Can Untrained Neural Networks Detect Anomalies?

Seunghyoung Ryu, *Member, IEEE*, Yonggyun Yu, and Hogeon Seo

***Abstract*—Anomaly detection (AD) plays a crucial role in identifying unusual data patterns indicative of potential issues or opportunities. Recent data-driven AD models require extensive training for satisfactory performance. This study explores the potential of untrained neural networks (UNNs) for AD tasks. UNNs are used for nonlinear random projection. The anomaly scores are derived from the randomly mapped features using the Mahalanobis distance. We conducted a series of experiments on 12 tabular and two image datasets, comparing the performance of UNNs with 12 established AD models, including state-of-the-art deep learning approaches. Our results demonstrate that UNNs can achieve competitive AD performance without training, which also underscores the importance of training to ensure higher performance beyond the untrained baseline. In addition, the proposed approach offers advantages in terms of time, computational costs, and accessibility, making it a compelling alternative for various applications.**

***Index Terms*—Anomaly detection (AD), deep learning, nonlinear random mapping, untrained neural networks (UNNs).**

Nomenclature

g_θ	Untrained neural network.
θ	Weights of the untrained neural network.
A_g	Anomaly scoring function of model g.
$\mathbf{x}$	Input to the untrained neural network.
$\mathbf{y}$	Output from the untrained neural network.
$\mathbf{x}'$	Unknown input sample for detection.
$\mathbf{x}_n$	Normal data point.
$\mathbf{X}_{\text{ref}}$	Reference set of normal data.
$\mathbf{Y}_{\text{ref}}$	Set of model outputs for reference normal data.
μ	Mean vector of the model outputs.
S	Covariance matrix of the model outputs.

Manuscript received 11 May 2023; revised 27 July 2023 and 27 November 2023; accepted 14 December 2023. Date of publication 11 January 2024; date of current version 4 April 2024. This work was supported in part by Korea Atomic Energy Research Institute R&D Program under Grant KAERI-524450-23, in part by the National Research Foundation of Korea under Grant NRF-2021R1F1A1051290 and Grant RS-2023-00253853 funded by the Korean government (Ministry of Science and ICT), and in part by the faculty research fund of Sejong University in 2023. Paper no. TII-23-1677. *(Corresponding author: Hogeon Seo.)*

Seunghyoung Ryu is with the Department of Artificial Intelligence and Robotics, Sejong University, Seoul 05006, South Korea (e-mail: shryu@sejong.ac.kr).

Yonggyun Yu and Hogeon Seo are with the Department of Artificial Intelligence, Korea National University of Science and Technology, Daejeon 34113, South Korea, and also with Applied Artificial Intelligence Section, Korea Atomic Energy Research Institute, Daejeon 34057, South Korea (e-mail: ygyu@kaeri.re.kr; hogeony@hogeony.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3345461.

Digital Object Identifier 10.1109/TII.2023.3345461

I. Introduction

ANOMALY refers to something that deviates from what is normal, and anomaly detection (AD) is the task of identifying such anomalies within data. Anomalies often have a negative impact on systems, making AD critical in various industrial applications, including finance [1], medical diagnostics [2], network security [3], and manufacturing [4]. For effective AD, AD models must grasp the concept of *normality* in the data they analyze. This understanding can be acquired through data-driven methods, which is why many modern AD models rely on machine learning and deep learning techniques.

When a sufficient number of labeled abnormal samples are available, AD can be treated as a typical classification task and addressed using a supervised learning framework. However, anomalies are rare, making it difficult and costly to obtain enough labeled samples for AD [5]. Given these challenges, unsupervised learning methods become a more practical and preferred choice for developing AD models. Based on the infrequent nature of anomalies, most data are assumed to be normal in the unsupervised AD framework. Then, the AD model learns normality from the data by training the model for unsupervised learning tasks such as clustering, reconstruction, and dimensionality reduction. After training, the model calculates the anomaly score of new samples and classifies any sample that exceed a threshold as an anomaly. Building on this foundation, traditional AD models utilize *machine learning* techniques, such as the isolation forest (ISOF) [6], one-class support vector machine (OCSVM) [7], and local outlier factor (LOF) [8]. Recent advancements in AD models leverage deep learning methods, which can extract high-level features from massive training data. This ability to recognize complex patterns helps the models learn normal characteristics, thereby enhancing the overall performance of AD. Several representative deep-learning-based AD (deep-AD) models include autoencoder (AE) [9], [10], [11], [12], [13], [14], [15], [16], deep support vector data description (DSVDD) [17], deep autoencoding Gaussian mixture model (DAGMM) [18], and generative adversarial network (GAN)-based AD models [2], [19], [20], [21], [22]. For example, AE-based AD models are trained to compress normal data into lower dimensional latent features, allowing for accurate reconstruction of the original data. Then, reconstruction error serves as an anomaly score, and samples with high scores are

identified as abnormal. The performance of an AD model is commonly assessed using classification metrics, such as the area under receiver operating characteristic curve (AUROC) and the area under the precision–recall curve (PRAUC). State-of-the-art deep-AD models have demonstrated enhanced performance in these metrics, due to the integration of advanced network architectures and customized loss functions [15], [23], [24].

Interestingly, we have found that simpler models have sometimes achieved competitive AUROC scores compared with sophisticated AD models. We also observed that an untrained AE could achieve an AUROC score over 0.5, exceeding that of random guessing. Motivated by these results, we formulate a provocative research question: "Can untrained neural networks (UNNs) detect anomalies?" To answer this, we present a simple and primitive AD strategy that takes advantage of the potential of UNNs with the Mahalanobis distance metric. First, a randomly initialized and fully connected neural network serves as a function that maps input data into nonlinear random space. Next, the Mahalanobis distance is employed to calculate the anomaly score, measuring the distance between a point and reference distribution of normal data mapped by the UNN. To verify this concept, we conducted experiments on various datasets: 12 tabular and two image datasets. We then compare the results with 12 established AD models, encompassing traditional models and state-of-the-art deep-AD models, including robust collaborative AE (RCA) [15], internal-contrastive-learning-based model (ICL) [23], and learnable unified neighborhood-based anomaly ranking (LUNAR) [24].

Our experimental results show that the proposed method demonstrates effective AD performance while offering accessibility advantages in practical use based on its training-free property (i.e., no need to train neural network weights). Consequently, the proposed model could serve as a baseline for developing new AD models, establishing a reference point for the empirical performance lower bound without training. Our research contributions can be summarized as follows.

1) We introduce an AD method based on UNNs that leverages randomly initialized neural networks in conjunction with the Mahalanobis distance. The integration of these two methods enables the proposed model to perform AD tasks effectively without training neural networks.
2) To validate UNN-based AD, we evaluate its performance on various tabular datasets by comparing AD models, including state-of-the-art deep learning approaches. Experimental results show that, despite its random and *training-free* nature, the proposed model consistently delivers competitive AD performance.
3) Considering the prevailing trend toward deep learning models, our work provides a counterintuitive observation and re-emphasizes the importance of training to ensure the higher performance of learning-based AD models.

The rest of this article is organized as follows. Section II introduces related works. Sections III and IV describe the problem formulation of the AD task and the proposed method, respectively. Section V presents the experiments, including ablation studies. Results and discussion are presented in Section VI. Finally, Section VII concludes this article.

II. Related Works

A. AD With Deep Learning

In order to compute anomaly score, deep learning models in unsupervised AD frameworks learn the normality of data through unsupervised tasks, which can generally be categorized into three types: probability distribution modeling, one-class classification, and reconstruction [25]. In some cases, these tasks are trained to perform together.

A conventional approach is to use AE-based reconstruction models. The AE is trained to minimize the reconstruction error of normal data; therefore, samples with a high reconstruction error are classified as anomalies. Due to its simple mechanism, the AE approach is widely adopted in various domains such as multivariate sensors [16], image [26], and time-series data [12]. Variations in AE have been developed to achieve better AD performance [9], [10], [11]. For example, convolutional AEs and long short-term memory AEs have been introduced to address challenges in more complex image and time-series data. Some models exploit additional features in anomaly scoring along with the reconstruction error, such as the error in latent spaces [13] and the uncertainty of the data [14]. Multiple AEs are collaboratively used to focus on normal samples with low reconstruction error within contaminated data [15]. AEs can also be associated with probability distribution modeling approaches. For example, DAGMM [18] leverages an AE to extract latent features and reconstruction errors. The combined distribution of these features and errors is then modeled using a mixture of multivariate Gaussians.

In terms of generative models, AD models based on variational AE [27], [28], [29] employ the reconstruction probability as an objective measure to identify anomalies, outperforming AD based on conventional AE and principal component analysis (PCA). GAN-based AD was first employed in AnoGAN [2], which utilizes reconstruction error along with discriminator loss. Following this, various GAN-based AD models have been developed, examples of which include efficient GAN-based AD [19], fast-AnoGAN [20], GANomaly [21], and their ensemble [22]. Other types of deep-AD models have evolved in different ways. For example, some models focus on solving one-class classification problems [17], [30], while others leverage architectures like graph neural networks [24]. A comprehensive overview of various deep-AD models can be found in [5], [25], and [31].

Advancements in algorithm and model structures have significantly improved the performance of deep-AD models. Despite these achievements, the development and optimization of problem-specific AD models for real-world applications still face several challenges. These include training-related factors such as the collection of training data, the embedding of domain-specific knowledge, limitations in computational time and resources, and the need to address issues like model drift. Automated machine learning [32] has emerged to address these challenges, but the training step remains necessary. In this context, the proposed UNNs offer distinct advantages, particularly in terms of efficiency and ease of implementation.

B. Utilizing Randomness

Random projection is a dimensionality reduction technique that linearly maps a point to a lower dimensional space by multiplying a random matrix. The theoretical foundation for this approach is provided by the Johnson–Lindenstrauss lemma [33] stating that distances between points can be approximately preserved after a linear random projection with high probability [34]. Thus, anomalies that are distinguishable in the original space can still be identified after the projection.

Lemma 1 (Johnson–Lindenstrauss lemma [35], [36]): For any integer n and tolerance $0 < \varepsilon < 1$, let m be a positive integer satisfying $m \geq O(\varepsilon^{-2}\log(n))$. Then, for any set of d-dimensional real-valued points $\mathcal{X} = \{x_i\}_{i=1}^{n}$, there exists a linear map $f : \mathbb{R}^d \to \mathbb{R}^m$ such that for all $u, v \in \mathcal{X}$

$$(1-\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2. \quad (1)$$

Early work of [37] used random projection to measure the outlyingness of data. The outlyingness is defined as the ratio of the median absolute deviation (MAD) of a given sample to the MAD of an entire dataset in randomly projected space, where a high outlyingness indicates abnormality. In [38], Fourier-based random features are combined with OCSVM to improve detection efficiency in large datasets. More recently, deep random projection outlyingness has been proposed, which combines random projection with DSVDD [39]. In this approach, the features extracted by DSVDD are subsequently transformed using random projection. These studies employed random projection for generating input vectors for subsequent AD models, necessitating further training.

Several recent studies have demonstrated the potential of using randomly initialized neural networks (i.e., nonlinear random projection) for AD. For example, an extreme learning machine (ELM) [40] is a type of neural network composed of random hidden layers and a learnable output layer. Instead of using backpropagation, the weights of the output layer are obtained by solving a least-squares problem. The ELM under unsupervised and semisupervised learning frameworks was also investigated in [41]. Related to AD, a random sparse neural network (i.e., sparse activation with randomly initialized weights) was used to generate a random binarized hash code, and its dot product to the moving average was used as the anomaly score [42]. A deep ISOF model [43] was introduced that uses a randomly initialized neural network as a feature generator in conjunction with an ISOF model.

The inherent efficacy of random neural networks can be found in some studies. For example, the ability to estimate numerosity from an untrained convolutional neural network was investigated in [44]. Kim et al. [45] highlighted the inadequacies of the point adjustment technique commonly used in recent time-series AD models, and it was observed that the reconstruction error of a random neural network showed comparable F1 scores. These works outline the potential of a randomly initialized neural network in different aspects. In this research, we investigate the direct use of UNN for AD in conjunction with the Mahalanobis distance. By combining them, the proposed method demonstrates robust and competitive AD performance across various tabular datasets.

Algorithm 1: AD via UNNs.

Require: $\mathbf{X}_{\text{ref}}$: Reference dataset for anomaly score calculation.
ϕ: Randomness method.
θ: Network weights.
$g_\theta^\phi(\mathbf{x})$: Randomly initialized neural network of ϕ.
d': Target dimensionality.
Ensure: $A_g(\mathbf{x}')$: Anomaly score of unknown sample $\mathbf{x}'$
procedure Model preparation
  Initialize neural network
  for all $\mathbf{x} \in \mathbf{X}_{\text{ref}}$ **do**
    $\mathbf{y} = g_\theta^\phi(\mathbf{x})$
    $\mathbf{Y}_{\text{ref}} = \mathbf{Y}_{\text{ref}} \cup \{\mathbf{y}\}$
  end for
  Calculate mean μ and covariance S of $\mathbf{Y}_{\text{ref}}$.
end procedure
procedure Score calculation
    Obtain a new sample $\mathbf{x}'$
    Performs nonlinear random mapping: $\mathbf{y}' = g_\theta^\phi(\mathbf{x}')$
    Calculate anomaly score $A_g(\mathbf{x}')$
    $A_g(\mathbf{x}') = \sqrt{(\mathbf{y}'-\mu)^T S^{-1}(\mathbf{y}'-\mu)}$
    return $A_g(\mathbf{x}')$
end procedure

III. Problem Formulation

In this section, we describe the general problem setting of AD tasks. Let $\mathbf{x} \in \mathbb{R}^d$ be a d-dimensional vector obtained from a system. Then, the deep-AD model f_θ maps $\mathbf{x}$ to $\mathbf{y} \in \mathbb{R}^{d'}$, where θ denotes the trainable weights of the neural network, and $\mathbf{y}$ differs according to the model. The value of θ is obtained by minimizing the loss function designed for AD such as reconstruction, classification, and clustering. After training, AD models derive the anomaly score of a new sample $\mathbf{x}'$, $a_{\mathbf{x}'} \in \mathbb{R}^+$ from the scoring function $A(\mathbf{x}', f_\theta(\mathbf{x}'))$. If $a_{\mathbf{x}'}$ exceeds a threshold a_{th}, it is classified as an anomaly; otherwise, it is considered normal. For example, $\mathbf{y}$ of an AE is a reconstruction of input $\mathbf{x}$ based on the lower dimensional latent feature $\mathbf{z}$. Then, f_θ is obtained by minimizing $\|\mathbf{x} - f_\theta(\mathbf{x})\|^2$, resulting in low reconstruction error $\|\mathbf{x} - f_\theta(\mathbf{x})\|^2$ for trained normal data and a relatively higher error for abnormal data. Besides reconstruction error, there are various ways to calculate anomaly scores, e.g., distance to the cluster [18], reconstruction probability [27], and Mahalanobis distance [14]. Deep-AD models generally employ a loss function closely tied to the anomaly score, allowing end-to-end AD.

IV. Proposed Methodology

The algorithm for the proposed methods is presented as pseudocode in Algorithm 1. Let g_θ be a UNN that maps input $\mathbf{x}$ to output $\mathbf{y}$, where θ denotes its weight parameters. In deep learning, θ is randomly initialized and then trained to minimize the predefined loss function. However, θ is not trained in the UNN; thus, the input $\mathbf{x}$ is randomly and nonlinearly transformed to $\mathbf{y} = g_\theta(\mathbf{x})$ according to random weights and activation functions. Here, we use a simple neural network with a single hidden layer and a sigmoid activation function.

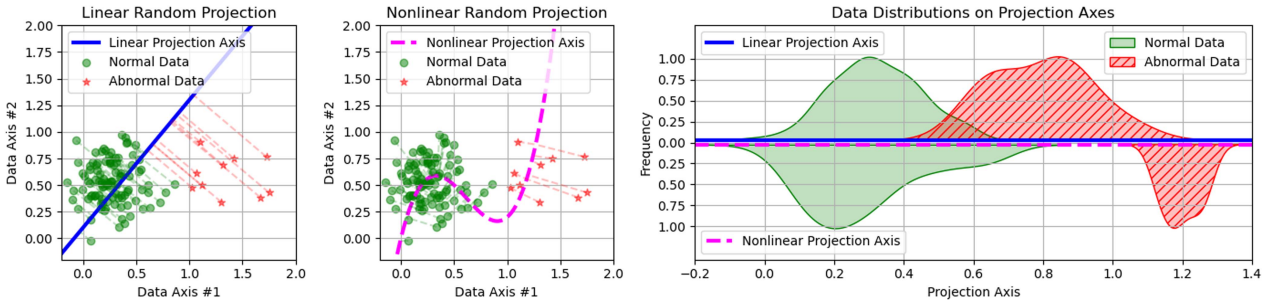

Fig. 1. Difference between linear and nonlinear projections: linear projection (left), nonlinear projection (center), and the distributions after projections (right). The nonlinear projection (right-bottom) provides better separation than the linear projection (right-top).

Then, the anomaly score of a new sample $\mathbf{x}'$ is derived from the Mahalanobis distance, which measures the distance between the mapped point and the reference distribution of normal data. Let $\mathbf{X}_{\text{ref}}$ be a set of reference normal data $\mathbf{x}_n$. We obtain $\mathbf{Y}_{\text{ref}} = \{g_\theta(\mathbf{x}) | \mathbf{x} \in \mathbf{X}_{\text{ref}}\}$ from the UNN, and the anomaly score of a new sample $\mathbf{x}'$ is derived using the following equation:

$$A_g(\mathbf{x}') = \sqrt{(\mathbf{y}' - \mu)^T S^{-1} (\mathbf{y}' - \mu)} \tag{2}$$

where $\mu \in \mathbb{R}^{d'}$ and $S \in \mathbb{R}^{d' \times d'}$ are a mean vector and a covariance matrix of $\mathbf{Y}_{\text{ref}}$, respectively. As a result, the anomaly score of UNN is the distance to the distribution of a reference set in a nonlinear random space. The combination with the Mahalanobis distance is a key point in utilizing UNN for AD. After random projection via UNN, the scale of each dimension can be different. In this regard, the Mahalanobis distance involves the normalization process, thus providing scale-invariant and normalized anomaly scores. It is worth noting that the Mahalanobis distance is proportional to the square root of the Gaussian negative log-likelihood. Therefore, a low-probability sample will have a larger distance when the distribution of normal data in random spaces is modeled as a multivariate Gaussian distribution.

A. Interpretation of the UNN-Based AD

Deep-AD models aim to identify nonlinear space through training where normal and anomalous data points can be easily distinguished. Trained neural networks are a subset of all possible randomly initialized models. In this respect, if the deviation between normal and abnormal data points can be preserved to some extent after projection onto a random space, feature vectors from UNNs can be used as is without training the weights of neural networks. However, compared to the linear random projection, theoretical modeling of distance-preserving property is difficult due to the complex nonlinearity of neural networks. In [46] and [47], the following bounds on Euclidean distance were studied with single-layer and rectified linear units (ReLU), which is given as [46, Corollary 5].

Corollary 1: Let $K \subset \mathbb{B}_1^d$ be the manifold of input data, and $\sqrt{m}M$ is an $m \times d$ random matrix of independent and identically distributed normal distribution. If we denote ρ indicating ReLU and $m \geq C\delta^{-4} w(K)^4$ for given constant C and the Gaussian mean width $w(K)$, then with high probability in the form of $1 - O(-\exp(\delta^2))$, the following inequality holds on $u, v \in \mathbb{R}^d$:

$$\tfrac{1}{4}\|u - v\|_2^2 - \delta \leq \|\rho(Mu) - \rho(Mv)\|_2^2 \leq \tfrac{1}{2}\|u - v\|_2^2 + \delta. \tag{3}$$

This suggests the distance-preserving capability of random neural networks with sufficiently large m. The proofs and detailed explanation are given in [46] and [47].

Next, we illustrate an example of how the nonlinear projection helps for AD in Fig. 1. Linear random projection maps data onto a random straight line drawn in observation space $\mathbb{R}^d$. We could separate normal and abnormal data by modeling the distribution of data points on that axis. The Euclidean distance between two points after linear projection is bounded by the distance in the original space. In the worst case, two classes may overlap completely, making it impossible to detect anomalies. However, this can be mitigated by drawing multiple lines to reduce such probability and using them together by calculating the Mahalanobis distance, i.e., the ensemble effect [14]. In this sense, the calculation of Mahalanobis distance with raw data is a special case of linear random projection where the straight lines correspond to the principal components defined by normal samples. In contrast to the linear case, a UNN draws a nonlinear curve and projects data onto that curve. Since a nonlinear curve does not impose an upper bound on the distance between data points after projection, the UNN has a chance to facilitate further separation of anomalies from normal samples. As in the examples in Fig. 1, compared with the linear case, abnormal and normal data can be further apart when stretching the curve to the line.

B. Approaches for Providing Randomness to Networks

Random initialization of weights inherently introduces randomness into neural networks. However, there can be variations in the process of obtaining random vectors from initialized neural networks. In this study, we compare the following three different models, as described in Algorithm 2.

1) *Iterative random model (IRM):* The IRM is composed of a single output neuron. As a result, a d'-dimensional vector is generated by concatenating the outputs from d' iterations. In each iteration, the weights of the neural network are re-initialized, which results in the utilization of varying hidden features.
2) *Dropout random model (DRM):* The DRM also has a single output neuron. Instead of re-initializing weights,

Algorithm 2: Variations of UNN

Require: d': Target dimensionality.
$g_\theta(\mathbf{x})$: Randomly initialized neural network.

Method 1. Iterative random model

Require: $\{\theta_i\}$: List of random weights for IRM.
for $i = 1$ to d
  $y_i = g_{\theta_i}(\mathbf{x})$
end for
$\mathbf{y} = \{y_i\}_{i=1}^{d}$
return $\mathbf{y}$

Method 2. Dropout random model

Require: p: Dropout rates for DRM.
$\{DM_i(p)\}$: List of dropout masks for DRM.
for $i = 1$ to d
  Apply dropout to the network with $DM_i(p)$
  $y_i = g_{\theta \times DM_i(p)}(\mathbf{x})$
end for
$\mathbf{y} = \{y_i\}_{i=1}^{d}$
return $\mathbf{y}$

Method 3. One-shot random model

$\mathbf{y} = g_\theta(\mathbf{x})$
return $\mathbf{y}$

the DRM utilizes the Monte Carlo dropout method [48], which is a common approach for measuring model uncertainty. In each inference iteration, the DRM randomly masks neurons based on the dropout layer, allowing the use of different features based on random subnetworks.

3) *One-shot random model (ORM):* The ORM is composed of d' output neurons. Thus, it generates a d'-dimensional random vector at once without iteration. Compared with the IRM and the DRM, it shares identical hidden features in obtaining d' output values.

V. Experiments

A. Datasets and Metrics

The AD performance of the proposed method is evaluated on 12 multivariate tabular datasets [13], [49], which are the primary targets of this method. In addition, to investigate the detection capability on image datasets, MNIST (*mnist*) and Fashion MNIST (*fmnist*) are also included. Basic information about these datasets is summarized in Table I.

All tabular datasets have predefined anomaly classes and go through the following preprocessing pipelines. First, the data are normalized using Z-scores, subtracting the mean and dividing by the standard deviation, column by column. Next, the dataset is divided into two subsets: the reference and test sets. The reference set $\mathbf{X}_{\text{ref}}$ is used to derive the mean vector μ and the covariance matrix S to compute the Mahalanobis distance. In doing this, a randomly sampled 70% of the normal data is used as $\mathbf{X}_{\text{ref}}$. The remaining 30% of the normal data is used as the test set $\mathbf{X}_{\text{test}}$ along with all the abnormal data.

TABLE I
Benchmark Datasets for Experiments

Dataset	N	d	$R_a(\%)$	Description
backdoor	95 329	196	2.4	Network backdoor attacks
campaign	41 188	62	11.2	Campaign calls success
celeba	202 599	39	2.2	Bald celebrities
census	299 285	500	0.2	Rare high-incomes
donors	619 326	10	0.1	Exceptional projects
eopt	90 515	20	11.3	Storage system failure
fraud	284 807	29	0.1	Credit card fraud detection
mi-f	25 286	58	8.5	Milling machine failure
mi-v	23 125	58	17.0	Milling machine validation
nasa	4 687	33	16.1	Hazardous asteroids
rarm	20 221	6	23.2	Robotarm malfunction
thyroid	7 200	21	7.4	Thyroid disease
mnist	70 000	784	–	Handwritten digits image
fmnist	70 000	784	–	Fashion products image

N: Number of data, d: Dimensionality, R_a: Ratio of anomalies.

The test set is used to evaluate the performance of AD models in terms of two standard classification metrics: the AUROC and the PRAUC. Both the metrics represent the overall detection performance of the models in a scalar value ranging from 0 to 1. AUROC and PRAUC can be represented by the following equations. First, we could generate a confusion matrix for the given threshold a_{th}; thus, true positive rate (TPR), false positive rate (FPR), and precision (PRC) can be denoted as functions of a_{th}. Then, AUROC and PRAUC are

$$\text{AUROC} = \int_0^1 \text{TPR}(a_{\text{th}}) d\text{FPR}(a_{\text{th}}), \tag{4}$$

$$\text{PRAUC} = \int_0^1 \text{PRC}(a_{\text{th}}) d\text{TPR}(a_{\text{th}}). \tag{5}$$

These metrics provide a comprehensive view of the model performance across all possible thresholds rather than being tied to a specific threshold value. This is particularly useful in AD, where the optimal threshold may vary depending on the specific application and the tradeoff between false positives and false negatives that the user is willing to accept. In both the cases, the perfect classifier has a score of 1, so a higher value indicates better performance. During the experiment, we report mean scores and standard deviations with $\pm$ sign after ten trials with different random seeds (0–9) for each dataset. Note that the UNN showed consistent results with no significant deviations as we increased the number of experiments from 10 to 1000.

B. Benchmarks Methods

We use AUROC and PRAUC to compare three proposed methods (IRM, DRM, and ORM) with the following three baselines: Mahalanobis distances in observation space (Raw), linear random mapping approach with Gaussian random projection (LRM_G), and uniform distribution (LRM_U).

For the Raw case, anomaly scores are derived by directly calculating the Mahalanobis distance on $\mathbf{X}_{\text{ref}}$. For the LRM cases, we generate a $d \times d'$ random matrix R from the zero-mean Gaussian distribution for LRM_G using the implementation in *scikit-learn* library [50]. LRM_U generates R from the uniform distribution ranging from 0 to 1 and normalizes the column sum by $\sum_{j=1}^{d'} R_{i,j} = 1$. Then, linear random mapping is conducted

TABLE II
DESCRIPTION OF UNN STRUCTURES

Model	Structure
IRM	Input(d)-Dense(128)-Sigmoid-Dense(1)
DRM	Input(d)-Drop(0.5)-Dense(128)-Sigmoid-Drop(0.5)-Dense(1)
ORM	Input(d)-Dense(128)-Sigmoid-Dense(100)

d is the dimensionality of the input vectors, p in Drop() is the dropout rate.

TABLE III
AUROC, PRAUC, AND COMPUTATION TIME ACCORDING TO THE RANDOMNESS METHODS FOR UNNS

Metric	Baselines			Proposed		
	Raw	LRM_G	LRM_U	IRM	DRM	ORM
AUROC	76.0 ±1.9	77.7 ±1.5	76.7 ±1.1	82.4 ±1.0	81.9 ±1.8	**82.5** ±0.6
PRAUC	51.9 ±1.7	52.3 ±2.1	52.7 ±1.6	59.8 ±1.6	59.0 ±2.0	**59.8** ±1.0
Time (s)	0.84 ±2.4	0.96 ±1.3	0.90 ±1.2	11.14 ±15.5	14.94 ±18.7	**0.58** ±0.7

Bold: The best. Underlined: The second best.

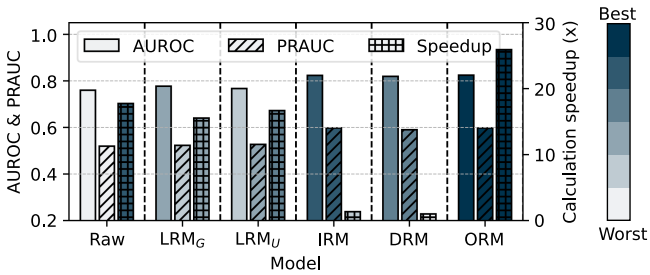

Fig. 2. AUROC and PRAUC on average and speedup indicating the ratio to the maximum calculation time of the UNN based on the DRM.

by the dot product with R, and its anomaly score is the Mahalanobis distance between $\mathbf{x}' \cdot R$ and $\mathbf{X}_{\text{ref}} \cdot R$.

For a UNN, we used a simple neural network consisting of only one hidden layer with a sigmoid activation function. The detailed structure differs depending on the randomness method and is described in Table II. The numbers in parentheses represent the hyperparameters of each corresponding layer, and common values were used for simplicity; the number of neurons for *Dense* layers and the dropout probability for *Drop* layers.

Next, we compared general AD models, including five machine-learning-based models and seven state-of-the-art deep-learning-based models. The machine-learning-based AD models include the following methodologies: LOF [8], clustering-based LOF (CBLOF) [51], ISOF [6], histogram-based outlier score (HBOS) [52], and PCA [53]. In the case of deep-AD models, we utilize DSVDD [17], DAGMM [18], GANomaly [21], random-distance-prediction-based AD model (RDP) [54], RCA [15], ICL [23], and LUNAR [24]. These benchmark models are implemented using its default parameters: *DeepOD* [43] for ICL, RCA, and RDP and *PyOD* [55] for the remaining models.

C. AD Performance on UNN Variations

First, we investigated the most effective way among UNN variations (IRM, ORM, and DRM) and compared it to the three baselines (Raw, LRM_G, and LRM_U). Table III and Fig. 2 describe the average performance in terms of AUROC, PRAUC, and speedup of computation time. According to the experimental results, even modeling of a multivariate Gaussian distribution on raw data achieves a certain level of AUROC. This indicates the inherent distinguishability of anomalies in observation space. In the context of linear random mapping, it has been observed that a generated random matrix does not have significant variations. The linear random mapping slightly increased the AUROC and PRAUC, but it could be further improved with UNNs.

The ORM model achieved the highest performance for all three metrics. IRM and DRM showed slightly lower AUROC and PRAUC performance and much longer computation time due to iterative initialization. The DRM also required longer computation time due to iterative masking of dropped neurons, and both AUROC and PRAUC were slightly reduced compared to the others. In terms of AUROC and PRAUC, there is only a marginal difference in performance between the different randomization methods. Therefore, we could simply use the ORM for a UNN-based AD; it is straightforward, fast, and also has a lower standard deviation.

D. Ablation Studies

Ablation studies were conducted on the structural changes of neural networks. We set the basic hyperparameter as the ORM described in Table II and changed one of the following list of hyperparameters while keeping the others fixed: 1) number of hidden neurons; 2) depth of hidden layers; 3) type of activation function; and 4) output dimensionality. Fig. 3 illustrates the distributions of AUROC and PRAUC on 12 tabular datasets in the form of box plots, and the average score is represented by a line graph. Note that the long whiskers in the box plot are due to the differences in the datasets. As in the previous experiments, we report the results after ten trials per dataset with different random seed settings.

1) Number of Hidden Neurons: The average AUROC and PRAUC increase as the number of hidden neurons increases. However, neurons more than 256 have no significant advantage in the results and tend to be saturated.

2) Depth of Hidden Layers: We fixed the number of hidden neurons to 128 and increased the number of hidden layers. The UNN with a single hidden layer had the highest scores and decreased as more layers were stacked.

3) Type of Activation Functions: We changed the activation function of the hidden layers to introduce different nonlinearity into the neural networks. As can be seen, sigmoid and tanh show similar performance, and the family of ReLU activation functions offers lower performance except for the exponential linear unit (ELU).

4) Dimension of Random Projection: We could obtain an ensemble effect by using the Mahalanobis distance on d'-dimensional output. We increase the output dimension (i.e., the number of output neurons) from 1 to 300. The average score increased with increasing dimensionality but became saturated as the number of hidden neurons increased. Using more than 100 output neurons has no significant advantage in the score.

E. Effect of Mahalanobis Distance

The Mahalanobis distance is used in AD to quantify the distance of an anomaly score from the center of the anomaly score distribution, taking into account the shape and orientation of the

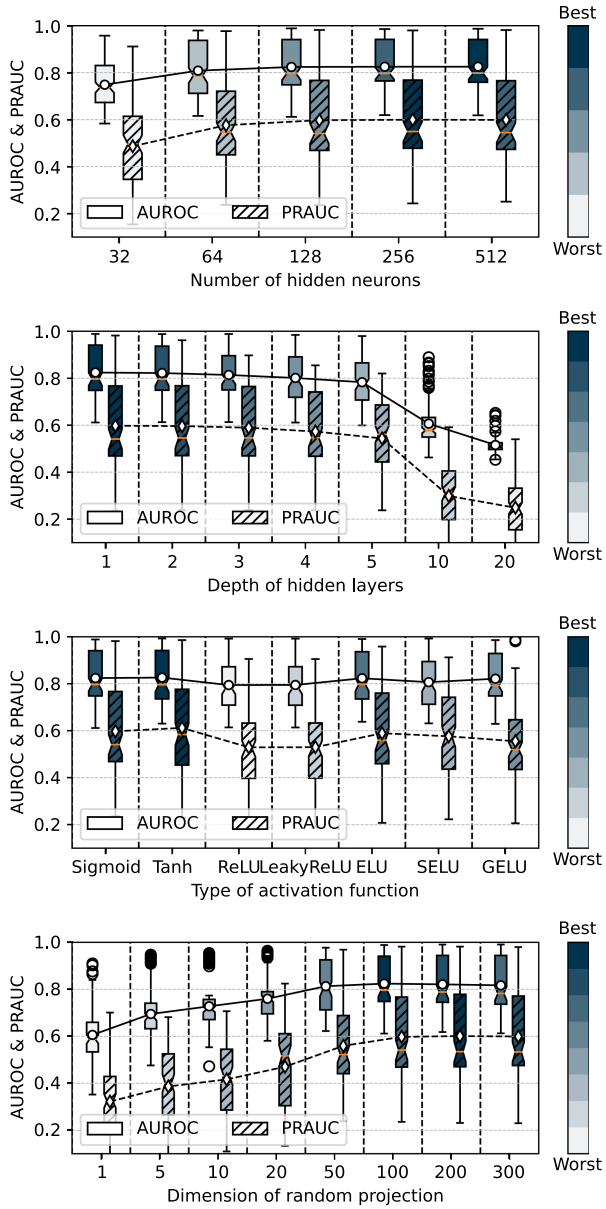

Fig. 3. AUROC and PRAUC with respect to modifications in hyperparameters of the ORM. Colors indicate the rank of the average score: the number of hidden neurons (top), the depth of hidden layers (second), the activation function (third), and the number of output neurons (bottom).

distribution. It assigns higher anomaly scores to points that are further away from the mean of the normal data distribution, effectively discriminating between normal and anomalous points. The effectiveness of Mahalanobis distance was investigated according to anomaly scoring methods based on UNNs. Four scoring methods were considered as follows: the reconstruction error ($\|\mathbf{y}' - \mathbf{x}'\|^2$), the norm score ($\|\mathbf{y}'\|^2$), the inner product to the average ($\mathbf{y}' \cdot \bar{\mathbf{y}}$, used in [42]), and the Euclidean distance to the average ($\|\mathbf{y}' - \bar{\mathbf{y}}\|$), where $\bar{\mathbf{y}}$ denotes the average vector of the reference output.

For the case of reconstruction error, the number of output neurons was changed to d in order to match the dimension. Table IV summarizes the averaged AUROC and PRAUC for different scoring methods. As can be seen, all scores showed average AUROC values above 0.5, but the norm and inner product-based

TABLE IV
DIFFERENCES IN AUROC AND PRAUC, ALONG WITH THEIR STANDARD DEVIATIONS, ACCORDING TO THE ANOMALY SCORING FUNCTIONS WHEN USING THE UNNS BASED ON THE ORM

Metrics	One-shot Random Model				
	R.E.	N.S.	I.P.	E.D.	M.D.
AUROC(%)	75.0 ±1.9	64.1 ±11.6	50.8 ±15.1	74.2 ±1.7	**82.5** ±0.6
PRAUC(%)	43.4 ±1.7	36.4 ±8.7	28.7 ±9.1	42.7 ±2.4	**59.8** ±1.0

N.S.: Norm score, R.E.: Reconstruction error, I.P.: Inner product, E.D.: Euclidean distance, M.D.: Mahalanobis distance.
Bold: The Best.

scores had relatively lower performance and higher deviation. The reconstruction error and the Euclidean distance showed similar values above 0.7. However, the performance was further improved with the Mahalanobis distance, which played an essential role in modeling the normal distribution.

According to the above experiments, a simple neural network (single hidden layer, sigmoid activation function, 128 hidden neurons, and 100 output neurons) has demonstrated sufficient AD performance and can be used as a good starting point. This structural simplicity also leads to computational advantages compared to more complex deep learning models.

VI. RESULTS AND DISCUSSION

A. Comparison to Various AD Models

The proposed ORM was compared to various machine learning and deep-AD models in terms of AUROC and PRAUC. Tables V and VI show the results for AUROC and PRAUC, respectively; the highest values are in bold, and the second highest values are underlined. For each dataset, the distribution of the anomaly scores of the normal and abnormal classes obtained from ORM is shown in Table VII.

Given the random nature and algorithmic simplicity, the limitations of the proposed model in AD performance seemed obvious. Interestingly, however, the ORM performed consistently well in terms of both AUROC and PRAUC, achieving the second highest average score across all datasets. Figs. 4 and 5 illustrate the performance differences of the ORM in percentage points relative to the score of each AD model. The relative score difference has positive values in most cases, indicating that the scores are lower than those of the ORM. Among the benchmark AD methods, the LUNAR models achieved the highest average AUROC and PRAUC in most cases. In addition, the ORM achieved a lower average standard deviation compared to deep-AD models; the standard deviation of AUROC and PRAUC are 0.6 and 1.0 for the ORM, while those of deep-AD models range from 0.5 to 7.8 for AUROC and 0.8 to 7.3 for PRAUC. This indicates that the training process of neural networks may have a greater impact on the performance deviation compared to random mapping.

Overall, the results demonstrate that even UNNs could exhibit reasonable AD performance compared to the existing methods, considering the simplicity of implementation and *training-free* and random characteristics. In this sense, the proposed model shows the potential as a baseline for the development of new AD models. Moreover, the total time to obtain AUROC and PRAUC

TABLE V
COMPARISON OF AUROC (%) OF THE PROPOSED UNN BASED ON THE ORM TO THE BENCHMARK AD MODELS

Dataset	Machine learning-based AD					Deep learning-based AD							UNN
	LOF	CBLOF	ISOF	HBOS	PCA	DAGMM	DSVDD	GANomaly	RDP	RCA	ICL	LUNAR	ORM
Average	76.8 ±0.6	73.0 ±1.0	75.8 ±1.7	72.4 ±0.5	75.0 ±0.3	59.6 ±7.8	61.4 ±7.6	72.0 ±7.1	77.7 ±1.9	76.1 ±0.5	73.2 ±3.3	**83.1** ±0.8	82.5 ±0.6
backdoor	**95.9** ±0.1	70.9 ±2.4	75.0 ±2.6	71.3 ±0.3	64.4 ±0.2	50.3 ±17.3	92.3 ±1.4	85.2 ±6.7	92.6 ±1.0	72.9 ±0.5	95.8 ±1.8	95.2 ±0.2	93.4 ±0.1
campaign	67.6 ±0.6	76.7 ±0.3	73.7 ±1.5	77.4 ±0.3	76.9 ±0.3	62.3 ±2.9	58.0 ±5.4	73.1 ±1.6	75.1 ±1.5	77.2 ±0.5	**82.4** ±0.6	72.2 ±0.8	78.7 ±0.4
celeba	47.7 ±0.3	77.0 ±1.9	71.3 ±1.5	76.1 ±0.1	**79.9** ±0.0	63.1 ±3.7	62.5 ±14.4	57.3 ±10.3	71.2 ±3.0	78.5 ±0.3	57.4 ±10.1	62.9 ±0.3	79.4 ±0.4
census	61.0 ±0.2	71.0 ±0.2	62.5 ±2.6	62.4 ±0.1	70.7 ±0.1	52.9 ±2.1	49.9 ±6.6	61.7 ±6.4	70.1 ±1.0	**71.6** ±0.1	65.9 ±1.0	67.2 ±0.1	71.6 ±0.3
donors	98.6 ±0.1	93.4 ±0.3	89.5 ±1.9	79.6 ±0.6	88.6 ±0.1	57.8 ±18.7	57.0 ±22.9	64.4 ±26.1	99.0 ±0.4	92.5 ±0.5	99.4 ±0.7	**100.0** ±0.0	98.2 ±0.4
eopt	64.0 ±0.2	59.6 ±0.2	60.2 ±0.9	55.0 ±0.1	58.2 ±0.2	49.9 ±2.7	50.0 ±5.7	54.6 ±1.3	62.2 ±1.0	63.4 ±0.4	61.6 ±1.7	**77.7** ±2.8	65.2 ±0.5
fraud	67.6 ±3.4	95.3 ±0.5	95.0 ±0.2	95.2 ±0.2	95.4 ±0.0	85.5 ±5.1	83.6 ±6.9	93.2 ±1.7	95.6 ±0.4	95.2 ±0.1	93.2 ±1.6	**96.8** ±0.1	95.9 ±0.1
mi-f	80.8 ±0.6	52.8 ±1.8	83.6 ±1.9	83.2 ±0.3	**84.0** ±0.3	59.5 ±11.3	54.7 ±12.3	79.8 ±9.5	64.0 ±4.1	68.0 ±0.8	45.1 ±3.2	80.9 ±0.8	79.1 ±1.9
mi-v	90.8 ±0.4	73.0 ±1.0	83.7 ±1.2	83.4 ±0.7	81.1 ±0.4	59.3 ±4.3	58.5 ±5.8	80.1 ±2.6	84.3 ±2.1	80.5 ±0.4	83.3 ±2.7	**95.4** ±0.4	87.5 ±0.5
nasa	70.3 ±0.6	68.3 ±0.7	64.8 ±2.0	50.6 ±1.4	53.4 ±0.9	52.0 ±6.6	55.6 ±5.2	64.8 ±5.3	67.6 ±3.4	63.0 ±0.7	63.0 ±5.6	**72.2** ±1.1	63.7 ±1.4
rarm	84.9 ±0.6	64.3 ±0.7	75.2 ±2.1	71.1 ±0.4	76.0 ±0.3	58.6 ±8.9	31.1 ±1.1	64.0 ±11.2	66.6 ±2.4	75.4 ±0.7	51.1 ±2.9	**97.9** ±1.5	97.4 ±0.4
thyroid	**92.2** ±0.3	74.0 ±1.6	75.2 ±2.5	63.0 ±1.1	71.0 ±1.0	63.7 ±9.5	81.1 ±4.0	86.4 ±2.4	83.4 ±2.2	74.7 ±1.1	80.3 ±8.2	77.4 ±1.8	79.3 ±0.7

Bold: The best. Underlined: The second best.

TABLE VI
COMPARISON OF PRAUC (%) OF THE PROPOSED UNN BASED ON THE ORM TO THE BENCHMARK AD MODELS

Dataset	Machine learning-based AD					Deep learning-based AD							UNN
	LOF	CBLOF	ISOF	HBOS	PCA	DAGMM	DSVDD	GANomaly	RDP	RCA	ICL	LUNAR	ORM
Average	54.9 ±0.7	42.9 ±1.3	42.7 ±2.3	41.3 ±0.9	43.6 ±0.5	31.8 ±6.5	44.1 ±6.5	44.3 ±7.3	54.1 ±2.9	44.9 ±0.8	56.1 ±3.2	**64.0** ±1.2	59.8 ±1.0
backdoor	77.0 ±1.5	14.1 ±1.1	14.2 ±1.6	12.9 ±0.1	12.1 ±0.1	10.0 ±3.9	86.0 ±1.9	32.8 ±11.3	56.4 ±6.7	15.2 ±0.2	**91.5** ±1.1	90.9 ±0.3	86.6 ±0.7
campaign	47.4 ±0.6	59.9 ±0.5	57.3 ±1.8	61.5 ±0.5	60.2 ±0.4	45.0 ±4.6	45.7 ±4.3	56.2 ±2.0	57.3 ±1.9	58.4 ±0.7	**62.7** ±1.3	54.8 ±1.0	61.7 ±0.6
celeba	6.5 ±0.1	22.2 ±5.1	19.0 ±1.2	24.5 ±0.2	**29.6** ±0.2	13.2 ±3.5	13.7 ±5.7	12.4 ±5.8	14.9 ±1.8	22.8 ±0.6	10.3 ±2.9	11.1 ±0.1	24.8 ±0.6
census	22.0 ±0.1	29.8 ±0.3	21.6 ±1.4	21.1 ±0.0	29.5 ±0.1	19.9 ±0.9	21.3 ±3.5	22.6 ±2.6	28.9 ±0.9	30.5 ±0.2	25.5 ±0.6	24.6 ±0.1	**30.9** ±0.6
donors	87.9 ±0.8	58.5 ±0.9	52.9 ±5.4	36.5 ±2.6	48.2 ±0.1	25.9 ±11.9	42.2 ±17.9	33.7 ±24.9	90.7 ±2.8	57.0 ±1.3	98.9 ±0.9	**100.0** ±0.0	84.7 ±2.1
eopt	50.1 ±0.3	35.2 ±0.2	36.7 ±0.6	33.4 ±0.1	35.9 ±0.2	30.3 ±1.2	36.8 ±4.3	33.2 ±0.9	42.6 ±1.0	38.7 ±0.3	47.3 ±1.0	**61.7** ±3.3	46.0 ±0.4
fraud	1.3 ±0.1	33.9 ±1.6	29.6 ±4.0	47.1 ±3.0	33.9 ±1.4	20.9 ±24.1	55.7 ±17.6	62.0 ±10.4	**69.1** ±6.3	39.9 ±1.5	62.7 ±6.8	65.3 ±2.1	57.3 ±2.6
mi-f	60.9 ±1.5	33.4 ±1.3	52.0 ±3.9	50.4 ±0.5	51.7 ±0.8	33.5 ±5.8	33.9 ±7.6	49.0 ±6.7	41.2 ±2.0	40.6 ±0.8	34.5 ±1.8	**63.9** ±1.4	49.8 ±1.5
mi-v	83.4 ±0.8	62.7 ±0.8	67.9 ±2.2	69.3 ±1.2	62.9 ±0.6	49.8 ±2.6	56.2 ±4.8	65.4 ±2.5	72.9 ±2.3	66.6 ±0.5	76.8 ±1.7	**89.0** ±0.7	73.8 ±0.8
nasa	**58.7** ±1.0	55.1 ±1.1	50.5 ±1.5	39.0 ±1.2	43.2 ±0.7	41.5 ±4.7	42.8 ±3.3	51.5 ±4.4	55.1 ±3.7	49.8 ±1.0	52.9 ±5.1	58.0 ±1.1	50.9 ±1.1
rarm	86.7 ±0.5	61.7 ±0.3	69.1 ±0.9	66.2 ±0.4	70.0 ±0.4	55.0 ±5.2	37.8 ±0.4	62.2 ±9.4	64.8 ±1.1	68.0 ±0.6	51.8 ±4.6	**98.2** ±1.2	97.9 ±0.3
thyroid	**77.6** ±1.2	48.8 ±2.2	41.0 ±3.6	33.5 ±0.9	45.9 ±1.3	37.1 ±9.8	52.8 ±6.4	50.4 ±6.6	55.3 ±4.0	51.1 ±1.4	57.8 ±10.1	47.8 ±2.7	52.9 ±1.2

Bold: The best. Underlined: The second best.

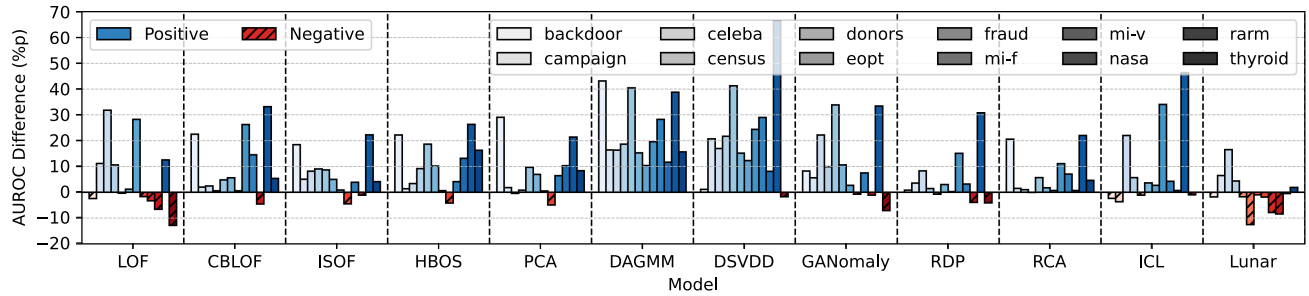

Fig. 4. Relative AUROC of UNN based on the ORM to those of the benchmark AD models. Positive values indicate that the UNN performs better.

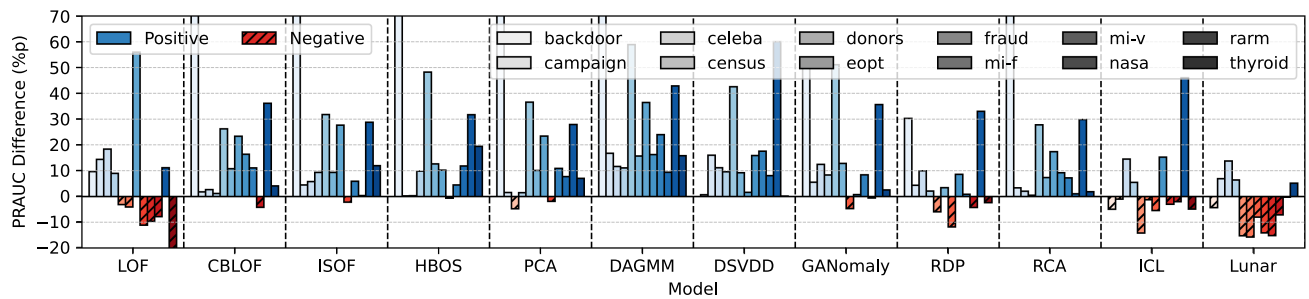

Fig. 5. Relative PRAUC of UNN based on the ORM to those of the benchmark AD models. Positive values indicate that the UNN performs better.

TABLE VII
MEAN AND STANDARD DEVIATION OF ANOMALY SCORE DISTRIBUTION

Dataset	Anomaly Score (mean ±std) Normal Class		Abnormal Class		Performance (%) AUROC	PRAUC
backdoor	0.72	±1.26	11.12	±4.47	93.4	87.4
campaign	0.87	±0.52	1.63	±1.06	78.9	62.2
celeba	0.92	±0.41	1.48	±0.59	79.7	24.9
census	0.86	±0.52	1.24	±0.52	71.5	30.6
donors	0.72	±0.70	3.05	±1.51	98.4	85.0
eopt	0.92	±0.38	1.36	±1.50	65.5	46.6
fraud	0.71	±0.67	7.90	±4.38	95.8	59.8
mi-f	0.81	±0.66	1.53	±1.91	80.8	51.1
mi-v	0.78	±0.61	1.37	±1.09	86.9	73.4
nasa	0.87	±0.53	1.11	±0.73	63.9	51.3
rarm	1.05	±0.37	24.03	±19.48	96.9	97.4
thyroid	0.74	±0.97	2.51	±3.28	79.0	52.2

Result on the experiment of the specific seed (0).

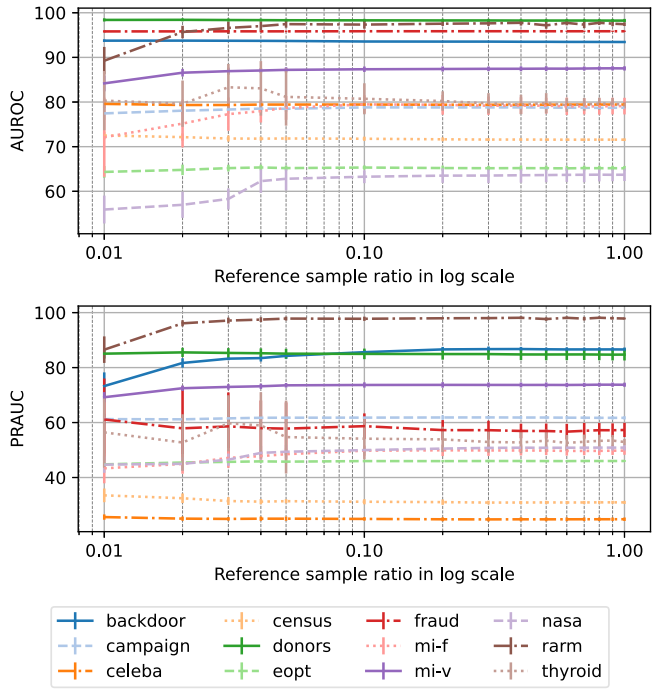

Fig. 6. AUROC and PRAUC with the relative size to the reference set.

using UNNs is significantly faster, about 43 times for the best performing machine learning model, LOF, and 218 times for the best performing deep learning model, LUNAR. For these tests, only the CPU (AMD Ryzen 5 3600) was used for UNNs and machine learning models, while both the CPU and the GPU were used for deep learning models. The longer time of other models is mainly due to the training process. This implies that using UNNs without requiring a training process can also serve as a practical and efficient alternative for AD tasks, particularly in scenarios where computational resources are constrained or where the model needs to adapt continuously or periodically to a changing normal environment.

B. Application of UNNs for AD in Image Datasets

The AD performance of using UNNs in two image datasets (MNIST and Fashion MNIST) was explored. For the image datasets, we adopt a slightly different setup. Initially, all data

TABLE VIII
COMPARISON OF AUROC (PRAUC) ON THE MNIST DATASET

Class	Machine learning-based AD LOF	ISOF	PCA	Deep learning-based AD GANomaly	RCA	ICL	LUNAR	UNN ORM
Average	97.5 (97.5)	85.0 (84.3)	88.1 (87.9)	56.7 (59.7)	82.5 (83.7)	96.0 (96.0)	95.4 (95.4)	94.4 (94.5)
0	99.6 (99.6)	95.4 (93.8)	96.1 (94.3)	96.9 (96.6)	79.6 (81.3)	99.6 (99.5)	99.4 (99.0)	99.0 (98.7)
1	99.5 (99.3)	99.3 (99.1)	98.7 (98.3)	88.3 (88.8)	99.4 (99.3)	99.5 (99.4)	99.7 (99.5)	99.8 (99.7)
2	95.2 (95.6)	73.0 (74.9)	80.3 (80.8)	36.3 (42.5)	70.0 (74.2)	92.7 (92.3)	91.7 (92.6)	89.6 (90.7)
3	97.2 (97.0)	81.1 (80.9)	83.0 (82.9)	42.3 (46.5)	78.8 (80.4)	92.4 (92.1)	93.6 (93.5)	93.6 (92.6)
4	96.7 (97.0)	87.0 (86.8)	87.8 (86.2)	56.1 (58.8)	84.1 (85.7)	97.3 (97.8)	94.3 (94.3)	93.6 (94.3)
5	98.0 (97.9)	74.4 (73.8)	73.8 (74.3)	52.5 (54.2)	79.4 (79.7)	90.6 (90.3)	95.3 (94.6)	95.0 (94.2)
6	99.8 (99.8)	88.8 (85.8)	95.1 (94.9)	44.2 (47.8)	85.8 (86.4)	99.3 (99.2)	98.0 (98.0)	98.1 (98.0)
7	97.8 (97.8)	90.7 (90.3)	94.2 (94.5)	59.3 (61.9)	90.5 (91.3)	96.0 (96.4)	96.8 (97.1)	96.6 (96.7)
8	93.0 (93.3)	73.5 (72.2)	80.4 (80.8)	36.0 (42.0)	71.6 (72.4)	95.1 (94.8)	89.4 (89.5)	83.3 (84.5)
9	97.9 (97.7)	87.2 (85.1)	91.8 (92.0)	55.1 (58.4)	86.0 (86.0)	97.7 (97.7)	96.1 (95.8)	95.7 (95.4)

TABLE IX
COMPARISON OF AUROC (PRAUC) ON THE FASHION MNIST DATASET

Class	Machine learning-based AD LOF	ISOF	PCA	Deep learning-based AD GANomaly	RCA	ICL	LUNAR	UNN ORM
Average	89.4 (89.6)	91.8 (91.7)	90.7 (90.5)	54.8 (60.4)	80.4 (83.3)	92.9 (93.9)	92.6 (92.9)	90.9 (91.6)
T-shirt	83.1 (80.1)	91.1 (90.3)	89.0 (87.9)	92.4 (92.1)	83.1 (83.6)	91.5 (91.0)	92.2 (91.6)	90.3 (89.7)
Trouser	95.1 (94.6)	97.8 (96.3)	98.3 (98.0)	58.8 (69.2)	91.3 (92.0)	98.6 (98.3)	99.1 (98.9)	98.6 (98.4)
Pullover	86.1 (86.6)	87.2 (87.9)	84.6 (85.9)	36.4 (40.1)	73.9 (75.8)	90.6 (92.5)	89.6 (89.9)	88.7 (89.0)
Dress	91.4 (91.7)	93.7 (94.4)	93.8 (94.6)	51.4 (58.7)	83.1 (86.9)	93.9 (94.1)	94.8 (95.2)	92.1 (92.7)
Coat	91.5 (92.9)	91.1 (91.7)	90.4 (91.5)	28.2 (38.4)	70.4 (75.8)	92.4 (93.9)	91.2 (92.5)	88.6 (90.6)
Sandal	88.9 (91.5)	92.6 (94.2)	92.5 (94.5)	84.9 (86.8)	88.1 (90.5)	88.6 (92.3)	92.8 (94.4)	88.3 (91.3)
Shirt	81.6 (82.1)	80.3 (81.2)	76.7 (76.5)	46.3 (45.4)	73.9 (74.4)	81.9 (85.6)	83.2 (83.9)	82.3 (83.2)
Sneaker	96.8 (97.5)	98.1 (98.3)	98.4 (98.6)	75.3 (81.5)	94.3 (95.7)	98.5 (98.8)	98.6 (98.8)	98.4 (98.6)
Bag	82.4 (81.0)	88.5 (84.2)	85.4 (79.0)	35.0 (41.6)	65.5 (72.5)	94.7 (93.3)	87.4 (86.3)	84.4 (84.2)
Ankle Boot	97.7 (97.9)	97.8 (98.2)	98.3 (98.5)	39.4 (50.4)	80.2 (85.4)	98.7 (99.0)	96.7 (97.3)	97.3 (97.7)

undergo min-max normalization, which modifies the grayscale values from a range of (0, 255) to a range of (0, 1). Subsequently, we assign a specific class as "normal" while treating the remaining nine classes as "abnormal." For instance, in *mnist*, if we designate "0" as the normal class, then the remaining classes from "1" through "9" are considered abnormal. We then split the normal data in the same way as we did for the tabular datasets, designating 70% of the data as the reference set and 30% as the normal test set. We randomly sampled data of the abnormal classes for the abnormal test set, ensuring an equal number of abnormal and normal test samples.

For performance comparison, we selected the several models that ranked first or second in Table V. These models represent the best performing models across the various datasets and provide a robust benchmark for evaluating the performance of our proposed method. The results of this comparison are presented in Tables VIII and IX for MNIST and Fashion MNIST, respectively. The proposed UNN method offers reasonable and competitive performance in AD in both image datasets, as evidenced by its comparable AUROC and PRAUC scores with both traditional and deep-learning-based models. However, the performance is limited in datasets with complex labels and significant pixel variation, as seen in color images where the same class may show diverse visual features. Addressing this in more intricate datasets remains a focus for future work.

C. Reference Set Size and Contamination

1) Reference Set Size: The influence of reference sample size on AD performance was analyzed. Initially, the reference set contains 70% of the normal data, and the remaining 30% is used to compute the AUROC for comparison as the test set. Keeping the test set constant, we modify the size of the reference set for the computation of the Mahalanobis distance. Fig. 6 illustrates the variations in AUROC according to changes in the size of the

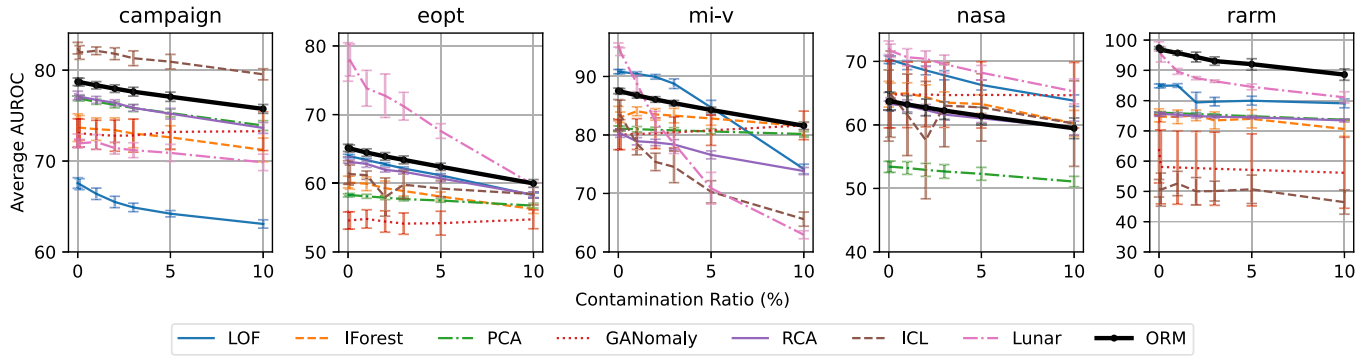

Fig. 7. Impact of the data contamination ratio in the reference set on AUROC (%) of the UNN based on the ORM.

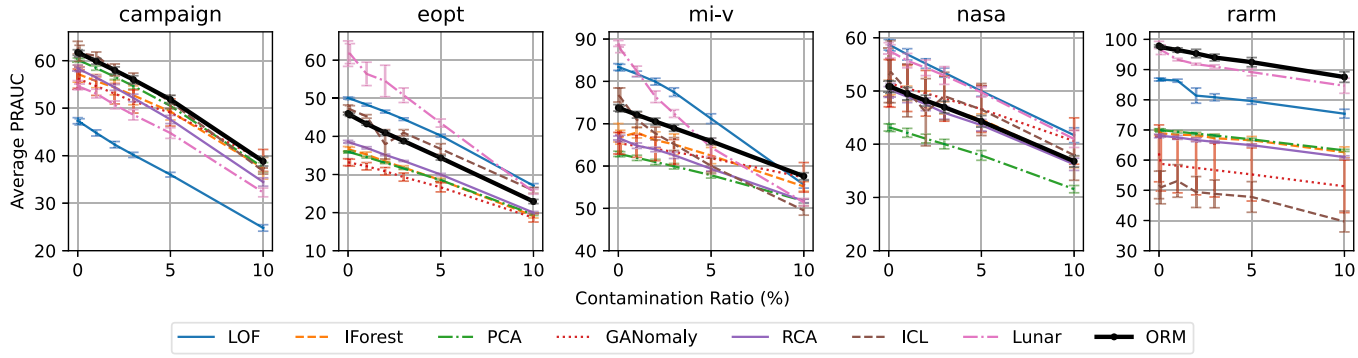

Fig. 8. Impact of the data contamination ratio in the reference set on PRAUC (%) of the UNN based on the ORM.

reference set, where the horizontal axis illustrates the ratio to the initial reference set. Significantly, the AUROC largely remains consistent, with a few exceptions. For instance, *rarm*, *mi-f*, *mi-v*, and *nasa* exhibit a decrease in AUROC when using less than 10% of the initial reference set. Typically, enhancing the performance of learning-based models necessitates increasing the amount of training data. However, the proposed method, which does not incorporate a learning process, largely maintains its performance even when the size of the reference sample decreases. This suggests that our approach is robust and can offer advantages even when data availability is limited.

2) Data Contamination: In real situations, the reference set may be contaminated with anomalies and outliers, making AD more difficult. To investigate the effects of such contamination, an experiment was carried out to explore the decline in performance relative to the contamination ratio for five datasets (*campaign*, *eopt*, *mi-v*, *nasa*, and *rarm*). The ratio represents the percentage of anomalies in the reference set, and 0.1%, 1%, 2%, 3%, 5%, and 10% were applied in this experiment. This ratio is denoted on the horizontal axes of Figs. 7 and 8. The abnormal data used for the contamination are excluded from the test set.

The average AUROC and PRAUC across the five datasets are depicted in Figs. 7 and 8, respectively. The standard deviation is represented as error bars. Each chart presents the results of each dataset using eight models, consistent with the experiment conducted for the image datasets. Both the AUROC and PRAUC generally exhibit a linear decrease as contamination increases. The PRAUC shows a more drastic decrease compared to the AUROC. Despite this degradation, the performance ranking of the models largely remains the same. These results suggest that our proposed model is affected by contamination to a similar degree as other models. A key implication of this, especially when combined with earlier findings regarding the reference set size, is that securing a contamination-free subset of normal data can ensure a comparable level of performance.

D. Relation to Unsupervised ELM

The initial motivation for the proposed model originates from the question of the inherent AD capability of neural networks. Thus, all weight parameters remain untrained in the proposed UNN AD model. However, when the Mahalanobis distance comes with the UNN for anomaly scoring, the proposed model operates in a similar context to the unsupervised ELM (US-ELM) proposed in [41]. If we denote the input data after the first hidden layer as $H \in \mathbb{R}^{N\times d_h}$, the US-ELM sets the weights of the output layer with d_o smallest eigenvectors by solving the generalized eigenvalue problem $A\nu = \lambda B\nu$, and the standard eigenvalue problem is a special case of $B = I$. The generalized eigenvalue problem of US-ELM is

$$(I_{d_h} + \gamma H^T L H)\nu = \lambda H^T H \nu. \tag{6}$$

In detail, I_{d_h} is the identity matrix of size d_h, γ is the regularization term, and L is graph Laplacian built from the training data X. Also, the Mahalanobis distance is the distance to the origin after projection to the principal components, i.e., the eigenvectors of the covariance matrix. In this regard, if we denote the zero-mean output matrix of the UNN as $\tilde{H} \in \mathbb{R}^{N\times d_o}$, the

TABLE X
AUROC AND PRAUC OF US-ELM AND UNN BASED ON THE ORM

Dataset	AUROC				PRAUC			
	US-ELM		ORM		US-ELM		ORM	
Average	80.1	±0.7	82.5	±0.6	57.5	±1.1	59.8	±1.0
backdoor	91.9	±0.4	93.4	±0.1	60.8	±3.4	86.6	±0.7
campaign	74.4	±0.3	78.7	±0.4	57.6	±0.3	61.7	±0.6
celeba	80.0	±0.2	79.4	±0.4	22.9	±0.6	24.8	±0.6
census	68.1	±0.7	71.6	±0.3	26.3	±0.7	30.9	±0.6
donors	99.3	±0.1	98.2	±0.4	91.8	±1.0	84.7	±2.1
eopt	62.5	±0.5	65.2	±0.5	40.6	±0.6	46.0	±0.4
fraud	95.9	±0.2	95.9	±0.1	77.1	±1.0	57.3	±2.6
mi-f	60.3	±2.9	79.1	±1.9	37.0	±1.5	49.8	±1.5
mi-v	81.9	±1.5	87.5	±0.5	70.7	±1.2	73.8	±0.8
nasa	70.5	±0.7	63.7	±1.4	57.1	±0.9	50.9	±1.1
rarm	97.6	±0.2	97.4	±0.4	97.9	±0.2	97.9	±0.3
thyroid	78.7	±0.5	79.3	±0.7	50.0	±1.2	52.9	±1.2

proposed model solves a standard eigenvalue problem of

$$\tilde{H}^T \tilde{H} \nu = \lambda \nu. \tag{7}$$

The US-ELM adopts graph Laplacian for feature representation; thus, it explicitly reflects data structure in raw space. However, when the amount of data grow, the creation of graph Laplacian is affected by computational resources since it requires pairwise distance calculation, which is associated with the $N \times N$ shaped matrix. On the other hand, the problem becomes simplified in the proposed method because it deals with $\tilde{H}^T \tilde{H}$ having the size of $d_o \times d_o$, where $d_o \ll N$ holds in general. Even with simplification, the proposed UNN AD model exhibited better results, which can be shown in Table X. Note that the number of training datasets is limited to 10 000 for US-ELM to solve (6).

VII. CONCLUSION

In this article, we explored the inherent AD capabilities of UNNs. To do so, the UNNs project data nonlinearly onto a random space. Next, the Mahalanobis distance between a point and the reference distribution in this random space is used as an anomaly score. The integration of these two methods achieves effective AD performance even without the training process that is essential in modern learning-based AD models. We validated the results through extensive experiments on various datasets with established AD models, including state-of-the-art deep learning models. The proposed model demonstrated competitive performance by achieving second best results in terms of AUROC and PRAUC. Furthermore, despite the random nature of the UNN, it demonstrated: 1) lower standard deviation; 2) maintained performance with limited data; and 3) robustness against contamination. These results highlight the AD ability of the proposed model.

Optimizing deep-learning- and machine-learning-based AD models can be resource intensive. However, our UNN-based AD approach offers good performance without the need for extensive training or complex design. In this sense, the proposed model can be an effective baseline for the development of AD models, which also underscores the importance of training to ensure higher performance beyond the untrained baseline. Since the proposed model is based on neural networks, it can also be seamlessly integrated with well-established deep learning libraries and environments, offering the potential for incorporation into more advanced deep-AD models. In addition, the simplicity and nontraining characteristics of the UNN offer advantages in scenarios where: 1) computational resources are very limited, such as in edge or sensor-level Internet of Things devices; 2) the model requires frequent updates, which is facilitated by adjusting only the scoring function; and 3) data security is a concern, as the model utilizes randomized data.

Our experiments were primarily conducted on tabular and grayscale image datasets. However, tackling more complex datasets, such as time series, color images, and videos, presents another challenge because of their unique complexity and higher dimensionality. Future work will focus on adapting our approach to those datasets. We are also developing hybrid techniques using trained and UNNs to improve AD performance further.

REFERENCES

[1] A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques," *Int. J. Syst. Assurance Eng. Manage.*, vol. 8, no. 2, pp. 937–953, 2017.

[2] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 146–157.

[3] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 22, no. 1, pp. 949–961, 2019.

[4] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines," *IEEE Access*, vol. 9, pp. 120043–120065, 2021.

[5] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*.

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

[7] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1999, pp. 582–588.

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

[9] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proc. 2nd Workshop Mach. Learn. Sens. Data Anal.*, 2014, pp. 4–11.

[10] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2017, pp. 665–674.

[11] C. Fan, F. Xiao, Y. Zhao, and J. Wang, "Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data," *Appl. Energy*, vol. 211, pp. 1123–1135, 2018.

[12] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for IoT time series," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 52, no. 1, pp. 112–122, Jan. 2022.

[13] K. H. Kim et al., "RaPP: Novelty detection with reconstruction along projection pathway," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–14.

[14] S. Ryu, J. Yim, J. Seo, Y. Yu, and H. Seo, "Quantile autoencoder with abnormality accumulation for anomaly detection of multivariate sensor data," *IEEE Access*, vol. 10, pp. 70428–70439, 2022.

[15] B. Liu, D. Wang, K. Lin, P.-N. Tan, and J. Zhou, "RCA: A deep collaborative autoencoder approach for anomaly detection," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1505–1511.

[16] S. Ryu, B. Jeon, H. Seo, M. Lee, J.-W. Shin, and Y. Yu, "Development of deep autoencoder-based anomaly detection system for hanaro," *Nucl. Eng. Technol.*, vol. 55, no. 2, pp. 475–483, 2023.

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

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

[19] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," 2018, *arXiv:1802.06222*.
[20] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks," *Med. Image Anal.*, vol. 54, pp. 30–44, 2019.
[21] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," in *Proc. 14th Asian Conf. Comput. Vis.*, 2019, pp. 622–637.
[22] X. Han, X. Chen, and L.-P. Liu, "GAN ensemble for anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, pp. 4090–4097.
[23] T. Shenkar and L. Wolf, "Anomaly detection for tabular data with internal contrastive learning," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–26.
[24] A. Goodge, B. Hooi, S.-K. Ng, and W. S. Ng, "LUNAR: Unifying local outlier detection methods via graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, pp. 6737–6745.
[25] L. Ruff et al., "A unifying review of deep and shallow anomaly detection," *Proc. IEEE*, vol. 109, no. 5, pp. 756–795, May 2021.
[26] D.-M. Tsai and P.-H. Jen, "Autoencoder-based anomaly detection for surface defect inspection," *Adv. Eng. Informat.*, vol. 48, Art. no. 101272.
[27] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture IE*, vol. 2, no. 1, pp. 1–18, 2015.
[28] H. Xu et al., "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," in *Proc. World Wide Web Conf.*, 2018, pp. 187–196.
[29] J. Sun, X. Wang, N. Xiong, and J. Shao, "Learning sparse representation with variational auto-encoder for anomaly detection," *IEEE Access*, vol. 6, pp. 33353–33361, 2018.
[30] S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain, "DROCC: Deep robust one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3711–3721.
[31] 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.
[32] M. Bahri, F. Salutari, A. Putina, and M. Sozio, "AutoML: State of the art with a focus on anomaly detection, challenges, and research directions," *Int. J. Data Sci. Anal.*, vol. 14, no. 2, pp. 113–126, 2022.
[33] W. B. Johnson, "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary Math.*, vol. 26, pp. 189–206, 1984.
[34] J. Matoušek, "On variants of the Johnson–Lindenstrauss lemma," *Random Struct. Algorithms*, vol. 33, no. 2, pp. 142–156, 2008.
[35] D. Achlioptas, "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," *J. Comput. Syst. Sci.*, vol. 66, no. 4, pp. 671–687, 2003.
[36] B. Ghojogh, M. Crowley, F. Karray, and A. Ghodsi, *Elements of Dimensionality Reduction and Manifold Learning*. New York, NY, USA: Springer, 2023.
[37] D. L. Donoho and M. Gasko, "Breakdown properties of location estimates based on halfspace depth and projected outlyingness," *Ann. Statist.*, vol. 20, pp. 1803–1827, 1992.
[38] S. M. Erfani, M. Baktashmotlagh, S. Rajasegarar, S. Karunasekera, and C. Leckie, "R1SVM: A randomised nonlinear approach to large-scale anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, pp. 432–438.
[39] M. Bauw, S. Velasco-Forero, J. Angulo, C. Adnet, and O. Airiau, "Deep random projection outlyingness for unsupervised anomaly detection," 2021, *arXiv:2106.15307*.
[40] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.
[41] G. Huang, S. Song, J. N. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.
[42] S. Leroux and P. Simoens, "Sparse random neural networks for online anomaly detection on sensor nodes," *Future Gener. Comput. Syst.*, vol. 144, pp. 327–343, 2022.
[43] H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep isolation forest for anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12591–12604, Dec. 2023.
[44] G. Kim, J. Jang, S. Baek, M. Song, and S.-B. Paik, "Visual number sense in untrained deep neural networks," *Sci. Adv.*, vol. 7, no. 1, 2021, Art. no. eabd6127.
[45] S. Kim, K. Choi, H.-S. Choi, B. Lee, and S. Yoon, "Towards a rigorous evaluation of time-series anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, pp. 7194–7201.
[46] R. Giryes, G. Sapiro, and A. M. Bronstein, "Deep neural networks with random Gaussian weights: A universal classification strategy," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3444–3457, Jul. 2016.
[47] R. Giryes, G. Sapiro, and A. M. Bronstein, "Corrections to "Deep neural networks with random Gaussian weights: A universal classification strategy," 2016 3444-3457 *IEEE Trans. Signal Process.*, vol. 68, pp. 529–531, 2020.
[48] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
[49] 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 Mining*, 2019, pp. 353–362.
[50] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[51] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, nos. 9/10, pp. 1641–1650, 2003.
[52] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," in *Proc. KI-2012: Poster Demo Track*, 2012, vol. 1, pp. 59–63.
[53] C. C. Aggarwal, *An Introduction to Outlier Analysis*. New York, NY, USA: Springer, 2017.
[54] H. Wang, G. Pang, C. Shen, and C. Ma, "Unsupervised representation learning by predicting random distances," in *Proc. 29th Int'l Joint Conf. on Artif. Intell.*, 2021, pp. 2950–2956.
[55] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A python toolbox for scalable outlier detection," *J. Mach. Learn. Res.*, vol. 20, no. 96, pp. 1–7, 2019.

Seunghyoung Ryu (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Sogang University, Seoul, South Korea, in 2014, 2016, and 2020, respectively.

He is currently an Assistant Professor with the Department of Artificial Intelligence and Robotics, Sejong University. He was a Senior Researcher with the Applied Artificial Intelligence Section, Korea Atomic Energy Research Institute, Daejeon, South Korea. His research interests include industrial artificial intelligence and energy information and communications technology, specifically focused on deep learning for anomaly detection and multivariate time-series forecasting.

Yonggyun Yu received the B.S. and Ph.D. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2001 and 2010, respectively.

He is currently the Principal Researcher and Research Director of the Applied Artificial Intelligence Section, Korea Atomic Energy Research Institute, Daejeon. He is also the President of AIFrenz and a full-time Faculty Member in Artificial Intelligence major with the Korea National University of Science and Technology, Daejeon. After completing his Ph.D. research, he joined the Mobile Harbor Center, KAIST, as a Postdoctoral Fellow and Research Assistant Professor. He possesses expertise in the design optimization of musical instruments and nuclear reactor components. His research interests include industrial application of deep learning.

Hogeon Seo received the B.S. degree in mechanical engineering and the Ph.D. degree in convergence mechanical engineering from Hanyang University, Seoul, South Korea, in 2013 and 2018, respectively.

He is currently an Associate Professor of Artificial Intelligence with the Korea National University of Science and Technology, Daejeon, South Korea. He is also a Senior Researcher with Korea Atomic Energy Research Institute, Daejeon. His research interests include AI for prognosis, nondestructive evaluation, and multimodal sensor fusion.

Dr. Seo is the Director of International Affairs with the Korean Society of Nondestructive Testing.