\mathbf{z} - \widehat{\mu}_r)}{\sqrt{|2\pi\widehat{\sigma}_r|}}\right), \tag{7}$$

where $|\cdot|$ is the determinant of the matrix.

3.3. Robust optimization for multiple domains

To make the model perform well in various domains, it is necessary to improve the performance of the domain with a small number of samples to prevent the samples in this domain from reducing the overall model performance. In this section, a robust optimization method, namely, EM^{3+}, based on the EM^3 framework is proposed. Multidomain learning aims to obtain common behaviors across related problems, so the core idea is to learn domain-specific parameters guided by shared parameters. Multidomain learning algorithms are simple to implement and scale to very large data sets. They process multiple streams of data from many different sources, transferring knowledge between domains through a shared model (Luong, Pham, & Manning, 2015). Given a data set $\mathbf{x} \in \left\{\mathbf{x}^1, \mathbf{x}^2, \ldots \mathbf{x}^k\right\}$ of k domains of cyberattacks, the data set of the m'th domain is $S_m = \left\{x_m^i, y_m^i\right\}$, where x_m^i is the i'th training sample and y_m^i is the corresponding label of the sample.

The objective function of multidomain learning can be formulated as the following empirical risk:

$$\min_{\mathbf{w}}\{\sum_{m=1}^{k} \max_{p} \mathbf{pf}(\mathbf{w}_m) + \beta \sum_{m=1}^{k} \frac{1}{k} \left\|\widehat{\mathbf{w}} - \mathbf{w}_m\right\|_F\}, \tag{8}$$

where $\left\{\mathbf{p} \in {}^k \left| \sum_{m=1}^{k} p_m = 1; \forall k, p_m \geq 0 \right.\right\}$ is the dual management factor and $\beta > 0$ is the penalty factor. $\mathbf{p}$ can also be treated as an adversarial distribution of different domains, and its default value is calculated as $p_m = \frac{1}{k}$. Then, we have Eq. (9):

$$\mathbf{f}\left(\mathbf{w}\right) = \left[f_1\left(\mathbf{w}\right), f_2\left(\mathbf{w}\right), \ldots, f_m\left(\mathbf{w}\right)\right]^T, \tag{9}$$

where $f_i\left(\mathbf{w}\right) = h(x_m^i, y_m^i; \mathbf{w})$.

The second term of Eq. (8) narrows the difference in the parameters of the multidomain model through the average value **w**. Obviously, it is a minimax problem that satisfies the Karush–Kuhn–Tucker (KTT) condition according to Yu (2020). Therefore, the optimization problem of the objective function can be considered equivalent to the Lagrangian dual problem. It is known that a key to minimax optimization is that the formula is very sensitive to outliers. If there is a domain with significantly lower performance than other domains, the objective function of the domain with poor performance will dominate the entire multidomain learning process, which will seriously affect the performance of the training model in different domains. For this reason, this paper normalizes the loss of each domain to prevent the **p** of a poor domain from being larger and to ensure that each domain has a smooth training process. To this end, we use the Lagrangian relaxation mode, as given in Eq. (10):

$$\max \mathbf{p}_m^T \mathbf{f}_m\left(\mathbf{w}\right) + \eta(p_1 + p_2 + \cdots + p_k - 1), \tag{10}$$

where $\eta \geq 0$ and $p_k \geq 0$.

The dual problem is formulated as Eq. (11):

$$\max_{p} \min_{\mathbf{w}} \mathbf{p}_m^T \mathbf{f}_m\left(\mathbf{w}\right) + \eta(p_1 + p_2 + \cdots + p_k - 1) \tag{11}$$

To solve the minimax optimization problem presented in Eq. (11), we use the gradient descent method to learn the model and the gradient ascent method to update the adversarial distribution. The p value can be obtained using the approximate gradient descent algorithm (Mariño & Míguez, 2006) in Eq. (12):

$$p_k = prox_{th}(p_{k-1} - t_{k-1} \nabla f(\mathbf{w})) \tag{12}$$

where t is the step size of each gradient.

Let $\chi(p) = \eta\left(p_1 + p_2 + \cdots + p_k - \mathbf{1}\right)$; according to the definition of $prox_{th}$ (Li, Zou, & Zhong, 2020), Eq. (12) is expanded to obtain Eq. (13):

$$\begin{gathered} p^+ = \underset{p}{\operatorname{argmin}} \left(\chi\left(p\right) + \frac{1}{2t} p + t \nabla f\left(\mathbf{w}\right)_2^2\right) \\ = \arg\min_{p}\left(\chi\left(p\right) + f\left(\mathbf{w}\right) + \nabla f\left(\mathbf{w}\right)^T p + \frac{1}{2t} p_2^2\right) \end{gathered} \tag{13}$$

The expression in parentheses is a second-order expansion of f near p, and p^+ is the minimum value of the approximate function. Furthermore, we have Eqs. (14) and (15):

$$0 \in t\partial\chi\left(p^+\right) + \left(p^+ - p + t\nabla f\left(\mathbf{w}\right)\right) \tag{14}$$

$$G_t(\mathbf{w}) := \frac{p - p^+}{t} \in \partial\chi\left(p^+\right) + \nabla f\left(\mathbf{w}\right), \tag{15}$$

where $G_t(\mathbf{w}) = \partial\chi\left(p^+\right) + \nabla g(\mathbf{w})$ is approximated as the subgradient of the function f and p^+ can be simplified as $p^+ = p - tG_t(\mathbf{w})$ so that the dual management factor p is obtained for each domain.

To find the shared model parameters **w**, the robust optimization algorithm is used as follows: The multidomain machine learning model initializes the model parameters of each domain. It randomly samples p*n samples for iteration to obtain the loss function of each domain and then uses the gradient descent algorithm to update the p value. When the update is complete, the average value of the model parameters $\widehat{\mathbf{w}}$ is reassigned to the model parameters in the corresponding domain until the end of training.

4. Experiments

The purpose of this study is to detect network intrusion in different domains using the proposed EM^3 framework. In this section, a series of experiments are conducted to test whether the proposed EM^3 is competitive with respect to specific measures compared to other algorithms. The data sets used in the experiment are detailed in Section 4.1, followed by a Section 4.2 that describes the baselines based on frequently used algorithms in the network intrusion anomaly detection area. Section 4.3 presents the experimental settings used in the experiment, and the results are reported in Section 4.4. All of the experiments are performed with PyTorch and the Python SK-learning library on a RHEL7.5 server with an Intel Xeon E5-2687w 3.1 GHz CPU and NVIDIA Quadro P4000 GPU.

4.1. Data sets

Three public data sets are used in this experiment. The first is KDDcup'99 (Zong et al., 2018), which comprises network connection and system audit data collected over 9 weeks by MIT using TCPdump. This data set simulates data samples of different user types, network traffic and attack methods. Each sample is described with 41 features. The data set includes 14,258 normal samples and 713 abnormal samples in total. The multiple domains can be divided into DoS, R2L, U2R and Probe domains. For the experiment, we randomly sample 20,000 data points for each of the DoS, R2L, and Probe types, where anomalies account for 10%. Since the size of the U2R data is small, we set 1000 U2R data points, and the abnormal proportion is 5%. For the nominal features in this data set, we coded them according to numerical categories. Taking protocol as an example, the Transmission Control Protocol (TCP) was coded as 0, the User Datagram Protocol (UDP) was coded as 1 and the Internet Control Message Protocol (ICMP) was coded as 2.

The second data set is CICMalDroid 2020 (Mahdavifar, Kadir, Fatemi, Alhadidi, & Ghorbani, 2020). This data set is for Android malware that contains the most complete static and dynamic network connection features among publicly available data sets, including the latest and most complex Android samples as of 2018. In this experiment, the top 9 features were selected for experimentation through principal component analysis (PCA). The multiple domains of this data set are Adware, Banking Malware, Short Message Service (SMS) Malware and Mobile Riskware. Each type contains 20,000 entities, of which abnormal data account for 10%.

The last data set is AWID (Vaca & Niyaz, 2018). It is collected from a small network environment for 802.11 networks, which typically involves 11 client computers. The data are a wireless local area network (WLAN) data stream captured in a packet-based format, and the multiple domains of this data set are Flooding, Impersonation, and Injection. Each domain contains 20,000 entities, where abnormal data account for 10%.

Since the KDDcup'99 and CICMalDroid 2020 data sets are structured numerical data, there are no garbled codes or symbols in the data, so this experiment normalizes them and encodes them with one hot encoding. However, the missing, garbled, and duplicated data in the AWID data set cannot be directly input to the model. We perform data cleaning, data transformation, and feature selection on the original data set. During data cleaning, this experiment uses replacement and zero padding to convert dirty data into quality data. Constants are used to fill the missing data in the set, and the existing special symbols and garbled codes are cleared or replaced. Since the AWID data set contains 154 features with both numerical and character data, we convert nonnumerical features such as wlan.ra and wlan.da into numerical features.

4.2. Baselines

To compare the performance difference of the proposed model with and without robust optimization, the experiment uses the following baselines:

- The one-class support vector machine (OCSVM) (Zhou & Paffenroth, 2017) is a commonly used kernel-based outlier detection method. When making predictions, the model searches for a hyperplane; the samples marked on this hyperplane are considered positive samples, and vice versa. All tasks in this experiment use radial basis function kernels.
- The deep autoencoding Gaussian mixture model (DAGMM) (Zong et al., 2018) combines a GMM with a deep autoencoder. The model uses the deep autoencoder to generate a low-dimensional representation and reconstructed error for each input data point and then inputs it to the GMM. The parameters of the deep autoencoder and the hybrid model are optimized simultaneously in an end-to-end manner, and a separate estimation network is used to promote the parameter learning of the hybrid model. The estimated density of the sample is used as an anomaly detection criterion.
- Adversarially learned anomaly detection (ALAD) (Zenati, Romain, Foo, Lecouat, & Chandrasekhar, 2018) combines an autoencoder with the standard generative adversarial network (GAN) algorithm and uses representative learning to measure the similarity in the data space. By combining the autoencoder with the GAN, the feature representation can be used as the basis for the reconstruction target in the GAN discriminator.

4.3. Settings

The learning rate of the models used on the KDDcup'99 and AWID data sets is set to 0.001, and the learning rate of the models used on the CICMalDroid 2020 data set is set to 0.0007, while the batch sizes are 256 and 128, respectively. The hyperparameters λ_1 and λ_2 of the regularized objective function are set to 0.1 and 0.0001 based on cross-validation experiments. To test the multichannel neural network in this study, the number of channels is set to 3 for simplification. The channels are the deep simple networks of [100, 64, 32, 16, 1, 16, 32, 64, 100], [112, 82, 56, 28, 1, 28, 56, 82, 112] and [60, 30, 10, 10, 30, 60], where the digits represent the number of neurons in the network.

The experiment consists of two parts. The first part of the experiment was conducted on data from the undifferentiated domain. In this setting, we did not divide the data set into domains following traditional solutions to show whether the performance of EM^3 proposed in this paper is competitive in a single domain. In the second part of the experiment, we divided the data set into domains to show whether the performance of the EM^3 framework proposed in this paper is competitive in multiple domains. We used the F1 value as the evaluation standard of the model performance based on the calculation of the accuracy and recall rate because it is a commonly used evaluation index for abnormality detection problems. The accuracy rate is the most intuitive performance measure; it is the ratio of the correctly predicted observations to the total observations. The recall rate is the ratio of correctly predicted positive observations to all observations in the actual class, and the F1 value is the weighted average of the accuracy rate and the recall rate. When the F1 value is higher, the model is better.

Table 2
F1 values of models.

	KDDcup'99	AWID	CICMalDroid 2020
OCSVM	0.7954	0.4354	0.6211
DAGMM	0.9369	0.4899	0.7516
ALAD	0.9377	0.5102	0.7881
EM^3	**0.9523**	**0.5612**	**0.8323**

Table 3
Results of the algorithms on multidomain data sets.

Data sets	KDDcup'99				CICMalDroid 2020				AWID		
Domains	DoS	Probe	R2L	U2R	Adware	Banking	SMS	Mobile	Flooding	Impersonation	Injection
OCSVM	0.7414	0.6912	0.5358	0.4887	0.5878	0.6157	0.5542	0.6712	0.3952	0.3276	0.4012
DAGMM	0.8282	0.7845	0.6641	0.7285	0.6745	0.6871	0.5927	0.6925	0.4118	0.3956	0.4571
ALAD	0.8325	0.7756	0.6714	0.6148	0.6642	0.6581	0.6155	0.7011	0.4214	0.3812	0.4482
EM^3	0.9021	0.8327	0.7895	0.7725	0.7144	0.6853	0.6657	0.7339	0.4857	0.4337	0.5017
EM^{3+}	**0.9211**	**0.8531**	**0.7975**	**0.8012**	**0.7158**	**0.7015**	**0.6987**	**0.7619**	**0.5042**	**0.4681**	**0.5359**

4.4. Results

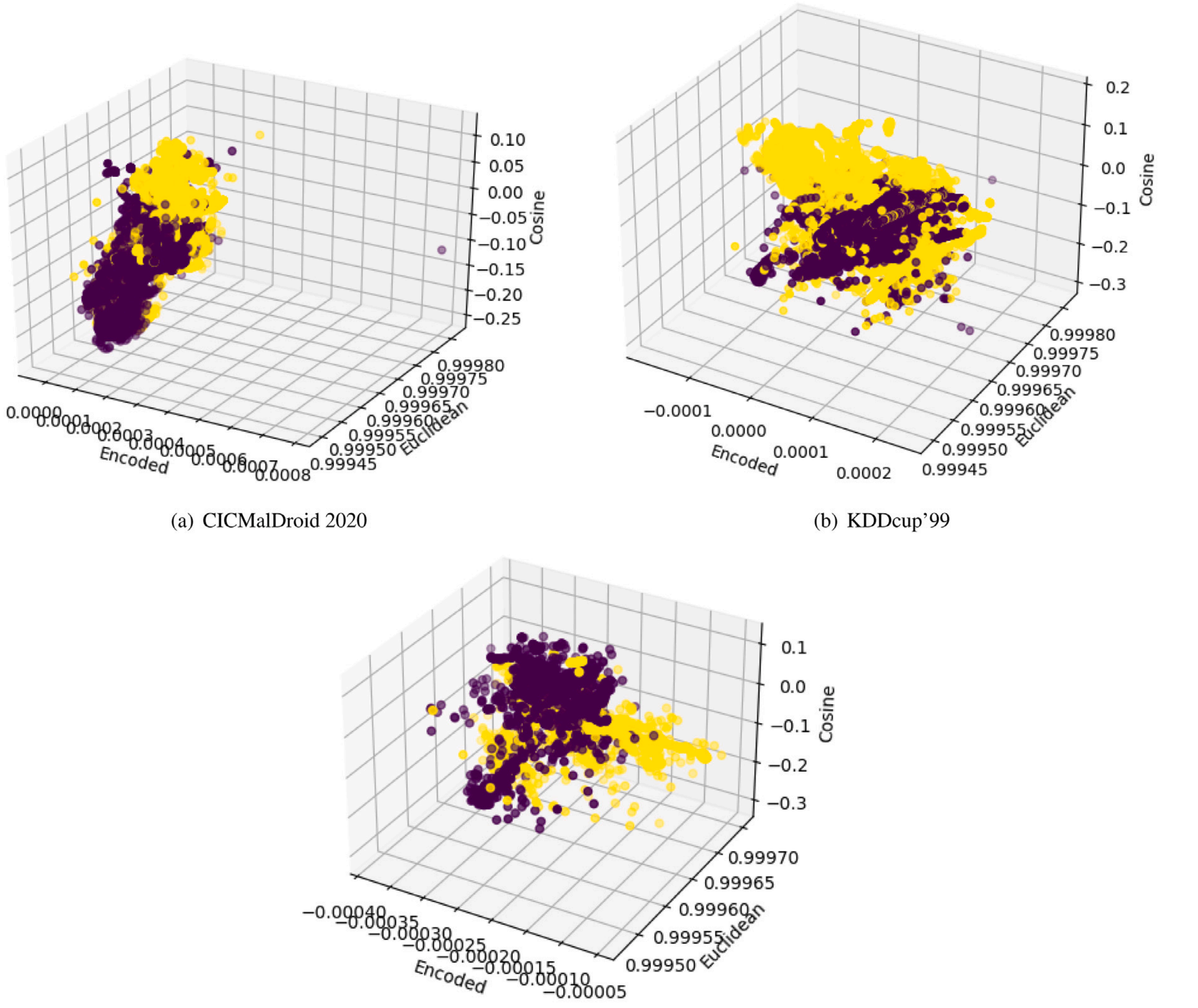
4.4.1. On an undifferentiated domain

Table 2 shows the result matrix for running the model on different data sets when no domain distinction is made on the data sets, as measured by F1. EM^3 has the highest F1 value on all data sets, followed by ALAD. This is because ALAD does not use the information represented by dimensionality reduction in the autoencoder hidden space. Although DAGMM uses a GMM and a deep autoencoder, its F1 value is reduced significantly, by approximately 1.6%, compared with EM^3 because EM^3 utilizes an attention model to select the optimal information of the multichannel autoencoder hidden space, and the ensemble autoencoder is more effective than the single autoencoder for data dimensionality reduction. The F1 value of the OCSVM model is the smallest. Compared with EM^3, the performance is reduced by approximately 16%, and the performance is reduced by approximately 15.2% compared with ALAD. In addition, the model has a higher F1 value on the KDDcup'99 data set and a lower F1 value on the AWID data set, which is attributed to the fact that there is less available information in the AWID data set. To the best of our knowledge, previous work has mainly applied this data set to classification tasks, and for the first time, we are using it for unsupervised anomaly detection testing.

In summary, the EM^3 proposed by this research obtained the best F1 value on the selected data set. The ensemble deep autoencoder produces a low-dimensional representation of data and the reconstructed errors, which effectively utilizes the hidden information in the network intrusion data. Furthermore, the attention mechanism is used to optimize the latent space distribution, which is an advantage that support vector machines do not have. In addition, EM^3 uses the reconstructed minimum error based on the Euclidean distance and cosine similarity measure to expand the difference distribution between normal samples and abnormal samples so that GMM can better learn the sample distribution.

The normal and abnormal samples in the hidden space of EM^3 are visualized in Fig. 2, which shows a low-dimensional representation of the hidden space of different channels in an ensemble network. The normal and abnormal samples can be better separated in low dimensions under the EM^3 framework because their samples overlap very little. EM^3 dynamically selects the autoencoder with the smallest reconstructed error for different data sets and jointly optimizes the autoencoder and GMM parameters in an end-to-end manner, which helps the autoencoder eliminate local optimization and obtain better compression results. In addition, the use of the GMM provides the model with more meaningful sample distributions for different domains. Fig. 3 presents the convergence of the models on different data sets. Clearly, the loss of each model decreases as the epoch increases. EM^3 decreases most rapidly compared to the other algorithms, which means that EM^3 has the best convergence ability. Specifically, the loss of EM^3 becomes constant after the algorithm runs approximately three epochs for all the selected data sets. However, the loss of the other algorithms depends on the data sets. SVM, DAGMM and ALAD have nearly the same loss changes for the KDD data set, and they reach stability when the algorithms run for more than eight epochs. For the Android data set, DAGMM is superior to SVM and ALAD. The loss of DAGMM reaches stability when the epoch number reaches two, while SVM and ALAD become stable when the epoch number reaches eleven and thirteen, respectively. For the AWID data set, the curve changes in different ways. SVM and ALAD converge faster than DAGMM, which reaches stability after 21 epochs. Moreover, the enlarged local charts indicate that EM^3 has the smallest loss value compared to the other algorithms, although all of them converge after several epochs.

When running the algorithm on the CICMalDroid 2020 data set, the model is stable after 3 epochs. The fast model learning speed mainly depends on two properties of the proposed method. First, we use 9 features of the data and a three-channel deep autoencoder, which can quickly learn the features of the data. Second, the attention mechanism chooses the hidden space so that the GMM can better learn the sample distribution of the data.



(a) CICMalDroid 2020

(b) KDDcup'99

(c) AWID

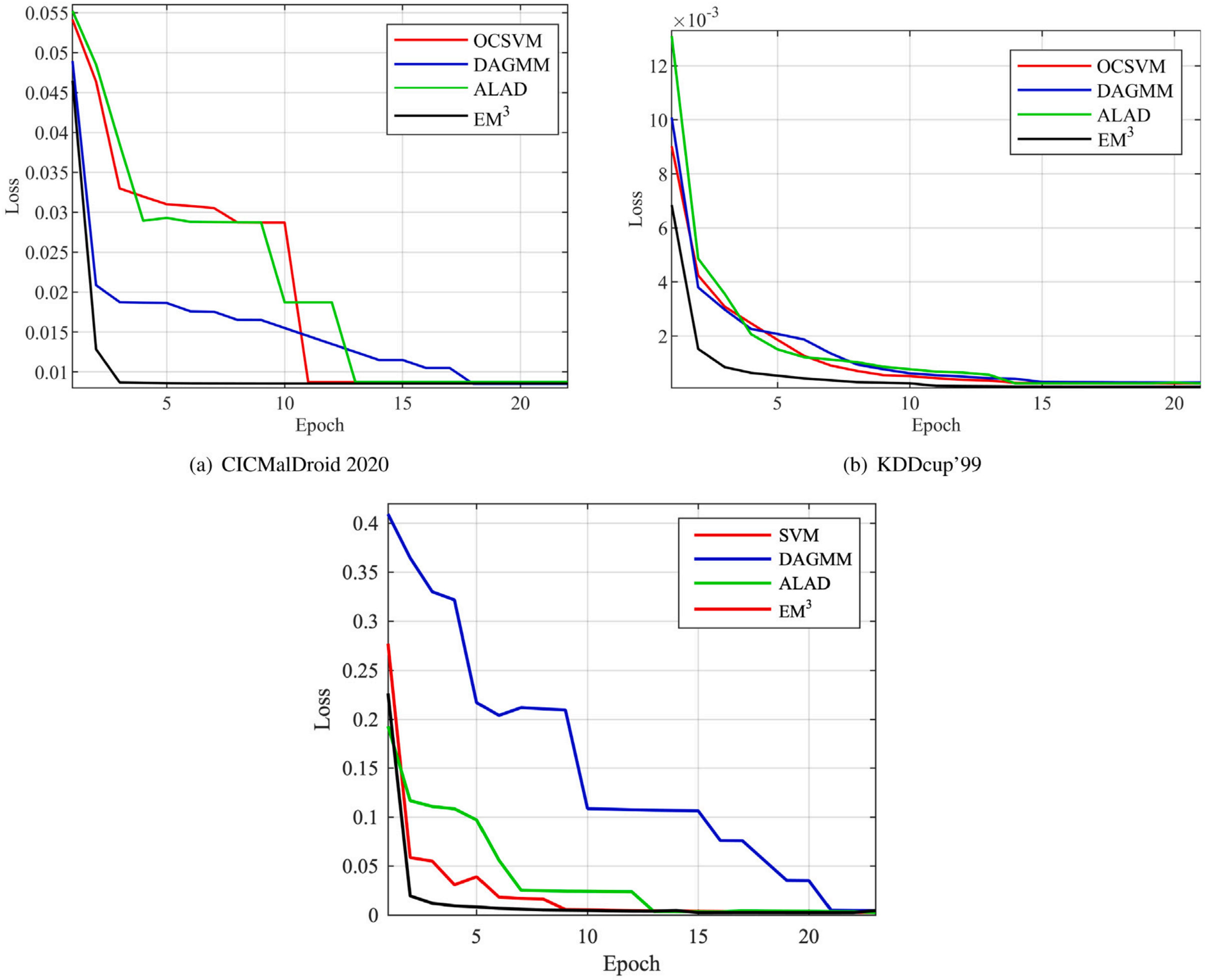
Fig. 2. Representation of data sets in the hidden space. Purple and yellow circles represent low-dimensional representations of normal and abnormal samples, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.4.2. On differentiated domains

Table 3 presents the results of the algorithms running on different domains. Clearly, the proposed EM^3 is superior to the other algorithms when the domains of the data sets are differentiated. For the KDDcup'99 data set, OCSVM is not able to recognize the abnormal samples of the U2R domain (F1 = 0.4887), while it obtains acceptable accuracy on the DoS domain (F1 = 0.7414), which means that the performance of the OCSVM algorithm depends on the distribution of each domain and is not robust to the data set. DAGMM and ALAD have relatively similar performances because they both use a combination of autoencoders and other models. For example, the DAGMM model combines an autoencoder and a GMM, and ALAD combines an autoencoder and a GAN.

Moreover, nearly all of the algorithms follow a similar trend when they are used on the domains of the CICMalDroid 2020 data set except that DAGMM outperforms ALAD in terms of specific domains such as Adware and Banking. On average, algorithms running on the domains of KDDcup'99 obtain the highest values, and algorithms running on the AWID data set have the lowest values. The reason for this is that the amount of data becomes small after it is cleaned, resulting in insufficient model training. In addition, some features are filled, which makes the model unable to learn the changes in each feature. EM^3 is optimal on each domain of the AWID data set. Fig. 4 shows the hidden space of the KDDcup'99 data, which shows that EM^3 can better distinguish the abnormal and normal samples in various domains of the data set.

It is worth noting that the performance of EM^{3+} is improved in each domain of the data set relative to that of EM^3. EM^{3+} improves by at least 0.2% on KDDcup'99 and at least 2% on CICMalDroid 2020. EM^{3+} is an updated version based on the robust



(a) CICMalDroid 2020

(b) KDDcup'99

(c) AWID

Fig. 3. Loss variation versus running epochs.

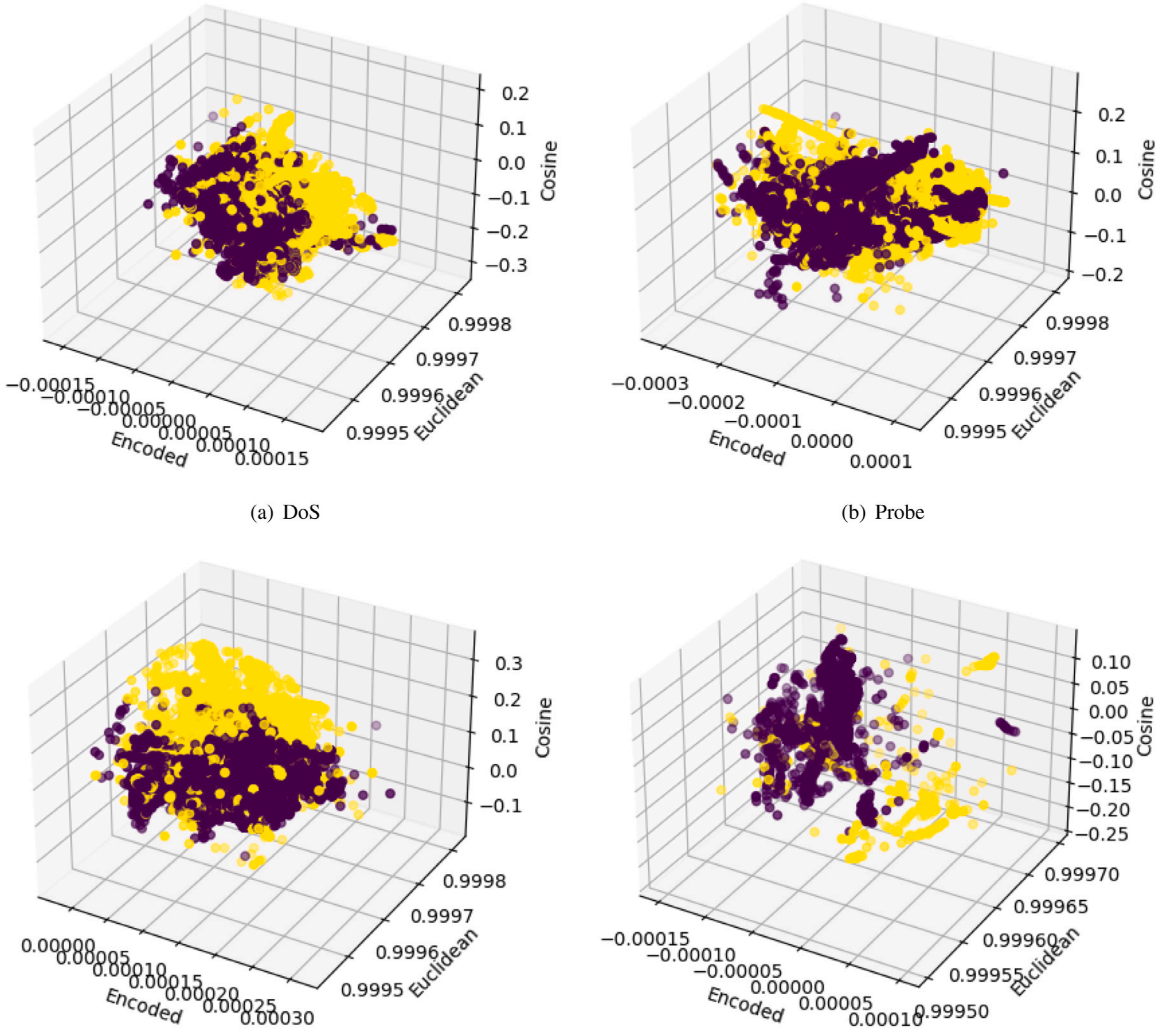
Table 4
Number of parameters that training is completed.

Algorithm	OCSVM	DAGMM	ALAD	EM^3
Number of parameters	836	2135	5032	3512

optimization of EM^3. It converts the multidomain objective function from the original problem to the dual problem and obtains the highest performance in almost every domain. It is concluded that the common feature representation of multidomain data can adapt to different domains, even if there are few data in a domain.

4.4.3. Complexity

Fig. 5(a) shows the time spent by each algorithm in the training and testing stages. The OCSVM has the least training time, and the ALAD has the most training time. The time spent by EM^3 is almost the same as DAGMM but is obviously shorter than ALAD. In the testing stage shown in Fig. 5(b), EM^3 takes less time than DAGMM and ALAD. These findings indicate that EM^3 has a faster detection speed for both normal and abnormal samples after the training stage is completed. The number of parameters of each model is reported in Table 4. The number of parameters of EM^3 and DAGMM is very close, but far less than that of ALAD. This indicates that EM^3 costs less than ALAD. Considering time and parameters together, overall EM^3 has a more competitive time complexity and fewer model parameters, illustrating the conclusion that EM^3 can be deployed flexibly and quickly on current hardware.



(a) DoS

(b) Probe

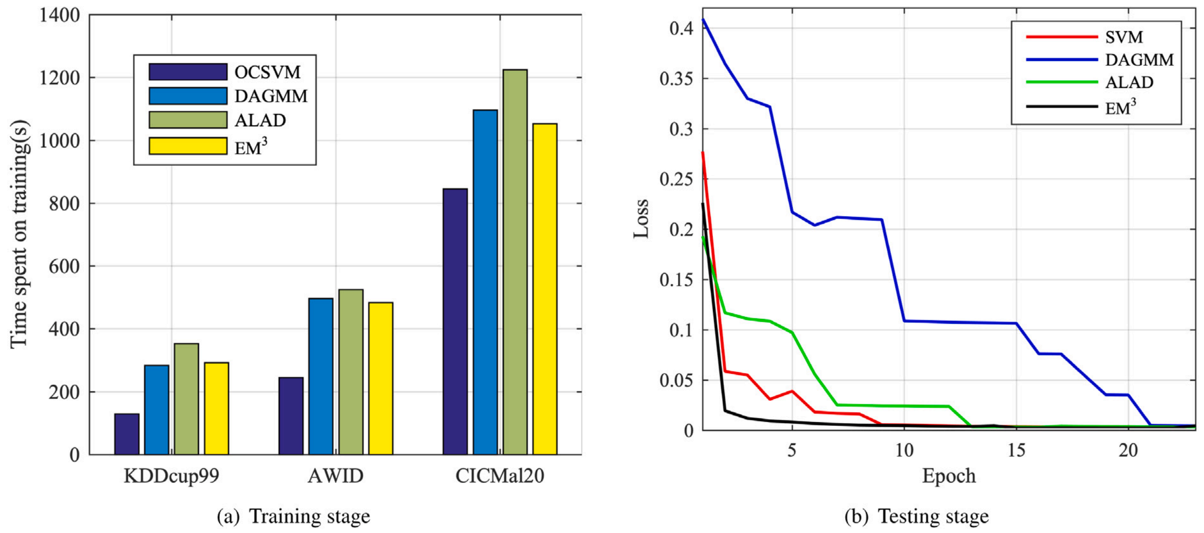
(c) U2R

(d) R2L

Fig. 4. Hidden space representation of the KDDcup'99 data domains as an example. Purple and yellow circles represent low-dimensional representations of normal and abnormal samples, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Conclusion

This paper proposes an unsupervised ensemble autoencoder Gaussian mixture model for cyberattack anomaly detection. It uses the latent representation of the attention mechanism and reconstructed features with minimal errors in the hidden space of the ensemble autoencoder. The expectation maximization algorithm is utilized to estimate the sample density of the Gaussian mixture model. To enhance the cross-domain adaptability of the model, this research transforms the training of multidomain data into a robust optimization problem. Experiments conducted on benchmark data sets show that the proposed model is significantly better than the three selected anomaly detection algorithms. This article focuses on multidomain data in heterogeneous networks, which plays an important role in maintaining the operation of the Internet and protecting Internet of Things devices, especially the privacy and security of a large number of dense mobile users. Future work includes solving the Lagrangian function to find an accurate solution instead of an approximate solution and further improving the ability to detect samples with approximate distributions.



(a) Training stage

(b) Testing stage

Fig. 5. Time spent on training stage and testing stage.

CRediT authorship contribution statement

Peng An: Supervision, Reviewing. **Zhiyuan Wang:** Writing – original draft, Model implementation, Experimentation. **Chunjiong Zhang:** Conceptualization, Methodology, Data curation, Experimentation, Visualization, Editing.

Acknowledgments

This research was supported by Ningbo Natural Science Foundation, China 2019A610106 and National Natural Science Foundation of China under Grant No. 61502256. Peng An and Zhiyuan Wang has equal contribution to this paper.

References

Andresini, G., Appice, A., Di Mauro, N., Loglisci, C., & Malerba, D. (2019). Exploiting the auto-encoder residual error for intrusion detection. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (pp. 281–290). IEEE.

Berriel, R., Lathuillere, S., Nabi, M., Klein, T., Oliveira-Santos, T., Sebe, N., et al. (2019). Budget-aware adapters for multi-domain learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 382–391).

Chen, Z., Yan, Q., Han, H., Wang, S., Peng, L., Wang, L., et al. (2018). Machine learning based mobile malware detection using highly imbalanced network traffic. *Information Sciences*, *433*, 346–364.

Dromard, J., Roudière, G., & Owezarski, P. (2016). Online and scalable unsupervised network anomaly detection method. *IEEE Transactions on Network and Service Management*, *14*(1), 34–47.

Fourure, D., Emonet, R., Fromont, E., Muselet, D., Neverova, N., Trémeau, A., et al. (2017). Multi-task, multi-domain learning: application to semantic segmentation and pose regression. *Neurocomputing*, *251*, 68–80.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, *17*(1), 2096–2030.

Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., et al. (2019). Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1705–1714).

Hu, D., Li, J., Liu, Y., & Li, Y. (2020). Flow adversarial networks: Flowrate prediction for gas-liquid multiphase flows across different domains. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(2), 475–487.

Injadat, M., Moubayed, A., Nassif, A. B., & Shami, A. (2020). Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Transactions on Network and Service Management*, 1. http://dx.doi.org/10.1109/TNSM.2020.3014929.

Injadat, M., Moubayed, A., & Shami, A. (2020). Detecting botnet attacks in IoT environments: An optimized machine learning approach. In *2020 32nd International Conference on Microelectronics (ICM)* (pp. 1–4). http://dx.doi.org/10.1109/ICM50269.2020.9331794.

Kim, Y.-g., Kwon, Y., Chang, H., & Paik, M. C. (2020). Lipschitz continuous autoencoders in application to anomaly detection. In *International Conference on Artificial Intelligence and Statistics* (pp. 2507–2517). PMLR.

Li, Q., Zou, S., & Zhong, W. (2020). Learning graph neural networks with approximate gradient descent. arXiv preprint arXiv:2012.03429.

Liao, W., Guo, Y., Chen, X., & Li, P. (2018). A unified unsupervised gaussian mixture variational autoencoder for high dimensional outlier detection. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 1208–1217). IEEE.

Liu, Y., Li, Z., Zhou, C., Jiang, Y., Sun, J., Wang, M., et al. (2020). Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, *32*(8), 1517–1528. http://dx.doi.org/10.1109/TKDE.2019.2905606.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.

Mahdavifar, S., Kadir, A. F. A., Fatemi, R., Alhadidi, D., & Ghorbani, A. A. (2020). Dynamic android malware category classification using semi-supervised deep learning. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)* (pp. 515–522). IEEE.

Majumdar, A., & Tripathi, A. (2017). Asymmetric stacked autoencoder. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 911–918). IEEE.

Mariño, I. P., & Míguez, J. (2006). An approximate gradient-descent method for joint parameter estimation and synchronization of coupled chaotic systems. *Physics Letters. A*, *351*(4–5), 262–267.

Peng, N., & Dredze, M. (2016). Multi-task multi-domain representation learning for sequence tagging. CoRR, abs/1608.02689.

Pratama, K., & Kang, D.-K. (2020). Trainable activation function with differentiable negative side and adaptable rectified point. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 1–18.

Qian, Q., Zhu, S., Tang, J., Jin, R., Sun, B., & Li, H. (2019). Robust optimization over multiple domains. In: *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, no. 01* (pp. 4739–4746).

Ren, Z., & Lee, Y. J. (2018). Cross-domain self-supervised multi-task feature learning using synthetic imagery. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 762–771).

Rezvy, S., Petridis, M., Lasebae, A., & Zebin, T. (2018). Intrusion detection and classification with autoencoded deep neural network. In *International Conference on Security for Information Technology and Communications* (pp. 142–156). Springer.

Schoenauer-Sebag, A., Heinrich, L., Schoenauer, M., Sebag, M., Wu, L. F., & Altschuler, S. J. (2019). Multi-domain adversarial learning. arXiv preprint arXiv:1903.09239.

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(1), 41–50.

Vaca, F. D., & Niyaz, Q. (2018). An ensemble learning based wi-fi network intrusion detection system (wnids). In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)* (pp. 1–5). IEEE.

Wang, K., Xu, L., Huang, L., Wang, C.-D., & Lai, J.-H. (2018). Stacked discriminative denoising auto-encoder based recommender system. In *International Conference on Intelligent Science and Big Data Engineering* (pp. 276–286). Springer.

Xu, R., Chen, Z., Zuo, W., Yan, J., & Lin, L. (2018). Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3964–3973).

Xu, S., Qian, Y., & Hu, R. Q. (2019). Data-driven network intelligence for anomaly detection. *IEEE Network*, *33*(3), 88–95.

Yu, W. (2020). Optimization of combined power and modeling attacks on VR PUFs with Lagrange multipliers. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *67*(11), 2512–2516.

Zenati, H., Romain, M., Foo, C.-S., Lecouat, B., & Chandrasekhar, V. (2018). Adversarially learned anomaly detection. In *2018 IEEE International Conference on Data Mining (ICDM)* (pp. 727–736). IEEE.

Zhang, Y., Li, X., Gao, L., Chen, W., & Li, P. (2020). Intelligent fault diagnosis of rotating machinery using a new ensemble deep auto-encoder method. *Measurement*, *151*, Article 107232.

Zhou, C., & Paffenroth, R. C. (2017). Anomaly detection with robust deep autoencoders. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 665–674).

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., et al. (2018). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*.