MSS-PAE: Saving Autoencoder-based Outlier Detection from Unexpected Reconstruction

Xu Tan[a], Jiawei Yang[b], Junqi Chen[a], Sylwan Rahardja[c], Susanto Rahardja[a,d,*]

[a]*School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, P.R.China*
[b]*Faculty of Technology, University of Turku, Turku, Finland*
[c]*School of Computing, University of Eastern Finland, Joensuu, Finland*
[d]*Engineering Cluster, Singapore Institute of Technology, Singapore*

Abstract

The Autoencoder (AE) is popular in Outlier Detection (OD) now due to their strong modeling ability. However, AE-based OD methods face the unexpected reconstruction problem: outliers are reconstructed with low errors, impeding their distinction from inliers. This stems from two aspects. First, AE may overconfidently produce good reconstructions in regions where outliers or potential outliers exist while using the mean squared error. To address this, the aleatoric uncertainty was introduced to construct the Probabilistic Autoencoder (PAE), and the Weighted Negative Log-Likelihood (WNLL) was proposed to enlarge the score disparity between inliers and outliers. Second, AE focuses on global modeling yet lacks the perception of local information. Therefore, the Mean-Shift Scoring (MSS) method was proposed to utilize the local relationship of data to reduce the false inliers caused by AE. Moreover, experiments on 32 real-world OD datasets proved the effectiveness of the proposed methods. The combination of WNLL and MSS achieved 45% relative performance improvement compared to the best baseline. In addition, MSS improved the detection performance of multiple AE-based outlier detectors by an average of 20%. The proposed methods have the potential to advance AE's development in OD.

Keywords:

*Corresponding author
Email addresses: xutan@ieee.org (Xu Tan), jiaweiyang@ieee.org (Jiawei Yang), jqchen@ieee.org (Junqi Chen), sylwanrahardja@ieee.org (Sylwan Rahardja), susantorahardja@ieee.org (Susanto Rahardja)

Outlier detection, Autoencoder, Uncertainty estimation, Mean-shift

1. Introduction

Outlier Detection (OD), also known as anomaly detection, is a fundamental technique in the field of data mining. An outlier is a data point that stands out from the rest of the dataset by exhibiting significantly different characteristics or behavior. The objective of OD is to identify the outliers from the normal samples (inliers), for the purpose of keeping the data clean and safe, or identifying the abnormal situations and behaviors. OD has many applications, such as fraud detection [1], video events detection [2], and abnormal trajectory analysis [3]. Generally, OD was treated as an unsupervised or semi-supervised task, since the ground-truth labels were hard to acquire in real-world applications [4]—which means most of the data consists of unidentified inliers and outliers.

Numerous OD methods have been proposed over the years. They could be categorized into different classes. Proximity-based methods [5] measured the similarities between inliers and outliers in the original data space. Statistical-based methods [6] modeled the statistic distribution of inliers. Classification-based methods [7] transformed the data into the substitute space, then identified outliers by dividing the substitute space. Ensemble-based methods [8, 9] combined several weak outlier detectors to produce a strong detector by utilizing the stochasticity and diversity of data. Probabilistic-based methods [10, 11] analyzed the probabilistic distribution of the attributes of the data.

Traditional OD methods usually encountered problems when facing high data dimensionality and complex data manifolds, due to their weak feature extraction abilities [12]. Thus in recent years, deep learning techniques with strong learning abilities piqued the curiosity of experts in OD, and the Autoencoder-based (AE-based) [13, 14, 15] methods became the most popular methods among them. AE-based methods utilized the strong learning abilities of the neural network to capture the inherent feature of the inliers, with the assumption that the inliers could be reconstructed better than the outliers.

Despite achieving remarkable performance in numerous applications, AE-based OD methods faced a common challenge: unexpected reconstruction, which significantly impacted their detection accuracy. Specifically, after training, certain outliers or potential outliers could be well reconstructed by AE, an undesirable outcome in OD tasks. After a thorough investigation, we found that this issue arose from two primary aspects.

Firstly, most AE-based OD methods utilized the Mean Squared Error (MSE) as the reconstruction error function. Optimizing MSE can be interpreted as a special case of maximizing likelihood with an underlying Gaussian error model. It assumes the variance term of the log-likelihood function to be 1, independent of the data instances or attributes. However, this assumption hinders AE from accurately modeling the true distribution of the dataset. Consequently, AE may produce good reconstructions in regions where outliers exist or even in areas devoid of the data. These overconfident yet unexpected reconstructions impede the detection of both existing outliers and potential outliers. To address this issue, this work investigates the impact of aleatoric uncertainty on AE's reconstruction results. Specifically, we modify conventional AE to the Probabilistic Autoencoder (PAE). Then we elucidate how the Negative Log-Likelihood (NLL) function mitigates the overconfidence problem, and further propose the *Weighted Negative Log-Likelihood (WNLL)* which enhances outlier discrimination and improves OD performance across various applications.

Secondly, while AE is a powerful tool for learning global data features, it fails to account for local relationships. Some outliers may share common characteristics with inliers, resulting in their accurate reconstructions. However, these outliers can be distinguished by incorporating local information. Therefore, in this work, we integrate the local relationships of data to refine AE's scoring strategy, introducing the *Mean-Shift outlier Scoring (MSS)* method. This method employed the mean-shifted data point instead of the original data point to calculate the outlier scores. Thus, when an outlier is reconstructed with a relatively high likelihood, the reconstruction result will not resemble the mean-shifted result of the input, leading to a high outlier score. This method is convenient and effective, and can be easily applied to most AE-based methods as a plug-and-play module.

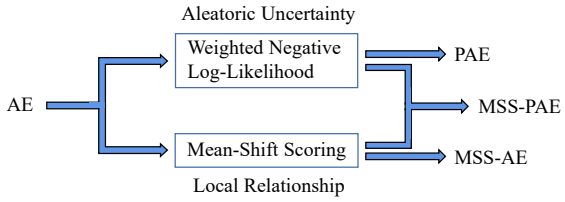

Figure 1: An illustration of how this work will improve AE-based OD methods. WNLL incorporates the aleatoric uncertainty to alleviate the overconfidence issue of AE, and MSS introduces the local relationship to detect well-reconstructed outliers.

An illustration of how this work will improve AE-based OD methods is shown in Figure 1. WNLL factors in aleatoric uncertainty to alleviate the overconfidence issue of AE, and MSS introduces the local relationship to detect well-reconstructed outliers.

In summary, the main contributions of this work are as follows:

- Theoretically analyzing the overconfidence issue in AE, and interpreting the effect of aleatoric uncertainty for multiple scenarios of AE-based OD.
- Proposing WNLL, which mitigates the overconfidence issue for AE-based OD , and makes the scores of outliers more distinguishable. It improves OD performance for different applications by balancing the effect of aleatoric uncertainty.
- Proposing MSS, which introduces local relationships to reduce the harm of unexpected well-reconstructed outliers in AE, thus improving the robustness and accuracy. In addition to its efficacy, it can be easily applied to other AE-based OD methods.
- Conducting experiments on 32 real-world OD datasets, comparing the proposed methods with 5 typical AE-based OD methods and 8 non-AE-based state-of-the-art (SOTA) OD methods. The experimental results proved the effectiveness and superiority of our methods.

The rest of the paper is organized as follows: In Section 2, AE-based OD methods proposed in recent years and studies of quantifying uncertainty in neural networks are reviewed. In Section 3, the principle of AE-based OD and the progress of the mean-shift technique are introduced. In Section 4, the proposed WNLL and MSS methods

are introduced in detail, and their theoretical advantages are analyzed. Then the experimental results including the case studies and empirical evaluations are reported in Section 5. Finally, conclusions are drawn in Section 6.

2. Related Works

2.1. AE-based OD methods

AE plays a crucial role in unsupervised learning-based OD methods. In addition to applying AE in various OD applications, some researchers have focused on enhancing the performance of AE itself to better align with the requirements of OD. Zhou *et al.* proposed the Robust Deep Autoencoders (RDA) [16], motivated by the robust principal component analysis. RDA proposed a new loss function that can iteratively remove anomalous components from the training set, reducing AE contamination. Chen *et al.* proposed the Randomized Neural Network for Outlier Detection (RandNet) [17] to improve the robustness of AE by training multiple AEs with random architectures on different subsets of training data. An *et al.* proposed a Variational-Autoencoder-based (VAE-based) OD method [14]. It used the variational inference and reconstruction probability to get more principled and objective outlier scores. Ishii *et al.* proposed the Low-Cost Autoencoder (LCAE) [18]. It only used data with low reconstruction error in each training epoch, reducing the contamination of the outliers during training. Gong *et al.* proposed a Memory-augmented Autoencoder (MemAE) [19], which improved the robustness of the AE by reconstructing samples from a limited number of recorded representative normal patterns. Lai *et al.* proposed an AE model based on the Robust Subspace Recovery layer (RSRAE) [20]. It used the robust subspace recovery layer to increase the reconstruction difficulty of outliers. Similarly, Yu *et al.* proposed an AE model based on the orthogonal projection constraints (OPCAE)[21], which used the Kautlr-Thomas transformation to preserve only inlier information after encoding. Guo *et al.* proposed an AE architecture called Feature Decomposition Autoencoder (FDAE) [15], which integrated the benefits of RDA and RSRAE.

In summary, most of the recent AE-based methods were dedicated to protecting AE against the detrimental effects of outliers during training. However, they mostly

have complicated network architectures and training procedures. This makes them difficult to use and thus achieve less satisfactory performance in practice. Additionally, they did not realize that AE sometimes makes overconfident yet unexpected decisions in unintended regions, and they did not utilize local information to complement the deficiencies arising from AE's global modeling, thus still leading to the occurrence of the false inliers.

2.2. Quantify uncertainty in neural networks

Neural networks have strong learning abilities, but they can produce overconfident decisions, potentially leading to serious consequences in high-risk applications like autonomous driving and medical diagnosis [22]. While better models and data can help, it's impractical to account for every scenario [23]. Thus, uncertainty quantification is crucial for neural networks to *know what they know* [22].

Uncertainty in neural networks is typically classified into aleatoric (data uncertainty) and epistemic (model uncertainty) [24]. Aleatoric uncertainty represents the noise or randomness inherent in observational data, and can be heteroscedastic or homoscedastic depending on whether the noise is data instance-dependent [23, 24]. Epistemic uncertainty reflects the network's ignorance about whether model parameters captured the underlying regularity of data, caused by erroneous training, weak model, or insufficient training data [25]. Different approaches are used to quantify these uncertainties: estimating noise variance for heteroscedastic aleatoric uncertainty [23, 24], while Bayesian neural networks [24] or ensemble techniques [22] are used for epistemic uncertainty.

Researchers have explored utilizing uncertainty quantification to enhance AE-based OD. For aleatoric uncertainty, An *et al.* used reconstruction probability in VAE, noting anomalies have higher uncertainty [14]. Pol *et al.* employed NLL-trained VAE with uncertainty-normalized reconstruction error [26]. Mao *et al.* detected abnormal pixels using NLL-trained AE and error-to-uncertainty ratio. For epistemic uncertainty, Legrand *et al.* used Bayesian AE with Monte-Carlo Dropout [27], combining uncertainty and reconstruction error. Daxberger *et al.* applied Bayesian VAE with MCMC [28], while Park *et al.* trained only the encoder of Bayesian VAE, using Monte-Carlo

Dropout for an integrated outlier score [29].

Although the uncertainty quantification had been applied to previous studies, the intricacies of its effect on the OD task had not been elucidated, especially for OD on general tabular data. It is worth noting that, the implication of uncertainty in various machine learning or OD scenarios may vary. For example, in regression tasks, aleatoric uncertainty reflects the noise in the input and target measurements that networks are unable to learn to correct [30]. But in classification tasks, it reflects the deficiency of information to identify one class of data [25]. Conventional uncertainty studies largely focus on regression, classification, or segmentation. However, the situation for OD tasks with reconstruction models such as AE is different. In this paper, we detailedly analyze the aleatoric uncertainty in AE, and explain its advantages for tabular OD.

3. Preliminary

3.1. Principles of AE-based OD

An AE is a neural network that is trained to reproduce its input to its output. It is comprised of two main components: the encoder and the decoder. Considering a traditional fully-connected AE, given a multivariate data vector $\mathbf{x} \in \mathbb{R}^D$ as the input, the encoder transforms it to a lower-dimensional latent representation, which retains the most critical features of the input data. The latent representation $\mathbf{z}$ output by the encoder is given by

$$\mathbf{z} = \text{Encoder}(\mathbf{x}, \theta), \tag{1}$$

where θ denotes the network parameters of the encoder. Then, the decoder reconstructs the input using the latent representation, and the output $\hat{\mathbf{x}}$ is expressed as

$$\hat{\mathbf{x}} = \text{Decoder}(\mathbf{z}, \phi), \tag{2}$$

where ϕ denotes the network parameters of the decoder.

Since the fully-connected layer can be viewed as the linear dimensionality reduction transformation, the effect of the encoder is similar to the principle components analysis, which aims to find the main components reflecting the features of the original data. However, complicated real-world data are difficult to analyze linearly, so a

nonlinear activation function is applied after each fully-connected layer to increase the expressing ability of the network. Common typical activation functions are Sigmoid, Tanh, Rectified Linear Unit (ReLU), and ReLU's variants. The optimization and outlier score computation of an AE both rely on its loss function. Since the target of the AE is to reconstruct the input data as the output, the bias between the input and output can be used as the loss function. This is also known as the reconstruction error. The most popular loss function of the conventional AE is the MSE function, which is established by the following equation:

$$\mathrm{MSE}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2, \tag{3}$$

where $\|\cdot\|_2$ denotes the L2-norm.

AE first transforms the input $\mathbf{x}$ to the latent $\mathbf{z}$, then transforms $\mathbf{z}$ to the output $\hat{\mathbf{x}}$. The dimension of the latent presentation $\mathbf{z}$ is designed to be smaller than the input layer. It served as a bottleneck, allowing only critical information to pass through. Thus, the AE is forced to capture the key features of the data to better reconstruct the input.

AE has found widespread applications across various domains of data mining [31, 32]. Unlike other fields that prioritize high reconstruction quality, OD emphasizes the disparity in reconstruction outcomes between inliers and outliers. Considering a dataset that contains both inliers and outliers. Inliers are abundant and generally show similar patterns, thus their features can easily be learned by an AE. Outliers are few and different in the dataset, hence their features are difficult for an AE to learn. As a consequence, inliers tend to be reconstructed with lower error margins, but outliers tend to have higher reconstruction errors.

3.2. Mean-shift technique

The concept of the mean-shift was first proposed by Fukunaga *et al.* [33], and applied to the estimation of the gradient of a density function. Then it was used in a variety of applications such as image segmentation and target tracking [34].

The process of the mean-shift could be described as follows. Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, mean-shift first calculated each sample's k-nearest-neighbors in $\mathbf{X}$. For example, for the sample $\mathbf{x}_i$, its k-nearest-neighbors and itself were included into

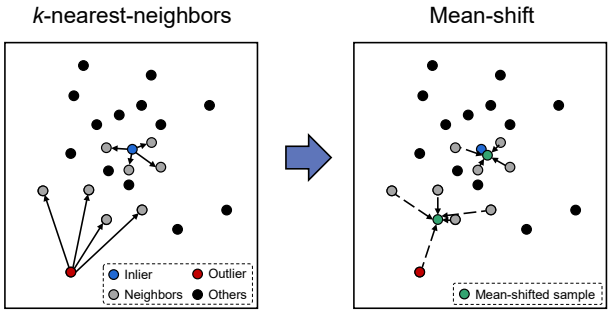

Figure 2: A 2-D illustration of the mean-shift with $m = 1$ and $k = 4$ for an inlier (top) and an outlier (bottom).

Algorithm 1 Mean-shift process

Input: $\mathbf{X}$ - original dataset, k - number of neighbors, m - shift times

Output: Shifted dataset $\mathbf{X}^{\mathrm{MS}(m,k)}$

function MeanShift($\mathbf{X}$, k, m)
  $t \leftarrow 0$, $\mathbf{X}^{\mathrm{MS}(t,k)} \leftarrow \mathbf{X}$.
  repeat
    for $\mathbf{x}_i$ in $\mathbf{X}^{\mathrm{MS}(t,k)}$ **do**
      Find $\mathbf{x}_i$'s k-nearest-neighbors set NL_i in $\mathbf{X}^{\mathrm{MS}(t,k)}$.
      Compute the mean-shifted sample $\mathbf{x}_i^{\mathrm{MS}(t+1,k)}$ according to Eq. 4.
    The mean-shifted dataset $\mathbf{X}^{\mathrm{MS}(t+1,k)} = \{\mathbf{x}_1^{\mathrm{MS}(t+1,k)}, \mathbf{x}_2^{\mathrm{MS}(t+1,k)}, \cdots, \mathbf{x}_N^{\mathrm{MS}(t+1,k)}\}$.
    $t \leftarrow t + 1$
  until $t > m$

one set. Then mean-shift calculated the mean value of the set, and combined all shifted samples as the shifted dataset, as expressed in Eq. 4:

$$\mathbf{x}_i^{\mathrm{MS}} = \frac{1}{|\mathrm{NL}_i|} \sum_{\mathbf{x} \in \mathrm{NL}_i} \mathbf{x}, \tag{4}$$

where $\mathrm{NL}_i = \{\mathbf{x}_i, \mathbf{x}_i^1, \mathbf{x}_i^2, \cdots, \mathbf{x}_i^k\}$ denotes the neighbor set, and $\mathbf{X}^{\mathrm{MS}}$ denotes the shifted dataset. It is worth noting that, this process can be repeated several times to get a more compact data distribution. The complete procedure of mean-shifting a dataset for m times is shown in Algorithm 1. In addition, a 2-D illustration of the mean-shift is shown in Figure 2.

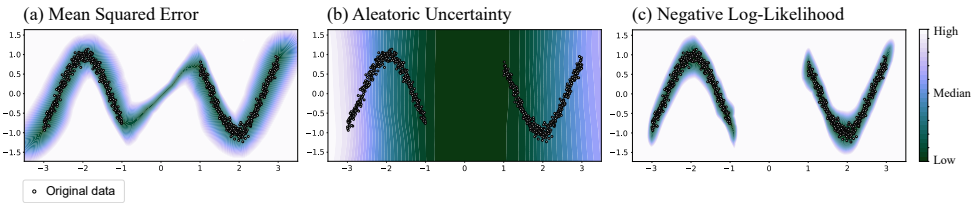

Figure 3: A 2-dimensional synthesized dataset containing two groups of sinusoidal-shaped data with equal density. (a) showed that AE was overconfident in making accurate reconstructions for instances outside the training data regions. (b) showed that the aleatoric uncertainty was low between two manifolds because of the lack of data. (c) showed that PAE with NLL could better capture the original shape of the dataset.

4. Methodology

In this section, we offered insights into the two reasons why conventional AE-based methods suffer from the unexpected reconstruction problem that lead to unsatisfied outlier detection performance. Moreover, we proposed two novel methods to address the issue effectively.

4.1. Addressing overconfidence and increasing score distinguishability with WNLL

AE-based OD methods consider instances with high reconstruction errors as outliers, and among all error functions, the MSE is the most widely used. However, we discovered that MSE can cause an overconfidence issue, which leads to disastrous results in certain OD scenarios. As illustrated in Figure 3.(a), we trained an AE using a set of 2-dimensional data, and the heatmap reflects its MSE in different regions of the data space. It can be observed that the MSE was low between the two groups of data, indicating that the AE was overconfident in making accurate reconstructions for instances outside the training data regions. When outliers appear in these areas, they are easily misclassified as inliers. This is because AE utilizes knowledge learned from the training set to infer information about data outside the training set. Although these inferences may not be accurate, we cannot distinguish them using MSE alone. These unexpected reconstructions can lead to poor OD performance if potential outliers lie in this region.

To further explore the reason for this issue, we need to revisit the optimization of

AE. The optimization target of AE can be interpreted as maximizing the reconstruction likelihood with certain underlying probability distributions according to the data noise or randomness. Assuming the distribution is the diagonal multivariate Gaussian distribution, Eq. 5 is established:

$$\begin{cases} \mathbf{x}^{\text{rec}} \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \\ \boldsymbol{\mu} = [\mu_1, \mu_2, \cdots, \mu_D]^T, \\ \boldsymbol{\sigma}^2 = [\sigma_1^2, \sigma_2^2, \cdots, \sigma_D^2]^T, \\ \boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma}^2), \end{cases} \tag{5}$$

where $\mathbf{x}^{\text{rec}}$ denotes the reconstruction random variable of input $\mathbf{x}$; $\mathrm{diag}(\cdot)$ denotes the diagonal matrix; $\boldsymbol{\mu}$ denotes the reconstruction mean of AE; $\boldsymbol{\sigma}^2$ denotes the reconstruction variance. This gives rise to Eq. 6:

$$P(\mathbf{x}^{\text{rec}}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}^{\text{rec}}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{\text{rec}}-\boldsymbol{\mu})}, \tag{6}$$

where $P(\cdot)$ denotes the probability density function, $|\boldsymbol{\Sigma}|$ denotes the determinant of the matrix $\boldsymbol{\Sigma}$, and $\boldsymbol{\Sigma}^{-1}$ denotes the inverse matrix of $\boldsymbol{\Sigma}$. The logarithmic probability density function may be expressed as Eq. 7:

$$\ln P(\mathbf{x}^{\text{rec}}) = -\frac{D}{2} \ln 2\pi - \frac{1}{2} \sum_{d=1}^{D} \ln \sigma_d^2 - \frac{1}{2} \sum_{d=1}^{D} \frac{(x_d^{\text{rec}} - \mu_d)^2}{\sigma_d^2}. \tag{7}$$

If all variance values are assumed to be 1, then

$$\begin{aligned} \sigma_1^2 = \sigma_2^2 = \cdots = \sigma_D^2 = 1, \\ \boldsymbol{\Sigma} = \mathbf{I}, \end{aligned} \tag{8}$$

where $\mathbf{I}$ denotes the identity matrix. Therefore, the logarithmic probability density function, also known as the log-likelihood of $\mathbf{x}^{\text{rec}}$ can be expressed by Eq. 9:

$$\begin{aligned} \ln P(\mathbf{x}^{\text{rec}}) &= -\frac{D}{2} \ln 2\pi - \frac{1}{2} \sum_{d=1}^{D} (x_d^{\text{rec}} - \mu_d)^2 \\ &= C - \frac{1}{2} \|\mathbf{x}^{\text{rec}} - \boldsymbol{\mu}\|_2^2, \end{aligned} \tag{9}$$

where C denotes a constant which will not affect the optimization process.

Since AE aims to reconstruct the input $\mathbf{x}$, which implies maximizing the log-likelihood $\ln P(\mathbf{x}^{\text{rec}})$ when $\mathbf{x}^{\text{rec}} = \mathbf{x}$, and the mean value $\boldsymbol{\mu}$ is treated as the output $\hat{\mathbf{x}}$

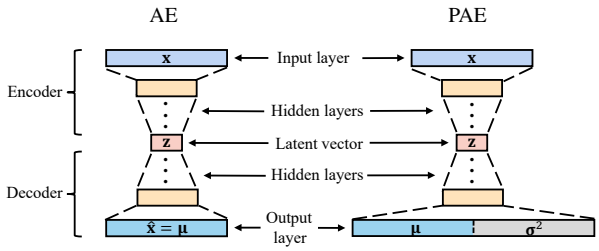

Figure 4: The typical structure of the AE and PAE. AE has a symmetrical structure and the size of the latent vector is much smaller than the input layer, while PAE has an output layer twice the size of its input layer.

of AE. Therefore, the optimization objective is to maximize $C - \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$. Evidently, it is equivalent to minimizing $\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$, which is exactly the form of general MSE. So given the training set $\mathbf{X} \in \mathbb{R}^{N \times D}$, the loss function of the conventional AE can be represented by Eq. 10:

$$\mathrm{MSE}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2, \tag{10}$$

where N denotes the number of samples in the training set.

From the derivation above it can be realized that, using the MSE loss is under the assumption that the variance values $\boldsymbol{\sigma}^2$ of all the reconstruction probability distributions are 1, which implies that the randomness of each data instance and attribute is constant and independent (i.e., homoscedastic). Hence this assumption is unrealistic and will cause the overconfidence issue, as noted in Figure 3.(a).

Consequently, it is necessary to force AE to learn the variance $\boldsymbol{\sigma}^2$ simultaneously, which elicits the NLL loss:

$$\mathrm{NLL}(\mathbf{X}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{d=1}^{D} \frac{(x_{i,d} - \mu_{i,d})^2}{\sigma_{i,d}^2} + \ln \sigma_{i,d}^2. \tag{11}$$

where $x_{i,d}$, $\mu_{i,d}$ and $\sigma_{i,d}^2$ denote the d-th attribute of the sample $\mathbf{x}_i$, output mean values $\boldsymbol{\mu}_i$, and output variance values $\boldsymbol{\sigma}_i^2$, respectively. It is a simple deformation of Eq. 7, after the elimination of the constant term, coefficients, and negation. Since the computation of NLL required two variables, $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i^2$, it required a network that has two outputs correspondingly. Therefore, the conventional AE was modified to the *Probabilistic Autoencoder (PAE)* as shown in Figure 4. The dimension of the output layer of PAE

is twice the size of the input layer, which consists of $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i^2$. A Softplus activation function is followed by the output variance to ensure positive values. $\boldsymbol{\mu}_i$ represents the reconstruction result. $\boldsymbol{\sigma}_i^2$ can be termed as heteroscedastic aleatoric uncertainty.

After training, the reconstruction results will fit the manifold of training data. At the same time, the estimated aleatoric uncertainty will reflect the quality of the training set, such as the randomness or noise inherent in the data. for example, looking back on the example in Figure 3.(b), the aleatoric uncertainty became quite low between the two manifolds. This was because this region did not even exist data, let alone randomness or noise. The low uncertainty made the first term of NLL quite large, leading to the result of Figure 3.(c), which avoided potential outliers between the two manifolds being misclassified as inliers.

Moreover, looking closely at the formula of NLL, aleatoric uncertainty has opposite impacts on the two terms. Higher aleatoric uncertainty will make the first term smaller, but the second larger. Since the behaviors of the aleatoric uncertainty become different for distinct applications, its effects on OD will also be different. Thus we proposed the weighted negative log-likelihood:

$$\mathrm{WNLL}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{d=1}^{D} \alpha \frac{(x_d - \mu_d)^2}{\sigma_d^2} + (1-\alpha) \ln \sigma_d^2, \tag{12}$$

where x_d, μ_d and σ_d^2 denote the d-th attribute of a sample, $\mathbf{x}$, output mean values, $\boldsymbol{\mu}$, and output variance values, $\boldsymbol{\sigma}^2$, respectively. $\alpha \in [0, 1]$ is a hyper-parameter that controls the trade-off between the two components of WNLL.

The setting of α is relevant to the specific application and the behaviors of the aleatoric uncertainty. When the dataset contains multiple inlier manifolds, the aleatoric uncertainty decreases in the hollow area within the manifolds; thus, α can be set higher to avoid the overconfidence issue. When the dataset is contaminated with outliers, the aleatoric uncertainty increases in the regions where outliers exist; thus, α can be set lower to make the scores of outliers more distinguishable. More detailed cases and analyses will be discussed later in Sec. 5.1.1.

In this work, we use WNLL as the score function in the testing phase. After adopting WNLL, the scores of outliers and potential outliers become much higher than inliers, leading to better detection performance. Additionally, by tuning α, PAE adapts to

the varying characteristics of the datasets. This delivers substantial benefits to various OD applications without extra modification to the training progress.

4.2. Integrating local relationships to reduce false inliers for AE

AE is trained to learn the normal patterns of the data, and outliers are intuitively expected to have higher reconstruction errors. Although the reconstruction error such as MSE can be used as the outlier score directly, the score may be less reliable in certain situations. First, if the training dataset is contaminated by outliers, AE will be misled to learn abnormal patterns of the data, resulting in lower reconstruction errors of outliers and increasing difficulty of distinguishing outliers from inliers. Second, some outliers exhibit latent features conforming to the patterns of inliers, except that their feature values deviate from the normal range. These outliers may be unexpectedly well reconstructed by AE, making them difficult to distinguish from inliers. These well-reconstructed outliers can be termed as the false inliers.

To address this issue, some methods have attempted to constrain the latent feature space using regularization terms or other sub-modules [15, 20, 35]. However, these methods either require the data to have specific properties or are complicated to implement. To overcome these limitations, we propose a new approach that can adapt to various data distributions, is easy to implement, and can be used as a plug-and-play module on any other AE-based OD method.

Specifically, we found the issue raised can be explained by AEs' capturing global features of data, which lacked explicit consideration of local relationships. Some outliers may share common features with inliers, leading to their good reconstruction. However, these outliers can be distinguished by incorporating local information, such as the density. Consequently, we leveraged the local relationships within the data as auxiliary information for AE, employing the mean-shift technique as the tool.

To apply the mean-shift on the AE-based OD method, the following procedure was proposed: During the training phase, AE could be trained using the loss function Eq. 10 or Eq. 11. During the scoring phase, the outlier score of a test sample $\mathbf{x}$ could be

computed using the score function:

$$\text{MSS-MSE}(\mathbf{x}) = \|\mathbf{x}^{\text{MS}(m,k)} - \hat{\mathbf{x}}\|_2^2, \tag{13}$$

or,

$$\text{MSS-WNLL}(\mathbf{x}) = \sum_{d=1}^{D} \alpha \frac{(x_d^{\text{MS}(m,k)} - \mu_d)^2}{\sigma_d^2} + (1-\alpha)\ln\sigma_d^2, \tag{14}$$

where $\mathbf{x}^{\text{MS}(m,k)}$ denotes the mean-shifted result of the test sample $\mathbf{x}$. We called this scoring method the Mean Shift Scoring.

The benefit of MSS can be analyzed from two perspectives. Upon analysis of the distance measurement, the reconstruction error could be viewed as the deviation of the object's position in the original data space. The reconstruction error of the inlier was relatively small if AE was well-trained, which means the reconstructed inlier lies within close proximity to the original position. Meanwhile, the mean-shifted result of an inlier was also close to its original position, hence the distance between the reconstructed inlier and the mean-shifted inlier would be small. For the outliers, the mean-shifted result was generally closer to inliers and further from its original locality. The outliers' reconstruction result deviated in a larger magnitude for distance, and with a random direction, since AE learning was inadequate. Thus, the distance between an outlier's mean-shifted result and reconstruction result would be significantly large compared to an inlier, which made them simple to distinguish. Upon analysis of the reconstruction likelihood, even though an outlier was reconstructed with a relatively high likelihood, its mean-shifted result was not similar to the reconstruction. This resulted in a high outlier score as the mean-shifted result fell into a position with low likelihood within the likelihood distribution.

In theory, usage of the distance between the mean of k-NN instead of the original data and the reconstructed data had two advantages. First, neighboring points in the data space should have similar outlier scores, so using mean-shift could avoid local variance in the outlier score space [36]. Second, the mean values were seen as the representatives of the neighbor area, thus they considered both object-level and group-level factors when scoring [37]. Therefore, it provided a robust way for the detection of both point outliers and collective outliers.

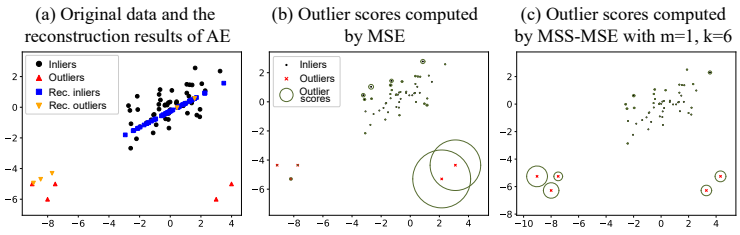

Figure 5: A 2-D example of the effect of the MSS. An AE was trained using these data with the loss function of MSE. (b) and (c) showed the outlier scores generated by MSE and MSS-MSE with $m = 1, k = 6$ respectively. The larger the radius of the green circle, the higher the outlier score.

An example of the effect of MSS is shown in Figure 5. The outliers lying on the bottom left were exactly situated along the feature direction of inliers, thus their reconstruction errors were unexpectedly small, making identification of these outliers from inliers challenging. However, since they were far way from inliers, they could easily be identified after applying MSS with $m = 1$ and $k = 6$. Another example of a real-world case could be found in Sec. 5.1.2

MSS can be integrated with PAE to create MSS-PAE, which combines the advantages of both proposed methods. This integration enhances the ability to eliminate unexpected reconstructions, thereby significantly improving OD performance. An illustration of the training and testing procedure of MSS-PAE is shown in Figure 6. The performance of MSS-PAE was evaluated in Section 5.

5. Experiments

In this section, we first conducted case studies to investigate the effects of the proposed WNLL and MSS. We then performed extensive empirical evaluations on 32 commonly used real-world OD datasets to demonstrate the superiority of the proposed methods compared to the existing SOTA OD methods.

5.1. Case Studies

5.1.1. Evaluation and analysis of WNLL

To reveal how PAE and WNLL benefit OD, two cases were studied via visualization. AE and PAE were trained with the same settings in each case. Since the cases

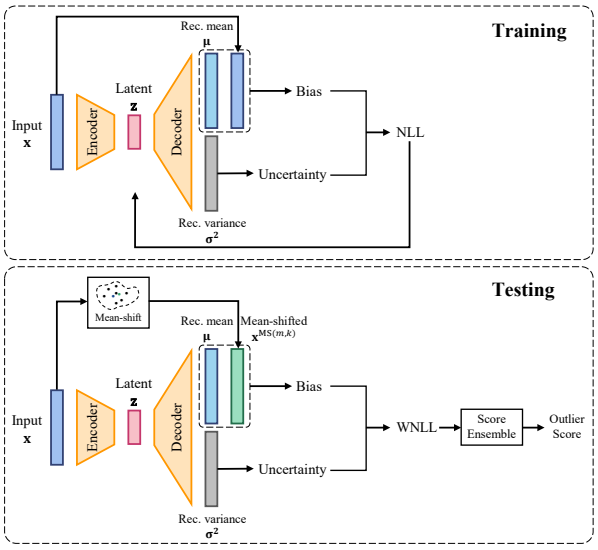

Figure 6: An illustration of the training and testing procedure of MSS-PAE. "Rec." in the figure denotes "Reconstruction".

are 2-dimensional and highly nonlinear, we set the number of units for each network layer for all models as [2,32,32,1,32,32,2], with scaled-exponential-linear-unit active function, except for the last layer. In both cases, the data included a large proportion of inliers and a small proportion of outliers. Besides testing the reconstruction performance of AE and PAE, OD performance with different score functions was also tested. Area Under the Receiver Operating characteristic Curve (AUROC) and Area Under the Precision-Recall Curve (AUPRC) were used as the OD performance metrics. They were both suitable for label-unbalanced classification tasks, and ranged from 0 to 1, where 1 indicates the best performance.

The first case included a large group of semicircular-shaped inliers and a small group of arcuate-shaped outliers, with an overlapping range on the horizontal axis. As shown in Figure 7, we can observe that AE was observed to erroneously fit outliers, resulting in poor OD performance. In contrast, PAE had different results. Its reconstruction results did not deviate significantly from the inliers. However, the overlap of inliers and outliers in parts of the latent space confused PAE, leading to regions with outliers having high aleatoric uncertainty. Therefore, even though PAE wanted to as-

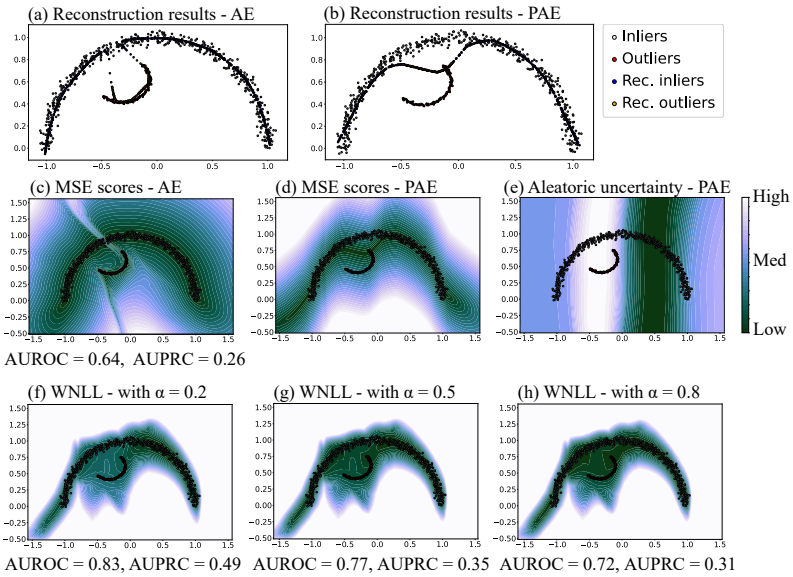

Figure 7: The first case included a large group of semicircular-shaped inliers and a small group of arcuate-shaped outliers, with an overlapping range on the horizontal axis. (a) and (c) showed that AE mistakenly fit the outliers, resulting in bad OD performance. (b) and (d) showed that the reconstruction results of PAE did not deviate a lot from the inliers manifold. (e) showed that PAE outputs high aleatoric uncertainty around outliers, making outliers easier to be discriminated. (f) to (h) showed that PAE achieved better OD performance with smaller α in WNLL.

sign low NLL scores for all data, it could only assign relatively high scores for the regions with a high concentration of outliers. In light of this, though some of the inliers were under suspicion, most outliers could be found out. To better prevent outliers from being erroneously identified as normal, we could decrease α in WNLL, since it improved the positive effect of aleatoric uncertainty. As a result, PAE achieved better OD performance when $\alpha = 0.2$, compared with $\alpha = 0.5$ (equivalent to NLL).

The second case included a large group of sinusoidal-shaped inliers and a small group of arcuate-shaped outliers, with no overlapping on the horizontal axis. It can be observed that, AE erroneously created regions with low MSE scores both around outliers, and between inliers and outliers, as shown in Figure 8.(c). This phenomenon impaired OD performance. In contrast, with PAE, the sparsity and minority of outliers

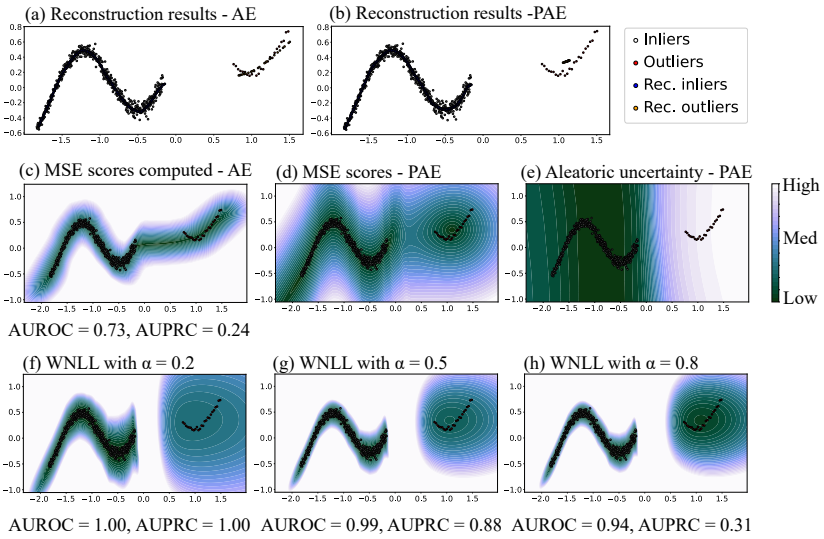

Figure 8: The second case included a large group of sinusoidal-shaped inliers and a small group of arcuate-shaped outliers, with no overlapping on the horizontal axis. (a) and (c) showed that AE reconstructed inliers well, but it output low MSE scores not only around outliers, but also between inliers and outliers. (b) and (d) showed that PAE did not reconstruct outliers well, but it still assigned relatively high MSE scores for some outliers. (e) showed that PAE output high aleatoric uncertainty in both of the aforementioned regions. (f) to (h) showed that PAE achieved better OD performance with smaller α in WNLL.

made PAE uncertain for the information in the region around outliers during training, thus output high aleatoric uncertainty there. For this situation, the positive effect of aleatoric uncertainty was essential for distinguishing, thus WNLL with smaller α yielded better OD performance.

Finally, we summarized the behaviors of aleatoric uncertainty for PAE as follows:

1) it increased in regions where data contained large randomness or noise;
2) it increased in regions where inliers and outliers overlapped in the latent space;
3) it increased in regions where data were sparse and few (most likely outliers);
4) it decreased in the hollow area within the data distribution.

Moreover, we give some recommendations for the usage of WNLL in OD tasks:

1) WNLL is effective in mitigating the overconfidence issue when training data contain multiple inlier manifolds (patterns), and α in WNLL is recommended to set a high

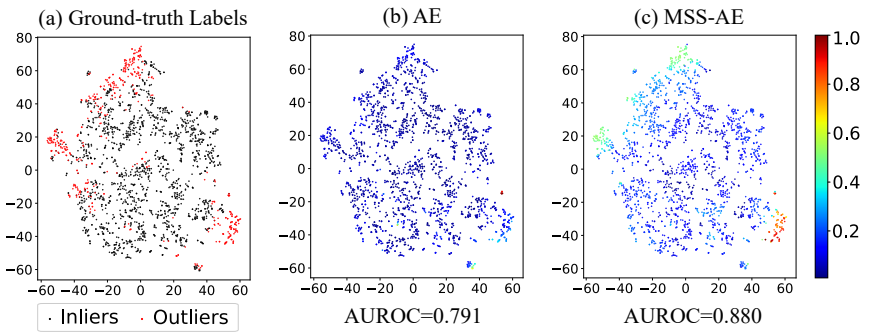

Figure 9: An illustration of the MSS method on the *Cardiotocography* dataset. The data in *Cardiotocography* were dimension-reduced to 2-D utilizing t-SNE. (a) showed the ground-truth label map of the data, in which the black points denote the inliers and the red points denote the outliers. (b) and (c) showed the outlier score map generated by AE and MSS-AE respectively.

value here;

2) WNLL is effective when training data contained outliers, and α in WNLL is recommended to set a low value here.

3) For applications with both multiple inlier patterns and outlier contamination, practitioners should carefully adjust α in WNLL according to the specific analysis.

5.1.2. Demonstration of the effect of MSS

To illustrate the effect of MSS, we studied the case of a real-word dataset *Cardiotocography*, which contained data from patients diagnosed with heart diseases, people suspected of having heart diseases, and normal individuals. Normal individuals were treated as inliers and the remaining as outliers. In this case, MSS was applied to the conventional AE, termed MSS-AE. For better presentation, all the data were dimension-reduced to 2-D utilizing t-distributed stochastic neighbor embedding (t-SNE), as shown in Figure 9.

Figure 9. (a) showed the ground-truth label map of the data, where the black points denoted the inliers and the red points denoted the outliers. The outliers were observed to be distributed in the top left region and bottom right region of the map. Figures. 9. (b) and (c) illustrated the outlier score map generated by AE and MSS-AE respectively. The color scale corresponded to the outlier score of each object. The performance was

evaluated by AUROC. AE could distinguish the part of the outliers lying at the bottom right and achieve an AUROC of 0.791, but could not distinguish the outliers at the top left either. The outlier scores were also not differentiable enough. In contrast, MSS-AE could distinguish both outliers lying at the bottom right and top left. MSS-AE provided more differentiable outlier scores and a significant improvement in AUROC performance to 0.880. In summary, MSS can effectively improve the OD performance of AE by reducing the false inliers generated by the regular scoring function.

5.2. Empirical evaluations

In this work, empirical experiments were conducted on 32 commonly used real-world outlier detection datasets. The proposed methods were compared with 5 typical AE-based OD methods (including AE itself), and proved the efficacy of applying MSS. A comparison was made against 8 classic non-AE-based SOTA OD methods.

The experiments answered the following questions:

1) How effective and flexible is WNLL in real-world scenarios? (Section 5.2.3.A.)

2) How do hyper-parameters k and m affect MSS? (Section 5.2.3.B.)

3) Is MSS beneficial for other AE-based methods? (Section 5.2.3.C.)

4) Comparing the proposed methods with 5 AE-based and 8 non-AE-based OD methods, which one yields the best performance? (Section 5.2.3.C.)

5.2.1. Datasets and baselines

32 real-world outlier detection datasets were obtained from DAMI[1] and ODDS[2] dataset repositories. The summary of these datasets was shown in TABLE 1. We divided each dataset into the training set and test set with a ratio of 3:1. Then, one-third of the samples in the training set were selected to form the auxiliary set. The outlier ratios were kept constant in all three sets, and the auxiliary set and test set included the data labels but the training set did not. In addition, each attribute of the data was normalized to zero-mean and unit-variance using the Mean-SD normalization. The mean value and the variance value were obtained from the training set. This operation

[1]DAMI: www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI

[2]ODDS: odds.cs.stonybrook.edu

Table 1: Information of 32 real-world outlier detection datasets.

Name	Dim	Sample	Ratio	Name	Dim	Sample	Ratio
ALOI	27	49,534	3.0%	Parkinson	22	195	75.4%
Annthyroid	21	7,129	7.5%	PenDigits	16	9,868	0.2%
Arrhythmia	259	450	45.8%	Pima	8	768	34.9%
Breastw	9	683	35.0%	Satellite	36	6,435	31.6%
Cardio.	21	2,114	22.0%	Satimage-2	36	5,803	1.2%
Glass	7	214	4.2%	Shuttle	9	1,013	1.3%
HeartDisease	13	270	44.4%	SpamBase	57	4,207	39.9%
InternetAds	1,555	1,966	18.7%	Speech	400	3,686	1.7%
Ionosphere	32	351	35.9%	Stamps	9	340	9.1%
Letter	32	1,600	6.3%	Thyroid	6	3,772	2.5%
Lympho.	3	148	4.1%	Vertebral	6	240	12.5%
Mammo.	6	11,183	2.3%	Vowels	12	1,456	3.4%
Mnist	100	7,603	9.2%	Waveform	21	3,443	2.9%
Musk	166	3,062	3.2%	Wilt	5	4,819	5.3%
Optdigits	64	5,216	2.9%	Wine	13	129	7.8%
PageBlocks	10	5,393	9.5%	WPBC	33	198	23.7%

was essential for the AE-based methods because it balanced the scale of each attribute of the data, lest the attributes with relatively large scale values had unexpectedly bigger impacts on the model optimization.

To show the superiority of the proposed methods, 5 typical AE-based methods and 8 non-AE-based SOTA OD methods were used as the baselines. The AE-based methods included AE [13], RDA [16], LCAE [18], RSRAE [20] and FDAE [15]. The details of these methods were introduced in Section 2.1. The non-AE-based OD methods included proximity-based Local Outlier Factor (LOF) [5], Mean-shift Outlier Detector (MOD) [36], and Density-increasing Path (DIP) [38], classification-based One-Class Support Vector Machines (OCSVM) [7], ensemble-based Isolation Forest (iForest) [8] and Deep Isolation Forest (DIF) [39], probabilistic-based Histogram-based Outlier Score (HBOS) [10] and empirical-Cumulative-distribution-based Outlier Detection (ECOD) [11]. We reproduced all the AE-based baselines, reported their original results, and applied the proposed MSS method to improve their performance. All the methods were implemented using Python, and the implementation from PyOD [40] was utilized for non-AE-based methods except DIP, which was implemented using its original codes. AUROC was used as the evaluation metric.

There are usually two ways of comparing different OD methods: either by using their default parameters that can be found in the respective papers, or using the op-

Table 2: Architecture setting of the AE-based methods.

Dimsension (D)	Number of layers	Number of units
<20	3	[D, D // 2, D]
≥20, <100	5	[D, D // 2, D // 4, D // 2, D]
≥100, <200	7	[D, D // 2, D // 4, D // 8, D // 4, D // 2, D]
≥200	7	[D, D // 2, D // 4, D // 16, D // 4, D // 2, D]

timal parameters tuned within specific value ranges. Since we wanted to conduct a comprehensive evaluation of different OD methods on the 32 datasets with distinct characteristics and structures, the default parameters of baselines given in the original papers were not appropriate for all these datasets. Thus, the second way was used in this paper. To obtain the optimal parameters, a subset or full set of labeled data has been adopted to tune the methods. Considering that labeling a small set of data is more practical in most real-world scenarios, it is feasible to select parameters using a small set of data. Therefore, we used the subset strategy to obtain the optimal parameters, and the subset was called the auxiliary set in this paper. To ensure a fair comparison, the auxiliary set was randomly sampled in a method-agnostic manner, and it was used to find the optimal parameters for all baseline methods and the proposed methods. The final evaluation and comparison were then conducted on the independent test set. It should be noted that the fairness could be further enhanced by sampling multiple auxiliary sets and test sets and evaluating the average performance across them.

5.2.2. Experimental setup

A. Network architecture. The architecture of PAE had been introduced in Section 4.1. The model used NLL as the training loss function and WNLL as the scoring function. MSS is a universal scoring method that can be applied to most AE-based methods including the proposed PAE. As notations, application to AE produced MSS-AE and application to RSRAE produced MSS-RSRAE, etc.

Considering all data were tabular data, fully-connected layers were used to construct the networks. The number of hidden layers depended on the dimension of the input data. The detailed setting of the network architecture was shown in TABLE 2. Then, the rectified linear unit (ReLU) activation function was added after each layer

of the network except the last layer to increase the nonlinearity of the network. ReLU is widely used as an activation function in deep neural networks. It mimics neuronal mechanisms in the human brain, and demonstrated good performance in many tasks while requiring minimal computational resources. The settings were applied to all the AE-based methods in this experiment.

B. Hyper-parameters. The proposed methods required three hyper-parameters, which were α, m, and k. For WNLL, nine different α were tested: $[0.1, 0.2, \cdots, 0.9]$, which were uniformly distributed between 0 and 1. For MSS, the mean-shift time m was varied from 1 to 3. The number of nearest neighbors k was selected from a candidate list $\mathbf{k}$. In our experiments, $\mathbf{k} = [1, 2, \cdots, \min(99, N-1)]$, where N denotes the number of the instances in the training set.

C. Training and evaluation. All baseline methods and the proposed methods were trained on the training set, and tuned with the aid of the auxiliary set. After training, the best models were tested on the testing set to obtain the final results. In addition, considering the influence of the initial states of the AE-based methods, 10 initial states were generated for AEs and trained separately. After training, the model that performed best on the auxiliary set was selected to be evaluated on the test set. The source codes of the proposed methods are available on Github for reproducibility [3].

5.2.3. Experimental results

A. Evaluation of the probabilistic autoencoder. In this section, the performance of PAE was evaluated. As shown in TABLE 3, the average performance over 32 real-world datasets of PAE with all nine values of α was better than the general AE. The largest relative improvement of the average AUROC was 27% (0.844 vs. 0.787), comparing PAE ($\alpha = 0.10$) and AE. Moreover, PAE significantly outperformed AE on some datasets, such as *Annothyroid* (0.842 vs. 0.592), *Glass* (0.922 vs. 0.794), *Optdigits* (0.946 vs. 0.767), *SpamBase* (0.820 vs. 0.561) and *Wilt* (0.886 vs. 0.451), which showed its superiority for specific applications.

[3] github.com/Ra1demmmm/Autoencoder-based-Outlier-Detection

Table 3: AUROC results of AE and PAE with different α.

Dataset	**AE**	**PAE*** with different α								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ALOI	0.567	0.552	0.546	0.547	0.547	0.543	0.536	0.537	0.537	**0.571**
Annthyroid	0.592	0.798	0.793	0.788	0.786	0.785	0.790	0.805	0.823	**0.842**
Arrhythmia	0.709	**0.750**	0.736	0.744	0.667	0.640	0.631	0.629	0.701	0.638
Breastw	0.978	**0.995**	0.994	0.993	0.990	0.982	0.981	0.976	0.970	0.980
Cardiotocography	0.812	0.816	0.818	0.818	0.819	0.819	0.819	0.820	0.820	**0.842**
Glass	0.794	0.775	0.765	0.755	0.725	0.725	**0.922**	0.912	0.882	**0.922**
HeartDisease	0.822	0.894	0.865	0.878	**0.907**	0.860	0.841	0.809	0.760	0.796
InternetAds	0.709	0.694	0.683	0.680	**0.731**	0.678	0.678	0.687	0.684	0.687
Ionosphere	0.969	0.975	**0.986**	0.979	0.944	0.903	0.816	0.945	0.956	0.945
Letter	0.824	0.697	0.721	0.764	0.797	**0.836**	0.822	0.804	0.806	0.790
Lymphography	**1.000**	0.943	0.943	0.943	0.943	0.943	0.943	0.943	0.943	0.943
Mammography	0.873	**0.901**	0.899	0.891	0.894	0.895	0.896	0.897	0.897	0.848
Mnist	0.887	0.923	**0.940**	0.934	0.936	0.930	0.915	0.874	0.877	0.882
Musk	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Optdigits	0.767	**0.946**	0.936	0.920	0.896	0.865	0.828	0.801	0.803	0.840
PageBlocks	0.950	0.957	**0.962**	0.945	0.948	0.949	0.946	0.939	0.954	0.947
Parkinson	0.662	0.711	**0.817**	0.678	0.787	0.706	0.669	0.727	0.616	0.595
PenDigits	0.868	**0.988**	0.981	0.946	0.943	0.931	0.919	0.904	0.889	0.875
Pima	0.651	**0.761**	0.742	0.699	0.694	0.656	0.641	0.630	0.632	0.631
Satellite	0.702	**0.838**	0.834	0.826	0.807	0.788	0.749	0.721	0.713	0.710
Satimage-2	**0.988**	0.985	0.987	0.987	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**
Shuttle	**1.000**	**1.000**	0.996	0.995	0.995	0.995	0.995	0.995	0.995	0.995
SpamBase	0.561	**0.820**	0.786	0.745	0.702	0.658	0.651	0.640	0.713	0.704
Speech	0.532	**0.643**	0.579	0.594	0.606	0.626	0.411	0.408	0.405	0.526
Stamps	0.833	0.866	0.844	0.842	**0.942**	0.894	0.892	0.892	0.933	0.931
Thyroid	0.985	0.986	**0.993**	0.992	0.991	0.979	0.979	0.979	0.979	0.979
Vertebral	0.604	0.632	0.679	**0.777**	0.750	0.690	0.731	0.747	0.753	0.712
Vowels	0.952	0.904	0.968	**0.976**	**0.976**	0.972	0.963	0.949	0.931	0.912
Waveform	0.738	**0.902**	0.881	0.827	0.801	0.776	0.773	0.770	0.771	0.778
Wilt	0.451	0.820	0.822	0.824	0.820	0.819	0.848	0.867	0.879	**0.886**
Wine	0.879	**0.914**	0.897	0.897	0.897	0.897	0.897	0.879	0.879	0.879
WPBC	0.509	**0.624**	0.580	0.558	0.523	0.514	0.514	0.516	0.516	0.516
AVG.	0.787	**0.844**	0.843	0.836	0.836	0.820	0.812	0.812	0.813	0.815
BEST	4	**15**	6	3	6	3	3	2	2	7

* denotes the proposed method.

The optimal value of α was application-dependent. For example, on datasets *ALOI*, *Annthyroid*, *Cardio.*, *Glass*, and *Wilt*, the best performance appeared when α increased, which implied that the reconstruction error or the negative effect of aleatoric uncertainty were more critical in these applications. In contrast, for datasets such as *BreastW*, *Optdigits*, *PenDigits*, *Satellite*, and *Waveform*, the performance improved as α decreased, which implied that the positive effect of aleatoric uncertainty was more critical in these applications.

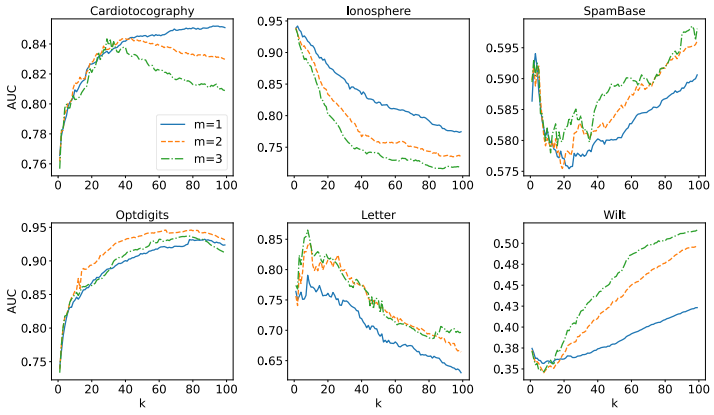

Figure 10: The performance curves of MSS-AE as k increased from 1 to 99 on 6 datasets (*Cardio.*, *Ionosphere*, *SpameBase*, *Optdigits*, *Letter*, and *Wilt*). Each individual graph has three curves with different m denoting the time of mean-shifting, and one line for the performance of the standard AE. For each dataset, the k-NN relationships were computed on the corresponding training set ($\frac{3}{4}$ size of the complete set) and the AUROC results were evaluated on the corresponding auxiliary set.

The experiments demonstrated that the average performance of the groups with smaller α values was generally better. The value of $\alpha = 0.1$ achieved the best performance on 15 datasets (47%) and the best average AUROC performance (0.844). This is because all the training sets contained different ratios of outliers; thus, the aleatoric uncertainty term played a positive role in discriminating the outliers. Although a larger α could more vigorously alleviate the overconfidence issue, the test sets may not have contained many outliers existing in the regions between manifolds, so a modest value of α is sufficient. In summary, a simple adjustment of α could bring significant improvement in most applications, which proved its effectiveness and flexibility in real-world scenarios. A suitable value of α should be selected based on the specific application to achieve the best OD performance. Generally, a relatively smaller α, such as 0.1 or 0.2, is recommended for general situations.

B. Evaluation of the mean-shift outlier scoring. The number of nearest neighbors k and the time of mean-shifting m were two important parameters for the MSS method. In practice, they were all application-dependent. The effect of k was demonstrated for MSS-AE on several datasets in Figure 10. The figure illustrated the performance vari-

ation as k increased across different datasets, and indicated the distinct optimal k for each dataset. Notably, for datasets *Cardio.*, *SpamBase*, *Optidigits*, and *Wilt*, MSS-AE preferred large k values. In the cases of *Ionosphere* and *Letter*, MSS-AE preferred small k values. As with all k-NN-based methods, the k factor was contingent upon the specific application's data properties, including the manifold structure, data density, and dataset scale. For instance, when k was set too large for datasets with relatively small-scale manifolds (e.g., *Ionosphere*) or for datasets where outlier regions did not exhibit significantly lower density (e.g., *Letter*), inliers near the manifold boundary might receive high outlier scores, thereby degrading overall performance. Nevertheless, when k was appropriately calibrated, the MSS method could enhance AE's OD performance in the vast majority of cases.

The effects of m were evaluated on MSS-AE and MSS-PAE ($\alpha = 0.20$), and their AUROC results were listed in TABLE 4. Variation of the time of mean-shifting $m = 1, 2, 3$ was tested for all three methods, and the optimal k for each case was found using the auxiliary set. The average results on 32 datasets demonstrated that MSS-AE and MSS-PAE triumphed their counterparts AE and PAE respectively. The best average AUROC performance for MSS-AE was 0.834 when $m = 3$, and for MSS-PAE, it was 0.869 when $m = 1$, which was the best among all competitors. For a total of 32 datasets, MSS-AE was better than AE for 25 datasets (78%), and MSS-PAE was better than PAE for 26 datasets (81%), which proved the effectiveness of the MSS. TABLE 4 illustrated that the MSS significantly improved the performance on the datasets where AE or PAE had poor performance, such as *InternetAds*, *Optidigits*, *Speech*, and *Vertebral*. Moreover, it can be observed that, in terms of average performance, larger values of m were generally better for MSS-AE, while smaller values of m were preferable for MSS-PAE. This phenomenon was attributed to the fact that after employing WNLL, the score distribution of PAE became more compact, and the issue of unexpected reconstruction was already mitigated. Consequently, less local information, corresponding to a smaller m, was sufficient for MSS-PAE.

In summary, with the appropriate selection of k and m, AE's OD performance was significantly improved on most datasets, demonstrating the effectiveness of the MSS method. Meanwhile, we recommend that practitioners, when applying this method in

Table 4: AUROC results of AE vs. MSS-AE, and PAE vs. MSS-PAE (α = 0.20).

Dataset	AE	MSS-AE			PAE	MSS-PAE		
		m = 1	m = 2	m = 3		m = 1	m = 2	m = 3
ALOI	**0.567**	0.561	0.563	0.563	0.546	0.619	0.647	**0.651**
Annthyroid	0.592	0.589	0.640	**0.649**	0.793	0.870	0.879	**0.885**
Arrhythmia	0.709	0.724	0.731	**0.738**	**0.736**	0.722	0.728	0.730
Breastw	0.978	**0.984**	0.981	0.979	0.994	**0.996**	**0.996**	0.994
Cardiotocography	0.812	**0.881**	**0.881**	0.877	0.818	0.841	**0.844**	**0.844**
Glass	0.794	0.735	0.755	**0.824**	0.765	0.804	0.824	**0.843**
HeartDisease	0.822	**0.841**	0.829	0.801	0.865	**0.897**	0.841	0.866
InternetAds	0.709	0.733	**0.798**	0.789	0.683	0.754	**0.775**	0.739
Ionosphere	0.969	**0.980**	0.975	0.976	**0.986**	0.962	0.968	0.963
Letter	0.824	0.821	**0.858**	0.852	0.721	0.748	0.739	**0.796**
Lymphography	**1.000**	**1.000**	**1.000**	**1.000**	0.943	**1.000**	**1.000**	0.714
Mammography	0.873	0.870	**0.887**	0.870	0.899	**0.901**	0.900	0.899
Mnist	**0.887**	0.875	0.873	0.872	0.940	**0.941**	0.940	**0.941**
Musk	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Optdigits	0.767	0.967	**0.980**	0.973	0.936	**0.994**	0.993	0.991
PageBlocks	0.950	0.943	0.941	**0.952**	**0.962**	0.958	0.957	0.954
Parkinson	0.662	0.718	**0.773**	0.708	0.817	0.773	0.796	**0.819**
PenDigits	0.868	0.859	0.968	**0.969**	0.981	0.984	0.992	**0.993**
Pima	0.651	0.661	0.661	**0.664**	0.742	**0.757**	0.740	0.722
Satellite	0.702	0.737	0.732	**0.772**	0.834	**0.840**	0.838	0.839
Satimage-2	**0.988**	0.986	0.982	0.975	**0.987**	0.986	0.982	0.977
Shuttle	**1.000**	0.997	**1.000**	0.995	0.996	**1.000**	0.997	0.997
SpamBase	0.561	0.587	0.649	**0.650**	0.786	**0.837**	0.823	0.822
Speech	0.532	0.582	0.656	**0.690**	0.579	**0.664**	0.477	0.531
Stamps	0.833	0.883	0.837	**0.957**	0.844	**0.954**	0.942	0.807
Thyroid	**0.985**	0.983	0.975	0.956	0.993	**0.994**	0.991	0.990
Vertebral	0.604	0.802	**0.821**	0.808	0.679	0.687	**0.712**	0.703
Vowels	0.952	0.909	0.965	**0.973**	**0.968**	0.940	0.910	0.957
Waveform	0.738	0.800	**0.813**	0.801	0.881	0.895	0.897	**0.901**
Wilt	0.451	0.493	0.537	**0.560**	0.822	**0.884**	0.873	0.675
Wine	0.879	0.966	**1.000**	0.983	0.897	0.966	**1.000**	**1.000**
WPBC	0.509	**0.563**	0.447	0.501	0.580	**0.629**	**0.629**	0.617
AVG.	0.787	0.813	0.828	**0.834**	0.843	**0.869**	0.863	0.849
BEST	7	7	12	14	6	16	8	11

real-world scenarios, choose parameters based on data structure analysis and domain-specific prior knowledge to maximize the potential of the proposed approach.

C. Comparison with baseline methods. To show the effectiveness of the proposed WNLL function and the versatility of the MSS method, the performance of PAE, 5 AE-based OD methods, and their mean-shifted versions were compared in TABLE 5. The parameters of all methods in this section were tuned to be optimal. The results of all MSS- versions were obtained with the optimal value of m, and the results of PAE

Table 5: AUROC results of 6 AE-based methods and their MSS- versions.

Dataset	AE [13]	MSS-AE *	PAE *	MSS-PAE *	RDA [16]	MSS-RDA *	LCAE [18]	MSS-LCAE *	RSRAE [20]	MSS-RSRAE *	FDAE [15]	MSS-FDAE *
ALOI	0.567	0.563	0.571	**0.695**	0.563	0.559	0.552	0.583	0.549	0.571	0.547	0.540
Annthyroid	0.592	0.649	0.842	**0.888**	0.592	0.629	0.749	0.688	0.617	0.613	0.603	0.599
Arrhythmia	0.709	0.738	0.750	**0.762**	0.709	0.731	0.724	0.707	0.711	0.678	0.707	0.615
Breastw	0.978	0.984	0.995	**0.996**	0.978	0.981	0.973	0.994	0.970	0.985	0.981	0.981
Cardiotocography	0.812	0.881	0.842	0.885	0.812	0.881	0.812	0.883	0.812	**0.901**	0.812	**0.901**
Glass	0.794	0.824	0.922	0.922	0.892	**0.980**	0.922	**0.980**	0.745	0.961	0.755	0.853
HeartDisease	0.822	0.841	0.907	0.907	0.822	0.846	0.822	**0.909**	0.818	0.845	0.750	0.848
InternetAds	0.709	**0.798**	0.731	0.785	0.717	**0.798**	0.701	0.776	0.653	0.754	0.653	0.715
Ionosphere	0.969	0.980	**0.986**	0.979	0.969	0.965	0.980	0.977	0.956	0.959	0.955	0.953
Letter	0.824	**0.858**	0.836	0.822	0.781	0.800	0.856	0.831	0.806	0.780	0.844	0.852
Lymphography	**1.000**	**1.000**	0.943	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.971	**1.000**	0.971	**1.000**
Mammography	0.873	0.887	0.901	0.902	0.873	0.872	0.896	0.895	**0.921**	0.919	0.910	0.910
Mnist	0.887	0.875	0.940	**0.947**	0.887	0.875	0.921	0.917	0.894	0.880	0.892	0.882
Musk	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Optdigits	0.767	0.980	0.946	0.995	0.767	0.980	0.722	0.950	0.769	**0.999**	0.810	0.993
PageBlocks	0.950	0.952	**0.962**	0.958	0.949	0.953	0.944	0.948	0.959	0.961	0.956	0.959
Parkinson	0.662	0.773	0.817	**0.921**	0.657	0.773	0.755	0.725	0.676	0.664	0.597	0.727
PenDigits	0.868	0.969	0.988	**0.996**	0.834	0.975	0.907	0.956	0.764	0.946	0.834	0.962
Pima	0.651	0.664	0.761	**0.771**	0.651	0.692	0.681	0.656	0.653	0.688	0.661	0.709
Satellite	0.702	0.772	0.838	**0.840**	0.700	0.773	0.816	0.836	0.817	0.830	0.830	0.839
Satimage-2	0.988	0.986	0.988	0.988	0.988	0.986	0.989	0.987	0.981	0.986	**0.990**	0.973
Shuttle	**1.000**	**1.000**	**1.000**	**1.000**	0.995	**1.000**	0.952	**1.000**	0.989	0.997	0.977	0.987
SpamBase	0.561	0.650	0.820	**0.841**	0.557	0.650	0.548	0.751	0.591	0.746	0.570	0.711
Speech	0.532	**0.690**	0.643	0.677	0.523	**0.690**	0.540	0.635	0.506	0.528	0.499	0.559
Stamps	0.833	**0.957**	0.942	0.954	0.814	**0.957**	0.941	0.879	0.941	0.931	0.941	0.917
Thyroid	0.985	0.983	0.993	**0.996**	0.984	0.967	0.980	0.980	0.973	0.979	0.971	0.970
Vertebral	0.604	0.821	0.777	**0.846**	0.604	0.821	0.604	0.830	0.602	0.745	0.703	0.739
Vowels	0.952	0.973	**0.976**	0.969	0.929	0.895	0.919	0.934	0.884	0.882	0.863	0.928
Waveform	0.738	0.813	**0.902**	0.901	0.738	0.812	0.785	0.754	0.795	0.825	0.796	0.811
Wilt	0.451	0.560	0.886	**0.891**	0.451	0.554	0.490	0.559	0.457	0.529	0.457	0.536
Wine	0.879	**1.000**	0.914	**1.000**	0.879	**1.000**	0.897	**1.000**	0.948	**1.000**	0.948	**1.000**
WPBC	0.509	0.563	0.624	**0.646**	0.509	0.563	0.496	0.484	0.489	0.590	0.533	0.536
AVG.	0.787	0.843	0.873	**0.896**	0.785	0.842	0.808	0.844	0.788	0.833	0.791	0.828
BEST	3	8	6	18	2	8	2	6	2	5	2	4
p-value	<0.001	-	0.001	-	<0.001	-	0.011	-	0.001	-	0.001	-

and MSS-PAE were obtained using the best value of α. The table conclusively showed that all MSS- versions outperformed their original counterpart with an average of 20% relative performance improvement on the average AUROC. Furthermore, t-tests comparing each MSS- version with its original counterpart revealed that the MSS- versions significantly outperformed their original counterparts ($p < 0.05$), demonstrating the versatility and effectiveness of the MSS method across various models. Then among all standard versions, PAE achieved the best average AUROC performance, which gained a 34% relative improvement compared to the second best method LCAE (0.873 vs. 0.808). Among all MSS- versions, MSS-PAE also achieved the best average AUROC performance, which gained a 33% relative improvement compared to the second best method MSS-LCAE (0.896 vs. 0.844). This result suggested that the proposed MSS was effective for all AE-based OD methods, especially when jointly incorporated with the WNLL method.

Table 6: AUROC results of the proposed methods and non-AE-based baselines.

Dataset	**ECOD** [11]	**iForest** [8]	**LOF** [5]	**OCSVM** [7]	**HBOS** [10]	**MOD** [36]	**DIF** [39]	**DIP** [38]	**AE** [13]	**MSS-AE** *	**PAE** *	**MSS-PAE** *
ALOI	0.540	0.544	0.785	0.543	0.494	0.769	0.546	**0.788**	0.567	0.563	0.571	0.695
Annthyroid	0.739	0.652	0.696	0.601	0.670	0.716	0.504	0.728	0.592	0.649	0.842	**0.888**
Arrhythmia	0.732	0.740	0.705	0.711	0.755	0.718	0.530	0.731	0.709	0.738	0.750	**0.762**
Breastw	0.992	0.989	0.696	0.995	0.983	0.987	0.975	0.993	0.978	0.984	0.995	**0.996**
Cardiotocography	0.802	0.706	0.628	0.767	0.619	0.598	0.689	0.556	0.812	0.881	0.842	**0.885**
Glass	0.853	0.922	0.706	0.863	0.848	**0.941**	0.931	0.814	0.794	0.824	0.922	0.922
HeartDisease	0.687	0.721	0.777	0.730	0.774	0.758	0.723	0.845	0.822	0.841	**0.907**	**0.907**
InternetAds	0.722	0.764	0.632	0.653	0.736	0.660	0.750	0.696	0.709	**0.798**	0.731	0.785
Ionosphere	0.789	0.895	0.931	0.885	0.892	0.932	0.903	0.906	0.969	0.980	**0.986**	0.979
Letter	0.549	0.575	0.878	0.571	0.555	**0.905**	0.554	0.872	0.824	0.858	0.836	0.822
Lymphography	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.914	**1.000**	**1.000**	**1.000**	0.943	**1.000**
Mammography	**0.914**	0.864	0.805	0.863	0.811	0.844	0.776	0.843	0.873	0.887	0.901	0.902
Mnist	0.751	0.819	0.846	0.857	0.534	0.860	0.584	0.851	0.887	0.875	0.940	**0.947**
Musk	0.967	**1.000**	0.997	**1.000**	**1.000**	0.997	0.966	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Optdigits	0.615	0.799	0.542	0.557	0.915	0.494	0.649	0.835	0.767	0.980	0.946	**0.995**
PageBlocks	0.914	0.905	0.933	0.930	0.766	0.921	0.929	0.915	0.950	0.952	**0.962**	0.958
Parkinson	0.350	0.414	0.789	0.384	0.676	0.669	0.484	0.609	0.662	0.773	0.817	**0.921**
PenDigits	0.392	0.758	0.936	0.539	0.761	0.976	0.764	0.985	0.868	0.969	0.988	**0.996**
Pima	0.537	0.592	0.641	0.631	0.656	0.634	0.637	0.690	0.651	0.664	0.761	**0.771**
Satellite	0.608	0.724	0.566	0.694	0.806	0.695	0.748	0.696	0.702	0.772	0.838	**0.840**
Satimage-2	0.969	0.992	0.976	0.996	0.980	0.998	0.996	**0.998**	0.988	0.986	0.988	0.988
Shuttle	0.723	0.820	0.993	0.989	0.803	0.989	0.900	0.988	**1.000**	**1.000**	**1.000**	**1.000**
SpamBase	0.679	0.622	0.474	0.537	0.690	0.538	0.552	0.662	0.561	0.650	0.820	**0.841**
Speech	0.507	0.527	0.698	0.506	0.518	**0.734**	0.582	0.627	0.532	0.690	0.643	0.677
Stamps	0.870	0.931	0.944	0.941	0.786	0.937	0.917	**0.961**	0.833	0.957	0.942	0.954
Thyroid	0.977	0.977	0.964	0.953	0.982	0.963	0.941	0.973	0.985	0.983	0.993	**0.996**
Vertebral	0.596	0.552	0.646	0.629	0.549	0.536	0.371	0.591	0.604	0.821	0.777	**0.846**
Vowels	0.632	0.767	0.939	0.792	0.682	**0.984**	0.760	0.979	0.952	0.973	0.976	0.969
Waveform	0.674	0.759	0.746	0.776	0.753	0.734	0.648	0.800	0.738	0.813	**0.902**	0.901
Wilt	0.350	0.437	0.690	0.295	0.323	0.667	0.478	0.640	0.451	0.560	0.886	**0.891**
Wine	0.897	0.862	0.931	0.879	0.966	0.931	0.897	0.879	0.879	**1.000**	0.914	**1.000**
WPBC	0.482	0.479	0.526	0.499	0.536	0.509	0.494	0.528	0.509	0.563	0.624	**0.646**
AVG.	0.713	0.753	0.782	0.736	0.744	0.800	0.722	0.812	0.787	0.843	0.873	**0.896**
BEST	2	2	1	2	2	5	0	4	3	5	6	20
p-value	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	0.001	0.001	-

To further demonstrate the superiority of the proposed methods, the performance comparisons of the proposed method against 8 non-AE-based OD methods were summarized in TABLE 6. The results showed that the proposed PAE, MSS-AE, and MSS-PAE outperformed all the baselines on average AUROC, where MSS-PAE achieved the best performance on 20 datasets among all 32 datasets (63%) and had the best average AUROC of 0.896 (45% relative improvement compared with DIP which was the best of 8 baselines). T-tests comparing MSS-PAE with all the other methods indicated that the MSS-PAE significantly outperformed other methods ($p < 0.05$), demonstrating its superiority.

Although the proposed method slightly underperformed baseline methods in a few instances, it's important to consider the well-known "No Free Lunch" theorem in ma-

chine learning and anomaly detection: datasets from diverse application scenarios often possess unique, complex structures and characteristics, making it impossible to design a single anomaly detector that outperforms all others across all tasks. Therefore, the results presented in TABLE 5 and TABLE 6, demonstrating that the proposed method achieved optimal performance on most datasets and exhibited the highest average performance, sufficiently proved its effectiveness and superiority.

Meanwhile, the results strongly proved the potency of combining the feature extraction (neural network), uncertainty estimation (probabilistic analysis), and local structure information (nearest-neighbor-graph). It indicated that the OD researchers should consider multiple instruments to design an outlier detector, to adapt various complex realistic scenarios.

6. Conclusion

In this paper, the issue of unexpected reconstruction in AE-based OD was mitigated by two novel strategies. First, the overconfidence issue and the role of aleatoric uncertainty for AE-based OD were analyzed. The analysis supported the usage of NLL to train AE, and WNLL for outlier scoring. Several recommendations were provided for the application of NLL and WNLL for various OD scenarios. Second, the MSS method was proposed to reduce the false inliers judged by the AE-based OD methods, by introducing the information of the local relationship.

The experimental results on 32 real-world OD datasets showed that the proposed methods significantly improved AE's performance on OD. Specifically, WNLL was quite effective and flexible when adapting to different OD applications, and the proposed MSS method could be effectively applied to other AE-based OD methods. Furthermore, MSS-PAE, which was the combination of the proposed two methods, performed best among all baselines. Therefore, the experiments supported the use of the proposed WNLL and MSS simultaneously to get the best performance in practice. We believe that the MSS method can be used as a general plugin for most AE-based OD methods. The proposed methods require hyper-parameter adjustment for various applications. Therefore, future work could focus on developing an auto-tuning technique

that adapts to specific dataset structures. In addition, WNLL may also be applied to improve the performance of other AE-based methods, and can be integrated with other loss functions, which were not evaluated in this paper but can be validated in future research.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the European Union's Horizon Europe research and innovation programme under Marie Sklodowska-Curie grant agreement n° [101126611].

References

[1] J. Yang, S. Rahardja, S. Rahardja, Click fraud detection: Hk-index for feature extraction from variable-length time series of user behavior, in: Proc. of the IEEE Int. Workshop on Mach. Learn. for Signal Process. (MLSP), 2022, pp. 1–6.

[2] Y. Chang, Z. Tu, W. Xie, B. Luo, S. Zhang, H. Sui, J. Yuan, Video anomaly detection with spatio-temporal dissociation, Pattern Recognit. 122 (2022) 108213.

[3] J. Yang, X. Tan, S. Rahardja, Mipo: How to detect trajectory outliers with tabular outlier detectors, Remote Sens. 14 (21) (2022) 5394.

[4] R. Domingues, M. Filippone, P. Michiardi, J. Zouaoui, A comparative evaluation of outlier detection algorithms: Experiments and analyses, Pattern Recognit. 74 (2018) 406–421.

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: identifying density-based local outliers, in: Proc. of the ACM SIGMOD Int. Conf. on Management of Data, 2000, pp. 93–104.

[6] X.-m. Tang, R.-x. Yuan, J. Chen, Outlier detection in energy disaggregation using subspace learning and gaussian mixture model, Int. J. Control Autom. 8 (8) (2015) 161–170.

[7] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471.

[8] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: Proc. of the IEEE Int. Conf. on Data Mining (ICDM), 2008, pp. 413–422.

[9] X. Tan, J. Yang, S. Rahardja, Sparse random projection isolation forest for outlier detection, Pattern Recognit. Lett. 163 (2022) 65–73.

[10] M. Goldstein, A. Dengel, Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm, KI-2012: poster and demo track 9 (2012).

[11] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, G. Chen, Ecod: Unsupervised outlier detection using empirical cumulative distribution functions, IEEE Trans. Knowl. Data Eng. (2022).

[12] D. Hu, Z. Dong, K. Liang, H. Yu, S. Wang, X. Liu, High-order topology for deep single-cell multi-view fuzzy clustering, IEEE Transactions on Fuzzy Systems (2024).

[13] S. Hawkins, H. He, G. Williams, R. Baxter, Outlier detection using replicator neural networks, in: Proc. of the Int. Conf. on Data Warehousing and Knowl. Discov. (DaWaK), 2002, pp. 170–180.

[14] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, Special Lecture on IE 2 (1) (2015) 1–18.

[15] Y. Guo, X. Zhu, Z. Hu, Z. Zhan, Unsupervised anomaly detection by autoencoder with feature decomposition, in: Proc. of the Int. Conf. on Mach. Learn. and Comput. (ICMLC), 2022, pp. 258–265.

[16] C. Zhou, R. C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proc. of the ACM SIGKDD Int. Conf. on Knowl. Discov. and Data Mining (KDD), 2017, pp. 665–674.

[17] J. Chen, S. Sathe, C. Aggarwal, D. Turaga, Outlier detection with autoencoder ensembles, in: Proc. of the SIAM Int. Conf. on Data Mining (SDM), 2017, pp. 90–98.

[18] Y. Ishii, M. Takanashi, Low-cost unsupervised outlier detection by autoencoders with robust estimation, J. Inf. Process. 27 (2019) 335–339.

[19] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, A. v. d. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: Proc. of the IEEE/CVF Int. Conf. on Comput. Vis. (ICCV), 2019, pp. 1705–1714.

[20] C.-H. Lai, D. Zou, G. Lerman, Robust subspace recovery layer for unsupervised anomaly detection, in: Proc. of the Eighth Int. Conf. on Learn. Representations (ICLR), 2020, pp. 1–28.

[21] Q. Yu, M. Kavitha, T. Kurita, Autoencoder framework based on orthogonal projection constraints improves anomalies detection, Neurocomputing 450 (2021) 372–388.

[22] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, Advances in neural information processing systems 30 (2017).

[23] N. Ståhl, G. Falkman, A. Karlsson, G. Mathiason, Evaluation of uncertainty quantification in deep learning, in: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, 2020, pp. 556–568.

[24] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, Advances in neural information processing systems 30 (2017).

[25] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al., A survey of uncertainty in deep neural networks, Artificial Intelligence Review 56 (Suppl 1) (2023) 1513–1589.

[26] A. A. Pol, V. Berger, C. Germain, G. Cerminara, M. Pierini, Anomaly detection with conditional variational autoencoders, in: 2019 18th IEEE international conference on machine learning and applications (ICMLA), IEEE, 2019, pp. 1651–1657.

[27] A. Legrand, H. Trannois, A. Cournier, Use of uncertainty with autoencoder neural networks for anomaly detection, in: 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), IEEE, 2019, pp. 32–35.

[28] E. Daxberger, J. M. Hernández-Lobato, Bayesian variational autoencoders for unsupervised out-of-distribution detection, arXiv preprint arXiv:1912.05651 (2019).

[29] S. Park, G. Adosoglou, P. M. Pardalos, Interpreting rate-distortion of variational autoencoder and using model uncertainty for anomaly detection, Annals of Mathematics and Artificial Intelligence 90 (7) (2022) 735–752.

[30] D. Hafner, D. Tran, T. Lillicrap, A. Irpan, J. Davidson, Noise contrastive priors for functional uncertainty, in: Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 905–914.

[31] D. Hu, K. Liang, S. Zhou, W. Tu, M. Liu, X. Liu, scdfc: a deep fusion clustering method for single-cell rna-seq data, Briefings in Bioinformatics 24 (4) (2023) bbad216.

[32] D. Hu, K. Liang, Z. Dong, J. Wang, Y. Zhao, K. He, Effective multi-modal clustering method via skip aggregation network for parallel scrna-seq and scatac-seq data, Briefings in Bioinformatics 25 (2) (2024) bbae102.

[33] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans. Inf. Theory 21 (1) (1975) 32–40.

[34] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.

[35] Y. Yao, J. Ma, Y. Ye, Regularizing autoencoders with wavelet transform for sequence anomaly detection, Pattern Recognit. 134 (2023) 109084.

[36] J. Yang, S. Rahardja, P. Fränti, Mean-shift outlier detection and filtering, Pattern Recognit. 115 (2021) 107874.

[37] J. Yang, Y. Chen, S. Rahardja, Neighborhood representative for improving outlier detectors, Inf. Sci. (2022).

[38] J. Zhao, F. Deng, J. Zhu, J. Chen, Searching density-increasing path to local density peaks for unsupervised anomaly detection, IEEE Transactions on Big Data (2023).

[39] H. Xu, G. Pang, Y. Wang, Y. Wang, Deep isolation forest for anomaly detection, IEEE Transactions on Knowledge and Data Engineering (2023).

[40] Y. Zhao, Z. Nasrullah, Z. Li, Pyod: A python toolbox for scalable outlier detection, J. Mach. Learn. Res. 20 (2019) 1–7.