UNSUPERVISED REPRESENTATION LEARNING BY PREDICTING RANDOM DISTANCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks have gained tremendous success in a broad range of machine learning tasks due to its remarkable capability to learn semantic-rich features from high-dimensional data. However, they often require large-scale labelled data to successfully learn such features, which significantly hinders their adaption into unsupervised learning tasks, such as anomaly detection and clustering, and limits their applications into critical domains where obtaining massive labelled data is prohibitively expensive. To enable downstream unsupervised learning on those domains, in this work we propose to learn features without using any labelled data by training neural networks to predict data distances in a randomly projected space. Random mapping is a highly efficient yet theoretical proven approach to obtain approximately preserved distances. To well predict these random distances, the representation learner is optimised to learn class structures that are implicitly embedded in the randomly projected space. Experimental results on 19 real-world datasets show our learned representations substantially outperform state-of-the-art competing methods in both anomaly detection and clustering tasks.

1 INTRODUCTION

Unsupervised representation learning aims at automatically extracting expressive feature representations from raw data without any manually labelled data. Due to the remarkable capability to learn semantic-rich features, deep neural networks have been becoming one widely-used technique to empower a broad range of machine learning tasks. One main issue with these deep learning techniques is that a massive amount of labelled data is typically required to successfully learn these expressive features. As a result, their transformation power is largely reduced for tasks that are unsupervised in nature, such as anomaly detection and clustering. This is also true to critical domains, such as healthcare and fintech, where collecting massive labelled data is prohibitively expensive and/or is impossible to scale. To bridge this gap, in this work we explore fully unsupervised representation learning techniques to enable downstream unsupervised learning methods on those critical domains.

In recent years, many unsupervised representation learning methods (Mikolov et al., 2013a; Le & Mikolov, 2014; Misra et al., 2016; Lee et al., 2017; Gidaris et al., 2018) have been introduced, of which most are self-supervised approach that formulates the problem as an annotation free pretext task. These methods explore easily accessible information, such as temporal or spatial neighbourhood, to design a surrogate supervisory signal to empower the feature learning. These methods have achieved significantly improved feature representations of text/image/video data, but they are often inapplicable to *tabular data* since it does not contain the required temporal or spatial supervisory information. We therefore focus on unsupervised representation learning of high-dimensional tabular data. Although many traditional approaches, such as random projection (Li et al., 2006), principal component analysis (PCA) (Rahmani & Atia, 2017), manifold learning (Donoho & Grimes, 2003; Hinton & Roweis, 2003) and autoencoder (Vincent et al., 2010), are readily available for handling those data, many of them (Donoho & Grimes, 2003; Hinton & Roweis, 2003; Rahmani & Atia, 2017) are often too computationally costly to scale up to large or high-dimensional data. Approaches like random projection and autoencoder are very efficient but they often fail to capture complex class structures due to its underlying data assumption or weak supervisory signal.

In this paper, we introduce a Random Distance Prediction (RDP) model which trains neural networks to predict data distances in a randomly projected space. When the distance information captures intrinsic class structure in the data, the representation learner is optimised to learn the class structure to minimise the prediction error. Since distances are concentrated and become meaningless in high dimensional spaces (Beyer et al., 1999), we seek to obtain distances preserved in a projected space to be the supervisory signal. Random mapping is a highly efficient yet theoretical proven approach to obtain such approximately preserved distances. Therefore, we leverage the distances in the randomly projected space to learn the desired features. We show this simple random distance prediction enables us to achieve expressive representations in a fully unsupervised manner. In addition, we also show some task-dependent auxiliary losses can be optionally added to further enhance the feature representations. In summary, this paper makes the following two main contributions.

- We propose a novel unsupervised representation learning framework that learns features by training neural networks to predict distances in a randomly projected space. Random distances can be obtained very efficiently and well preserve original feature space information, which enables us to learn expressive representations efficiently.
- The framework is instantiated into a model termed RDP to learn representations of high-dimensional tabular data. We show the learned features substantially improve the performance of downstream anomaly detection and clustering tasks on 19 real-world datasets.

2 Random Distance Prediction Model

2.1 The Proposed Framework

We propose to learn representations by training neural networks to predict distances in a randomly projected space without involving any manually labelled data. The key intuition is that, given some distance information that faithfully encapsulates the underlying class structure in the data, the representation learner is forced to learn the class structure in order to yield distances that are as close as the given distances. Therefore, one key ingredient is how to obtain such trustworthy distance information. Also, to efficiently optimise the model, it is important that the distance can be obtained in a computationally efficient way. One simple strategy is to calculate the distance in the original space, but it is ineffective in high-dimensional spaces. In this work, we use the inner products in a randomly projected space as the source of distance/similarity since it is very efficient and there is strong theoretical support of its capacity in preserving the genuine distance information.

Our proposed framework is illustrated in Figure 1. Specifically, given data points $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$, we first feed them into a weight-shared Siamese-style neural network $\phi(\mathbf{x}; \Theta)$. $\phi : \mathbb{R}^D \mapsto \mathbb{R}^M$ is a representation learner with the parameters Θ to map the data onto a M-dimensional new space. Then we formulate the subsequent step as a distance prediction task and define a loss function as:

$$L_{rdp} = l(< \phi(\mathbf{x}_i; \Theta), \phi(\mathbf{x}_j; \Theta) >, < \eta(\mathbf{x}_i), \eta(\mathbf{x}_j) >), \tag{1}$$

where η is an existing data mapping function and $l(\cdot, \cdot)$ is a loss function of the difference between its two input terms. Our extensive experimental results show this loss enables effective learning of expressive representations. In addition, a task-dependent auxiliary loss L_{aux}, such as reconstruction loss (Hinton & Salakhutdinov, 2006) for clustering or novelty loss (Burda et al., 2019) for anomaly detection, may be added to further enhance the representation learning.

2.2 The Instantiated Model

Although our framework is generic and may work on different data modalities, in this work we focus on learning representations of tabular data which is less explored than other types of data. Therefore, the following instantiated model, termed RDP, is tailored for handling tabular data.

$$\arg\min_{\Theta} \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}} L_{rdp} + L_{aux}. \tag{2}$$

The specific random distance prediction loss is defined as[1]:

$$L_{rdp} = (\phi(\mathbf{x}_i; \Theta) \cdot \phi(\mathbf{x}_j; \Theta) - \eta(\mathbf{x}_i) \cdot \eta(\mathbf{x}_j))^2 \tag{3}$$

[1]Since we operate on real-valued vector space, the inner product is implemented by the dot product. The dot product is used hereafter to simplify the notation.

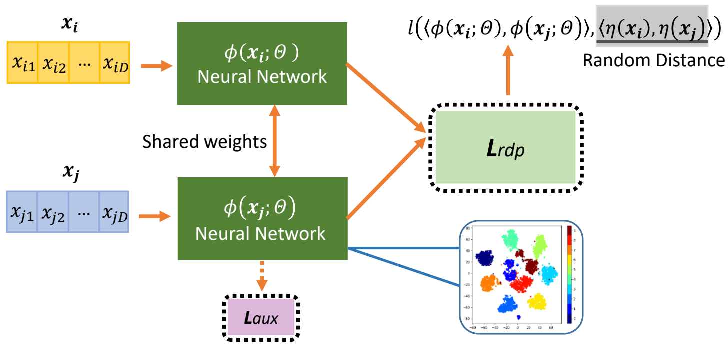

Figure 1: The proposed random distance prediction framework. L_{rdp} is a random distance prediction loss, L_{aux} is a task-dependent auxiliary loss, and η denotes an existing random mapping function.

where ϕ is implemented by multilayer perceptron for dealing with tabular data and $\eta : \mathbb{R}^D \mapsto \mathbb{R}^K$ is an off-the-shelf random data mapping function (see Sections 3.1 and 3.2 for detail).

For clustering, an auxiliary reconstruction loss is defined as:

$$L_{aux}^{clu} = \left(\mathbf{x} - \phi'(\phi(\mathbf{x};\Theta);\Theta')\right)^2, \tag{4}$$

where ϕ is an encoder and $\phi' : \mathbb{R}^M \mapsto \mathbb{R}^D$ is a decoder. This loss may be *optionally* added into RDP to better capture global feature representations.

Similarly, in anomaly detection a novelty loss may be *optionally* added, which is defined as:

$$L_{aux}^{ad} = \left(\phi(\mathbf{x}_i;\Theta) - \eta(\mathbf{x}_i)\right)^2. \tag{5}$$

Since L_{aux}^{ad} involves a mean squared error between two vectors, the dimension of the projected space resulted by ϕ and η is required to be equal in this case. Therefore, when this loss is added into RDP, the M in ϕ and K in η need to be the same. We do not have this constraint in other cases. L_{aux}^{ad} is introduced in (Burda et al., 2019) to capture the novelty of states in the reinforcement learning framework. By using a fixed η, minimising L_{aux}^{ad} helps learn the frequency of underlying patterns in the data. As a result, anomalies or novel points are expected to have substantially larger $\left(\phi(\mathbf{x}_i;\Theta^\star) - \eta(\mathbf{x}_i)\right)^2$ than normal points, so this value can be directly leveraged to detect anomalies. Incorporating L_{aux}^{ad} well complements the pairwise distance information modelled by L_{rdp} as it enables RDP to capture those pattern frequency information which is crucial in anomaly detection.

3 Theoretical Analysis of RDP

In this section, we show the original distance information can be effectively approximated using random distances derived from the inner products in either linearly or non-linearly projected spaces. Also, to accurately predict these distances, the representation learner is forced to learn the class structure embedded in the data.

3.1 When Linear Projection Is Used

Random projection is a simple yet very effective linear feature mapping technique which has proven the capability of distance preservation. Let $\mathcal{X} \subset \mathbb{R}^{N\times D}$ be a set of N data points, random projection uses a random matrix $\mathbf{A} \subset \mathbb{R}^{K\times D}$ to project the data onto a lower K-dimensional space by $\mathcal{X}' = \mathbf{A}\mathcal{X}^\intercal$. The Johnson-Lindenstrauss lemma (Johnson & Lindenstrauss, 1984) guarantees the data points can be mapped to a randomly selected space of suitably lower dimension with the distances between the points are approximately preserved. More specifically, let $\epsilon \in (0, \frac{1}{2})$ and $K = \frac{20 \log n}{\epsilon^2}$. There exists a linear mapping $f : \mathbb{R}^D \mapsto \mathbb{R}^K$ such that for all $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$:

$$(1-\epsilon)||\mathbf{x}_i - \mathbf{x}_j||^2 \leq ||f(\mathbf{x}_i) - f(\mathbf{x}_j)||^2 \leq (1+\epsilon)||\mathbf{x}_i - \mathbf{x}_j||^2. \tag{6}$$

Furthermore, assume the entries of the matrix $\mathbf{A}$ are sampled independently from a Gaussian distribution $\mathcal{N}(0,1)$. Then, the norm of $\mathbf{x} \in \mathbb{R}^D$ can be preserved as:

$$\Pr\left((1-\epsilon)||\mathbf{x}||^2 \leq ||\frac{1}{\sqrt{K}}\mathbf{A}\mathbf{x}||^2 \leq (1+\epsilon)||\mathbf{x}||^2\right) \geq 1 - 2e^{\frac{-(\epsilon^2-\epsilon^3)K}{4}}. \tag{7}$$

Under such random projections, the norm preservation helps well preserve the inner products:

$$\Pr\left(|\hat{\mathbf{x}}_i \cdot \hat{\mathbf{x}}_j - f(\hat{\mathbf{x}}_i) \cdot f(\hat{\mathbf{x}}_j)| \geq \epsilon\right) \leq 4e^{\frac{-(\epsilon^2-\epsilon^3)K}{4}}, \tag{8}$$

where $\hat{\mathbf{x}}$ is a normalised $\mathbf{x}$ such that $||\hat{\mathbf{x}}|| \leq 1$.

The proofs of Eqns. (2)-(4) can be found in (Vempala, 1998).

Eqn. (8) states that the inner products in the randomly projected space can largely preserve the inner products in the original space, particularly when the projected dimension K is large.

3.2 When Non-linear Projection Is Used

Here we show that some non-linear random mapping methods are approximate to kernel functions which are a well-established approach to obtain reliable distance/similarity information. The key to this approach is the kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, which is defined as $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$, where ψ is a feature mapping function but needs not to be explicitly defined and $\langle \cdot, \cdot \rangle$ denotes a suitable inner product. Given a dataset $\mathcal{X}$, the kernel function yields a $|\mathcal{X}| \times |\mathcal{X}|$ kernel matrix where each entry represents the similarity between a pair of data points. A non-linear kernel function such as polynomial or radial basis function (RBF) kernel is typically used to project linear-inseparable data onto a linear-separable space.

The relation between non-linear random mapping and kernel methods is justified in (Rahimi & Recht, 2008), which shows that an explicit randomised mapping function $g : \mathbb{R}^D \mapsto \mathbb{R}^K$ can be defined to project the data points onto a low-dimensional Euclidean inner product space such that the inner products in the projected space approximate the kernel evaluation:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle \approx g(\mathbf{x}_i) \cdot g(\mathbf{x}_j). \tag{9}$$

Let $\mathbf{A}$ be the mapping matrix. Then to achieve the above approximation, $\mathbf{A}$ is required to be drawn from Fourier transform and shift-invariant functions such as cosine function are finally applied to $\mathbf{A}\mathbf{x}$ to yield a real-valued output. By transforming the two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ in this manner, their inner product $g(\mathbf{x}_i) \cdot g(\mathbf{x}_j)$ is an unbiased estimator of $k(\mathbf{x}_i, \mathbf{x}_j)$.

3.3 Learning Class Structure By Random Distance Prediction

Our model using only the random distances as the supervisory signal can be formulated as:

$$\arg\min_{\Theta} \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}} \left(\phi(\mathbf{x}_i; \Theta) \cdot \phi(\mathbf{x}_j; \Theta) - y_{ij}\right)^2, \tag{10}$$

where $y_{ij} = \eta(\mathbf{x}_i) \cdot \eta(\mathbf{x}_j)$. Let $\mathbf{Y}_\eta \in \mathbb{R}^{N \times N}$ be the distance/similarity matrix of the N data points resulted by η. Then to minimise the prediction error in Eqn. (10), ϕ is optimised to learn the underlying class structure embedded in $\mathbf{Y}$. When $\mathbf{Y}_\eta$ well preserves the distance information, our learner ϕ is driven to learn feature representations that effectively capture the genuine class structure. The properties demonstrated in Eqns. (8) and (9) enable us to effectively learn such feature representations when η is set to be either the random projection-based f function or the kernel method-based g function.

4 Experiments

This section evaluates the learned representations through two typical unsupervised tasks: anomaly detection and clustering. Some preliminary results of classification can be found in Appendix E.

4.1 PERFORMANCE EVALUATION IN ANOMALY DETECTION

4.1.1 EXPERIMENTAL SETTINGS

Our RDP model is compared with five state-of-the-art unsupervised anomaly/novelty detection methods, including iForest (Liu et al., 2008), autoencoder (AE) (Hinton & Salakhutdinov, 2006), REPEN (Pang et al., 2018), DAGMM (Zong et al., 2018) and RND (Burda et al., 2019). iForest and AE are two of the most popular baselines used in anomaly detection. The other three methods are recently proposed methods that learn representations specifically for anomaly detection.

As shown in Table 1, the comparison is performed on 14 publicly available datasets of various domains, including network intrusion, credit card fraud detection, disease detection and bank campaigning. Many of the datasets contain real anomalies, including *DDoS*, *Donors*, *Backdoor*, *Creditcard*, *Lung*, *Probe* and *U2R*. Following (Liu et al., 2008; Pang et al., 2018; Zong et al., 2018), the rare class(es) is treated as anomalies in the other datasets to create semantically real anomalies. The Area Under Receiver Operating Characteristic Curve (AUC-ROC) and the Area Under Precision-Recall Curve (AUC-PR) are used as our performance metrics. Larger AUC-ROC/AUC-PR indicates better performance. The reported performance is averaged over 10 independent runs.

Table 1: AUC-ROC (mean±std) performance of RDP and its five competing methods on 14 datasets.

Data Characteristics				Our Method RDP and Its Five Competing Methods					
Data	**N**	**D**	**Anomaly (%)**	**iForest**	**AE**	**REPEN**	**DAGMM**	**RND**	**RDP**
DDoS	464,976	66	3.75%	0.880 ± 0.018	0.901 ± 0.000	0.933 ± 0.002	0.766 ± 0.019	0.852 ± 0.011	**0.942 ± 0.008**
Donors	619,326	10	5.92%	0.774 ± 0.010	0.812 ± 0.011	0.777 ± 0.075	0.763 ± 0.110	0.847 ± 0.011	**0.962 ± 0.011**
Backdoor	95,329	196	2.44%	0.723 ± 0.029	0.806 ± 0.007	0.857 ± 0.001	0.813 ± 0.035	**0.935 ± 0.002**	0.910 ± 0.021
Ad	3,279	1,555	13.99%	0.687 ± 0.021	0.703 ± 0.000	0.853 ± 0.001	0.500 ± 0.000	0.812 ± 0.002	**0.887 ± 0.003**
Apascal	12,695	64	1.38%	0.514 ± 0.051	0.623 ± 0.005	0.813 ± 0.004	0.710 ± 0.020	0.685 ± 0.019	**0.823 ± 0.007**
Bank	41,188	62	11.26%	0.713 ± 0.021	0.666 ± 0.000	0.681 ± 0.001	0.616 ± 0.014	0.690 ± 0.006	**0.758 ± 0.007**
Celeba	202,599	39	2.24%	0.693 ± 0.014	0.735 ± 0.002	0.802 ± 0.002	0.680 ± 0.067	0.682 ± 0.029	**0.860 ± 0.006**
Census	299,285	500	6.20%	0.599 ± 0.019	0.602 ± 0.000	0.542 ± 0.003	0.502 ± 0.003	**0.661 ± 0.003**	0.653 ± 0.004
Creditcard	284,807	29	0.17%	0.948 ± 0.005	0.948 ± 0.000	0.950 ± 0.001	0.877 ± 0.005	0.945 ± 0.001	**0.957 ± 0.005**
Lung	145	3,312	4.13%	0.893 ± 0.057	0.953 ± 0.004	0.949 ± 0.002	0.830 ± 0.087	0.867 ± 0.031	**0.982 ± 0.006**
Probe	64,759	34	6.43%	0.995 ± 0.001	0.997 ± 0.000	0.997 ± 0.000	0.953 ± 0.008	0.975 ± 0.000	**0.997 ± 0.000**
R8	3,974	9,467	1.28%	0.841 ± 0.023	0.835 ± 0.000	**0.910 ± 0.000**	0.760 ± 0.066	0.883 ± 0.006	0.902 ± 0.002
Secom	1,567	590	6.63%	0.548 ± 0.019	0.526 ± 0.000	0.510 ± 0.004	0.513 ± 0.010	0.541 ± 0.006	**0.570 ± 0.004**
U2R	60,821	34	0.37%	**0.988 ± 0.001**	0.987 ± 0.000	0.978 ± 0.000	0.945 ± 0.028	0.981 ± 0.001	0.986 ± 0.001

Our RDP model uses the optional novelty loss for anomaly detection task by default. Similar to RND, given a data point $\mathbf{x}$, its anomaly score in RDP is defined as the mean squared error between the two projections resulted by $\phi(\mathbf{x}; \Theta^{\star})$ and $\eta(\mathbf{x})$. Also, a boosting process is used to filter out 5% likely anomalies per iteration to iteratively improve the modelling of RDP. This is because the modelling is otherwise largely biased when anomalies are presented. In the ablation study in Section 4.1.3, we will show the contribution of all these components.

4.1.2 COMPARISON TO THE STATE-OF-THE-ART COMPETING METHODS

The AUC-ROC and AUC-PR results are respectively shown in Tables 1 and 2. It is clear that RDP outperforms all the five competing methods in both of AUC-ROC and AUC-PR in at least 12 out of 14 datasets. This improvement is statistically significant at the 95% confidence level according to the two-tailed sign test (Demšar, 2006). Remarkably, RDP obtains more than 10% AUC-ROC/AUC-PR improvement over the best competing method on six datasets, including *Donors*, *Ad*, *Bank*, *Celeba*, *Lung* and *U2R*. RDP can be thought as a high-level synthesis of the two competing methods REPEN and RND, because REPEN leverages a triplet loss with pseudo anomaly and normal class labels to learn representations for anomaly detection while RND is built using L_{aux}^{ad}. In nearly all the datasets, RDP well leverages the distance-based loss L_{rdp} and L_{aux}^{ad} to achieve significant improvement over both REPEN and RND. In very limited cases, such as on datasets *Backdoor* and *Census* where RND performs very well while REPEN performs less effectively, RDP is slightly downgraded due to the use of L_{rdp}. In the opposite case, such as *Probe*, on which REPEN performs much better than RND, the use of L_{aux}^{ad} may drag down the performance of RDP a bit.

Table 2: AUC-PR (mean±std) performance of RDP and its five competing methods on 14 datasets.

Data	iForest	AE	REPEN	DAGMM	RND	RDP
DDoS	0.141 ± 0.020	0.248 ± 0.001	0.300 ± 0.012	0.038 ± 0.000	0.110 ± 0.015	**0.301 ± 0.028**
Donors	0.124 ± 0.006	0.138 ± 0.007	0.120 ± 0.032	0.070 ± 0.024	0.201 ± 0.033	**0.432 ± 0.061**
Backdoor	0.045 ± 0.007	0.065 ± 0.004	0.129 ± 0.001	0.034 ± 0.023	**0.433 ± 0.015**	0.305 ± 0.008
Ad	0.363 ± 0.061	0.479 ± 0.000	0.600 ± 0.002	0.140 ± 0.000	0.473 ± 0.009	**0.726 ± 0.007**
Apascal	0.015 ± 0.002	0.023 ± 0.001	0.041 ± 0.001	0.023 ± 0.009	0.021 ± 0.005	**0.042 ± 0.003**
Bank	0.293 ± 0.023	0.264 ± 0.001	0.276 ± 0.001	0.150 ± 0.020	0.258 ± 0.006	**0.364 ± 0.013**
Celeba	0.060 ± 0.006	0.082 ± 0.001	0.081 ± 0.001	0.037 ± 0.017	0.068 ± 0.010	**0.104 ± 0.006**
Census	0.071 ± 0.004	0.072 ± 0.000	0.064 ± 0.005	0.061 ± 0.001	0.081 ± 0.001	**0.086 ± 0.001**
Creditcard	0.145 ± 0.031	**0.382 ± 0.004**	0.359 ± 0.014	0.010 ± 0.012	0.290 ± 0.012	0.363 ± 0.011
Lung	0.379 ± 0.092	0.565 ± 0.022	0.429 ± 0.005	0.042 ± 0.003	0.381 ± 0.104	**0.705 ± 0.028**
Probe	0.923 ± 0.011	0.964 ± 0.002	**0.964 ± 0.000**	0.409 ± 0.153	0.609 ± 0.014	0.955 ± 0.002
R8	0.076 ± 0.018	0.097 ± 0.006	0.083 ± 0.000	0.019 ± 0.011	0.134 ± 0.031	**0.146 ± 0.017**
Secom	0.106 ± 0.007	0.093 ± 0.000	0.091 ± 0.001	0.066 ± 0.002	0.086 ± 0.002	**0.096 ± 0.001**
U2R	0.180 ± 0.018	0.230 ± 0.004	0.116 ± 0.007	0.025 ± 0.019	0.217 ± 0.011	**0.261 ± 0.005**

4.1.3 Ablation Study

This section examines the contribution of L_{rdp}, L_{aux}^{ad} and the boosting process to the performance of RDP. The experimental results in AUC-ROC are given in Table 3, where RDP\X means the RDP variant that removes the 'X' module from RDP. In the last two columns, *Org_SS* indicates that we directly use the distance information calculated in the original space as the supervisory signal, while *SRP_SS* indicates that we use SRP to obtain the distances as the supervisory signal. It is clear that the full RDP model is the best performer. Using the L_{rdp} loss only, i.e., RDP$\setminus L_{aux}^{ad}$, can achieve performance substantially better than, or comparably well to, the five competing methods in Table 1. This is mainly because the L_{rdp} loss alone can effectively force our representation learner to learn the underlying class structure on most datasets so as to minimise its prediction error. The use of L_{aux}^{ad} and boosting process well complement the L_{rdp} loss on the other datasets.

In terms of supervisory source, RDP and SRP_SS perform substantially better than Org_SS on most datasets. This is because the distances in both the non-linear random projection in RDP and the linear projection in SRP_SS well preserve the distance information, enabling RDP to effectively learn much more faithful class structure than that working on the original space.

Table 3: AUC-ROC (mean±std) performance of RDP and its variants in the anomaly detection task. Similar results can also be observed in AUC-PR in Table 8 in Appendix C.

	Decomposition				Supervision Signal	
Data	**RDP**	**RDP**$\setminus L_{rdp}$	**RDP**$\setminus L_{aux}^{ad}$	**RDP\Boosting**	**Org_SS**	**SRP_SS**
DDoS	**0.942 ± 0.008**	0.852 ± 0.011	0.931 ± 0.003	0.866 ± 0.011	0.924 ± 0.006	0.927 ± 0.005
Donors	**0.962 ± 0.011**	0.847 ± 0.011	0.737 ± 0.006	0.910 ± 0.013	0.728 ± 0.005	0.762 ± 0.016
Backdoor	0.910 ± 0.021	0.935 ± 0.002	0.872 ± 0.012	**0.943 ± 0.002**	0.875 ± 0.002	0.882 ± 0.010
Ad	**0.887 ± 0.003**	0.812 ± 0.002	0.718 ± 0.005	0.818 ± 0.002	0.696 ± 0.003	0.740 ± 0.008
Apascal	**0.823 ± 0.007**	0.685 ± 0.019	0.732 ± 0.007	0.804 ± 0.021	0.604 ± 0.032	0.760 ± 0.030
Bank	**0.758 ± 0.007**	0.690 ± 0.006	0.684 ± 0.004	0.736 ± 0.009	0.684 ± 0.002	0.688 ± 0.015
Celeba	**0.860 ± 0.006**	0.682 ± 0.029	0.709 ± 0.005	0.794 ± 0.017	0.667 ± 0.033	0.734 ± 0.027
Census	0.653 ± 0.004	**0.661 ± 0.003**	0.626 ± 0.006	0.661 ± 0.001	0.636 ± 0.006	0.560 ± 0.006
Creditcard	**0.957 ± 0.005**	0.945 ± 0.001	0.950 ± 0.000	0.956 ± 0.003	0.947 ± 0.001	0.949 ± 0.003
Lung	**0.982 ± 0.006**	0.867 ± 0.031	0.911 ± 0.006	0.968 ± 0.018	0.884 ± 0.018	0.928 ± 0.008
Probe	0.997 ± 0.000	0.975 ± 0.000	**0.998 ± 0.000**	0.978 ± 0.001	0.995 ± 0.000	0.997 ± 0.001
R8	0.902 ± 0.002	0.883 ± 0.006	0.867 ± 0.003	0.895 ± 0.004	0.830 ± 0.005	**0.904 ± 0.005**
Secom	**0.57 ± 0.004**	0.541 ± 0.006	0.544 ± 0.011	0.563 ± 0.008	0.512 ± 0.007	0.530 ± 0.016
U2R	0.986 ± 0.001	0.981 ± 0.001	0.987 ± 0.000	**0.988 ± 0.002**	0.987 ± 0.000	0.981 ± 0.002
#wins/draws/losses (RDP vs.)		13/0/1	13/0/1	12/0/2	10/2/2	6/0/8

4.2 Performance Evaluation in Clustering

4.2.1 Experimental Settings

For clustering, RDP is compared with four state-of-the-art unsupervised representation learning methods in four different areas, including HLLE (Donoho & Grimes, 2003) in manifold learning, Sparse Random Projection (SRP) (Li et al., 2006) in random projection, autoencoder (AE) (Hinton & Salakhutdinov, 2006) in data reconstruction-based neural network methods and Coherence Pursuit (COP) (Rahmani & Atia, 2017) in robust PCA. These representation learning methods are first used to yield the new representations, and K-means (Hartigan & Wong, 1979) is then applied to the representations to perform clustering. Two widely-used clustering performance metrics, Normalised Mutual Info (NMI) score and F-score, are used. Larger NMI or F-score indicates better performance. The clustering performance in the original feature space, denoted as Org, is used as a baseline. As shown in Table 4, five high-dimensional real-world datasets are used. Some of the datasets are image/text data. Since here we focus on the performance on tabular data, they are used as tabular data by treating each pixel as a feature or using bag-of-words representation. The reported NMI score and F-score are averaged over 30 times to address the randomisation issue in K-means clustering.

In this section RDP uses the optional reconstruction loss L_{aux}^{clu} by default, but RDP also works very well without the use of L_{aux}^{clu}. We will discuss this in Section 4.2.3.

4.2.2 Comparison to the-state-of-the-art competing methods

Table 4 shows the NMI and F-score performance of K-means working on the original feature space (Org) and on the projected spaces returned by the five representation learning methods. Our method RDP achieves the best performance on three datasets and ranks in second in the other two datasets. RDP-enabled clustering performs substantially and consistently better than that based on AE in terms of both NMI and F-score. This demonstrates that the random distance loss enables RDP to effectively capture some class structure in the data which cannot be capture by using the reconstruction loss. RDP also consistently outperforms the random projection method, SRP, and the robust PCA method, COP. It is interesting that K-means clustering performs best in the original space on *Sector*. This may be due to that this data contains many relevant features, resulting in no obvious curse of dimensionality issue. *Olivetti* may contain complex manifolds which require extensive neighbourhood information to find them, so only HLLE can achieve this goal in such cases. Nevertheless, RDP performs much more stably than HLLE across the five datasets.

Table 4: NMI and F-score performance of K-means on the original space and projected spaces.

Data Characteristics			NMI Performance					
Data	**N**	**D**	**Org**	**HLLE**	**SRP**	**AE**	**COP**	**RDP**
R8	7,674	17,387	0.524 ± 0.047	0.004 ± 0.001	0.459 ± 0.031	0.471 ± 0.043	0.025 ± 0.003	**0.539 ± 0.040**
20news	18,846	130,107	0.080 ± 0.004	0.017 ± 0.000	0.075 ± 0.002	0.075 ± 0.006	0.027 ± 0.040	**0.084 ± 0.005**
Olivetti	400	4,096	0.778 ± 0.014	**0.841 ± 0.011**	0.774 ± 0.011	0.782 ± 0.010	0.333 ± 0.018	0.805 ± 0.012
Sector	9,619	55,197	**0.336 ± 0.008**	0.122 ± 0.004	0.273 ± 0.011	0.253 ± 0.010	0.129 ± 0.014	0.305 ± 0.007
RCV1	20,242	47,236	0.154 ± 0.000	0.006 ± 0.000	0.134 ± 0.024	0.146 ± 0.010	N/A	**0.165 ± 0.000**
Data Characteristics			**F-score Performance**					
Data	**N**	**D**	**Org**	**HLLE**	**SRP**	**AE**	**COP**	**RDP**
R8	7,674	17,387	0.185 ± 0.189	0.085 ± 0.000	0.317 ± 0.045	0.312 ± 0.068	0.088 ± 0.002	**0.360 ± 0.055**
20news	18,846	130,107	0.116 ± 0.006	0.007 ± 0.000	0.109 ± 0.006	0.083 ± 0.010	0.009 ± 0.004	**0.119 ± 0.006**
Olivetti	400	4,096	0.590 ± 0.029	**0.684 ± 0.024**	0.579 ± 0.022	0.602 ± 0.023	0.117 ± 0.011	0.638 ± 0.026
Sector	9,619	55,197	**0.208 ± 0.008**	0.062 ± 0.001	0.187 ± 0.009	0.184 ± 0.010	0.041 ± 0.004	0.191 ± 0.007
RCV1	20,242	47,236	0.519 ± 0.000	0.342 ± 0.000	0.508 ± 0.003	0.514 ± 0.057	N/A	**0.572 ± 0.003**

4.2.3 Ablation Study

Similar to anomaly detection, this section examines the contribution of the two loss functions L_{rdp} and L_{aux}^{clu} to the performance of RDP, as well as the impact of different supervisory sources on the performance. The F-score results of this experiment are shown in Table 5, in which the notations have exactly the same meaning as in Table 3. The full RDP model that uses both L_{rdp} and L_{aux}^{clu} performs more favourably than its two variants, RDP$\backslash L_{rdp}$ and RDP$\backslash L_{aux}^{clu}$, but it is clear that using L_{rdp} only performs very comparably to the full RDP. However, using L_{aux}^{clu} only may result in large performance drops in some datasets, such as *R8*, *20news* and *Olivetti*. This indicates L_{rdp} is a more important loss function to the overall performance of the full RDP model. In terms of supervisory source, distances obtained by the non-linear random projection in RDP are much more effective than the two other sources on some datasets such as *Olivetti* and *RCV1*. Three different supervisory sources are very comparable on the other three datasets.

5 Related Work

Self-supervised Learning. Self-supervised learning has been recently emerging as one of the most popular and effective approaches for representation learning. Many of the self-supervised methods

Table 5: F-score performance of RDP and its variants in the clustering task. Similar results can also be observed in NMI in Table 9 in Appendix D.

	Decomposition			Supervision Signal	
Data	**RDP**	**RDP**$\setminus L_{rdp}$	**RDP**$\setminus L_{aux}^{clu}$	**Org_SS**	**SRP_SS**
R8	0.360 ± 0.055	0.312 ± 0.068	0.330 ± 0.052	0.359 ± 0.028	$\mathbf{0.363 \pm 0.046}$
20news	$\mathbf{0.119 \pm 0.006}$	0.083 ± 0.010	0.117 ± 0.005	0.111 ± 0.005	0.111 ± 0.007
Olivetti	$\mathbf{0.638 \pm 0.026}$	0.602 ± 0.023	0.597 ± 0.019	0.610 ± 0.022	0.601 ± 0.023
Sector	0.191 ± 0.007	0.184 ± 0.010	$\mathbf{0.217 \pm 0.007}$	0.181 ± 0.007	0.186 ± 0.009
RCV1	$\mathbf{0.572 \pm 0.003}$	0.514 ± 0.057	0.526 ± 0.011	0.523 ± 0.003	0.532 ± 0.001

learn high-level representations by predicting some sort of ‘context’ information, such as spatial or temporal neighbourhood information. For example, the popular distributed representation learning techniques in NLP, such as CBOW/skip-gram (Mikolov et al., 2013a) and phrase/sentence embeddings in (Mikolov et al., 2013b; Le & Mikolov, 2014; Hill et al., 2016), learn the representations by predicting the text pieces (e.g., words/phrases/sentences) using its surrounding pieces as the context. In image processing, the pretext task can be the prediction of a patch of missing pixels (Pathak et al., 2016; Zhang et al., 2017) or the relative position of two patches (Doersch et al., 2015). Also, a number of studies (Goroshin et al., 2015; Misra et al., 2016; Lee et al., 2017; Oord et al., 2018) explore temporal contexts to learn representations from video data, e.g., by learning the temporal order of sequential frames. Some other methods (Agrawal et al., 2015; Zhou et al., 2017; Gidaris et al., 2018) are built upon a discriminative framework which aims at discriminating the images before and after some transformation, e.g., ego motion in video data (Agrawal et al., 2015; Zhou et al., 2017) and rotation of images (Gidaris et al., 2018). There have also been popular to use generative adversarial networks (GANs) to learn features (Radford et al., 2015; Chen et al., 2016). The above methods have demonstrated powerful capability to learn semantic representations. However, most of them use the supervisory signals available in image/video data only, which limits their application into other types of data, such as traditional tabular data. Although our method may also work on image/video data, we focus on handling high-dimensional tabular data to bridge this gap.

Other Approaches. There have been several well-established unsupervised representation learning approaches for handling tabular data, such as random projection (Arriaga & Vempala, 1999; Bingham & Mannila, 2001; Li et al., 2006), PCA (Wold et al., 1987; Schölkopf et al., 1997; Rahmani & Atia, 2017), manifold learning (Roweis & Saul, 2000; Donoho & Grimes, 2003; Hinton & Roweis, 2003) and autoencoder (Hinton & Salakhutdinov, 2006; Vincent et al., 2010). One notorious issue of PCA or manifold learning approaches is their prohibitive computational cost in dealing with large-scale high-dimensional data due to the costly neighbourhood search and/or eigen decomposition. Random projection is a computationally efficient approach, supported by proven distance preservation theories such as the Johnson-Lindenstrauss lemma (Johnson & Lindenstrauss, 1984). We show here that the preserved distances by random projection can be harvested to effectively supervise the representation learning. Autoencoder networks are another widely-used efficient feature learning approach which learns low-dimensional representations by minimising reconstruction errors. One main issue with autoencoders is that they focus on preserving the global feature information only, which may result in loss of local structure information. There have been some representation learning methods designed specifically for anomaly detection (Pang et al., 2018; Zong et al., 2018; Burda et al., 2019). By contrast, we aim at generic representations learning while being flexible to incorporate optionally task-dependent losses to learn task-specific semantic-rich representations.

6 Conclusion

In this paper, we introduce a novel Random Distance Prediction (RDP) model which learns features in a fully unsupervised fashion by predicting data distances in a randomly projected space. The key insight is that random mapping is a theoretical proven approach to obtain approximately preserved distances, and to well predict these random distances, the representation learner is optimised to learn class structures that are implicitly embedded in the randomly projected space, achieving semantic-rich feature representations. Our idea is justified by thorough and systematic experiments in two typical unsupervised tasks, anomaly detection and clustering, which show RDP-enabled anomaly detectors and clustering substantially outperform their counterparts on 19 real-world datasets. We plan to extend RDP to other types of data to broaden its application scenarios.

REFERENCES

Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 37–45, 2015.

Rosa I Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *40th Annual Symposium on Foundations of Computer Science*, pp. 616–623. IEEE, 1999.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbor meaningful? In *International Conference on Database Theory*, pp. 217–235. Springer, 1999.

Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 245–250. ACM, 2001.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *ICLR*, 2019.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

David L Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE international conference on computer vision*, pp. 4086–4093, 2015.

John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1367–1377, 2016.

Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pp. 857–864, 2003.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pp. 1188–1196, 2014.

Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 667–676, 2017.

Ping Li, Trevor J Hastie, and Kenneth W Church. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 287–296. ACM, 2006.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. IEEE, 2008.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013b.

Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pp. 527–544. Springer, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2041–2050. ACM, 2018.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2015.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Mostafa Rahmani and George Atia. Coherence pursuit: Fast, simple, and robust subspace recovery. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2864–2873. JMLR. org, 2017.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pp. 583–588. Springer, 1997.

Santosh Vempala. Random projection: A new approach to vlsi layout. In *Proceedings 39th Annual Symposium on Foundations of Computer Science*, pp. 389–395. IEEE, 1998.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11:3371–3408, 2010.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1058–1067, 2017.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1851–1858, 2017.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*, 2018.

A Implementation Details

RDP-enabled Anomaly Detection. The RDP consists of one fully connected layer with 50 hidden units, followed by a leaky-ReLU layer. It is trained using Stochastic Gradient Descent (SGD) as its optimiser for 200 epochs, with 192 samples per batch. The learning rate is fixed to 0.1. We repeated the boosting process 30 times to obtain statistically stable results. In order to have fair comparisons, we also adapt the competing methods AE, REPEN, DAGMM and RND into ensemble methods and perform the experiments using an ensemble size of 30.

RDP-enabled Clustering. RDP uses a similar network architecture and optimisation settings as the one used in anomaly detection, i.e., the network consists of one fully connected layer, followed by a leaky-ReLU layer, which is optimised by SGD with 192 samples per batch and 0.1 learning rate. Compared to anomaly detection, more semantic information is required for clustering algorithms to work well, so the network consists of 1,024 hidden units and is trained for 1,000 epochs. Clustering is significant yet common analysis method, which aims at grouping samples close to each other into the same clusters and separating far away data points into different clusters. Compared to anomaly detection that often requires pattern frequency information, clustering has a higher requirement of the representation expressiveness. Therefore, if the representative ability of a model is strong enough, it should also be able to learn representations that enable clustering to work well on the projected space.

Note that the representation dimension M in the ϕ function and the projection dimension K in the η function are set to be the same to alleviate parameter tuning. This means that $M = K = 50$ is used in anomaly detection and $M = K = 1024$ is used in clustering. We have also tried deeper network structures, but they worked less effectively than the shallow networks in both anomaly detection and clustering. This may be because the supervisory signal is not strong enough to effectively learn deeper representations.

B Datasets

The statistics and the accessible links of the datasets used in the anomaly detection and clustering tasks are respectively presented in Tables 6 and 7. *DDoS* is a dataset containing DDoS attacks and normal network flows. *Donors* is from KDD Cup 2014, which is used for detecting a very small number of outstanding donors projects. *Backdoor* contains backdoor network attacks derived from the UNSW-NB15 dataset. *Creditcard* is a credit card fraud detection dataset. *Lung* contains data records of lung cancer patients and normal patients. *Probe* and *U2R* are derivd from KDD Cup 99, in which probing and user-to-root attacks are respectively used as anomalies against the normal network flows. The above datasets contain real anomalies. Following (Liu et al., 2008; Pang et al., 2018; Zong et al., 2018), the other anomaly detection datasets are transformed from classification datasets by using the rare class(es) as the anomaly class, which generates semantically real anomalies.

Table 6: Datasets used in the anomaly detection task

Data	N	D	Anomaly (%)	Link
DDoS	464,976	66	3.75%	http://www.csmining.org/cdmc2018/index.php
Donors	619,326	10	5.92%	https://www.kaggle.com/c/kdd-cup-2014-predicting-excitement-at-donors-choose
Backdoor	95,329	196	2.44%	https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity
Ad	3,279	1,555	13.99%	https://archive.ics.uci.edu/ml/datasets/internet+advertisements
Apascal	12,695	64	1.38%	http://vision.cs.uiuc.edu/attributes/
Bank	41,188	62	11.26%	https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
Celeba	202,599	39	2.24%	http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
Census	299,285	500	6.20%	https://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29
Creditcard	284,807	29	0.17%	https://www.kaggle.com/mlg-ulb/creditcardfraud
Lung	145	3,312	4.13%	https://archive.ics.uci.edu/ml/datasets/Lung+Cancer
Probe	64,759	34	6.43%	http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
R8	3,974	9,467	1.28%	http://csmining.org/tl_files/Project_Datasets/r8_r52/r8-train-all-terms.txt
Secom	1,567	590	6.63%	https://archive.ics.uci.edu/ml/datasets/secom
U2R	60,821	34	0.37%	http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

R8, *20news*, *Sector* and *RCV1* are widely used text classification benchmark datasets. *Olivetti* is a widely-used face recognition dataset.

Table 7: Datasets used in the clustering task

Data	N	D	#Classes	Link
R8	7,674	17,387	8	http://csmining.org/tl_files/Project_Datasets/r8_r52/r8-train-all-terms.txt
20news	18,846	130,107	20	https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html
Olivetti	400	4,096	40	https://scikit-learn.org/0.19/datasets/olivetti_faces.html
Sector	9,619	55,197	105	https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/multiclass.html#sector
RCV1	20,242	47,236	2	https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/binary.html#rcv1.binary

C AUC-PR Performance of Ablation Study in Anomaly detection

The experimental results of AUC-PR performance of RDP and its variants in the anomaly detection task are shown in Table 8. Similar to the results shown in Table 3, using the L_{rdp} loss only, our proposed RDP model can achieve substantially better performance over its counterparts. By removing the L_{rdp} loss, the performance of RDP drops significantly in 11 out of 14 datasets. This demonstrates that the L_{rdp} loss is heavily harvested by our RDP model to learn high-quality representations from random distances. Removing L^{ad}_{aux} from RDP also results in substantial loss of AUC-PR in many datasets. This indicates both the random distance prediction loss L_{rdp} and the task-dependent loss L^{ad}_{aux} are critical to RDP. The boosting process is also important, but is not as critical as the two losses. Consistent with the observations derived from Table 3, distances calculated in non-linear and linear random mapping spaces are more effective supervisory sources than that in the original space.

Table 8: AUC-PR performance of RDP and its variants in the anomaly detection task.

	Decomposition				Supervision Signal	
Data	**RDP**	**RDP**$\backslash L_{rdp}$	**RDP**$\backslash L^{ad}_{aux}$	**RDP\Boosting**	**Org_SS**	**SRP_SS**
DDoS	0.301 ± 0.028	0.110 ± 0.015	0.364 ± 0.013	0.114 ± 0.001	0.363 ± 0.007	**0.380 ± 0.030**
Donors	**0.432 ± 0.061**	0.201 ± 0.033	0.104 ± 0.007	0.278 ± 0.040	0.099 ± 0.004	0.113 ± 0.010
Backdoor	0.305 ± 0.008	0.433 ± 0.015	0.142 ± 0.006	**0.537 ± 0.005**	0.143 ± 0.005	0.154 ± 0.028
Ad	**0.726 ± 0.007**	0.473 ± 0.009	0.491 ± 0.014	0.488 ± 0.008	0.419 ± 0.015	0.530 ± 0.007
Apascal	**0.042 ± 0.003**	0.021 ± 0.005	0.031 ± 0.002	0.028 ± 0.003	0.016 ± 0.003	0.035 ± 0.007
Bank	**0.364 ± 0.013**	0.258 ± 0.006	0.266 ± 0.018	0.278 ± 0.007	0.262 ± 0.016	0.265 ± 0.021
Celeba	**0.104 ± 0.006**	0.068 ± 0.010	0.060 ± 0.004	0.072 ± 0.008	0.050 ± 0.009	0.065 ± 0.010
Census	0.086 ± 0.001	0.081 ± 0.001	0.075 ± 0.001	**0.087 ± 0.001**	0.077 ± 0.002	0.064 ± 0.001
Creditcard	0.363 ± 0.011	0.290 ± 0.012	**0.414 ± 0.02**	0.329 ± 0.007	0.362 ± 0.016	0.372 ± 0.024
Lung	**0.705 ± 0.028**	0.381 ± 0.104	0.437 ± 0.083	0.542 ± 0.139	0.361 ± 0.054	0.464 ± 0.053
Probe	0.955 ± 0.002	0.609 ± 0.014	0.952 ± 0.007	0.628 ± 0.011	0.937 ± 0.005	**0.959 ± 0.011**
R8	0.146 ± 0.017	0.134 ± 0.031	0.109 ± 0.006	**0.173 ± 0.028**	0.067 ± 0.016	0.134 ± 0.019
Secom	**0.096 ± 0.001**	0.086 ± 0.002	0.096 ± 0.006	0.090 ± 0.001	0.088 ± 0.004	0.093 ± 0.004
U2R	0.261 ± 0.005	0.217 ± 0.011	**0.266 ± 0.007**	0.238 ± 0.009	0.187 ± 0.013	0.239 ± 0.023
#wins/draws/losses (RDP vs.)		13/0/1	11/0/3	11/0/3	12/0/2	5/0/9

D NMI Performance of Ablation Study in Clustering

Table 9 shows the NMI performance of RDP and its variants in the clustering task. It is clear that our RDP model with the L_{rdp} loss is able to achieve NMI performance that is comparably well to the full RDP model, which is consistent to the observations in Table 5. Without using the L_{rdp} loss, the performance of the RDP model has some large drops on nearly all the datasets. This reinforces the crucial importance of L_{rdp} to RDP, which also justifies that using L_{rdp} alone RDP can learn expressive representations. Similar to the results in Table 5, RDP is generally more reliable supervisory sources than Org_SS and SRP_SS in this set of results.

E Performance Evaluation in Classification

We also performed some preliminary evaluation of the learned representations in classification tasks using a feed-forward three-layer neural network model as the classifier. We used the same datasets

Table 9: NMI performance of RDP and its variants in the clustering task.

	Decomposition			Supervision Signal	
Data	**RDP**	**RDP**$\backslash L_{rdp}$	**RDP**$\backslash L_{aux}^{clu}$	**Org_SS**	**SRP_SS**
R8	0.539 ± 0.040	0.471 ± 0.043	0.505 ± 0.037	0.567 ± 0.021	$\mathbf{0.589 \pm 0.039}$
20news	$\mathbf{0.084 \pm 0.005}$	0.075 ± 0.006	0.081 ± 0.002	0.075 ± 0.002	0.074 ± 0.003
Olivetti	$\mathbf{0.805 \pm 0.012}$	0.782 ± 0.010	0.784 ± 0.010	0.795 ± 0.011	0.787 ± 0.011
Sector	0.305 ± 0.007	0.253 ± 0.010	$\mathbf{0.340 \pm 0.007}$	0.295 ± 0.009	0.298 ± 0.008
Rcv1	0.165 ± 0.000	0.146 ± 0.010	$\mathbf{0.168 \pm 0.000}$	0.154 ± 0.002	0.147 ± 0.000

as in the clustering task. Specifically, the representation learning model first outputs the new representations of the input data, and then the classifier performs classification on the learned representations. RDP is compared with the same competing methods HLLE, SRP, AE and COP as in clustering. F-score is used as the performance evaluation metric here.

The results are shown in Table 10. Similar to the performance in clustering and anomaly detection, our model using only the random distance prediction loss L_{rdp}, i.e., RDP$\backslash L_{aux}^{clu}$, performs very favourably and stably on all the five datasets. The incorporation of $\backslash L_{aux}^{clu}$ into the model, i.e., RDP, helps gain some extra performance improvement on datasets like *20news*, but it may also slightly downgrade the performance on other datasets. An extra hyperparameter may be added to control the importance of these two losses.

Table 10: F-score performance of classification on five real-world datasets.

Data	**HLLE**	**SRP**	**AE**	**COP**	**RDP**$\backslash L_{aux}^{clu}$	**RDP**
R8	0.246	0.895	0.874	0.860	0.900	**0.906**
20news	0.005	0.733	0.709	0.718	0.735	**0.753**
Olivetti	0.895	0.899	0.820	0.828	**0.900**	0.896
Sector	0.037	0.671	0.645	0.689	0.690	**0.696**
RCV1	0.766	0.919	0.918	N/A	**0.940**	0.926