Masked Memory Network for Semi-Supervised Anomaly Detection in Internet of Things

Jiaxin Yin, Yuanyuan Qiao, *Member, IEEE*, Zunkai Dai, Zitang Zhou, Xiangchao Wang, Wenhui Lin, and Jie Yang

***Abstract*—With the rapid development of Internet of Things (IoT), an increasing volume of data is generated across diverse IoT devices. Within these data, an extremely limited subset may manifest notable deviations from the majority of data, such as network intrusion data in traffic monitoring devices. The identification of such anomalous data assumes considerable importance across diverse domains. In this article, we propose masked memory network, a semi-supervised anomaly detection method which can be applied to various IoT devices. Our approach leverages a masked memory module combined with a soft masking strategy to acquire discriminative patterns capable of distinguishing anomalies from normal data. We also devise a anomaly scoring strategy which can exploits the characteristics of attention weights between test samples and memory items to detect both known and unknown anomalies. Experimental results on various AD data sets collected by IoT devices demonstrate the effectiveness of our work in numerous IoT applications.**

***Index Terms*—Anomaly detection (AD), Internet of Things (IoT), neural networks, semi-supervised learning.**

I. Introduction

INTERNET of Things (IoT) describes devices with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. As the development of IoT continues to advance, it brings about a substantial increase in data volume, inspiring various researches based on IoT data, such as technologies for user behavior prediction [1], smart agriculture [2], and improving IoT security [3], [4]. The surge in data volume has also led to the increased emergence of abnormal data patterns within IoT systems. For example, for IoT in healthcare, deviations from regular heart rate patterns could serve as indicators of potential health concerns. For IoT in industrial production, deviations from expected characteristics indicate defects in manufactured products. Detecting these anomalies is crucial as it enables timely identification and mitigation of potential issues, prevents the adverse consequences across diverse domains, such as security, healthcare, and industrial production.

Most anomaly detection (AD) methods utilized in IoT applications are implemented in an unsupervised setting where all the training data are unlabeled [5], [6], [7], [8], [9], [10], [11], [12] or only normal samples are available in training set [13], [14], [15], [16], [17], [18]. However, for a lack of prior knowledge of true anomalies, these methods struggle in learning discriminative patterns that can differentiate anomalies from normal data, leading to high false positives or low detection recall [19]. Semi-supervised AD is one of the most promising paradigms to address this weakness, where the training set consists of a large set of unlabeled data and a small set of labeled anomalies. It has been proven with merely 1% labeled anomalies, suitable semi-supervised methods can outperform the best unsupervised method [20]. For example, DevNet [21], FEAWAD [22], and PReNet [19] leverage deviation loss to map representations of anomalies to a score higher than that of normal data. Deep SAD [23] learns a hypersphere in representation space, where anomalies are distributed outside the sphere while normal data are inside. Overlap [24] constrains the overlap of anomaly score between anomalies and normal data to enhance the differences between them. However, these works can detect known anomalies well, i.e., anomalies of the same type with labeled anomalies, but cannot generalize well to detect unknown anomalies, i.e., anomalies of different types with labeled anomalies. To this end, a method is needed to solve the two major challenges in AD simultaneously: 1) How to learn discriminative patterns that can differentiate anomalies from normal data and 2) How to detect unknown anomalies uncovered by the labeled anomaly data [25].

In this article, we propose masked memory network (MMNet) to address these challenges. Memory network is first presented for question answering [26], which consists of a set of learnable embeddings to record important information as a knowledge base. For AD, memory network is usually set between an encoder and a decoder to learn patterns of normal data, leading to larger reconstruction errors for anomalies [27], [28]. To solve the first challenge, we propose a soft masking strategy for memory network. Specifically, we design a mask loss, forcing the query features of normal data

Manuscript received 11 March 2024; revised 6 May 2024; accepted 5 June 2024. Date of publication 13 June 2024; date of current version 25 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62272057, and in part by the Major Science and Technology Projects in Anhui Province under Grant 202203a05020025. *(Corresponding author: Yuanyuan Qiao.)*

Jiaxin Yin, Yuanyuan Qiao, Zunkai Dai, Zitang Zhou, and Jie Yang are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yinjx@bupt.edu.cn; yyqiao@bupt.edu.cn; daizk@bupt.edu.cn; zhouzitang@bupt.edu.cn; janeyang@bupt.edu.cn).

Xiangchao Wang is with the School of Electronic Information, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: 20041926@hdu.edu.cn).

Wenhui Lin is with the Technology Research Institute, Aisino Corporation, Beijing 100195, China (e-mail: linwenhui@aisino.com).

Digital Object Identifier 10.1109/JIOT.2024.3413676

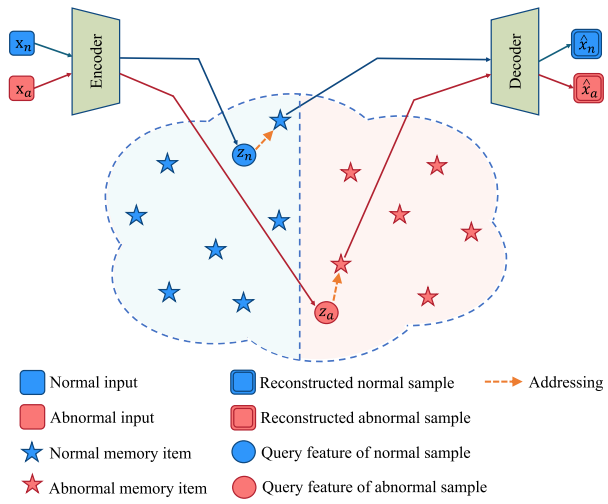

Fig. 1. Learning process of memory items. In traditional autoencoders, representation **z** from the encoder will be sent to the decoder directly for reconstruction. In our methods, **z** is used as a query, memory items are used as keys and values, and the weighted combination of **z**'s closest memory items (To simplify the visualization, we assume only one memory item is retrieved here) will be sent to the decoder for reconstruction. Simultaneously, through soft mask strategy, we encourage normal data to retrieve the first (blue) part of the memory items for reconstruction, and abnormal data to retrieve the second (red) part. To reconstruct corresponding data well, the memory items in the first and second parts will be encouraged to record patterns of normal and abnormal data, respectively.

to only focus on one part of memory items when perform addressing, while the query features of known anomalies only focus on the other part. Under the constraints of reconstruction loss and mask loss, the first part of memory items will be encouraged to capture prototypical patterns of normal data, the other part of memory items will be encouraged to record patterns of known anomalies. At the same time, the distance between these two parts of memory items in latent space will be enlarged, rendering them discriminative in distinguishing between normal and abnormal data. The process is shown in Fig. 1.

To address the second challenge, we design an anomaly scoring strategy. Most AD methods based on AE use reconstruction error as anomaly score [27], [29], [30]. However, this metric exclusively leverages information from input space, ignoring the latent space information which is more expressive. We propose to detect anomalies by exploring the characteristics of attention weights, which are calculated by the similarity between the query feature of current input and all the memory items. Since the first part of memory items capture normal patterns, the sum of the first part of attention weights can characterize how far a sample deviates from normal patterns in latent space. Another term is the Kullback–Leibler (KL) divergence between the distribution of attention weights and the uniform distribution. Given that unknown anomalies are dissimilar to both parts of memory items, the distribution of their attention weights tends to be closer to the uniform distribution, so this term can highlight unknown anomalies. Overall, the main contributions can be summarized as follows.

1) We present MMNet for semi-supervised AD in IoT applications. To learn discriminative normal and abnormal patterns, we propose a soft masking strategy, encouraging the memory to capture prototypical patterns of normal data and known anomalies separately, and enlarging the distance of two parts of memory items in latent space at the same time. To our best known, it is the first time to leverage memory network for semi-supervised AD.
2) We propose a novel scoring strategy, exploring the characteristics of attention weights to detect both known and unknown anomalies. We use the sum of the first part of attention weights to measure the extent of deviation from learned normal patterns. Besides, we calculate the KL-divergence between the distribution of attention weights and the uniform distribution to highlight unknown anomalies.
3) Extensive experiments on various data sets collected by IoT devices demonstrate the effectiveness of our method and show that it achieves better performance than state-of-the-art methods. We also provide an extensive experimental analysis with ablation studies.

II. Related Work

A. Unsupervised Anomaly Detection

A plethora of literature has emerged to address the problem of AD in the last few decades [20], [25], [31], [32], [33], where most works focus on unsupervised setting. In general, unsupervised AD can be categorized into four groups: 1) density-based methods [8], [9], [12], [34]; 2) distance-based methods [5], [35]; 3) classification-based methods [6], [10], [36]; and 4) reconstruction-based methods [11], [37], [38].

Recently, a new setting has attracted increasing attention where only normal samples are available in training set [13], [14], [15], [16], [17], [18]. E^3Outlier [13] uses surrogate supervision to train a NN by creating multiple pseudo classes of different transformations of images, and proposes a negative entropy score based on inlier priority. CSI [14] leverages contrastive learning with distributionally-shifted transformations, which leads to better representation in terms of AD. GOAD [15] extends the applicability of transformation-based methods to nonimage data using random affine transformations. NeuTralAD [16] derives a single objective function for jointly learning useful data transformations and representations, the function is also used for the calculation of anomaly score. ICL [17] learns mappings that maximize the mutual information between each sample and the part that is masked out. The mappings are learned by employing an internal contrastive loss. MCM [18] generalizes mask modeling methods to tabular AD, generates masks by learning and proposes a diversity loss to avoid redundant masks. ADRIoT [39] proposes an AD framework for IoT networks, which leverages edge computing to uncover potential threats. GTA [40] integrates graph convolution and temporal dependency modeling, based on which a transformer-based architecture is designed for multivariate time-series AD in IoT. However, a common limitation of the above methods is

that discriminative normal/abnormal patterns are hard to learn for a lack of prior knowledge or true anomalies.

B. Semi-Supervised Anomaly Detection

Semi-supervised setting can solve the limitation by introducing limited labeled anomalies. In real-world IoT applications, a small number of anomalies are usually accessible, but very little attention has been paid to it. DevNet directly maps data to an anomaly score, of which labeled anomalies are higher than unlabeled data by a predefined margin. Deep SAD first uses a center of hypersphere to characterize normality, then penalizes the distance of unlabeled data to the center as well as the reciprocal of the distance of labeled anomalies to the center. FEAWAD leverages an AE to extract three factors, which are then combined to generate an anomaly score. Based on DevNet, PReNet packs two samples into a pair, and maps the pair to a score that depends on the number of labeled anomalies in the pair. SOEL [41] proposes a block coordinate ascent scheme that alternates between inferring the unknown labels and minimizing the semi-supervised outlier exposure loss. Overlap first estimates the score distribution of normal and abnormal samples, then designs a loss to minimize the overlap area of them. However, all these methods are overfitting and biased toward known anomalies, thus hard to detect unknown anomalies. In DRA [42], Ding el al. try to detect unknown anomalies by introducing pseudo anomalies, but this method is only available for image data and its performance would substantially deteriorate if the unknown anomalies are not similar to the pseudo anomalies. To sum up, current AD works either cannot learn discriminative patterns or fail to detect unknown anomalies well. Different from above works, we propose MMNet, which can learn discriminative normal/abnormal patterns and detect both known and unknown anomalies without any strict condition.

C. Memory Network

Memory network is first presented to describe long term dependencies in sequential data for question answering [26]. Sukhbaatar et al. [43] proposed a continuous form of the memory network, which can be successfully trained end-to-end via backpropagation. Since then, memory networks have been used in various applications. MemAE [27] is the first work to introduce memory network to AD. Given an input, MemAE first obtains the encoding from the encoder and then uses it as a query to retrieve the most relevant memory items for reconstruction. Memory is encouraged to record the prototypical normal patterns, so MemAE can increase the reconstruction error of the anomalies while reconstruct normal samples well. Considering the various patterns of normal data, Park et al. [28] proposed feature compactness and separateness losses, enforcing the memory items to record more diverse and more discriminative features. Two works [30], [44] employ multiple memory modules to memorize normal patterns at different feature levels in an AE structure with skip connections. Cai et al. [45] introduced memory network to model the information in both appearance and motion signals, and then use a transfer module to realize the fusion of two modalities' information. PatchCore [46] samples in distribution features into a memory module and performs nearest neighbor searching to discover anomalies. In SQUID [47], a visual pattern dictionary is dynamically maintained that classifies recurring anatomical patterns based on the spatial location of the training data. However, these works all focus on unsupervised settings in which memory items only record the pattens of normal data. In our work, we introduce memory network to semi-supervised AD, and enforce the memory to capture prototypical patterns of normal data and known anomalies separately.

III. Methods

A. Problem Statement and Notations

Let $\mathcal{X}$ represents the domain of the data samples and $\mathcal{Y} \in \{0, 1\}$ represents the domain of labels. We define samples with $y = 0$ to be normal data and $y = 1$ to be anomalies. We are given a training data set containing a large set of unlabeled data and a small set of labeled anomalies, $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N, (\mathbf{x}_{N+1}, y_{N+1}), \ldots, (\mathbf{x}_{N+M}, y_{N+M})\}$, where unlabeled data set are denoted by $\mathcal{D}_u = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, while labeled data set are denoted by $\mathcal{D}_l = \{(\mathbf{x}_{N+1}, y_{N+1}), (\mathbf{x}_{N+2}, y_{N+2}), \ldots, (\mathbf{x}_{N+M}, y_{N+M})\}$, $M << N$. A consensus in AD is that normal instances usually account for an overwhelming proportion of the data set. To conform to this consensus and real-world applications in IoT, in our setting, the majority of data in $\mathcal{D}_u$ are normal samples while the remaining few are abnormal samples. For $\mathcal{D}_l$, all the data are with the label $y = 1$, providing prior information of anomalies. The objective is to generate an anomaly score for every test sample, a higher value indicates a higher probability of anomalies.

B. Network Architecture

The framework of our proposed MMNet is shown in Fig. 2. Given an input, it will first be encoded to an encoding by the encoder. Then, the encoding will act as a query feature to perform attention-based addressing operation on the memory items to obtain the attention weights $\mathbf{w}$ and value feature $\hat{\mathbf{z}}$. During this process, we apply soft masking strategy, according to whether the input is from D_u or D_l, one part of the memory items will be softly masked and $\mathbf{z}$ will focus on the other part of memory items. Finally, $\hat{\mathbf{z}}$ will be sent to the decoder for reconstruction and $\mathbf{w}$ will be used to calculate the anomaly score. The model is trained end-to-end by minimizing reconstruction loss, mask loss and KL loss. The encoder and decoder, memory module and addressing operation, soft masking strategy, loss function, and anomaly scoring strategy will be elaborated in the following parts, meanwhile, we will also present some implicit assumptions used in our method.

C. Encoder and Decoder

Assumption 1: The manifold assumption asserts that the data lives (approximately) on some lower dimensional manifold that is embedded within the data space.

This assumption implies the existence of a low-dimensional latent space which can capture intrinsic structure of high-dimensional data. In AD, AE is the most common network

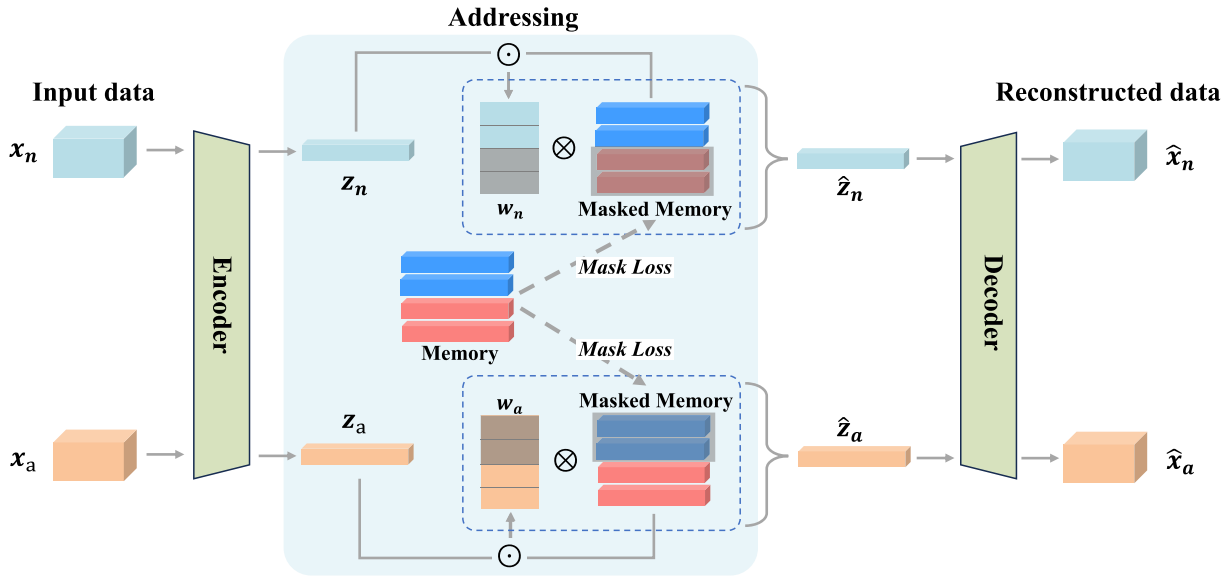

Fig. 2. Overview of MMNet. When inputs a sample, the encoder will encode it to a representation $\mathbf{z}$, then $\mathbf{z}$ will perform attention-based addressing operation with memory items, where $\mathbf{z}$ acts as a query, memory items act as keys and values. By attention mechanism, $\hat{\mathbf{z}}$, the weighted combination of $\mathbf{z}$'s most relevant memory items, will be obtained and sent to decoder for reconstruction. During this process, a constraint is imposed: when the input is a normal sample, we force it focus on the first part of memory items, so we mask the other part, on the contrary, when the input is an abnormal sample, we force it focus on the second part of memory items and mask the first part. By doing so, two parts of memory items will be encouraged to record prototypes of normal and abnormal data, respectively.

to find such latent space, in which an encoder maps the input data to a latent representation, while an decoder maps the representation from latent space back to the original space. By minimizing the reconstruction error, the representation learnt will capture the most important representations of the original data.

In our method, we also encode the data from original space $\mathcal{X}$ to latent space $\mathcal{Z}$ by an encoder ϕ_e

$$\mathbf{z} = \phi_e(\mathbf{x}, \theta_e) \tag{1}$$

where $\mathbf{x}$ is a input sample, $\mathbf{z}$ is the representation of the input sample, and θ_e denotes parameters of the encoder.

Different from vallina AE, we do not feed $\mathbf{z}$ to the decoder directly, but use it as a query to retrieve the weighted combination of relevant memory items $\hat{\mathbf{z}}$, then an decoder ϕ_d maps $\hat{\mathbf{z}}$ from $\mathcal{Z}$ to $\mathcal{X}$

$$\hat{\mathbf{x}} = \phi_d(\hat{\mathbf{z}}, \theta_d) \tag{2}$$

where θ_d denotes parameters of the decoder.

D. Memory Module and Addressing Operation

Assumption 2: The prototype assumption asserts that there exists a finite number of prototypical elements in the data that characterize the data well.

Memory module $\mathcal{M} \in \mathbb{R}^{(K+L)\times C}$ is a real-value matrix, initialized randomly by Xavier initialization [48], where $K+L$ and C denotes the size and the dimension of memory module, respectively. Each row of the memory module is defined as a memory item $\mathbf{m}_j$ with fixed dimension C, where $j \in \{1, 2, \ldots, (K+L)\}$. Memory items are learnable embeddings that can be updated during training. We denote the first part $\mathcal{M}_u \in \mathbb{R}^{K\times C}$, and the second part $\mathcal{M}_l \in \mathbb{R}^{L\times C}$, which will be trained to record prototypical elements of $\mathcal{D}_u$ and $\mathcal{D}_l$ separately.

Addressing operation is based on attention mechanism, converting the input of the decoder from $\mathbf{z}$ to $\hat{\mathbf{z}}$, where $\mathbf{z}$ acts as a query, memory items are keys, and the value $\hat{\mathbf{z}}$ is the linear combination of $\mathbf{z}$'s most relevant items in the memory. In this way, memory items participant in the training phase and can be updated through back-propagation. During training, combinations of different memory items are going to reconstruct all the samples in D after decoder, consequently, memory items will be trained to record common and salient representations in D, i.e., prototypical elements.

Specifically, addressing operation first computes the similarity d_j of the query feature $\mathbf{z}$ with each item $\mathbf{m}_j$ in memory module

$$d_j = \mathbf{z} \cdot \mathbf{m}_j^T \tag{3}$$

here $\cdot$ denotes linear product operation. Then, corresponding weights are calculated by softmax operation

$$w_j = \text{softmax}(d_j) = \frac{\exp(d_j)}{\sum_k \exp(d_k)}. \tag{4}$$

We denote all the attention weights $\mathbf{w} = [w_1, \ldots, w_K, w_{K+1}, \ldots, w_{K+L}]$. Following [27], we constrain the sparsity of $\mathbf{w}$ by the shrinkage operation:

$$\tilde{w}_j = \frac{\max(w_j - \gamma, 0) \cdot w_j}{| w_j - \gamma | + \epsilon} \tag{5}$$

where ϵ is a tiny constant to avoid a zero denominator, and γ is a threshold adjusting the degree of shrinkage. Shrinkage operation is to promote the sparsity of attention weights, enforcing the query feature to focus on the memory items that are most relevant to them, which is conductive to the

reconstruction of samples as well as the updates of memory items. Then, we renormalize the attention weights:

$$\hat{w}_j = \tilde{w}_j / \|\tilde{\mathbf{w}}\|_1. \tag{6}$$

Finally, $\hat{\mathbf{z}}$ will be obtained by weighted linear combination of memory items with the normalized attention weights $\hat{\mathbf{w}}$:

$$\hat{\mathbf{z}} = \sum_j \hat{w}_j \mathbf{m}_j. \tag{7}$$

E. Soft Masking Strategy

Assumption 3: The data imbalance assumption asserts that in AD data sets, anomalies are typically rare data instances, contrasting to normal instances usually account for an overwhelming proportion of the data.

Although memory items can capture prototypical patterns in D after addressing operation, we have no idea which items capture normal patterns and which capture abnormal patterns. The purpose of masking strategy is to force the memory to record representations that capture normal and abnormal patterns separately.

First, we propose a soft masking strategy, encouraging memory items in M_u to record the prototypes of D_u, while memory items in M_l to record prototypes of D_l. Specifically, we make the query features perform addressing on all the memory items and then encourage them to focus on one part by the following constraint:

$$l_{\text{mask}} = (1-l)w_l + l w_u \tag{8}$$

where $w_l = \sum_{j=K+1}^{K+L} w_j$, $w_u = \sum_{j=1}^{K} w_j$, w_j is the jth items in attention weights $\mathbf{w} = [w_1, \ldots, w_K, w_{K+1}, \ldots, w_{K+L}]$. $l = 0$ denotes the sample comes from D_u while $l = 1$ denotes the sample comes from D_l. When input is an unlabeled sample, minimizing (8) means w_l will be smaller. Since $w_l + w_u = 1$, w_u will be larger. On the one hand, a larger w_u will encourage unlabeled samples to pay attention to memory items in $\mathcal{M}_u$, while a smaller w_l will encourage them to stay away from items in $\mathcal{M}_l$. On the other hand, the memory items in $\mathcal{M}_u$ and $\mathcal{M}_l$ will, respectively, shorten and increase the distance from the representations of unlabeled data.

In the following formulas, we take the first memory item $\mathbf{m}_1$ as an example to show the update of items in $\mathcal{M}_u$ under the constraint of (8):

$$\begin{aligned}
\frac{\partial l_{\text{mask}}}{\partial \mathbf{m}_1} &= \frac{\partial w_l}{\partial \mathbf{m}_1} = -\frac{\partial w_u}{\partial \mathbf{m}_1} = -\frac{\partial\left(\sum_{j=1}^{K} w_j\right)}{\partial \mathbf{m}_1} \\
&= -\left[\frac{\partial w_1}{\partial \mathbf{m}_1} + \frac{\partial\left(\sum_{j=2}^{K} w_j\right)}{\partial \mathbf{m}_1}\right] \\
&= -\left[\frac{\partial w_1}{\partial d_1}\frac{\partial d_1}{\partial \mathbf{m}_1} + \frac{\partial\left(\sum_{j=2}^{K} w_j\right)}{\partial d_1}\frac{\partial d_1}{\partial \mathbf{m}_1}\right] \\
&= -\left[(1-w_1)w_1\mathbf{z} - \sum_{j=2}^{K} w_j w_1 \mathbf{z}\right] \\
&= -\left(1 - \sum_{j=1}^{K} w_j\right) w_1 \mathbf{z} \\
\mathbf{m}_1 &:= \mathbf{m}_1 + \alpha\left(1 - \sum_{j=1}^{K} w_j\right) w_1 \mathbf{z}
\end{aligned} \tag{9}$$

where α is the learning rate above 0, and it is obvious $(1 - \sum_{j=1}^{K} w_j)$ and w_1 are both over 0. Thus, items in $\mathcal{M}_u$ will be updated to decrease their distance with the latent feature $\mathbf{z}$ of unlabeled samples. To analyse the updates of items in $\mathcal{M}_l$, we take the $K+1$ memory item $\mathbf{m}_{K+1}$ as an example. Similarly, $\mathbf{m}_{K+1}$ is updated by

$$\begin{aligned}
\frac{\partial l_{\text{mask}}}{\partial \mathbf{m}_{K+1}} &= \left(1 - \sum_{j=K+1}^{K+L} w_j\right) w_{K+1}\mathbf{z} \\
\mathbf{m}_{K+1} &:= \mathbf{m}_{K+1} - \alpha\left(1 - \sum_{j=K+1}^{K+L} w_j\right) w_{K+1}\mathbf{z}
\end{aligned} \tag{10}$$

where α, $(1 - \sum_{j=K+1}^{K+L} w_j)$ and w_{K+1} are all over 0. Thus, items in $\mathcal{M}_l$ will increase their distance with latent feature $\mathbf{z}$ of unlabeled samples.

Conversely, when inputs an anomaly, the memory items in $\mathcal{M}_u$ and $\mathcal{M}_l$ will, respectively, increase and shorten the distance from the latent features of the anomaly. The distance between memory items in $\mathcal{M}_u$ and $\mathcal{M}_l$, which represent the patterns of D_u and D_l in the latent space, are also increased.

Note although we do not know the label of samples in D_u, based on the data imbalance assumption and the actual situation in IoT applications, most data in D_l are normal. A vital setting here is $K << N$, i.e., the number of memory items in M_u is much smaller than the number of unlabeled data in D_u. In this way, M_u cannot capture all the patterns of unlabeled data in D_u for a lack of enough memory items. In order to reduce the reconstruction loss for the majority of the data, and based on inlier priority [13], the limited memory items in Mu will record the prominent patterns shared by the majority of the data in Du, i.e., normal patterns. Thus, memory items in M_u and M_l can capture prototypical patterns of normal data and labeled anomalies, respectively. Experiments will show that our method can perform well with different ratio of anomaly contamination in D_u.

F. Loss Function

We present the following overall loss to train our model:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_m \mathcal{L}_{\text{mask}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}} \tag{11}$$

where $\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{mask}}$, and $\mathcal{L}_{\text{KL}}$ are reconstruction loss, mask loss, and KL loss, respectively, λ_m and λ_{KL} are hyperparameters to balance the them. We set $\lambda_m = 1$ and $\lambda_{\text{KL}} = 0.0001$ by default.

Reconstruction Loss: Reconstruction loss is extensively used in AE. In unsupervised setting, it usually minimizes the reconstruction error of all the data. Considering a small set of labeled anomalies is available now, we constrain the reconstruction error of anomalies to explicitly deviate from the

unlabeled data with a margin a, so as to push the distribution of anomalies in latent space away from the normal objects

$$\mathcal{L}_{\text{rec}} = \sum_i (1 - l_i)e_i + l_i \max(0, a - e_i) \tag{12}$$

where e_i is the reconstruction error between sample $\mathbf{x}_i$ and its reconstruction $\hat{\mathbf{x}}_i$. l_i decides whether x_i is an labeled data ($l_i = 1$) or not ($l_i = 0$). We use l_2 norm to measure the reconstruction error here

$$e_i = \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2. \tag{13}$$

Mask Loss: Based on the mask loss of single sample in (8), we use w_{il}, w_{iu} to denote w_l, w_u of input sample x_i. So, the mask loss of all the training samples here is

$$\mathcal{L}_{\text{mask}} = \sum_i (1 - l_i)w_{il} + l_i w_{iu}. \tag{14}$$

KL Loss: KL loss aims to minimize the negative of the KL-divergence between the distribution of attention weights $\mathbf{w}_i$ and the uniform distribution $\mathbf{u}$

$$\begin{aligned}\mathcal{L}_{\text{KL}} &= -D_{\text{KL}}(\mathbf{u}||\mathbf{w_i}) \\ &= \frac{1}{K+L}\sum_{j=1}^{K+L} \log\big(w_{ij}\big) + H(\mathbf{u})\end{aligned} \tag{15}$$

where $\mathbf{u} = [(1/[K+L]), (1/[K+L]), \ldots, (1/[K+L])] \in \mathbb{R}^{[K+L]}$, H is the entropy operation. Neglecting the constant term $H(\mathbf{u})$ and coefficient, KL loss can be simplified as

$$\mathcal{L}_{\text{KL}} = \sum_{j=1}^{K+L} \log\big(w_{ij}\big). \tag{16}$$

This term applies constraint on all the training data. After training, if a test sample is of the same type with some training data, whether they are normal samples or known anomalies, its KL-divergence will also be larger due to this constraint. But for unknown anomalies whose types are different with all the training data, their KL-divergence will be smaller because they are not subject to this constraint. Thus, $\mathcal{S}_{\text{KL}}$ will be more effective in detecting unknown anomalies.

G. Anomaly Scoring Strategy

Most AE-based methods employ reconstruction error as anomaly score, however, a limitation is that it only leverages information of input space, but ignores latent space representations, which are more semantic and more expressive. Here, we present a novel scoring strategy based on attention weights, which are calculated by the similarity between the query feature of a test sample and all the memory items. Specifically, during testing, the query feature z_i of a test sample x_i performs addressing on all the $K+L$ items in $\mathcal{M}$ to obtain the attention weights $\mathbf{w}_i = [w_{i1}, \ldots, w_{iK}, w_{i(K+1)}, \ldots, w_{i(K+L)}]$, where the first K items in $\mathbf{w}_i$ are calculated with items in $\mathcal{M}_u$, and the latter L items in $\mathbf{w}_i$ are calculated with items in $\mathcal{M}_l$. Because items in $\mathcal{M}_u$ are prototypical normal patterns, the first K items in $\mathbf{w}_i$ can measure how far a sample deviates from normal patterns in latent space. Thus, we sum the first K attention weights as the main term of our anomaly score

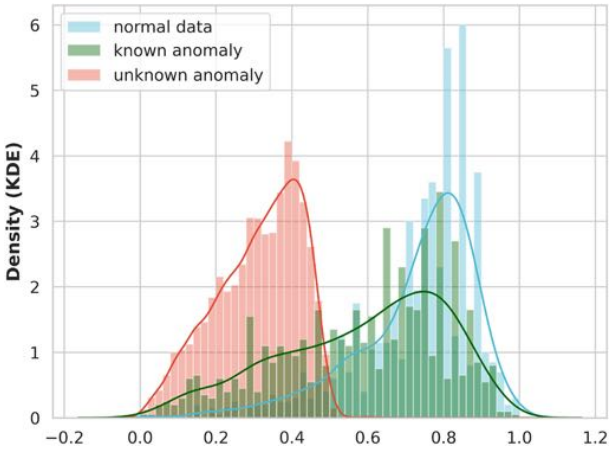

Fig. 3. Distribution of $\mathcal{S}_{\text{KL}}$ on carpet objects of MVTec AD data set, which has five types of anomalies. Here, data with color defects are chosen as known anomalies, and the left four types of abnormal data are unknown anomalies.

$$\mathcal{S}_{\text{sum}} = \sum_{j=1}^{K} w_{ij}. \tag{17}$$

To highlight the unknown anomalies, another term of our anomaly score is the KL-divergence between the distribution of attention weights $\mathbf{w}_i$ and the uniform distribution $\mathbf{u}$. Similar to $\mathcal{L}_{\text{KL}}$, $\mathcal{S}_{\text{KL}}$ is defined as

$$\mathcal{S}_{\text{KL}} = -\frac{1}{K+L}\sum_{j=1}^{K+L} \log\big(w_{ij}\big) - H(\mathbf{u}) \tag{18}$$

where H is the entropy operation.

The reason why this term works in detecting unknown anomalies is twofold. First, after training, the prototypes of normal data and known anomalies are learnt by and stored in the memory. So for a test sample with the same type of normal data or known anomalies, the dot product between its representation and similar memory items tends to be large, leading to several attention weights much bigger than others. But for unknown anomalies, the prototypes of which have not be learnt by memory, dot product of their representations and all the memory items will be smaller, resulting in small gaps between the attention weights after softmax calculation. Therefore, $\mathcal{S}_{\text{KL}}$ of unknown anomalies are smaller than normal data and known anomalies. The second reason is attributed to the constraint of $\mathcal{L}_{\text{KL}}$, which has been elaborated in the last section.

Fig. 3 shows an instance of the distribution of $\mathcal{S}_{\text{KL}}$ of normal data, known anomalies and unknown anomalies. Obviously, $\mathcal{S}_{\text{KL}}$ of unknown anomalies concentrate on an interval of smaller values, showing significant statistical difference from normal data and known anomalies, which are mainly distributed over areas with larger values.

We expect that a higher anomaly score indicates a higher probability of anomaly, so the final anomaly score is given by

$$\mathcal{S} = -(\mathcal{S}_{\text{sum}} + \lambda \mathcal{S}_{\text{KL}}) \tag{19}$$

TABLE I
DETAILS OF DATA SETS

Dataset	Samples	Dims	Anomaly	Ratio	Category
NSL-KDD	148,517	122	77,054	0.05%	Network
Spambase	4061	57	1813	1.30%	Document
Arrhythmia	452	279	66	9.12%	Healthcare
Fraud	284,807	29	492	0.01%	Finance
Backdoor	95,329	196	2329	0.04%	Network
Census	299,285	500	18,568	0.01%	Sociology
InternetAds	1966	1555	368	2.29%	Image
Speech	3686	400	61	1.02%	Linguistics

where λ is a hyperparameter to balance them. We set the default value of λ to 0.1.

IV. EXPERIMENTS

A. Experimental Setup

1) Data Set: For data generated from IoT devices, tabular data is the most common and widespread data type, we choose eight tabular data sets from various IoT applications to evaluate the performance of our method. For example, NSL-KDD is a network intrusion data set captured by traffic monitoring devices. The detailed information of these data sets is exhibited in Table I, where Ratio denotes the ratio of labeled anomalies in training data.

2) Evaluation Metric: We employ two metrics for performance evaluation: 1) area under receiver operating characteristic curve (AUC-ROC) and 2) area under precision-recall curve (AUC-PR). AUC-ROC is the area under the ROC curve, which is drawn according to the false positive rate and true positive rate under different thresholds. PR curve characterizes the relationship of precision and recall, similarly, the area under the PR curve is defined as AUC-PR. Both metrics vary from 0-1, and a value closer to 1 means a better performance.

3) Implementation Details: Referring to [21], we split the whole data set into training set and test set with a ratio of 8:2. The training set consists of $\mathcal{D}_u$ and $\mathcal{D}_l$. In $\mathcal{D}_u$, we first preserve all the normal data. Then, to simulate real-world applications, we randomly add some anomalies, i.e., anomaly contamination, and make them account for 2% of the $\mathcal{D}_u$. As for $\mathcal{D}_l$, we exclusively preserve 30 labeled anomalies, which can provide prior knowledge of true anomalies. For each data set, we normalize the data to 0-1, run ten times and calculate the means as the final result.

The encoder and the decoder are both composed of two linear layers with ReLU activations. The number of memory items in memory module $\mathcal{M}$, i.e., memory size, is 32 for Spambase, 64 for NSLKDD, Fraud, Speech, and 128 for the other data sets. We set the memory size of $\mathcal{M}_u$ equal to $\mathcal{M}_l$, both are half of that of $\mathcal{M}$. The hyperparameter λ_m, λ_{KL} in $\mathcal{L}$ are fixed to 1 and 0.0001, respectively. The margin a in $\mathcal{L}_{\text{rec}}$ is set to 5, while λ in $\mathcal{S}$ is set to 0.1. Our model is optimized by Adam optimizer, which is bound to an exponentially decaying learning rate controller. We train the network on one Tesla V100 GPU on the Ubuntu18.04 system. Our code is implemented based on PyTorch 1.8.1 framework with Python 3.6.

B. Comparison With Existing Methods

We compare our model with seven AD methods: 1) DevNet [21]; 2) Deep SAD [23]; 3) FEAWAD [22]; 4) PReNet [19]; 5) SOEL [41]; 6) Overlap [24]; and 7) MCM [18]. The first six are latest semi-supervised methods, which are implemented in the same setting of our method. The last one is an unsupervised method for tabular data, which represents the state-of-the-art performance under unsupervised setting. As shown in Table II, our proposed method achieves the best AUC-ROC performance on seven of eight data sets. As for AUC-PR performance, our method also ranks first on six data sets. Furthermore, our MMNet surpasses all the competitive methods on both average AUC-ROC and AUC-PR. Compared to the unsupervised methods, we obtain average AUC-ROC and AUC-PR gains of 17% and 24%, respectively. These evaluation results demonstrate the effectiveness of our method.

C. Ablation Study

In our method, soft masking strategy and anomaly scoring strategy are two main innovative components. In this section, we conduct ablation experiments on NSL-KDD data set to show the effectiveness of them and their improvements over existing methods.

1) Impact of Soft Masking Strategy: To analyse the impact of soft masking strategy, we compare our method with two traditional memory-based AD methods, MemAE [27] and MNAD [28], which are without a masking strategy and can be regarded as baselines. Besides, we also replace our soft masking strategy with hard masking strategy for comparison. Recall that soft masking strategy makes every query feature perform addressing operation on all the memory items and then use $\mathcal{L}_{\text{mask}}$ to constrain their attention on different parts of memory items. Differently, hard masking strategy makes query features of unlabeled data perform addressing operation only on M_u, and labeled anomalies only on M_l, with no other constraints imposed.

As shown in Table III, compared with two baselines, our method achieves an absolute AUC-ROC and AUC-PR gain of at least 8% and 14%, which is quite significant. The reason is that masking strategy can obtain a part of memory items recording the patterns of labeled anomalies specifically, which enables the model to leverage the prior information of anomalies, thus improving the model's perception ability to anomalies. Notably, soft masking strategy also outperforms hard masking strategy on both AUC-ROC and AUC-PR, because soft masking strategy can not only learn normal and abnormal memory items separately, but also enlarge the distance of them in latent space, making them more discriminative for distinguishing anomalies from normal data.

2) Impact of Anomaly Scoring Strategy: To analyse the impact of anomaly scoring strategy, we compare our score with the traditional and most used anomaly score, mean squared

TABLE II
COMPARISON OF MMNET WITH THE STATE-OF-THE-ART METHODS, WE CALCULATE AUC-ROC AND AUC-PR ON EIGHT TARBULAR DATA SETS

Dataset	AUC-ROC Performance								AUC-PR Performance							
	MMNet	Overlap	SOEL	PReNet	FEAWAD	DeepSAD	DevNet	MCM	MMNet	Overlap	SOEL	PReNet	FEAWAD	DeepSAD	DevNet	MCM
NSL-KDD	**0.978**	0.845	0.759	0.966	0.955	0.969	0.952	0.855	**0.981**	0.911	0.766	0.969	0.970	0.965	0.946	0.867
Spambase	**0.951**	0.730	0.726	0.921	0.921	0.898	0.890	0.762	**0.917**	0.752	0.655	0.890	0.910	0.890	0.828	0.749
Arrhythmia	**0.911**	0.823	0.670	0.785	0.820	0.762	0.788	0.789	**0.693**	0.563	0.304	0.452	0.586	0.482	0.462	0.578
Fraud	**0.982**	0.836	0.776	0.978	**0.982**	0.946	0.977	0.936	**0.800**	0.180	0.339	0.683	0.692	0.563	0.688	0.514
Backdoor	**0.971**	0.866	0.943	0.951	0.962	0.938	0.970	0.967	0.875	0.504	0.393	0.870	0.773	0.570	**0.879**	0.837
Census	**0.867**	0.433	0.387	0.798	0.669	0.718	0.836	0.758	0.300	0.097	0.049	0.299	0.248	0.184	**0.322**	0.242
InternetAds	**0.945**	0.805	0.764	0.894	0.890	0.893	0.942	0.773	**0.868**	0.719	0.510	0.835	0.819	0.782	0.866	0.748
Speech	0.709	0.573	0.625	0.865	0.633	0.562	**0.879**	0.379	**0.254**	0.126	0.190	0.210	0.071	0.032	0.133	0.029
Average	**0.914**	0.739	0.706	0.895	0.854	0.832	0.904	0.777	**0.711**	0.481	0.401	0.651	0.634	0.558	0.640	0.570

TABLE III
ABLATION STUDY OF SOFT MASKING STRATEGY

Memory Network	MemAE	MNAD	Hard Mask	Soft Mask
AUC-ROC	0.897	0.776	0.950	**0.978**
AUC-PR	0.844	0.821	0.946	**0.981**

TABLE IV
ABLATION STUDY OF ANOMALY SCORING STRATEGY

Anomaly Score	MSE	$\mathcal{S}_{sum}$	$\mathcal{S}_{KL}$	Ours
AUC-ROC	0.947	0.959	0.926	**0.978**
AUC-PR	0.961	0.967	0.862	**0.981**

error (MSE). Given that our score contains two terms, $\mathcal{S}_{\text{sum}}$ and $\mathcal{S}_{\text{KL}}$, we also show the performance when using one of these two terms alone. The results in Table IV indicate that our proposed attention weights-based anomaly score can improve the performance of MSE obviously. The reason is that attention weights are calculated by query features and memory items in latent space, so the information leveraged is more semantic and expressive than that of input space. Combining $\mathcal{S}_{\text{sum}}$ and $\mathcal{S}_{\text{KL}}$ can bring further performance improvement compared to using them alone, because $\mathcal{S}_{\text{sum}}$ can measure how far a sample deviates from normal patterns, $\mathcal{S}_{\text{KL}}$ can highlight unknown anomalies, which are both conducive to improving the final performance.

Furthermore, for better understanding the role of $\mathcal{S}_{\text{sum}}$ and $\mathcal{S}_{\text{KL}}$, we conduct a case study on MVTec AD data set, the number of memory items is set to 6, the first three are used to capture normal patterns and the last three capture abnormal patterns. We randomly select a normal sample, a known anomaly, an unknown anomaly, and visualize the distributions of their attention weights, as shown in Fig. 4. In this case, $\mathcal{S}_{\text{sum}}$ is the sum of the first three weights, it can be seen for both known and unknown anomaly, their first three weights are much smaller than that of the normal sample, because both of them deviate from normal patterns. $\mathcal{S}_{\text{KL}}$ is the KL-divergence between the distribution of the attention weights and the uniform distribution. As shown in the right column, because the patterns of unknown anomalies have not been learnt and stored in any memory items, for the attention weights of the unknown anomaly, there is no weights much bigger or smaller than others, so the distribution is closer to the uniform distribution. Therefore, $\mathcal{S}_{\text{KL}}$ can distinguish unknown anomalies from other samples.

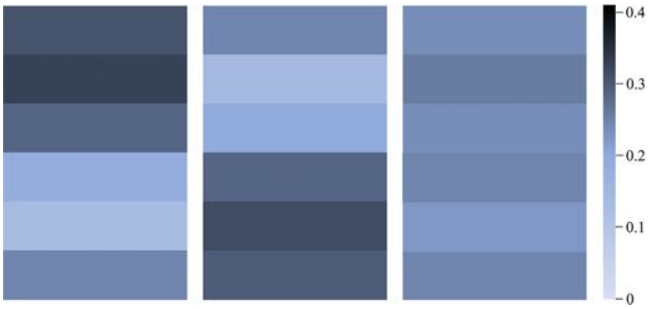

Fig. 4. Distribution of the attention weights of a normal data (left), a known anomaly (middle), and an unknown anomaly (right) from MVTec AD data set.

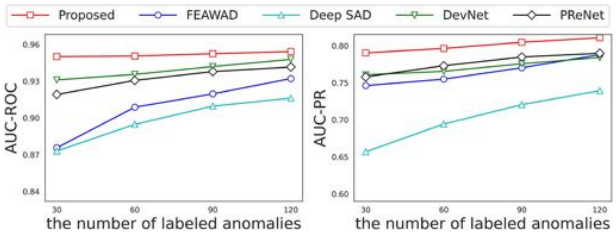

Fig. 5. AUC-ROC and AUC-PR performance with different number of labeled anomalies.

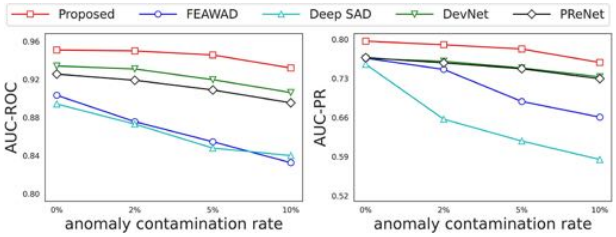

Fig. 6. AUC-ROC and AUC-PR performance with different ratio of anomaly contamination.

D. Sensitivity Study

1) Impact of the Number of Labeled Anomalies: To evaluate the data efficiency of our MMNet, we conduct experiments with different number of labels, i.e., 30, 60, 90, and 120 respectively. Since the total amount of anomalies in Arrhythmia and Speech data set is not enough, we draw the average AUC-ROC and AUC-PR of the other six data sets in

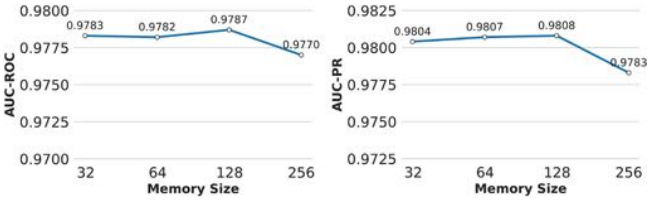

Fig. 7. Sensitivity to memory size on NSL-KDD.

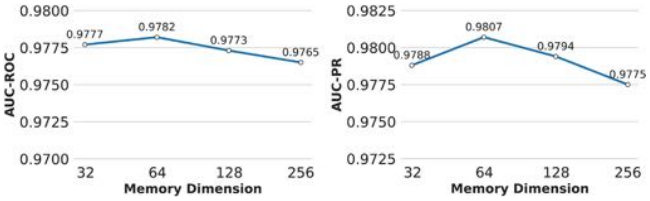

Fig. 8. Sensitivity to memory dimension on NSL-KDD.

Fig. 5. As it is shown, the performance of all the methods are boosted with the increasing numbers of labeled anomalies. Our proposed MMNet performs consistently better than others. It is worth noting that the gap of MMNet and other methods is more obvious when proportion of labeled data is smaller, which proves MMNet can make better use of the limited number of labeled data.

2) Impact of the Ratio of Anomaly Contamination: To analyse the robustness of our model to anomaly contamination, we conduct experiments in the case of no anomaly contamination and anomaly contamination ratio of 2%, 5% and 10% in unlabeled set D_u. Similarly, Arrhythmia and Speech data set are excluded, the average results of other data sets are calculated and drawn in Fig. 6. With the increasing of the ratio of anomaly contamination, the performance of Deep SAD and FEAWAD plummets, other methods also have a slightly drop. MMNet is relatively more stable and consistently shows the best performance.

The reason here is that the number of normal part of memory items is set to be much smaller than the number of unlabeled data, rendering them hard to reconstruct all the unlabeled data well. By doing so, to minimize the overall reconstruction error of unlabeled data as much as possible, the memory items will capture the most common and salient patterns of the input, i.e., patterns of normal data, such that it can reconstruct most samples well. As a result, memory items will not record the patterns of anomalies, which only account for a small proportion of unlabeled data. Therefore, MMNet has a remarkable robustness to anomaly contamination.

3) Sensitivity to Memory Size and Memory Dimension: The memory module in our method is a real-value matrix, each row of which is a memory item. We denote the number of memory items as memory size, and the dimension of each memory item as memory dimension. We conduct an experiments to show the sensitivity of our method to memory size and memory dimension on NSL-KDD data set. As show in Figs. 7 and 8, the performance of MMNet is relatively stable with the varying of memory size and memory dimension. Memory size of 128 and memory dimension of 64 achieve best performance. A lower number will limit the capacity of the model while a higher number will make the model difficult to optimize.

4) Sensitivity to Other Important Hyperparameters: In the design of the loss function for MMNet, three crucial hyperparameters are incorporated: the margin a in the reconstruction loss, as well as the coefficients λ_m and λ_{KL} controlling the weights of mask loss and KL loss. Sensitivity experiments regarding these three hyperparameters are conducted on the NSL-KDD data set, as depicted in Figs. 9–11.

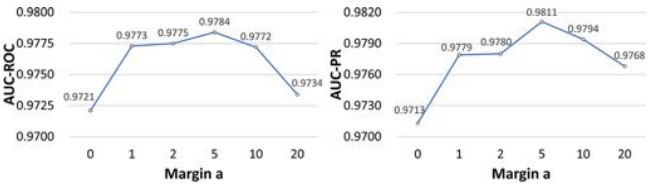

Fig. 9. Sensitivity to margin a on NSL-KDD.

Fig. 10. Sensitivity to λ_m on NSL-KDD.

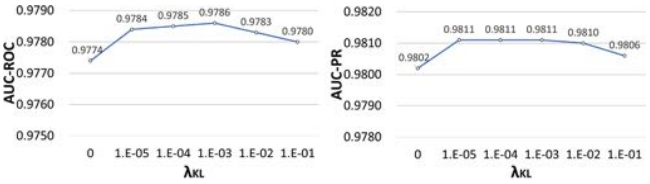

Fig. 11. Sensitivity to λ_{KL} on NSL-KDD.

The margin a controls the deviation of the anomaly scores between normal data and anomalies. As is shown in Fig. 9, $a = 5$ achieves the best performance, a smaller value will reduce the difference of representations between normal and abnormal data, while a larger value will make the model prone to overfitting and limit the generalization ability of the model. λ_m controls the weight of mask loss. Compared to $\lambda_m = 0$, all the other settings have a evident performance boost, which demonstrates the significance of mask loss. When λ_m is above 0, the performance is fairly insensitive to changes of it, so we just set $\lambda_m = 1$. As for λ_{KL}, it governs the balance the KL loss and the other losses. With the increasing of it, the performance first increases and then decreases. In our experiments, we set $\lambda_m = 0.0001$, a smaller value will weaken the role of $\mathcal{S}_{\mathrm{KL}}$, while a larger value will influence the impact of reconstruction loss and mask loss.

E. Performance on Unknown Anomalies

In this section, we evaluate the performance of our model in detecting unknown anomalies. In general, labels of tabular data set in AD exclusively have the information for whether it is abnormal, but lack of precise information for which category of anomalies it is. Therefore, we choose MVTec AD [49], a data set for industrial defect detection,

TABLE V
AUC-ROC AND AUC-PR PERFORMANCE OF UNKNOWN ANOMALIES ON 14 EXPERIMENTS CONSTRUCTED FROM MVTEC AD. EACH TYPE OF ANOMALY WILL BE SELECTED AS KNOWN CATEGORY AND BE ADDED TO THE TRAINING SET IN TURN, TEST SET CONTAINS NORMAL TEST DATA AND ABNORMAL DATA ONLY FROM THE REMAINING ANOMALY TYPES

Object	Known	AUC-ROC Performance							AUC-PR Performance						
		MMNet	**Overlap**	**SOEL**	**PReNet**	**FEAWAD**	**DeepSAD**	**DevNet**	**MMNet**	**Overlap**	**SOEL**	**PReNet**	**FEAWAD**	**DeepSAD**	**DevNet**
Carpet	Color	**0.817**	0.424	0.505	0.428	0.315	0.301	0.574	**0.933**	0.691	0.671	0.641	0.601	0.598	0.777
	Cut	**0.784**	0.627	0.470	0.441	0.537	0.365	0.687	**0.924**	0.807	0.713	0.664	0.744	0.654	0.865
	Hole	**0.791**	0.468	0.604	0.480	0.408	0.340	0.332	**0.928**	0.722	0.777	0.737	0.686	0.647	0.630
	Metal	**0.858**	0.470	0.628	0.446	0.436	0.329	0.550	**0.939**	0.687	0.825	0.646	0.646	0.630	0.710
	Thread	**0.754**	0.688	0.657	0.505	0.546	0.309	0.643	**0.865**	0.799	0.849	0.665	0.739	0.618	0.810
	Average	**0.801**	0.536	0.573	0.460	0.448	0.329	0.557	**0.918**	0.741	0.767	0.670	0.683	0.629	0.758
Capsule	Crack	**0.739**	0.472	0.504	0.456	0.324	0.548	0.472	**0.920**	0.795	0.793	0.762	0.682	0.820	0.762
	Imprint	**0.712**	0.598	0.707	0.523	0.374	0.396	0.629	**0.888**	0.804	0.868	0.795	0.703	0.747	0.833
	Poke	**0.773**	0.594	0.736	0.343	0.370	0.484	0.564	**0.906**	0.837	0.913	0.725	0.734	0.799	0.828
	Scratch	**0.736**	0.603	0.582	0.562	0.381	0.456	0.601	**0.915**	0.815	0.826	0.800	0.710	0.796	0.822
	Squeeze	**0.813**	0.664	0.625	0.571	0.413	0.368	0.560	**0.919**	0.848	0.876	0.808	0.758	0.707	0.805
	Average	**0.753**	0.586	0.631	0.491	0.372	0.450	0.565	**0.908**	0.820	0.855	0.778	0.717	0.774	0.810
Hazelnut	Crack	**0.869**	0.132	0.292	0.089	0.564	0.429	0.106	**0.924**	0.390	0.493	0.378	0.728	0.522	0.382
	Cut	**0.712**	0.042	0.091	0.160	0.176	0.637	0.057	**0.854**	0.373	0.382	0.401	0.458	0.733	0.376
	Hole	**0.702**	0.067	0.138	0.069	0.355	0.511	0.188	**0.832**	0.373	0.403	0.374	0.577	0.603	0.416
	Print	**0.881**	0.705	0.721	0.655	0.871	0.392	0.734	**0.930**	0.776	0.767	0.750	0.924	0.514	0.831
	Average	**0.791**	0.237	0.310	0.243	0.491	0.492	0.271	**0.885**	0.478	0.511	0.476	0.672	0.593	0.501

to verify the detection performance of MMNet on unknown anomalies. MVTec AD consists of images of a variety of objects in industrial production, each objects includes defect-free images and defective images of different types of defects, attached with label information for specific anomaly types.

Referring to the experimental protocol in [42], we select three objects: Carpet (with 5 anomaly category), Capsule (with 5 anomaly category), Hazelnut (with 4 anomaly category), and conduct 14 independent experiments. In each experiment, one type of anomaly is designated as the known anomaly category, while the remaining anomaly categories are considered as unknown anomalies. The training set comprise normal training samples and abnormal samples only from the known anomaly category. During testing, we evaluate the detection performance solely on normal test samples and unknown anomaly categories. To adapt to methods for tabular data, a ResNet18 pretrained on ImageNet is used to obtain a fixed feature vector for each image, which is the input to MMNet and all the compared methods. As shown in Table V, our MMNet achieves the best performance on all the 14 experiments, and outperforms other methods on both average AUC-ROC and average AUC-PR.

Remarkably, we also compare MMNet with the variants without $\mathcal{S}_{\mathrm{KL}}$. With $\mathcal{S}_{\mathrm{KL}}$, the average AUC-ROC is improved from 0.523 to 0.782, while the average AUC-PR is improved from 0.693 to 0.904, obtaining a gain of 30.54%. These results fully demonstrate the effectiveness of $\mathcal{S}_{\mathrm{KL}}$ and MMNet in detecting unknown anomalies.

F. Performance on Different Types of Synthetic Anomalies

Although the types of anomalies are infinite, existing works [20], [50] have summarized four common types of anomalies and proposed methods for generating them. Following the same setup, we conduct experiments to generate synthetic anomalies based on a realistic Fraud data set and assess the performance of MMNet on specific types of synthetic anomalies.

1) *Local anomalies* refer to anomalies that deviate from their local neighborhoods.
2) *Global anomalies* are samples scattered throughout the entire space while being far from the distribution of normal data.
3) *Dependency anomalies* represent samples that do not adhere to the correlation structure among normal data.
4) *Clustered anomalies* refer to groups of data with similar characteristics but significantly different from normal data.

Table VI displays the performance of MMNet and other methods. It can be seen almost all methods perform well on cluster anomalies. It is because clustered anomalies are groups of data with similar characteristics, and they belong to a same type. Therefore, when trained on several such anomalies, all the anomalies in test set are known anomalies, so most methods can perform well due to their ability in detecting known anomalies. Differently, for local, global, and dependency anomalies, the characteristics of different anomalies can be disparate and the types of anomalies are countless, so most anomalies in test set are unknown anomalies. In this case, the performance of other methods drops significantly, because they bias toward known anomalies and can not detect unknown anomalies. MMNet outperforms other methods by a large margin owing to its superior generalization ability to unknown anomalies.

V. CONCLUSION

In this article, we propose MMNet for semi-supervised AD in IoT. To learn discriminative normal and abnormal

TABLE VI
RESULTS OF OUR METHOD AND BASELINES ON DIFFERENT TYPES OF SYNTHETIC ANOMALIES. WE INITIALLY BUILT A DISTRIBUTION BY FITTING IT TO THE TRAINING DATA OF THE FRAUD DATA SET. NORMAL SAMPLES GENERATED ARE DIRECTLY SAMPLED FROM THIS DISTRIBUTION, WHILE ABNORMAL SAMPLES GENERATED ARE OBTAINED FROM DISTRIBUTIONS THAT MAKE SPECIFIC ALTERATIONS FOR CERTAIN ANOMALY TYPES

	Metrics	MMNet	Overlap	SOEL	PReNet	FEAWAD	DeepSAD	DevNet	MCM	PReNet	Overlap	MCM
Local	AUC-ROC	0.9737	0.5519	0.9345	0.5243	**0.9753**	0.4521	0.5670	0.9127	0.5243	0.5519	0.8065
	AUC-PR	**0.5040**	0.0171	0.2223	0.0178	0.0402	0.0846	0.0083	0.4978	0.0178	0.0171	0.2167
Cluster	AUC-ROC	**1.0000**	0.9958	1.0000	1.0000	1.0000	1.0000	1.0000	0.9463	1.0000	0.9958	0.9463
	AUC-PR	**1.0000**	0.9950	0.9975	1.0000	1.0000	1.0000	1.0000	0.9170	1.0000	0.9950	0.9170
Dependency	AUC-ROC	**1.0000**	0.7483	0.9990	0.9790	0.7267	0.9925	0.1522	0.9048	0.9790	0.7483	0.5471
	AUC-PR	**1.0000**	0.0048	0.5000	0.0455	0.0036	0.4087	0.0012	0.0146	0.0455	0.0048	0.0043
Global	AUC-ROC	**0.9955**	0.1002	0.9248	0.5183	0.8900	0.8903	0.1898	0.9623	0.5183	0.1002	0.9344
	AUC-PR	0.4217	0.0015	0.0323	0.0746	0.0087	0.1634	0.0012	**0.9617**	0.0746	0.0015	0.6310

patterns, we design a soft masking strategy, which encourages the memory to capture prototypical patterns of normal data and known anomalies separately, and increases the distance between normal and abnormal memory items in latent space. In addition, we propose a novel scoring strategy to detect both known anomalies and unknown anomalies by leveraging the characteristics of attention weights between test samples and memory items. Extensive experimental results on data sets from various IoT applications demonstrate the effectiveness of our method. In the future, we will dynamically update existing memory parts using corresponding types of data encountered in real-world applications. Moreover, we will also consider methods for dynamically add new memory parts for recording data from significantly different distributions. These changes will make MMNet more suitable for online learning setting, which could bolster the method's applicability across a broader spectrum of IoT applications.

REFERENCES

[1] T. K. Rodrigues, S. Verma, Y. Kawamoto, N. Kato, M. M. Fouda, and M. Ismail, "Smart handover with predicted user behavior using convolutional neural networks for WiGig systems," *IEEE Netw.*, early access, Jan. 12, 2024, doi: 10.1109/MNET.2024.3353301.

[2] F. K. Shaikh, S. Karim, S. Zeadally, and J. Nebhen, "Recent trends in internet-of-things-enabled sensor technologies for smart agriculture," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23583–23598, Dec. 2022.

[3] S. Verma, Y. Kawamoto, and N. Kato, "A network-aware Internet-wide scan for security maximization of IPV6-enabled WLAN IoT devices," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8411–8422, May 2021.

[4] S. Verma, Y. Kawamoto, and N. Kato, "A smart Internet-wide port scan approach for improving IoT security under dynamic WLAN environments," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11951–11961, Jul. 2022.

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2000, pp. 93–104.

[6] D. M. Tax and R. P. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, pp. 45–66, 2004.

[7] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Min.*, 2008, pp. 413–422.

[8] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," in *Proc. Poster Demo Track*, vol. 1, 2012, pp. 59–63.

[9] M. Pavlidou and G. Zioutas, "Kernel density outlier detector," in *Proc. 1st Conf. Int. Soc. Nonparametric Statist.*, 2014, pp. 241–250.

[10] L. Ruff et al., "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.

[11] B. Zong et al., "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–19.

[12] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, "ECOD: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12181–12193, Dec. 2023.

[13] S. Wang et al., "Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–14.

[14] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: Novelty detection via contrastive learning on distributionally shifted instances," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 11839–11852.

[15] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–12.

[16] C. Qiu, T. Pfrommer, M. Kloft, S. Mandt, and M. Rudolph, "Neural transformation learning for deep anomaly detection beyond images," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8703–8714.

[17] Y. Chen, Y. Tian, G. Pang, and G. Carneiro, "Deep one-class classification via interpolated Gaussian descriptor," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, 2022, pp. 383–392.

[18] J. Yin, Y. Qiao, Z. Zhou, X. Wang, and J. Yang, "MCM: Masked cell modeling for anomaly detection in tabular data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024, pp. 1–23.

[19] G. Pang, C. Shen, H. Jin, and A. van den Hengel, "Deep weakly-supervised anomaly detection," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2023, pp. 1795–1807.

[20] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "Adbench: Anomaly detection benchmark," 2022, *arXiv:2206.09426*.

[21] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019, pp. 353–362.

[22] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, and L. Liu, "Feature encoding with autoencoders for weakly supervised anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2454–2465, Jun. 2022.

[23] L. Ruff et al., "Deep semi-supervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–23.

[24] M. Jiang, S. Han, and H. Huang, "Anomaly detection with score distribution discrimination," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2023, pp. 984–996.

[25] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.

[26] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[27] D. Gong et al., "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1705–1714.

[28] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14372–14381.

[29] F. Ye, C. Huang, J. Cao, M. Li, Y. Zhang, and C. Lu, "Attribute restoration framework for anomaly detection," *IEEE Trans. Multimedia*, vol. 24, pp. 116–127, Dec. 2022, doi: 10.1109/TMM.2020.3046884.
[30] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, "A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13588–13597.
[31] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*.
[32] L. Ruff et al., "A unifying review of deep and shallow anomaly detection," *Proc. IEEE*, vol. 109, no. 5, pp. 756–795, May 2021.
[33] H. Hojjati, T. K. K. Ho, and N. Armanfard, "Self-supervised anomaly detection in computer vision and beyond: A survey and outlook," *Neural Netw.*, vol. 172, Apr. 2024, Art. no. 106106.
[34] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based GMM," in *Proc. SIAM Int. Conf. Data Min.*, 2009, pp. 145–154.
[35] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2000, pp. 427–438.
[36] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
[37] C. O'Reilly, A. Gluhak, and M. A. Imran, "Distributed anomaly detection using minimum volume elliptical principal component analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2320–2333, Sep. 2016.
[38] W. Liu, H. Chang, B. Ma, S. Shan, and X. Chen, "Diversity-measurable anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2023, pp. 12147–12156.
[39] R. Li, Q. Li, J. Zhou, and Y. Jiang, "ADRIoT: An edge-assisted anomaly detection framework against IoT-based network attacks," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 10576–10587, Jul. 2022.
[40] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, "Learning graph structures with transformer for multivariate time-series anomaly detection in IoT," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9179–9189, Jun. 2022.
[41] A. Li, C. Qiu, M. Kloft, P. Smyth, S. Mandt, and M. Rudolph, "Deep anomaly detection under labeling budget constraints," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 19882–19910.
[42] C. Ding, G. Pang, and C. Shen, "Catching both gray and black swans: Open-set supervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7388–7398.
[43] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–11.
[44] J. Hou, Y. Zhang, Q. Zhong, D. Xie, S. Pu, and H. Zhou, "Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8791–8800.
[45] R. Cai, H. Zhang, W. Liu, S. Gao, and Z. Hao, "Appearance-motion memory consistency network for video anomaly detection," in *Proc. AAAI*, 2021, pp. 938–946.
[46] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14318–14328.
[47] T. Xiang et al., "SQUID: Deep feature in-painting for unsupervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23890–23901.
[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Stat.*, 2010, pp. 249–256.
[49] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9592–9600.
[50] G. Steinbuss and K. Böhm, "Benchmarking unsupervised outlier detection with realistic synthetic data," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 4, pp. 1–20, 2021.

Jiaxin Yin received the bachelor's degree from the School of Information and Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Artificial Intelligence, supervised by Prof. Jie Yang.

His research topic contains anomaly detection, out-of-distribution detection, and self-supervised learning.

Yuanyuan Qiao (Member, IEEE) received the B.E. degree from Xidian University, Xi'an, China, in 2009, and the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014.

From September 2019 to October 2020, she was a Visiting Scholar with the Senseable City Laboratory, MIT, Cambridge, MA, USA. She is currently a Professor with the School of Artificial Intelligence, BUPT. Her research focuses on multimodal anomaly detection and data mining.

Zunkai Dai received the B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China in 2022, where he is currently pursuing the Ph.D. degree with the School of Artificial Intelligence.

His current research interests include anomaly detection and big data.

Zitang Zhou received the B.E. degree in international school from Beijing University of Posts and Telecommunications, Beijing, China, in 2023, where she is currently pursuing the M.E. degree with the School of Artificial Intelligence, supervised by Prof. Yuanyuan Qiao.

Her research topic contains anomaly detection, out-of-distribution detection, and self-supervised learning.

Xiangchao Wang received the B.E. degree from the School of Electronic Information, Hangzhou Dianzi University, Hangzhou, China.

His research topics include anomaly detection and self-supervised learning.

Wenhui Lin received the B.E., M.E., and Ph.D. degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 2006, 2009, and 2014, respectively.

He is currently a Chief Engineer and Researcher with Aisino Corporation, Beijing. His current research interests include traffic measurement and classification, cloud computing, and big data analytics.

Jie Yang received the B.E., M.E., and Ph.D. degrees from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1993, 1999, and 2007, respectively.

She is currently a Professor with the School of Artificial Intelligence, BUPT and the Director of the Teaching and Research Center of Intelligent Perception and Computing. Her research focuses on deep learning-based big data analytics.