ISLAMIC AZAD UNIVERSITY
MASHHAD BRANCH

Faculty of Engineering- Department of Computer

Thesis for receiving «M.Sc.» degree on Artificial Intelligence

Subject:

Distance Metric Learning for dimensionality reduction and performance improvement based on the structural neighborhoods

Thesis Advisor:
Mohammad Hossein Moattar Ph.D.

Consulting Advisor:
Yahya Forghani Ph.D.

By:
Mostafa Razavi Ghods

Summer 2017

ISLAMIC AZAD UNIVERSITY
MASHHAD BRANCH

Faculty of Engineering- Department of Computer

Thesis for receiving «M.Sc.» degree on Artificial Intelligence

Subject:

Distance Metric Learning for dimensionality reduction and performance improvement based on the structural neighborhoods

By:
Mostafa Razavi Ghods

Approved by:

Assoc. Prof. Dr. Mohammad Hossein Moattar: (Thesis Advisor)	**……**
Assoc. Prof. Dr. Yahya Forghani:	**……**
Assoc. Prof. Dr. Reza Godaz:	**……**

Abstract

Distance metric learning can be viewed as one of the fundamental interests in pattern recognition and machine learning, which plays a pivotal role on the performance of many learning methods. One of the effective methods in learning such a metric is to learn it from a set of labeled training samples. The issue of data imbalance is the most important challenge of the recent methods. This research tries not only to preserve the local structures but also covers the issue of imbalanced datasets. To do this, the proposed method first tries to extract a low dimensional manifold from the input data. Then, it learns the local neighborhood structures and the relationship of the data points in the ambient space based on the adjacencies of the same data points on the embedded low dimensional manifold. Using the local neighborhood relationships extracted from the manifold space, the proposed method learns the distance metric in a way which minimizes the distance between similar data and maximizes their distance from the dissimilar data points. The evaluations of the proposed method on numerous datasets from the UCI repository of machine learning, and also the KDDCup98 dataset as the most imbalance dataset, justify the supremacy of the proposed approach in comparison with other approaches especially when the imbalance factor is high.

Keywords: Distance metric learning; Imbalanced data; Manifold learning; Mahalanobis distance; Locally Linear Embedding (LLE).

Table of contents

Abstract ..... **iii**

Table of contents ..... **iv**

List of figures ..... **vii**

List of tables ..... **viii**

Chapter 1: Introduction **11**

1.1 Background and aims 11

1.2 Statement of the problem 12

1.3 Importance and necessity of doing this research 13

1.4 A summary of the proposed method 14

1.5 Thesis structure 14

Chapter 2: Literature review **16**

2.1 Introduction 16

2.2 Definitions of the problem and literature review (dimensionality reduction and manifold learning problems) 16

2.2.1 Dimensionality reduction 16

2.3 Main features of distance learning algorithms 17

2.3.1 Learning scope 18

2.3.2 Form of metric 18

2.3.3 Dimensionality reduction 19

2.4 A brief on distance metric learning 19

2.5 Mahalanobis Distance Metric 19

2.6 Distance metric learning algorithms 21

2.7 Unsupervised distance metric learning approaches 21

2.7.1 Principle Component Analysis (PCA) 22

2.7.2 Nonlinear PCA 23

2.7.3 Autoencoder 23

2.7.4 Locality Preserving Projections (LPP) 25

2.7.5 Laplacian Embedding (LE) 27

2.7.6 Locally Linear embedding (LLE) 27

2.7.7 Isometric feature mapping (Isomap) 28

2.8 Supervised distance metric learning approaches 29

2.8.1 Linear discriminant analysis (LDA) 29

2.8.2 Relevant Component Analysis (RCA) 30

2.8.3 Information Theoretic Metric Learning (ITML) 31

2.8.4 Neighborhood Component Analysis (NCA) 32

2.8.5 Discriminative Least Squares Regression (DLSR) 32

2.9 Semi-supervised distance metric learning 34

2.9.1 Laplacian Regularized Metric Learning (LRML) 34

2.9.2 Constraint Margin Maximization (CRM) 35

2.10 Summarization and conclusion of this chapter 36

Chapter 3: Methodology 38

3.1 Introduction 38

3.2 Proposed method 38

3.2.1 Data neighborhood structures after distance metric learning 40

3.3 The objective of distance metric learning 42

3.4 Advantages of the proposed method 44

Chapter 4: Experiments 45

4.1 Introduction 45

4.2 Dataset 45

4.3 Evaluation criteria 46

4.4 Evaluation scenarios and experimental results 46

4.5 Representation of the data after reduction 51

4.6 Evaluations on the KDD data 53

Chapter 5: Conclusion and Future Work 59

5.1 Introduction 59

5.2 Future work 59

References 61

Appendix 68

Results of the SVM (the remainder) 68

Sensitivity 68

Specificity 70

results of the k-NN+S 72

Accuracy 72

Sensitivity 75

Specificity 77

Comparison of the average runtime of the k-NN, k-NN+S, and SVM 79

List of figures

Figure 2-1. An example of Manifold learning while maintaining a structural neighborhood and manifold learning with lower dimensions embedded in the original space with higher dimensions [35]. 17

Figure 2-2. Diagram of an Autoencoder [47]. 24

Figure 3-1. Overall process of the proposed method. 39

Figure 3-2. The local patch consisting of the dissimilar neighbors. 39

Figure 3-3. The local neighborhoods after distance metric learning with the proposed approach. 40

Figure 3-4. The representation of data points after manifold embedding and similarity calculation. 40

Figure 4-1. Data distribution visualization after the reduction to a new 2D space using different approaches. 53

Figure 0-1. Comparison of the averagre accuracy of the proposed method on 10-fold settings based on the neighborhoods on the manifold learnt by autoenocders and DLSR on the vehicle dataset using the k-NN+S classifier. 72

List of tables

Table 2-1. Unspervised distance metric learning algorithms. 22

Table 2-2. Supervised distnace metric learning algorithms. 29

Table 2-3. Semi-supervised distance metric learning algorithms. 34

Table 4-1. The properties of the datasets. 45

Table 4-2. Accuracy comparison between different approaches versus the proposed using 10-fold cross validation and 7-NN classifier with (d,r) indicating the best latent dimensionality and the rank of the approach, respectively (AE denotes auto-encoder approach). 47

Table 4-3. Sensitivity comparison between different approaches versus the proposed using 10-fold cross validation and 7-NN classifier with (d,r) indicating the best dimensionality and the rank of the approach, respectively (AE denotes auto-encoder approach). 48

Table 4-4. Specificity comparison between different approaches versus the proposed using 10-fold cross validation and 7-NN classifier with (d,r) indicating the best dimensionality and the rank of the approach, respectively (AE denotes auto-encoder approach). 49

Table 4-5. Accuracy comparison between different approaches versus the proposed using 10-fold cross validation and SVM classifier with (d,r) indicating the best latent dimensionality and the rank of the approach, respectively (AE denotes auto-encoder approach). 50

Table 4-6. The average confusion matrix of the 10-fold cross validation using DLSR approach on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 28). 54

Table 4-7. The average confusion matrix of the 10-fold cross validation using the proposed PCA+DSLR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 10). 54

Table 4-8. The average confusion matrix of the 10-fold cross validation using the proposed LDA+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 28). 55

Table 4-9. The average confusion matrix of the 10-fold cross validation using the proposed MDS+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1). 55

Table 4-10. The average confusion matrix of the 10-fold cross validation using the proposed Isomap+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1).....55

Table 4-11. The average confusion matrix of the 10-fold cross validation using the proposed LLE+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1). 55

Table 4-12. The average confusion matrix of the 10-fold cross validation using the proposed KPCA+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1).....56

Table 4-13. The average confusion matrix of the 10-fold cross validation using the proposed Autoencoder+DLSR approach on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1). 56

Table 4-14. The average confusion matrix of the 10-fold cross validation using the proposed Autoencoder+DLSR approach on KDD dataset using the SVM classifier for the best latent dimensionality (i.e. 9). 57

Table 4-15. A comparison between the accuracy of the proposed method and some other recent works on different classes of the KDD based on the 10-fold cross validation. 58

Table 0-1. Comparison of the results for the sensitivity criterion using the SVM classifier. 68

Table 0-2. Ranking of the results for the sensitivity criterion based on the average of the 10-fold cross-validation. 69

Table 0-3. Comparison of the results for the specificity criterion using the SVM classifier. 70

Table 0-4. Ranking of the results for the specificity criterion based on the average of the 10-fold cross-validation. 71

Table 0-5. Comparison of the results for the accuracy criterion using the 7-NN classifier. 73

Table 0-6. Ranking of the results for the accuracy criterion based on the average of the 10-fold cross-validation using the 7-NN classifier. 74

Table 0-7. Comparison of the results for the sensitivity criterion using the 7-NN classifier. 75

Table 0-8. Ranking of the results for the sensitivity criterion based on the average of the 10-fold cross-validation using the 7-NN classifier. 76

Table 0-9. Comparison of the results for the specificity criterion using the 7-NN classifier. 77

Table 0-10. Ranking of the results for the specificity criterion based on the average of the 10-fold cross-validation using the 7-NN classifier. 78

Table 0-11. Comparison of the average execution time per test data on 10-fold over different datasets between the three classification methods k-NN, k-NN + S, and SVM in the proposed similarity space. 79

Chapter 1: Introduction

1.1 Background and aims

Distance metric learning (DML) for many years has been considered as one the main research interests in works which try to define the similarity and dissimilarity criteria between patterns. Distance metric learning approaches are employed to define an appropriate metric which can reflect the similarity and the dissimilarity of the data points with respect to the application in which they are used. The goal of distance metric learning is to find a real-valued metric function of data pairs under which the data pair with the same label are as close and the data pair from different classes are as far as possible. In this work, the main goal is to learn a function which can transform the input data onto the learned manifold with the least possible amount of changes in the relative distance of data-points from the same class [1].

The application of the distance metric learning the in pattern recognition includes algorithms such as k-means, k-nearest neighbors and kernel-based algorithms such as support vector machines (SVMs) [2]–[9] .Distance metric learning approaches can be categorized into three classes of: fully-supervised, unsupervised, and semi supervised methods. In fully-supervised learning, the ultimate goal is to use the class discriminative information between the data-pairs in order to keep all data within a class as close and the data from different classes as diverse as possible. Zhang et al. [10] have shown that learning the distance metric based on the class discriminative information usually shows better performance than using the classical Euclidean distance.

Supervised distance metric learning itself could be divided into the two categories of local and global approaches. An approach is to learn a global distance metric from the training data in order to satisfy the constraints between all data-pairs simultaneously [5], [11]. The most expressive work in this field is Xing's [11] algorithm which learns a distance metric in the global scale where the distances between the data-pairs are in turn minimized and maximized under the equivalence and inequivalence constraints, respectively. Equivalence and inequivalence constraints may conflict when the data from different classes have multiple distributions. Thus, it is hard to satisfy the whole constrains in the global scale. In order to confront with this phenomenon, local distance metric learning approaches, which take account of the local constraints, are introduced

[12]–[14]. These local algorithms only consider the pairwise constraints while avoiding the conflicting ones.

The aforementioned approaches try to present one single metric for all instances of the data. However, learning only single metric may have the deficiencies like: (1) is barely probable to find a metric appropriate for all the training data; (2) a local metric may not be immune to noisy data; (3) a local metric cannot be used in the multi-modal problems. Therefore, it is recommended to use different metrics for multiple distributions of the training data [4], [14], [15].

Generally, supervised distance metric learning could be divided into the two groups of local and global approaches. The local methods could also be subcategorized into the single-metric and multiple-metric approaches. The global methods try to keep the similar samples as close and dissimilar samples as far as possible. Xing's algorithm [11] is a good representative of global approaches which optimizes some equal and inequality constraints simultaneously using the convex optimization methods.

The advantage of using the global approaches is in their ability to capture the distributions from different classes when all the samples belonging to the same class do not obey the same distribution. However, the global approaches may not be able to learn the optimal distance metric when data have multimodal distributions.

Local approaches use the neighborhood information to cope with the multimodal distribution problems. Local Fisher Discriminant Analysis (LFDA) [13], according to the local information, give more weight to the pairwise neighborhood constraints. Yang et al. [3], proposed a probabilistic approach to optimize the local pairwise constraints. Goldberger et al. [12] utilized a stochastic variant of the KNN classifier to calculate the leave-one-out classification error.

1.2 Statement of the problem

Dimensionality reduction (DR) approaches try to find a low dimensional representation of the data in order to satisfy some goals. Size reduction of the feature vectors for data compression (from the unsupervised perspective) as well as avoiding the curse of dimensionality (from the supervised view) are two main objectives of the dimensionality reduction approaches. However, problems happen when the number of data-points is not sufficient to cover the whole initial high dimensional space. Data

visualization is one other goal of the DR approaches, in which the DR is used to project the high dimensional data onto a space with at most two or three dimensions in a way which is comprehensible and visualizable. In data classification application, the DR methods could be used to find a low-dimensional manifold on which the data with the same label are compact, while the data from different classes are discriminant with respect to each other, which itself improves the classification accuracy.

DML has been one of the fields of research for many years, the main purpose of which is to properly define the criteria of similarity (or dissimilarity). Thus, DML methods are used to define an appropriate distance metric that reflects the similarity or dissimilarity in each application. The goal of DML is to find a metric function of pairs of data with real values in such a way that the data points with the same label be as close to each other and the data with the dissimilar labels become as separated from each other as possible. Here, the goal is to learn a transformation, that is capable of mapping the data points to the manifold space with the least change in the locality and neighborhood of the points with respect to each other.

1.3 Importance and necessity of doing this research

The proposed method in this research tries to cover three of the challenges and gaps in distance learning standard methods. Therefore, the importance and necessity of this research will be in three ways:

1. **Neighborhood Preservation**: In this research, we try to learn the distance metric in such a way that the locality as well as local neighborhoods between similar points are preserved as much as possible and only dissimilar data pairs get separated from each other.

2. **Unbalanced data**: The proposed method in this research is such that distance learning is done in such a way that for each data point the number of similar and dissimilar data points are equal.

3. **Independence of the problem**: The proposed method tries to do the DML in such a way that can be used for any application, whether learning with supervision or without supervision.

In this research, the proposed method tries to cover a triple of the challenges in the distance metric learning dimensionality reduction. This research is important as it tries to learn the distance metric in such a way that after transformation, which is done by the learned metric function, the data from the same class are as close and the data from different classes are as far from each other as possible. Besides, nowadays many of the real-world datasets are found to be imbalanced in terms of the number the points associated to different classes. Thus, the proposed method tries to learn the distance metric with respect to this phenomenon. Furthermore, one other goal of the proposed method is to learn the distance metric in a way that it could be used in any application, independent from the presence or absence of the labeling information.

1.4 A summary of the proposed method

In this study, we have attempted to learn a low-dimensional manifold out of the data in the initial space. Then, similar, dissimilar and irrelevant data-points are found based their local neighborhood on the manifold. Consequently, based on these neighborhood relationships which are found on the manifold and based the coordinates of the data points on the initial space, distance metric learning is done using a Mahalanobis distance metric learning approach.

1.5 Thesis structure

The remainder of this thesis is organized as follows. Chapter 2 primarily deals with the concept of distance metric learning and dimensionality reduction followed by some discussion on the different manifold learning approaches. The proposed method will be introduced in Chapter 3. Chapter 4 describes the experimental setup and analyzes

its performance and summarizes its results and finally, Chapter 5 discusses the main findings and concludes this study besides giving some directions for future studies.

Chapter 2: Literature review

2.1 Introduction

Distance metric plays a key role in the success of many machine learning algorithms. For example, the classification techniques such as the k-nearest neighbors (k-NN) [16] and the clustering approaches like k-means algorithm are highly dependent on the applied metric in order to model the structural models between the input data. A tangible example in this field could be the visual object recognition problem. Lots of the applications in machine learning could be considered as implicit distance metric learning approaches which are capable of learning the similarities and dissimilarities between visual input objects. This typical example includes classification [17] and Content-based image retrieval [18] where a distance metric is required to discriminate between different classes of related or irrelevant objects.

In this chapter we will touch upon some basic ideas about the distance metric learning and dimensionality reduction approaches. Then we will discuss about some of the most promising approaches in this discipline and finally we will conclude this chapter with a short review on each of the triple of the distance metric learning approaches as supervised, unsupervised and the semi-supervised.

2.2 Definitions of the problem and literature review (dimensionality reduction and manifold learning problems)

2.2.1 Dimensionality reduction

Dimensionality reduction or feature extraction has been widely used in data mining, computer vision, and pattern recognition [19]. Classical dimensionality reduction methods, such as the Principal Component Analysis (PCA) [20]–[22] as well as Linear Resolution Analysis (LDA) [23]–[25] and their modified methods [26]–[28], are simple and effective and have been widely used in various fields such as face recognition, palm recognition, etc. However, classical methods such as PCA and LDA focus only on the global structure of a dataset in order to reduce the dimensionality.

Roweis and Saul [29] and Tenenbaum et al. [30] state that the images of different objects are placed on a low-dimensional manifold that is laid in a high-dimensional space. To identify the intrinsic geometry of a data set, manifold learning methods have been widely used, the most popular of which are LLE, ISOMAP, and Laplacian eigenmap [31]. Based on these nonlinear methods, many linear dimensional reduction methods based on manifold learning for feature extraction have been proposed. Among these methods, Neighborhood Preserving Embedding (NPE) [32], Orthogonal Neighborhood-Preserving Projection (ONPP) [33], Locality Preserving Projections (LPP) [34] can be considered as methods that preserve the local geometric structure of the data and map them onto the manifold space by applying a simple linear estimation to maintain nonlinear maps.

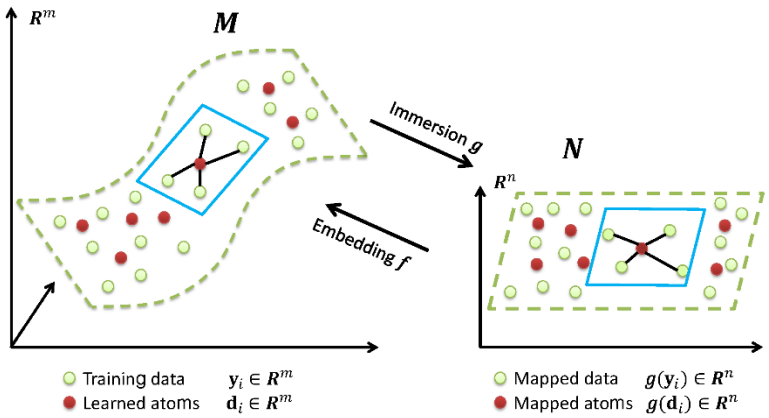


Figure 2-1. An example of Manifold learning while maintaining a structural neighborhood and manifold learning with lower dimensions embedded in the original space with higher dimensions [35]**.**

LLE-based methods have been developed in a variety of forms and thanks to their simplicity are widely usage in a variety of applications such as facial expression recognition, image prediction and retrieval, feature fusion, face recognition, gait recognition, human motion recognition, etc. [19].

2.3 Main features of distance learning algorithms

With the exception of some basic methods, most distance learning algorithms are essentially competitive in such a way that they can achieve the state-of-the-art performance in some areas. However, each algorithm has its own inherent characteristics (such as, type of metric, ability to use unsupervised data, scalability in different dimensions, guarantee of generalization, etc.) and on the decision to choose a method that

fits the type of the problem given should be considered with special attention. In this section, we examine three important features of distance learning algorithms:

2.3.1 Learning scope

In this section, we examine three important features of distance learning algorithms:

Fully Supervised: the distance learning algorithm has full access to the training data set $\{z_i = (x_i, y_i)\}_{i=1}^{n}$, where each training sample $z_i \in Z = x \times y$ is consisted of an instance $x_i \in X$ and a label (or class) $y_i \in Y$. Y is a discrete and finite set of labels $|Y|$ (unless noted otherwise). In practice, label information is commonly used to construct specific sets of pair/triplet constraints S, D, R, for example based on the notion of neighborhood.

Weakly supervised: the algorithm has no access to the label information of any training data and only has information about constraints S, D, R. That is a meaningful approach in many applications where obtaining labeling data is costly while ancillary information is inexpensive. For example, implicit user feedback (such as clicking on search engine results), referrals between articles, or links within a network.

Semi-supervised: in addition to fully and semi-supervised methods, the algorithm has access to a large number of unlabeled instances for which no additional information is available. This learning method is useful to avoid over-fitting when tagged information or ancillary information are very limited.

2.3.2 Form of metric

Clearly, the form of the learned metric is a key choice. The three main families of distance learning metrics are:

Linear metrics, such as the Mahalanobis distance. Their expressive power is limited, but they are easier to optimize (they often lead to convex formulation, and as a result the global optimization of the solution) and are less likely to over-fit.

Nonlinear metrics: such as the X^2 histogram distance. They usually lead to nonconvex formulations (due to the local optimalities) as well as overfitting, but they can maintain nonlinear variations in the data.

Local metrics: in which several local metrics (linear or nonlinear, usually simultaneously) are learned to deal with complex issues such as heterogeneous data.

However, they are more prone to over-fitting than global methods as the number of parameters they have to learn can be very large.

2.3.3 Dimensionality reduction

As mentioned earlier, distance metric learning is in some cases formulated as a visualization of data into a new feature space. An attractive achievement in this case is finding lower-dimensionally projected space, which allows us to perform calculations faster as well as more compact representations of them. This is usually done by adjusting the distance matrix of the learned distance to a lower rank.

2.4 A brief on distance metric learning

Although the roots of distance metric learning can be traced back to some previous work (e.g. [36], [37]), distance learning was actually emerged in 2002 by the pioneering work of Xing et al. [11] that formulates as a convex optimization problem.

The purpose of learning the distance criterion of matching a function with real values, for example the distance Mahalanobis $d_M(x, x') = \sqrt{(x - x')^T M (x - x')}$, with the problem using information that They are created by educational data. Most methods learn the distance criterion (which here is the positive semi-definite matrix M) with a weakly supervised method based on pairwise or triple constraints as follows:

Must-link/cannot link constraints (sometimes called positive/negative pairs):

$$S = \{(x_i, x_j): x_i \text{ and } x_j \text{ should be similar}\}$$

$$D = \{(x_i, x_j): x_i \text{ and } x_j \text{ should be dissimilar}\}$$

Relative constrains (sometimes called training triplets):

$$R = \{(x_i, x_j, x_k): x_i \text{ should be more similar to } x_j \text{ than } x_k\}$$

2.5 Mahalanobis Distance Metric

This section deals with the distance metric of the supervised Mahalanobis (complete or weak), which has gained a lot of attention due to its simplicity and excellent interpretability due to linear projection. Here we present the Mahalanobis distance as well as the two important challenges related to this distance metric.

Mahalanobis distance: This concept is derived from Mahalanobis (1936) [38] and basically refers to the measurement of distances that includes the correlation between features:

$$d_{maha}(x, \acute{x}) = \sqrt{(x - \acute{x})^T \mathbf{\Omega}^{-1} (x - \acute{x}),} \qquad \textit{Equation 2-1}$$

Where x and $\acute{x}$ are random vectors of the same distribution as the covariance matrix $\mathbf{\Omega}$. With a slight change in the terms that are common in the distance metric learning literature, we will actually be able to derive from the concept of Mahalanobis distance to refer to generalized square distances, which are defined as follows:

$$d_M(x, x') = \sqrt{(x - x')^T M (x - x')} \qquad \textit{Equation 2-2}$$

Which is parameterized with $M \in S_+^d$, where S_+^d is a symmetric positive semi-definite cone d × d (PSD). M must ensure that d_M satisfies the properties of the pseudo-distance $\forall\, x, x', x'' \in X$,

1. $d_M(x, x') \geq 0$
2. $d_M(x, x') = 0$
3. $d_M(x, x') = d(x', x)$
4. $d_M(x, x'') \leq d(x, x') + d(x', x'')$

Interpretation: It should be noted that when M is the same matrix, we obtain the Euclidean distance. Otherwise, M can be expressed by$L^T L$, where $L \in R^{K \times d}$ where k is the rank of the matrix M. We can also write $d_M(x, x')$ as follows:

$$d_M(x, x') = \sqrt{(x - x')^T M (x - x')} \qquad \textit{Equation 2-3}$$
$$= \sqrt{(x - x')^T L^T L (x - x')}$$
$$= \sqrt{(Lx - Lx')^T (Lx - Lx')}$$

Thus, a Mahalanobis distance can be implicitly expressed as a calculation of the Euclidean distance after the linear representation of the data defined by the M conversion matrix. It should be noted that M is a low-order matrix, in other words, $rank(M) = r < d$, so it provides a linear conversion of information into a space with smaller dimensions r. Thus, it allows us to get a more concise representation of cheaper distance information and calculations, especially when the main feature space is large. These excellent features

testify to the great attractiveness of the Mahalanobis distance, which is used in many distance learning applications.

Challenges: There are two major challenges in learning the Mahalanobis distance standard. The first challenge is to maintain $M \in S_+^d$ somehow during the optimization process. A simple way to do this is to use the illustrated rotation method, which is between a gradient step and an imaging step on the PSD cone by setting the negative eigenvalues to zero. But this method is expensive for high-dimensional problems, because the analysis of eigenvalues will be compared to $O(d^3)$. The second challenge is learning a low-order matrix (which, as mentioned earlier, involves an illustration of a smaller space) instead of a full-order matrix. Unfortunately, optimizing M against rank constraint or NP-regulation is difficult and therefore cannot be done effectively.

2.6 Distance metric learning algorithms

In this section, we review some of the distance learning methods that have been introduced recently. We divide these algorithms into unsupervised, supervised, or semi-supervised categories based on the supervised information they use. In this research, we use $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$ to represent the data matrix. $x_i \in \mathbb{R}^d$ is the i vector of the data vector. It should be noted that in the following demonstration we will use D to represent the distance measurement we want to learn.

Although there are many distance learning standard algorithms, almost all of them optimize the following objective function. In which $L(D)$ is a supervised term containing regulatory information, $U(D)$ is an unsupervised term and interacts only with data indicators. $\lambda_1 \geq 0, \lambda_2 \geq 0$ are balancing parameters. With this in mind, we will discuss the details of algorithms and how to formulate them.

$$J(D) = \lambda_1 L(D) + \lambda_2 U(D) \qquad \textit{Equation 2-4}$$

2.7 Unsupervised distance metric learning approaches

Unsupervised methods of distance metric learning do not require any supervision information. In other words, they perform distance metric learning only by having a matrix X, so that a discriminative or geometric optimality or is obtained. In *Equation 2-4*, unsupervised methods $J(D)$ are optimized by considering $\lambda_1 = 0$. Based on the properties of our existing unsupervised methods, compare them with Table 2 1. Unsupervised distance learning algorithms. We classify in which distance learning

algorithms are divided into linear and nonlinear categories based on the linearity or nonlinearity of their imagery. The concepts of learning the local and global distance criterion also depend on whether the algorithm applies some constraints to the local or global data structure.

Table 2-1. Unspervised distance metric learning algorithms.

	Local	Global
Linear	LPP [39]	PCA [40], [41]
Nonlinear	LE [42], LLE [29], Isomap [43], SNE [44], KLPP [39]	KPCA [45], KUMMP [41]

2.7.1 Principle Component Analysis (PCA)

PCA [30] is a method that tries to extract directions from the data in such a way that the maximum variances can be achieved. Assuming that $\bar{X}$ be a zero-mean matrix (matrix with mean equal zero), then the first principal component W_1 can be obtained from the following equations.

$$W_1 = argmax_{\|W\|=1} = Var(W^T \bar{X}) \qquad \textit{Equation 2-5}$$

$$argmax_{\|W\|=1} \frac{1}{n-1} W^T \bar{X} \bar{X}^T W$$

By having the first $k-1$ principal components, the k principal of the principal component can be obtained by subtracting the first principal component k-1 from $\bar{X}$.

$$\hat{X}_{k-1} = \hat{X} - \sum_{i=1}^{k-1} W_i W_i^T \hat{X}_i \qquad \textit{Equation 2-6}$$

Using $\hat{X}_{k-1}$ as the new data set, we can obtain the k-th principal component according to the following equation.

$$W_k = argmax_{\|W\|=1} W^T \hat{X}_{k-1} \hat{X}_{k-1} W \qquad \textit{Equation 2-7}$$

Finally, we want to obtain the matrix W with the following properties:

$$\max_{w} tr(W^T \bar{X}\bar{X}^T W), \;\; s.t. W^T W = I \qquad \textit{Equation 2-8}$$

We observe that the PCA directions continue with a sequential search for the directions on which the data changes are the most. Given that $\mathrm{tr}(\mathrm{W}^{\mathrm{T}}\, \bar{X}\bar{X}^T W)$ considers the set of variances in all directions, the desired W can be obtained by the eigen decomposition $\bar{X}\bar{X}^T$. PCA is a linear and global learning method. The distance learned between x_i and x_j is the Euclidean distance between $W^T\, \bar{x_i}$ and $W^T\, \bar{x_j}$. Also, since W is explicitly learned, PCA can also be applied to out-of-sampledata.

2.7.2 Nonlinear PCA

One of the limitations of PCA is that it is linear. In order to be able to learn the direction of distributions of nonlinear data, we can use a kernel trick [45], which is a common method in machine learning and data mining, that to tries to convert the nonlinear distribution of data in the initial space to a linear distribution in the mapped feature space. Assuming this mapping is $\phi: \mathbb{R}^d \rightarrow F$. F is a Reproducing Kernel Hilbert Space (RKHS), we can express this mapping as $\phi: x \rightarrow k(.,x)$, where $k\,(.,x)$ is a function in RKHS as explained in the following:

$$-\langle k(.,x)\rangle = f(x) \qquad \textit{Equation 2-9}$$

$$-\langle k(.,x), k(.,y)\rangle = k(x,y) \qquad \textit{Equation 2-10}$$

Where, f is a function in the same RKHS. Based on the theorem presented in [45], we can write each function $f \in F$ as follows:

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \qquad \textit{Equation 2-11}$$

2.7.3 Autoencoder

Autoencoder is a type of neural network with a generally narrow (bottleneck) hidden layer. This network tries to reconstruct the input data in the output and generally

is used for novelty detection and deep learning [46]. This network initially encodes the input and then decodes it to reconstruct it in the output. The goal of the autoencoder is to reconstruct the input itself.

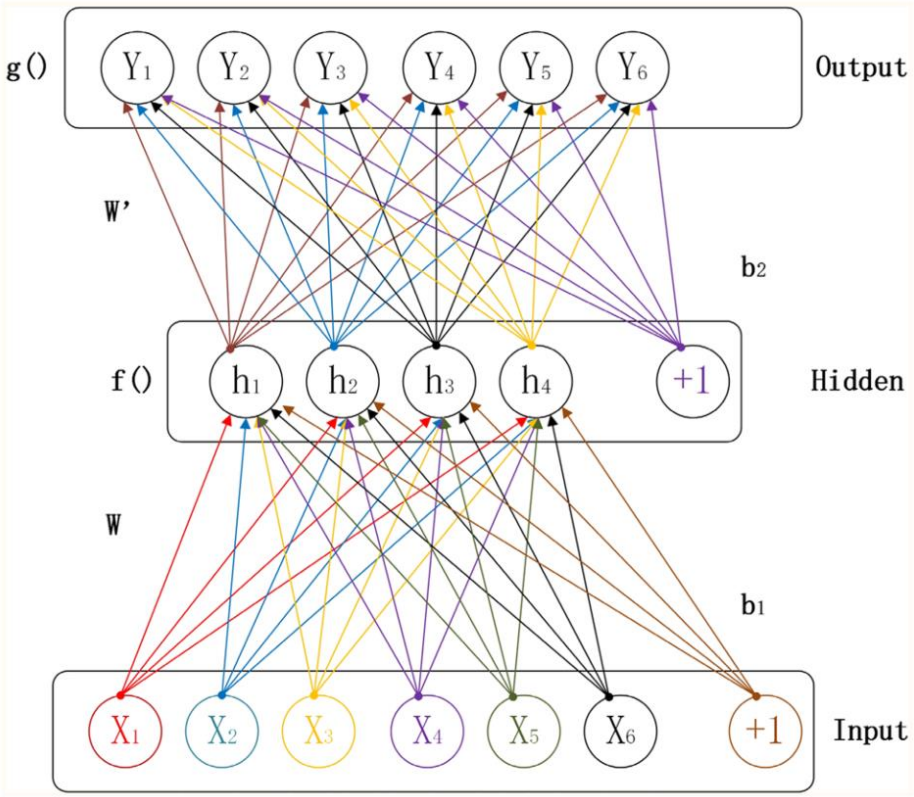


***Figure 2-2. Diagram of an Autoencoder* [47].**

The autoencoder tries to learn the function $S(.)$ as follows:

$$S_{W,W',b_1,b_2}(X) \approx X \qquad \textit{Equation 2-12}$$

In which W, W', b_1, b_2 are the model parameters. W is a weighted matrix connected to the input and hidden layers and W' is the output layer weights matrix. b_1 and b_2 are also the bias vectors of the hidden and output layers, respectively. $S(.)$ is divided into two phases. First phase, or the encoding phase is from the input to the hidden layer (*Equation 2-13*) and the second phase or the decoding phase is from the hidden layer to the output (*Equation 2-14*). Autoencoder finds the latent space (i.e., hidden variables) embedded in the input data, from the outputs of the hidden layer as denoted by h in *Equation 2-13*.

$$h = f(W \times X + b_1) \quad \textit{Equation 2-13}$$

$$Y = g(W' \times h + b_2) \quad \textit{Equation 2-14}$$

In practice, we could use the tied weight $W' = W$ to reconstruct the input X i.e., $Y \approx X$. To do this we could use the square error (*Equation 2-15*) and cross entropy loss function (*Equation 2-16*).

$$L_s(W, W', b_1, b_2; X) = \frac{1}{2}Y - X^2 \quad \textit{Equation 2-15}$$

$$L_c(W, W', b_1, b_2; X) = -[X \log Y + (1 - X) \log(1 - Y)] \quad \textit{Equation 2-16}$$

In these equations, if X is a matrix with real values then we would usually use the Least square loss function and in case the values are binary then the use of the cross-entropy loss function would be more appropriate. Y could be calculated from the combination of equations (*Equation 2-13*) and (*Equation 2-14*) as shown in *Equation 2-17*:

$$Y = g(W' \times X + b_1) + b_2) \quad \textit{Equation 2-17}$$

Generally, in order to control the weights' scale and to stop the overfitting the regularization term is added to the loss functions *Equation 2-15* or *Equation 2-16* where the loss function would be finally as follows.

$$L(W, W', b_1, b_2; X) = L_t(W, W', b_1, b_2; X) + \frac{\lambda}{2}\sum_{l=1}^{nl}\sum_{i=1}^{sl}\sum_{j=1}^{sl+1}\left(W_{ij}^{l}\right)^2 \quad \textit{Equation 2-18}$$

In which, L_t shows the squared error L_s or the cross-entropy L_c. Additionally, nl shows the layer number and sl and $sl + 1$ show the units on the lth and $l + 1$th layer, respectively.

2.7.4 Locality Preserving Projections (LPP)

Transformations such as PCA find directions in which data is optimally distributed. In other words, PCA seeks directions in which total data changes are maximal. Another way to find the direction of imagery is to preserve the geometry and

proximity of the data to the original space. An example of this type of learning is the Retention Location Imaging (LPP) method [39].

The LPP tries to find a W imaging matrix that preserves the local position of the data in the original space. In this method, the local position of the data is considered based on the similarity of the data pair $\{\{\omega_{ij}\}_{i,j=1}^{n}$ in a neighborhood. This neighborhood is usually calculated by a Gaussian function as follows.

$$\omega_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) (if\ x_i \in N_j\ or\ x_j \in N_i) \qquad \textit{Equation 2-19}$$

Here N_i and N_j are neighbors around points x_i and x_j, respectively. This means that we need to maintain distances in only one local neighborhood.

The purpose of LPP is to solve the following optimization problem.

$$\min_{w} \sum_{ij:x_i \in N_j\ or\ x_j \in N_i} \|W^T x_i - W^T x_j\|^2 \omega_{ij} = tr(W^T X L X^T W) \qquad \textit{Equation 2-20}$$

$$s.t. W^T X D X^T W = I,$$

where $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix with $D_{ii} = \sum_j \omega_{ij}$. $L = D - \Omega$ is a Laplacian matrix with $\Omega_{ij} = \omega_{ij}$ if $x_i \in N_i$, otherwise $\Omega_{ij} = 0$. The optimal solution for *Equation 2-20* can be done by performing generalized special analysis on (XLX, XDX).

And the direction of optimal imaging is obtained by adding k special vectors whose corresponding eigenvalues have the smallest values. LPP is therefore a linear and local method, as it only tries to preserve the local geometry of the data. The distance learned between x_i and x_j is the Euclidean distance between $W^T x_i$ and $W^T x_j$. The computational method used in LPP is eigenvalue analysis, and with W it can also be used for off-sample data.

$$XLX^T w = \lambda XDX^T w \qquad \textit{Equation 2-21}$$

we can also use the kernel trick to build a nonlinear LPP, in which case we need to solve the problem of parsing the following eigenvalues.

$$\Phi L \Phi^T v = \lambda \Phi D \Phi^T v \Rightarrow \Phi L \Phi^T \Phi \alpha = \Phi D \Phi^T \Phi \alpha, \qquad \textit{Equation 2-22}$$

where Φ is the data matrix in the property space after kernel mapping and we have $v = \Phi\alpha$. Therefore, we have:

$$KLK\alpha = KDK\alpha \Rightarrow Ly = \lambda Dy, \qquad \textit{Equation 2-23}$$

where $y = K\alpha$ represents the embedded data after KLPP. Thus, KLPP is a local and nonlinear method.

2.7.5 Laplacian Embedding (LE)

In fact, before LPP emerged, a method called LE [42] was proposed that focused on embedding while maintaining a structural neighborhood. Here, too, neighborhood is defined as stated in *Equation 2-19*. Assuming we want to map the data in a one-dimensional space with mapped coordinates $y = [y_1, y_2, \dots, y_n]$, then LE's goal is to obtain y by solving the following optimization problem.

$$\min_{y} \sum_{i=1}^{n} (y_i - y_j)^2 \omega_{ij} = y^T L y \qquad \textit{Equation 2-24}$$

$$s.t. y^T D y = 1,$$

where L, like LLP, is the Laplacian matrix. Because LE performs mapped coordinates without any obvious mapping, LE is a nonlinear, local method. The distance learned between x_i and x_j is the Euclidean distance between y_i and y_j. The computational method used in LE is the eigenvalue decomposition method. Given that LE performs mapped coordinates without obtaining any explicit mapping, it is simply not possible to generalize this mapping to off-sample data.

2.7.6 Locally Linear embedding (LLE)

LLE [29] is another approach to achieve the embedded space which tries to preserve the local neighborhoods of the input data. The difference between LLE and the LE [42] is in the way that they calculate the neighborhoods between the points. LLE is based on the assumption of linear neighborhood between the points, which assumes that each point, $x_i (i = 1, 2, \dots, n)$, could be reconstructed using the location of its neighbors, $N_i (i = 1, 2, \dots, n)$.

$$\min_{\omega_{ij}} \sum_{i} \left\| x_i - \sum_{x} \omega_{ij} x_j \right\|^2 \qquad \textit{Equation 2-25}$$

$$s.t. \sum_{j} \omega_{ij} = 1 \ (\forall\, i = 1, 2, \dots, n)$$

In the second step, LLE tries to retrieve the mappings in a lower dimension while preserving the local relations by solving the optimization problem in *Equation 2-26*.

$$\min_{\{y_i\}_{i=1}^{n}} \sum_{i} \left\| y_i - \sum_{x} \omega_{ij} y_j \right\|^2 \qquad \textit{Equation 2-26}$$

$$s.t. \sum_{i=1}^{n} y_i = 0, \quad \sum_{i=1}^{n} y_i y_j = nI$$

LLE is also a local and non-linear method. In this approach, like LE the learned distance between x_i and x_j is the Euclidean distance between y_i and y_j. The computation method used in the LLE utilizes both quadratic programming and eigen analysis. Generalization of the LLE for the out-of-sample data is not easy as it calculates the mapped coordinates of the data directly and without calculating any explicit mappings.

2.7.7 Isometric feature mapping (Isomap)

Isomap [43] is another approach for learning the low-dimensional spaces where the geodesic distances are devised on a weighted graph with classical scaling (metric Multidimensional Scaling [48]). The main difference between the Isomap, LE and LLE is in their approach of learning the similar data-pairs. In Isomap, in addition to the similarity, the distance between the data-pairs (i.e., the dissimilarities) are first calculated, and then the classic MDS approach is used to calculate the coordinates of the mappings in a way that the pairwise distances are preserved with the best way possible.

Here, the distance between the data-pairs is measured as followed. First, a connected neighborhood graph is constructed on the dataset, this graph could be weighted or unweighted. Then the geodesic distances would be the shortest path between the data-pairs. These computations could be considered as the discrete approximation of the real geodesic distances of the data-pairs on the manifold. Thus, Isomap is a nonlinear and

global approach. The learned distance is measured by the Euclidean distance on the low-dimensional space. The computation method used in Isomap is Eigen decomposition. As in Isomap the mapped coordinates of the data are learned directly and without any explicit mappings; thus, like LE and LLE, it is not that easy to extend the Isomap to the out-of-sample data. Geodesic distance has been previously applied successfully for dimensionality reduction in classification and clustering application [22].

2.8 Supervised distance metric learning approaches

Supervised distance metric learning algorithms, which preform the learning process based on the data points and their corresponding labels, are discussed in this chapter. Referring to *Equation 2-4*, the supervised approaches perform $J(D)$ with $\lambda_2 = 0$. Like the unsupervised approaches, we divide the supervised approaches to different categories based on their characteristics.

Table 2-2. Supervised distnace metric learning algorithms.

	Local	Global
Linear	NCA [49], ANMM [50], LMNN [51]	LDA [52], LSI [11], ITML [53], MMDA [54], RCA [55]
Nonlinear	KANMM [50] , KLMNN [51]	KLDA [56], KMMDA [54], KRCA [57]

2.8.1 Linear discriminant analysis (LDA)

LDA [52] is one the popular supervised embedding approaches. This approach, searches for the directions where the data belonging to different classes are discriminated in the best way possible. To be more precise, with the assumption that the data are from C different classes, LDA defines the compactness and separation matrices as follows:

$$\sum_{C} = \frac{1}{C}\sum_{c}\frac{1}{n_c}\sum_{x_i \in c}(x_i - \bar{x}_c)(x_i - \bar{x}_c)^T \qquad \textit{Equation 2-27}$$

$$\sum_{S} = \frac{1}{C}\sum_{c}(\bar{x}_c - \bar{x})(\bar{x}_c - \bar{x})^T \qquad \textit{Equation 2-28}$$

The goal of LDA is to find W which could be calculated by solving the following equation:

$$\min_{W^TW=I} \frac{tr(W^T \sum_C W)}{tr(W^T \sum_S W)} \qquad \textit{Equation 2-29}$$

By extending the numerator and denominator of *Equation 2-29*, it could be seen that the numerator corresponds to the sum of the distances between data points and its class center after the mapping, and the denominator corresponds to sum of the distances between the center of each class and the total mean of the data after projection. Therefore, by minimizing *Equation 2-29* the inter-class scatter increases and at the same time the intra-class scatter decreases. As it is hard to solve *Equation 2-29*, some researchers [58], [59] have conducted some research on this problem. LDA is a linear global approach. The learned distance between x_i and x_j is the Euclidean distance between $W^T x_i$ and $W^T x_j$. The generalization of LDA to the out-of-sample data is easy as it learns the transformation matrix W explicitly through eigenvalue decomposition.

2.8.2 Relevant Component Analysis (RCA)

RCA [55] is another distance metric learning method that uses data pairwise constraints. The RCA's goal is to find a mapping that reinforces related variances and eliminates the unrelated ones. Here, variances are just sample variances. We assume that data variances are correlated with a specific task, if removing these variances from the data (on average) worsens the clustering or retrieval results. They will be irrelevant if they are stored in the data but not correlated with a specific task [55]. We also define small clusters and call them chunklets, which are connected components derived from must-links. The steps that the RCA includes are:

- Build chunklets based on must-link constraints, so that the data inside each chunklet is paired with must-link constraints.

- Assuming p there is a point in k chunklet, where chunklet j contains the points $\{x_{ji}\}_{i=1}^{n_j}$ and its mean $\overline{m_j}$. RCA Calculates the weighted covariance matrix inside the chunklet.

$$C = \frac{1}{p}\sum_{j}^{k}\sum_{i}^{n}(x_{ji} - \overline{m_j})(x_{ji} - \overline{m_j})^T \qquad \textit{Equation 2-30}$$

- Calculation of the whitening transformation $W = C^{\frac{1}{2}}$, and apply it to the original data: $\tilde{x} = Wx$. Intermittent use of inverse C as a precision matrix of a generalized Mahalanobis distance.

Therefore, RCA is a global and linear method. It is easy to generalize the RCA to the out-of-sampledatapoints, since this method clearly learns the mapping matrix W. The computational method used in RCA is also eigen analysis.

2.8.3 Information Theoretic Metric Learning (ITML)

The objective function based on information theory is an approach to developing a supervised distance criterion. An example of this type of approach is ITML [53]. Assuming we have a generalized Mahalanobis spaced parameterized by the M_0 precision matrix, an M set of must-link constraints and a set of cannot cannot-link constraints. ITML solves the following optimization problem:

Equation 2-31

$$\min_{M \geq 0} d_{logdet}(M, M_0)$$

$$s.t.(x_i - x_j)^T M(x_i - x_j) \geq l, (x_i, x_j) \in \mathsf{C}$$
$$(x_u - x_v)^T M(x_u - x_v) \leq u, (x_u, x_v) \in \mathcal{M}$$

d_{logdet} is the LogDet divergence, also called Stein loss. It can be shown that Stein loss is a constant-scale constant loss function for which the neutral estimator of the least uniform changes is also an equivalent minimum error estimator [53]. The authors in [53] also present a Bergman illustration method for problem solving. ITML is a global, linear method. The distance learned is the distance of the Mahalanobis from the M accuracy matrix. Because the M accuracy matrix, which is used to estimate distance between data pairs, is learned, ITML generalization to off-sample data is easy. Also, the computational method used in ITML is Bergman illustration.

$$d_{logdet}(M, M_0) = tr(MM_0^{-1}) - logdet(MM_0^{-1}) - n \qquad \textit{Equation 2-32}$$

2.8.4 Neighborhood Component Analysis (NCA)

Unlike global learning methods such as LDA, NCA [49] is a local method for supervised distance learning. In this method, each point x_i selects another point x_j with the probability of p_{ij}, and assigns its class label based on the selected point. NCA defines the probability of selecting a point as a neighbor as follows. Under this random selection rule, the NCA calculates the probability that point i will be classified correctly.

$$p_{ij} = \frac{\exp(-\left\|W^T x_i - W^T x_j\right\|^2)}{\sum_{k \neq i}(-\|W^T x_i - W^T x_k\|^2)} \qquad \textit{Equation 2-33}$$

Where $L_i = \{j | l_i = l_j\}$ is a set of points in the same class as point i.

$$p_i = \sum_{j \in L_i} p_{ij} \qquad \textit{Equation 2-34}$$

The goal of the NCA is to maximize the number of correctly classified points as follows:

$$J(W) = \sum_i p_i = \sum_i \sum_{j \in L_i} p_{ij} \qquad \textit{Equation 2-35}$$

In [49], the authors have proposed a truncated gradient descent method to minimize $J\,(W)$. NCA is a local and linear approach. The distance learned between x_i and x_j is the Euclidean distance between $W^T x_i$ and $W^T x_j$. Generalizing the NCA to the out-of-sample data is easy, because the mapping matrix W is explicitly learned, and the computational method used is the eigen decomposition.

2.8.5 Discriminative Least Squares Regression (DLSR)

discriminative least square approach proposed in [60] is a framework for computing the least square regression (LSR) for multiclass classification. The main goal of this approach is to enlarge the distances between different classes under the framework of the LSR. To do so, [26] has utilized a method called the ϵ-dragging to push the regression objective of different classes back in different directions in a way that the distance between different classes is increased. With the assumption of having n training

samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ in $c(\geq 2)$ classes, where $\boldsymbol{x}_i$ is a datapoint in $\mathbb{R}^m$ and $y_i \in \{1, 2, \dots, c\}$ is the label of $\boldsymbol{x}_i$. The main goal of the DLSR is to learn the following linear function:

$$\boldsymbol{y} = \boldsymbol{W^T x} + \boldsymbol{t} \qquad \textit{Equation 2-36}$$

Note that an arbitrary set of c independent vectors in $\mathbb{R}^c$ is capable of identifying c classes independently. Thus, $0/1$ class label vectors cloud be used as the regression objective for the multiclass classification. In other words, for the jth class, $j = 1, 2, \dots, c$, $\boldsymbol{f}_j = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^c$ could be de fined by making the jth element equal to one in a way that for n training examples we would have:

$$\boldsymbol{f}_{y_i} \approx \boldsymbol{W}^T \boldsymbol{x}_i + \boldsymbol{t}, i = 1, 2, \dots, n, \qquad \textit{Equation 2-37}$$

Where, W is a transformation matrix in $\mathbb{R}^{m \times c}$ and $\boldsymbol{t}$ is a translation vector in $\mathbb{R}^c$.

In order to develop a compressed optimization method for multiclass classification, assume that $\boldsymbol{B} \in \mathbb{R}^{n \times c}$ be a constant matrix where the ith row and the jth column and is defined as follows:

$$B_{ij} = \begin{cases} +1, & if\ y_i = j \\ -1, & otherwise. \end{cases} \qquad \textit{Equation 2-38}$$

From the geometrical viewpoint, each element in B corresponds to a dragging direction. In other words, "+1" indicates the dragging towards the positive direction, whereas "-1" shows the dragging in the negative direction. By performing the mentioned dragging on each element of $\boldsymbol{Y}$ and recording this epsilon with matrix $\boldsymbol{M}$, we would have the following equation:

$$\boldsymbol{XW} + \boldsymbol{e}_n \boldsymbol{t}^T - (\boldsymbol{Y} + \boldsymbol{B} \odot \boldsymbol{M}) \approx 0 \qquad \textit{Equation 2-39}$$

Where $\odot$ indicates the Hadamard (or elementwise) multiplication and is $\boldsymbol{e}_n = [1, 1, \dots, 1]^T \in \mathbb{R}^n$ a vector of ones.

Now by obtaining the regularized framework of the LSR, we would have the following learning model:

$$\min_{\boldsymbol{W},\boldsymbol{t},\boldsymbol{M}} \|\boldsymbol{XW} + \boldsymbol{e_n t^T} - \boldsymbol{Y} - \boldsymbol{B} \odot \boldsymbol{M}\|_F^2 + \lambda \|\boldsymbol{W}\|_F^2, \qquad \textit{Equation 2-40}$$

Where, λ is a positive regularization term and $\|.\|_F$ indicates the Fresenius norm.

By adding the term $\boldsymbol{B} \odot \boldsymbol{M}$ in *Equation 2-40* which is related to the ϵ-dragging for enlarging the inter-class distances, this model could be used for a constrained optimization problem. Based the convex optimization theory, the convexity of *Equation 2-40* could be easily justified [60] and on this basis it would have one unique answer. For more details on the DLSR algorithm, one could refer to [60]. The ϵ-dragging method is applied as one of the key ideas in our proposed approach.

2.9 Semi-supervised distance metric learning

The main purpose of semi-supervised methods is to learn the distance criterion on the data when only monitoring information is provided for a small part of the data. These algorithms use both observational and unsupervised data in the learning process. Therefore, a simple method for constructing a semi-supervised algorithm using an object such as *Equation 2-4* with the values $\lambda_1 \neq 0, \lambda_2 \neq 0$, in which case $U(D)$ can be applied to all data in some way. Unsupervised and $L(D)$ can also be made on the part of the labeled data. Finally, some constraints can be applied to the distance learning criteria to balance the two parts. Table 2-3, shows the classification of some semi-supervised methods along with their characteristics.

Table 2-3. Semi-supervised distance metric learning algorithms.

	Local	Global
Linear	LRML [61]	LRML [61], CMM [62]
Nonlinear	KLRML [61], SSDM [63], MPCK-means [64]	KLRML [61], KCMM [62], SSDM [63]

2.9.1 Laplacian Regularized Metric Learning (LRML)

LRML [61] is a semi-supervised method of distance learning. LRML uses the LPP formulation [39] for the unsupervised term $U(D)$, and for the supervised term the ANMM method [50]. Shows the optimization problem that LRML is trying to solve. Where the smoothing term is defined as follows.

$$\min_{M} t + \gamma_1 t_2 - \gamma_2 t_3 \qquad \textit{Equation 2-41}$$
$$s.t.\, t_1 \le t$$
$$M \ge 0,$$

where $M = WW^T$. Considerable terms including compaction and dispersion are:

$$t_1 = \sum_{i,j} \left\| W^T x_i - W^T x_j \right\|^2 \omega_{ij} = tr(W^T XLX^T W) = tr(XLX^T M) \qquad \textit{Equation 2-42}$$

Equation 2-43

$$t_2 = \sum_{(x_i, x_j) \in M} \left\| W^T x_i - W^T x_j \right\|^2$$
$$= tr\left[M \sum_{(x_i, x_j) \in M} (x_i - x_j)(x_i, x_j)^T \right]$$

in which, M and C are must-link and cannot-link sets, respectively.

$$t_3 = \sum_{(x_i, x_j) \in C} \left\| W^T x_i - W^T x_j \right\|^2 = tr\left[M \sum_{(x_i, x_j) \in C} (x_i - x_j)(x_i, x_j)^T \right] \qquad \textit{Equation 2-44}$$

In [61] a semi-deterministic programming method for solving the problem of *Equation 2-41* is presented. LRML is a combination of local (unsupervised) and global (supervised) and linear methods. The distance learned is the distance of the Mahalanobis to the accuracy matrix M. Therefore, since the M accuracy matrix is learned, LRML can also be generalized to off-sample data. The computational method used in this method is quadratic programming.

2.9.2 Constraint Margin Maximization (CRM)

Similarly, CMM [62] uses PCA as its unsupervised term in its objective function and ANMM for the observable term as well as LRML. Thus, the CMM optimization problem will be as follows:

Where the unsupervised term $t_4 = tr(W^T X \Sigma X^T W)$ is the same as the PCA target. It should be noted that before applying CMM, the average of all data must be zero. The goal of CMM is to maximize data point variations while meeting the limitations of data pairs in mapped space. The authors in [62] have shown that the optimal W is obtained by

the standard eigenvalue decomposition process. Also shown in [62] is how to obtain a kernel-based version for dealing with nonlinear data. The distance learned between x_i and x_j is the Euclidean distance between $W^T x_i$ and $W^T x_j$. Generalization of CMM is easy for out-of-sample data, as the mapping matrix W is explicitly learned, and the computational method used is eigen analysis.

$$\max_{W} t_4 - \gamma_1 t_2 - \gamma_2 t_3 \qquad \textit{Equation 2-45}$$
$$s.t. W^T W = I,$$

Where the unsupervised term $t_4 = tr(W^T X \Sigma X^T W)$ is the PCA target function. It should be noted that all data must have a mean of zero before applying CMM. The goal of CMM is to maximize data changes in the mapped space while meeting binary constraints.

2.10 Summarization and conclusion of this chapter

In this chapter, different methods of learning distance criteria and dimensional reduction are examined and classified into three categories, which are briefly described as follows:

• Unsupervised methods whose learning is based on unlabeled data, all of which formulate the learning process as an optimization problem, in which the objective function can be based on spatial geometry or information theory. In all of these methods except PCA, there are release parameters such as kernel parameters for kernel-based methods, neighborhood size, or scaling parameters that must be determined in advance. From the point of view of generalization to non-sample data (data that does not have a training set), linear and core-based methods are preferable to other methods, because these methods can be directly and indirectly. Learn matrix mapping and illustration. But for methods such as Isomap, LLE, and LE that implicitly learn mapped coordinates, additional work is needed to learn the mapping matrix [65].

• Supervised methods, in which learning is done on labeled training data, similar to unsupervised methods, require the initial adjustment of free parameters, and most of them involve some heavy computational processes such as parsing. Are special or semi-definitive programming. The ITML method is an exception to this, as it uses Bergman's imaging strategy, which can increase computational efficiency [66].

• Semi-supervised methods, in which the algorithm accesses a large number of unlabeled samples for which no additional information is available, can be used as a

combination of supervised methods and Considered without an observer. These methods can be used when the supervised information on the data is very limited and sparse. However, not all regulatory information is necessarily useful for distance learning. Accordingly, some researchers have conducted research on the usefulness of supervised information [67]. Research has also been done on when unlabeled data can be useful [68]. In practice, the balance between regulatory data usage and unlabeled data will vary depending on the data distribution infrastructure. And as far as we know, there are no universal rules for optimizing parameters.

Chapter 3: Methodology

3.1 Introduction

This chapter will describe the proposed method of distance metric learning in detail. The proposed method tries to learn the distance metric in way that the structures between the data-points are preserved as much as possible. In this approach, in order to encounter with the problem of the imbalanced distributions of different classes, for each given data point, two neighborhoods are created, each of which consisting of the data with similar and dissimilar labels to the given data point, respectively. The proposed method tries to preserve the spatial locality of the similar data in relation with each other and to push back the dissimilar data from each data-point. On this basis, and with respect to the fact that the number of the data points in the similar neighborhood is equal to the number of points in the dissimilar neighborhood, the problem of the imbalanced data distributions could also be covered.

3.2 Proposed method

As it can be seen in Figure 3-1, in the proposed method, in order to increase the manifold and distance metric learning speed besides reaching a feasible amount of system memory on today's computers, first the number of the training data is down-sampled, otherwise the size of the similarity matrix would as big and bulky that it could not be implemented and, as a result, the execution of the proposed method would not be possible. In order to encounter with such big data, a uniform random sampling of the training data is preformed through which the share of each class in training samples will be remained intact. The down-sampling factor is considered to be 0.1 of all samples.

After sample reduction, manifold learning is conducted on these data using one of the manifold learning approaches in order to extract the local neighborhoods of the nodes based on their adjacencies on the manifold. Consequently, based on these extracted local neighborhoods, two neighborhoods are created for each given data point. As it can be

seen in Figure 3-2, one of the created neighborhoods is dedicated to the data with the same label whereas the other neighborhood consists of the dissimilar neighbors to the given data point. Other data points are regarded as so called unrelated set. Finally, as it is depicted in Figure 3-3, distance metric learning based on the initial coordinates of the given data point in ambient space and with respect to the similarity and dissimilarity relations thanks to constructed similar and dissimilar neighborhoods is conducted in a way that the similar data points to the given point would be more close to it than the other dissimilar points.

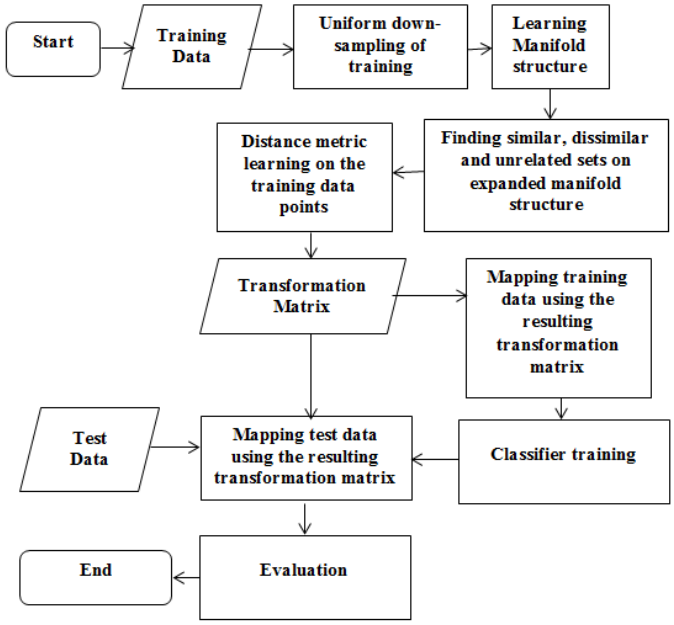


Figure 3-1. Overall process of the proposed method.

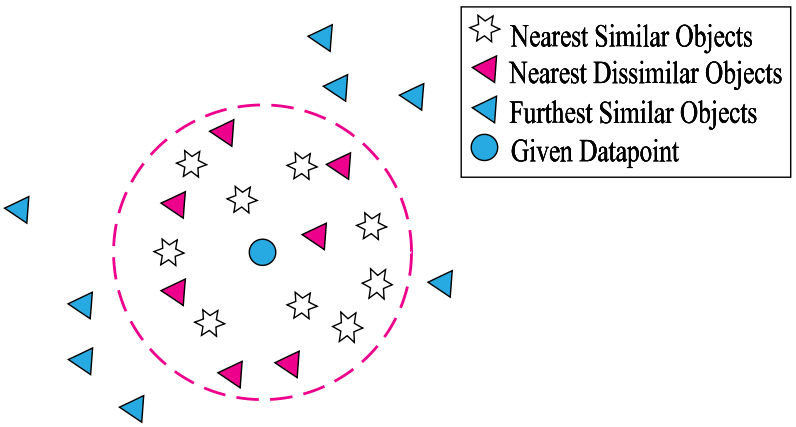


Figure 3-2. The local patch consisting of the dissimilar neighbors.

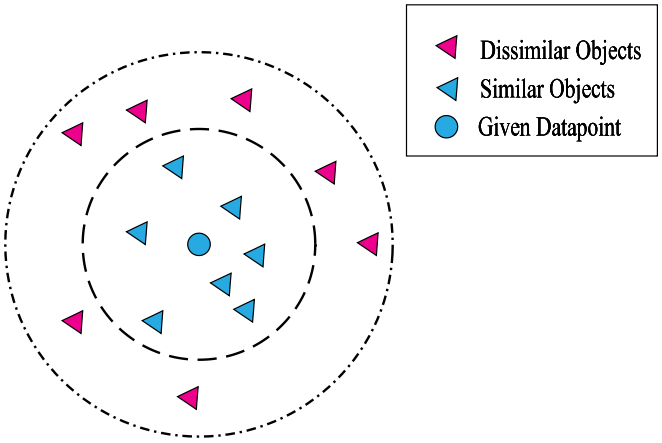


Figure 3-3. The local neighborhoods after distance metric learning with the proposed approach.

3.2.1 Data neighborhood structures after distance metric learning

After discriminating the similar and dissimilar neighborhoods, as well as the unrelated data points which are not contained in either of similar and dissimilar neighborhoods, they are ordered as shown in Figure 3-4, based on their distance to the given data point and the following relation vector is created.

x_i	S_i	D_i	U_i

Figure 3-4. The representation of data points after manifold embedding and similarity calculation.

In Figure 3-4, x_i shows the given data point, S_i shows the faraway points from x_i with the similar class labels and, D_i shows the neighbors with the dissimilar data and U_i indicates the unrelated data which are not included in either of the similar and dissimilar sets with respect to the given data point. In the other words, if a data point is not a member of either of their similar or dissimilar neighborhoods, it is said to be unrelated.

At this stage one of the distance metric learning methods e.g., the Mahalanobis distance, could be used. In the proposed framework, we have adopted the Discrete Least Square Regression (DLSR), proposed in [60] and modified the approach in it in order to be compatible with the proposed distance metric learning. Having the above

similar/dissimilar/unrelated sets the proposed approach can be formulated as the following optimization problem as inspired from [26]:

$$\min\|XW + e_n t^T - Y - B \odot M\|_F^2 + \lambda\|W\|_F^2 \qquad \textit{Equation 3-1}$$

In our proposed method, $X \in \mathbb{R}^{n\times m}$ is the input data matrix, $W \in \mathbb{R}^{m\times n}$ is the transformation matrix to the similarity space (resulted from the distance metric learning) and $e_n = [1, 1, \dots, 1]^T \in \mathbb{R}^n$ is a vector consisting of ones. Also, $Y \in R^{n\times n}$ and $B \in R^{n\times n}$ are two constant matrices each of which are in the *i*th row and the *j*th column as follows:

$$Y_{i,j} = \begin{cases} 1, & if\ l_i = j\ , j \in P_{s_i} \\ 0, & otherwise \end{cases} \qquad \textit{Equation 3-2}$$

$$B_{i,j} = \begin{cases} +1, & if\ l_i = j\ , j \in P_{s_i} \\ -1, & if\ l_i = j\ , j \in P_{d_i} \\ 0, & otherwise \end{cases} \qquad \textit{Equation 3-3}$$

Where P_{s_i} and P_{d_i} show the similar and dissimilar sets for each given data point. In other words, each element $Y_{i,j}$ will be equal to one in case that the *j*th data point which has the same label as i be in the furthest neighborhood of the given it. Also, matrix B shows the similar/dissimilar/unrelated set (+1,-1 and 0 respectively) information as gathered from the previous stage. The other matrices and variables included in [60], as well as the calculations of transformation matrix, W, are done precisely based on the assumptions contained in the DLSR algorithm [60].

As you can see in Figure 3-1, after calculating the mapping matrix W, all the training/test data are mapped to the similarity space using the following equation.

$$X' = X \times W + e_n \times t^T \qquad \textit{Equation 3-4}$$

Where X' is the transformed data matrix, showing the data mapped onto the similarity space and also $t \in \mathbb{R}^n$ is a translation vector according to the assumptions in [60].

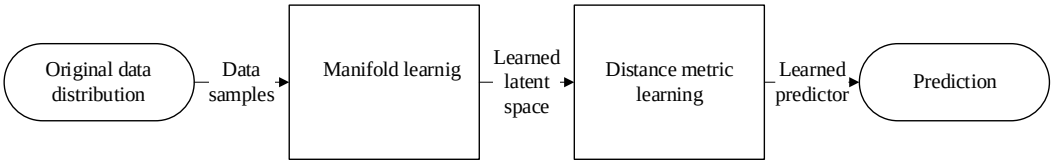


Figure 3-5. The general process in metric learning.

3.3 The objective of distance metric learning

As it is shown in Figure 3-2 and Figure 3-3, the main objective of a distance metric learning algorithm, is to learn the parameters of the metric which are best suited for the constraints in such a way that it is the best approximation of the distance embedded between the data points. Distance metric learning is commonly expressed as an optimization problem, as the general form below:

$$\min_{M} l(M, S, D, R) + \lambda R(M) \qquad \textit{Equation 3-5}$$

Where $l(M, S, D, R)$ is a loss function that acquires a penalty in case the training constraints are violated and $R(M)$ regularizes the parameters M of the learned metric and $\lambda \geq 0$ is a regularization parameter.

After the learning phase, the resulted function is used to improve the performance of a metric-based algorithm, which is most commonly k-Nearest Neighbors (k-NN). The main goal of using the k-NN is to preserve the symmetry in the distance metric learning phase, with the sense that, as seen in Figure 3-2 and Figure 3-3, the number of the similar neighbors is equal with the number of the neighbors from other classes around each data point. As a result, the supremacy of the proposed method is that it learns the distance

metric in a balanced way as it uses an equal number of the similar and dissimilar data points to learn the distance metric.

3.4 Advantages of the proposed method

One of the advantages of the proposed method is that it preserves the structure of similar local data, so that the proposed method tries to locate the data that are in the vicinity of each point and on the patch of data with the same label as the data point. Have the least change of location and remove only neighboring data labeled as dissimilar to the data point. Thus, when entering test data, this data will be mapped in the new space with minimal change in location relative to their neighbors.

Another advantage of the proposed method is that the distance learning criterion is balanced over similar and dissimilar data. Because in the proposed method, considering the number of identical data points on similar and dissimilar patches with each data point, an attempt has been made to learn the distance criterion by observing the balance between similar and dissimilar samples with each data point.

Chapter 4: Experiments

4.1 Introduction

This chapter will make a comparison between the proposed method and the Discriminative Least Squares Regression (DLSR) [60] and some other fundamental methods of dimensionality reduction.

4.2 Dataset

In order to evaluate the proposed method in this research the following numeric datasets which are obtained from the UCI repository of machine learning are employed.

Table 4-1. The properties of the datasets.

Dataset	#Samples	# Class	# Features	Imbalance ratio
Vehicle	846	۴	18	1.09
Bupa	345	2	6	1.37
Glass	214	6	8	8.44
Ionosphere	351	2	34	1.78
Iris	150	3	4	1
KDD	494021	۵	۴۱	7528.03
Monks	124	2	6	1
New-thyroid	215	3	5	5
Pima	768	2	8	1.86
WDBC	569	2	30	1.68
Wholesale	440	2	7	2.09
Wine	178	3	13	1.47

In which the imbalance ratio is the proportion of the population of the majority class to the population of the minority class which could be calculated from the following equation.

$$R_{Im} = \frac{n_{major}}{n_{minor}} \qquad \textit{Equation 4-1}$$

Where R_{Im} is the imbalance ratio and n_{major} and n_{minor} are the population of the majority class to the minority class, respectively.

4.3 Evaluation criteria

In order to compare the proposed method with other approaches we have employed the following evaluation criteria:

Accuracy or the correct rate is the proportion of the correctly classified data to the total number of the items in the dataset.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$ *Equation 4-2*

Sensitivity, true positive rate (TPR), recall, or the hit rate, is the proportion of the data which are correctly classified in the positive class to the total of the positive data.

$$SEN = \frac{TP}{TP + FN}$$ *Equation 4-3*

Specificity or true negative rate is the proportion of the negative points which are correctly classified in the negative class to the total number of the negative samples.

$$SPC = \frac{TN}{TN + FP}$$ *Equation 4-4*

4.4 Evaluation scenarios and experimental results

In this chapter we will analyze and make a comparison between the performance results of the proposed method and some other well-known approaches of distance metric learning and dimensionality reduction and also the original DLSR algorithm with respect to the evaluation measures. To do this, the 10-fold cross validation is utilized. The results are based on the performance of the two k-NN classifier and SVM classifier with the RBF kernel. The accuracy of different approaches including DLSR and the proposed approach are depicted in Table 2. In these experiments the proposed method has employed different manifold learning approaches such as PCA, LDA, MDS, Isomap, LLE, Kernel PCA and

Autoencoder. The experiments are performed for different latent dimensions and the best results are reported in the tables. Note that, in the following tables (d, R) respectively show the best latent dimension and the rank of the method on the corresponding dataset.

Table 4-2. Accuracy comparison between different approaches versus the proposed using 10-fold cross validation and 7-NN classifier with (d,r) indicating the best latent dimensionality and the rank of the approach, respectively (AE denotes auto-encoder approach).

Dataset	Dimensionality Reduction				Feature Selection			Proposed Method						
	PCA	LLE	Kernel PCA	AE	Fisher	Gini	DLSR	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	AE
Vehicle	0.6823 (13, 9)	0.6117 (17, 12)	0.2588 (9, 14)	0.5294 (5, 13)	0.6823 (17, 9)	0.6705 (17, 11)	0.9183 (17, 1)	0.8235 (1,4)	0.8705 (1, 3)	0.8235 (1, 4)	0.8235 (1, 4)	0.8941 (13, 2)	0.8 (1, 8)	0.8235 (13, 4)
Bupa	0.5714 (1, 12)	0.6571 (3, 9)	0.4285 (5, 14)	0.5714 (3, 12)	0.6857 (5, 7)	0.7428 (1, 9)	0.7573 (3, 2)	0.6285 (1, 10)	0.7714 (5, 1)	0.6285 (1, 10)	0.6857 (1, 7)	0.7142 (3, 5)	0.7142 (1, 5)	0.7428 (5, 3)
Glass	0.5454 (1, 10)	0.5454 (5, 10)	0.5 (7, 12)	0.4545 (1, 13)	0.7272 (9, 3)	0.7272 (9, 3)	NA	0.7272 (1, 3)	0.7272 (3, 3)	0.7272 (1, 3)	0.7272 (5, 3)	0.7727 (3, 1)	0.7727 (5, 1)	0.7272 (9, 3)
Ionosphere	0.8888 (8, 11)	0.8055 (22, 14)	0.9166 (8, 5)	0.8888 (15, 11)	0.9166 (15, 5)	0.9166 (15, 5)	0.8694 (29, 13)	0.9166 (15, 5)	0.9722 (1, 1)	0.9166 (15, 5)	0.9444 (15, 4)	0.9722 (29, 1)	0.9722 (1, 1)	0.9166 (29, 5)
Iris	1 (1, 1)	1(2, 1)	1(2, 1)	1 (1, 1)	1(2, 1)	1(2, 1)	NA	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
KDD	0.9879 (1, 10)	0.9839 (28, 12)	0.7915 (10, 14)	0.9819 (37, 13)	0.9919 (10, 6)	0.9919 (19, 6)	0.9901 (28, 9)	0.9939 (10, 2)	0.9939 (10, 10)	0.9939 (1, 2)	0.9939 (1, 2)	0.9959 (19, 1)	0.9939 (10, 2)	0.9919 (1, 6)
Monks	0.8333 (5, 8)	0.9166 (5, 4)	0.8333 (3, 8)	0.5 (1, 14)	0.5833 (3, 12)	0.5833 (3, 12)	0.7916 (5, 11)	1(5, 1)	0.8333 (3, 8)	1(5, 1)	0.9166 (5, 4)	0.9166 (3, 4)	0.9166 (3, 4)	1(3, 1)
New-thyroid	0.9545 (5, 1)	0.9090 (3, 3)	0.5909 (1, 13)	0.9090 (3, 3)	0.9090 (1, 3)	0.9090 (1, 3)	NA	0.9090 (1, 3)	0.9090 (1, 3)	0.9090 (1, 3)	0.9090 (1, 3)	0.9090 (1, 3)	0.9545 (5, 1)	0.9090 (1, 3)
Pima	0.7532 (3, 10)	0.7142 (7, 12)	0.6493 (3, 14)	0.6883 (7, 13)	0.7922 (1, 2)	0.7922 (1, 2)	0.7597 (1, 9)	0.7792 (5, 4)	0.8051 (1, 1)	0.7792 (5, 4)	0.7792 (7, 4)	0.7792 (1, 4)	0.7792 (3, 4)	0.7532 (7, 10)
WDBC	0.9298 (7, 9)	0.9122 (8, 12)	0.6315 (22, 14)	0.8596 (8, 13)	0.9298 (22, 9)	0.9298 (22, 9)	0.9807 (1, 4)	0.9473 (1, 5)	0.9473 (1, 5)	0.9473 (1, 5)	0.9473 (1, 5)	0.9824 (15, 1)	0.9824 (1, 1)	0.9824 (1, 1)
Wine	0.7777 (4, 9)	0.9444 (13, 8)	0.3888 (1, 13)	0.7222 (1, 12)	0.7777 (13, 9)	0.7777 (13, 9)	NA	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(4, 1)	1(1, 1)
Wholesale	1(3, 1)	0.9545 (3, 11)	0.6818 (3, 13)	0.7045 (5, 12)	1(5, 1)	1(7, 1)	NA	1(1, 1)	0.9772 (1, 8)	1(1, 1)	1(3, 1)	1(7, 1)	0.9772 (1, 8)	0.9772 (1, 8)
Average rank	7.66	9	11.25	10.83	5.67	5.5	7.14	3.33	3.08	3.33	3.25	2.08	3.08	3.92

As it can be seen in Table 4-2, from the total of 12 experiments on different datasets, the proposed method of distance metric learning using the LLE, Kernel PCA and LDA approaches for manifold learning has gained the first rank on 7, 6 and 5 datasets, respectively. While, under the same circumstances the other methods such as the pure manifold learning, feature selection and the DLSR have achieved the best accuracy only in one experiment which is still equal to the result of the proposed method.

Therefore, from total of 12 experiments, the proposed framework, has totally gained the first rank, whereas the base approaches have the first rank only in one experiment which is a testimony of the absolute excellence of the proposed approach from the accuracy viewpoint using 7-NN classifier.

Also, with respect to the fact that among different manifold learning methods combined with DML, LLE has gained the maximum rank, it could be concluded that this approach has got the best performance in finding the structural neighborhoods in comparison with the other manifold learning approaches in terms of the accuracy using the 7-NN classifier.

Table 4-3. Sensitivity comparison between different approaches versus the proposed using 10-fold cross validation and 7-NN classifier with (d,r) indicating the best dimensionality and the rank of the approach, respectively (AE denotes auto-encoder approach).

Dataset	Dimensionality Reduction				Feature Selection			Proposed Method						
	PCA	LLE	ernel PCA	E	Fisher	Gini	DLSR	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	AE
Vehicle	0.95 (13, 9)	0.75 (4, 14)	1(1, 1)	0.95 (9, 9)	0.9846 (5, 7)	0.9692 (17, 8)	1(13, 1)	0.95 (1, 9)	1 (1, 1)	0.95 (1, 9)	0.95 (5, 9)	1(1, 1)	1(1, 1)	1 (13, 1)
Bupa	0.4667 (3, 13)	0.6666 (3, 6)	1 (1, 1)	0.5333 (5, 9)	0.75 (5, 4)	0.8 (1, 2)	0.7616 (3,3)	0.5333 (1, 9)	0.7333 (5, 5)	0.5333(1, 9)	0.5333 (1, 9)	0.6 (3, 7)	0.4667 (1, 13)	0.6 (3, 7)
Glass	0.8571 (1, 8)	0.8571 (1, 8)	0.5 (7, 13)	0.7142 (1, 12)	0.8571 (5, 8)	0.8571 (5, 8)	NA	1(3, 1)	1(5, 1)	1(3, 1)	1(1, 1)	1(3, 1)	1(3, 1)	1(1, 1)
Ionosphere	0.9565(8, 10)	0.9565 (22, 10)	0.9130 (1, 14)	0.9522 (15, 10)	1(1, 1)	1(1, 1)	0.9913 (29, 9)	1 (15, 1)	1(1, 1)	1 (15, 1)	1(1, 1)	1(1, 1)	0.9522(1, 10)	1 (29, 1)
Iris	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	NA	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
KDD	1(10, 1)	1(10, 1)	1 (10, 1)	1(19, 1)	0.9919 (10, 14)	0.9974 (10, 12)	0.9971 (19, 13)	1 (10, 1)	1 (1, 1)	1 (1, 1)	1(1, 1)	1(1, 1)	1 (10, 1)	1(1, 1)
Monks	1(5, 1)	1(1, 1)	0.8333 (3, 12)	1(3, 1)	1(5, 1)	1(1, 1)	0.7333 (5, 13)	1(5, 1)	0.6666(1, 14)	1(5, 1)	1(5, 1)	1(3, 1)	1(3, 1)	1(3, 1)
New-thyroid	1(5, 1)	1(1, 1)	0.8666(1, 13)	1(5, 1)	1(1, 1)	1(1, 1)	NA	1 (1, 1)	1 (1, 1)	1 (1, 1)	1 (1, 1)	1 (1, 1)	1 (1, 1)	1 (1, 1)
Pima	0.4444(3, 13)	0.4444 (3, 13)	1(1, 1)	0.9259 (3, 3)	0.7037 (1, 3)	0.7037 (1, 3)	0.6185 (1, 5)	0.5185 (1, 8)	0.5555 (7, 6)	0.5185 (5, 8)	0.5185 (1, 8)	0.5555 (1, 8)	0.5185 (1, 8)	0.5185 (5, 8)
WDBC	1(8, 1)	1(8, 1)	1 (22, 1)	1(1, 1)	1(22, 1)	1(22, 1)	0.9861 (1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
Wine	0.85 (1, 10)	0.8333 (4, 11)	1(4, 1)	1(4, 1)	0.8333 (4, 11)	0.8333 (1, 11)	NA	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
Wholesale	1(3, 1)	0.9666 (3, 11)	1(3, 1)	1(1, 1)	1(5, 1)	1(7, 1)	NA	1(1, 1)	0.9666 (1, 11)	1(1, 1)	1(3, 1)	1(7, 1)	0.9666 (1, 11)	1(5, 1)
Average rank	5.75	6.5	5	4.08	4.42	4.42	8.29	2.92	3.67	2.92	2.92	1.92	4	2.08

Table 4-3, denotes the comparison between the proposed methods and other approaches of distance metric learning and dimensionally reduction in term of sensitivity. As it can be seen in Table 4-3, from the total of 12 experiments on different datasets, the proposed method of distance metric learning using LLE, Auto-encoder and the PCA

approaches of manifold learning has gained the first rank on 10, 10 and 9 datasets, respectively. Whereas, under the same circumstances from the other methods, approaches such Auto-encoder, Gini and Fisher has gained the first rank in 7, 6 and 6 experiments, respectively.

Table 4-4. Specificity comparison between different approaches versus the proposed using 10-fold cross validation and 7-NN classifier with (d,r) indicating the best dimensionality and the rank of the approach, respectively (AE denotes auto-encoder approach).

Dataset	Dimensionality Reduction				Feature Selection			Proposed Method						
	PCA	LLE	Kernel PCA	AE	Fisher	Gini	DLSR	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	AE
Vehicle	0.9692 (13, 11)	0.8923 (17, 13)	1 (5, 1)	0.8307(1, 14)	0.9846(5, 10)	0.9692 (17, 11)	1 (17, 1)	1(1, 1)	1 (1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
Bupa	0.7 (1, 11)	0.65 (3, 13)	0 (1, 14)	0.7 (1, 11)	0.75 (5, 8)	0.8 (1, 3)	0.7564 (3, 7)	0.75 (3, 8)	0.8 (1, 3)	0.75 (3, 8)	0.8(1, 3)	0.8 (3, 3)	0.9(1, 1)	0.9 (5, 1)
Glass	0.6666 (7, 13)	0.7333 (3, 10)	0.8 (1, 1)	0.8 (5, 1)	0.7333 (1, 10)	0.7333 (1, 10)	NA	0.8 (1, 1)	0.8 (3, 1)	0.8(1, 1)	0.8(3, 1)	0.8 (1, 1)	0.8(5, 1)	0.8 (1, 1)
Ionosphere	0.7692 (8, 8)	0.6153 (15, 14)	1(8, 1)	0.7692 (15, 8)	0.8461 (15, 4)	0.8461 (15, 4)	0.7076(8, 13)	0.7692(1, 8)	0.9230(1, 2)	0.7692 (1, 8)	0.8461 (15, 4)	0.9230(29 , 2)	0.8461 (1, 4)	0.7692(1, 8)
Iris	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	NA	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
KDD	0.9906 (10, 4)	0.9439 (10, 13)	0.0373 (10, 14)	1 (37, 1)	1 (10, 1)	0.9906 (10, 4)	0.9953 (37, 3)	0.9906 (10, 4)	0.9906 (1, 4)	0.9906 (1, 4)	0.9906 (1, 4)	0.9906 (1, 4)	0.9906 (1, 4)	0.9906 (1, 4)
Monks	0.8333 (3, 9)	0(1, 14)	0.8333 (1, 9)	1(3, 1)	0.6666 (5, 12)	0.6666 (5, 12)	0.85 (5, 8)	1(1, 1)	1(1, 1)	1(3, 1)	1(1, 1)	0.8333 (1, 9)	1(1, 1)	1(1, 1)
New-thyroid	0.8571 (3, 2)	0.7142 (3, 7)	0.8666 (1, 1)	0.8571 (5, 2)	0.8571 (3, 2)	0.8571 (3, 2)	NA	0.7142 (1, 7)	0.7142 (1, 7)	0.7142 (1, 7)	0.7142 (1, 7)	0.7142 (1, 7)	0.8571 (5, 2)	0.7142 (1, 7)
Pima	0.92 (5, 5)	0.9 (7, 12)	1 (3, 11)	0.82 (7, 14)	0.92 (3, 5)	0.92 (3, 5)	0.836 (7, 13)	0.92 (5, 5)	0.96 (1, 2)	0.92 (5, 5)	0.92 (7, 5)	0.94 (7, 3)	0.92 (3, 5)	0.94 (7, 3)
WDBC	1(1, 1)	0.7619 (1, 13)	1(1)	0.7619 (8, 13)	0.8095 (8, 11)	0.8095 (8, 11)	0.9714 (1, 3)	0.8571 (1, 7)	0.8571 (1, 7)	0.8571 (1, 7)	0.8571 (1, 7)	0.9523 (15, 4)	0.9523 (15, 4)	0.9523 (1, 4)
Wine	1(4, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	NA	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
Wholesale	1(3, 1)	0.9285 (3, 12)	1(1, 1)	0.0714(5, 13)	1(5, 1)	1(5, 1)	NA	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
Average rank	5.58	10.25	3.83	6.67	5.5	5.42	6.86	3.75	2.58	3.75	3	3.01	2.17	2.75

In a comparison between the proposed method and the base methods in terms of the specificity according to Table 4-4, from the total of the 12 experiments, thanks to the data mapping on the manifold learnt by the kernel PCA, LDA, and Auto-encoder methods, the proposed method has gained the first rank in 7, 6, and 7 experiments, respectively. Yet, under the same circumstances out of the base methods of manifold learning, the methods of

Kernel PCA, Gini index, and Fisher have earned the first rank in 5, 3, and 4 experiments, respectively.

Also, the proposed methods have gained a lower on average rank than the base approaches. Also, we can claim kernel PCA as the best approach in finding the neighborhoods on the manifold in terms of the specificity by using the 7-NN classifier.

Table 4-5. Accuracy comparison between different approaches versus the proposed using 10-fold cross validation and SVM classifier with (d,r) indicating the best latent dimensionality and the rank of the approach, respectively (AE denotes auto-encoder approach).

Dataset	Dimensionality Reduction				Feature Selection		Proposed Method						
	PCA	LLE	Kernel PCA	AE	Fisher	Gini	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	AE
Vehicle	0.4588 (1, 9)	0.4470 (17, 10)	0.2588 (1, 13)	0.3411 (13, 11)	0.4941 (5, 8)	0.3058 (5, 12)	0.8(5, 6)	0.8235 (1, 2)	0.8(5, 6)	0.8117 (5, 4)	0.8470 (1, 1)	0.8117 (9, 4)	0.8235 (13, 2)
Bupa	0.6 (3, 9)	0.5714 (1, 11)	0.5714 (1, 11)	0.5714 (1, 11)	0.6 (3, 9)	0.7714 (1, 1)	0.7142 (1, 2)	0.6857 (5, 5)	0.7142 (1, 2)	0.7142 (1, 2)	0.6857 (1, 5)	0.6571 (5, 8)	0.6857 (3, 5)
Glass	0.5909 (7, 1)	0.5454 (9, 2)	0.5454 (9, 2)	0.3181 (9, 13)	0.5454 (7, 2)	0.4545 (5, 12)	0.5238 (1, 5)	0.5238 (1, 5)	0.5238 (1, 5)	0.5238 (1, 5)	0.5238 (3, 5)	0.5238 (3, 5)	0.5238 (1, 5)
Ionosphere	0.9444 (15, 7)	0.7222 (22, 12)	0.9166 (22, 10)	0.6388 (8, 13)	0.9722 (8, 1)	0.9444 (8, 7)	0.9722 (15, 1)	0.9722 (1, 1)	0.9722 (15, 1)	0.9166 (8, 10)	0.9722 (15, 1)	0.9444 (1, 7)	0.9722 (22, 1)
Iris	1(1, 1)	1(2, 1)	0.9333 (2, 11)	0.6666 (1, 13)	1(1, 1)	1(1, 1)	1(1, 1)	0.9333 (1, 11)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
KDD	0.9779 (1, 9)	0.9839 (19, 8)	0.7855 (1, 13)	0.9338 (37, 12)	0.9378 (19, 11)	0.9679 (1, 10)	0.9939 (1, 3)	0.9939 (10, 3)	0.9939 (1, 3)	0.9919 (1, 7)	0.9959 (1, 1)	0.9939 (10, 3)	0.9959 (1, 1)
Monks	0.8333 (3, 5)	0.5 (1, 13)	0.8333 (5, 5)	0.75 (1, 11)	0.9166 (5, 2)	0.9166 (5, 2)	0.8333 (3, 5)	0.75 (1, 11)	0.8333 (3, 5)	0.9166 (5, 2)	0.8333 (3, 5)	0.8333 (3, 5)	1(3, 1)
New-thyroid	0.7272 (1, 12)	0.8636 (5, 9)	0.6818 (1, 13)	0.8181 (5, 11)	0.9090 (1, 6)	0.9090 (1, 6)	0.9545 (5, 1)	0.8636 (1, 9)	0.9545 (5, 1)	0.9545 (5, 1)	0.9545 (5, 1)	0.9545 (5, 1)	0.9090 (3, 6)
ima	0.6883 (1, 10)	0.6623 (7, 11)	0.6493 (1, 12)	0.6493 (1, 12)	0.7272 (1, 8)	0.7272 (1, 8)	0.7532 (3, 2)	0.7662 (1, 1)	0.7532 (3, 2)	0.7532 (3, 2)	0.7532 (3, 2)	0.7532 (1, 2)	0.7532 (3, 2)
WDBC	0.6491 (1, 8)	0.8947 (29, 11)	0.6315 (1, 12)	0.6315 (1, 12)	0.8596 (1, 10)	0.8771 (1, 9)	0.9649 (1, 4)	0.9649 (1, 4)	0.9649 (1, 4)	0.9649 (1, 4)	0.9824 (15, 1)	0.9824 (1, 1)	0.9824 (15, 1)
Wine	0.4888(1, 11)	0.8888 (7, 8)	0.3888 (7, 12)	0.3333 (1, 13)	0.8888 (1, 8)	0.5555 (4, 10)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)	1(1, 1)
Wholesale	0.6818 (1, 9)	0.7954 (5, 8)	0.6818 (1, 9)	0.6818 (1, 9)	0.6818 (1, 9)	0.6818 (1, 9)	1(3, 1)	0.9772 (1, 3)	1(3, 1)	0.9772 (1, 3)	0.9772 (1, 3)	0.9772 (1, 3)	0.9772 (1, 3)
Average rank	7.83	8.41	10.25	11.75	6.25	7.25	2.66	4.66	2.66	3.5	2.25	3.41	2.41

Comparing the proposed method with the base approaches in terms of the accuracy using the SVM classifier according to Table 4-5, out of the total

of 12 experiments, the proposed method using the LLE, Auto-encoder and the PCA, has gained the first rank in 6, 6, and 5 experiments respectively. Whereas, under the same circumstances, from the base approaches the Fisher, Gini, and PCA have received the first rank in 2, 2, and 2 experiments, respectively.

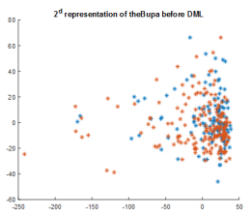
Totally, out of the 12 experiments, the proposed approaches have earned the first rank in 10 experiments. As you can see, the proposed methods have generally a better average ranking in comparison with the base approaches. Having these observations, we can announce LLE as the best approach in finding the neighborhoods on the manifold in terms of accuracy and using SVM as the classifier.

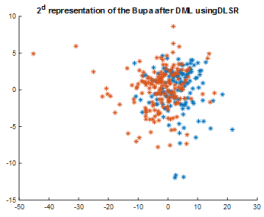
4.5 Representation of the data after reduction

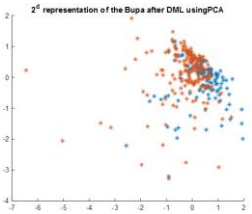
For better visualization of the achievements of the approach after feature mapping and reduction, the data distribution after using the proposed approach is plotted in a 2D space and compared with the original distribution after feature selection and data distribution after DLSR. The plots are shown in Fig. 5.

Data name	Original 2D view	DLSR	PCA + DLSR

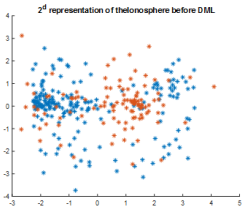
Bupa

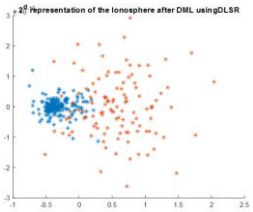


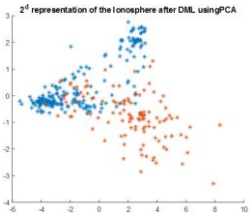




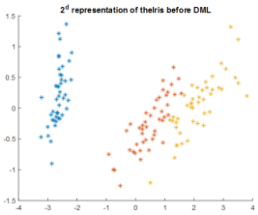
Ionosphere

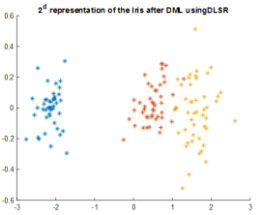


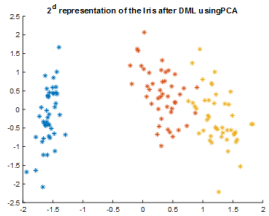




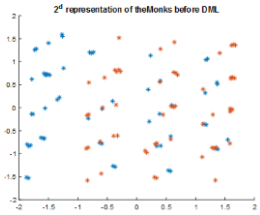
Iris

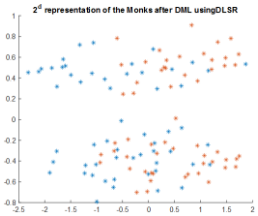


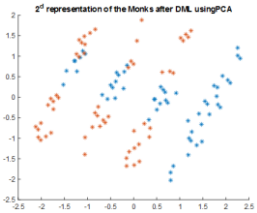




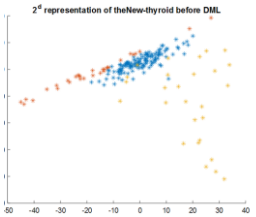
Monks

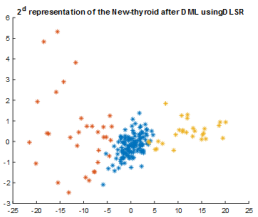


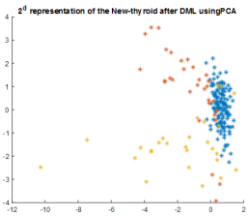




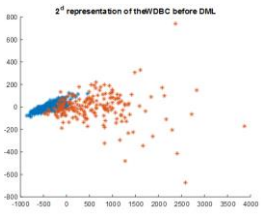
New-thyroid

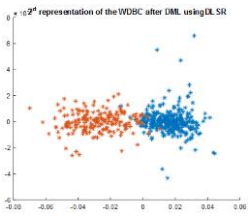


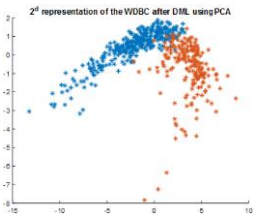




WDBC







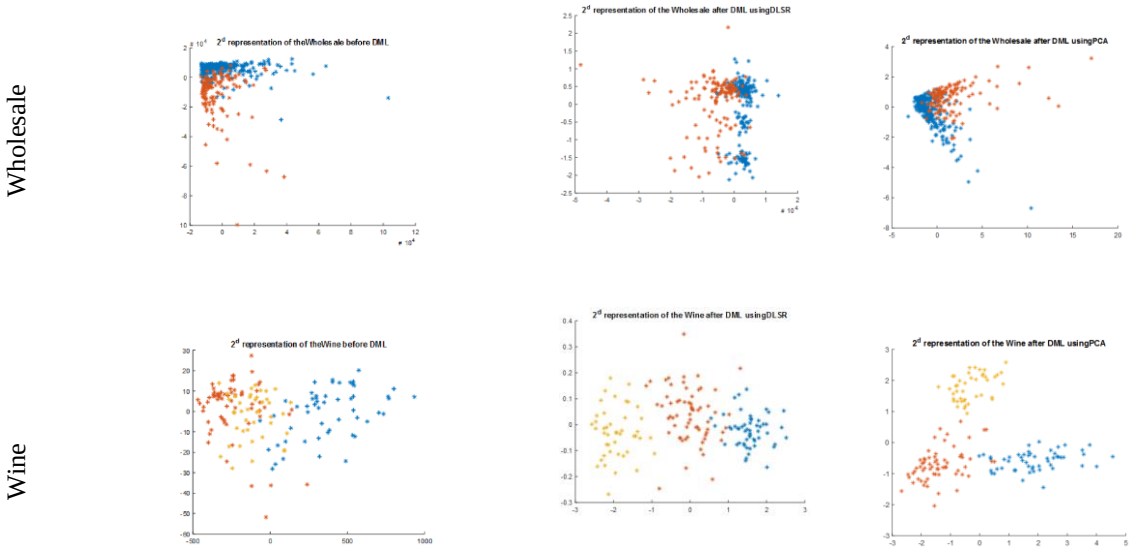


Figure 4-1. Data distribution visualization after the reduction to a new 2D space using different approaches.

In the above illustrations, PCA is applied as the manifold learning approach, so that the discriminative nature of the approach does not affect the final distribution. As seen in the illustrations, the proposed method discriminates the data from different classes far better than the traditional DLSR approach.

4.6 Evaluations on the KDD data

In this chapter we will specifically compare the average results with respect to the confusion matrix of the proposed algorithm and DSLR [25] on KDD dataset which has the highest imbalance ratio among the other datasets studied in this research. The results are shown in Tables 4-6 to 4-13.

Note that for the sake of computational facility, as the number of samples in KDD dataset is too great, we have conducted an under sampling before executing the process, i.e., we have only sampled one hundredth of each in the KDD dataset, except the U2R class which is the minority class.

In the following tables, Table 4-6 denotes DLSR approach results while the others denote the results of the proposed framework using different methods in the manifold learning phase. As seen in Tables 4-7 to 4-13, the integer average values for the number of samples correctly classified to each of the classes testifies the predictability and class-wisely equal performance of the proposed methods which signifies the robustness of the approach independent from the fold on which it is tested. Whereas, under the same conditions DLSR method, shown in Table 4-6, has the non-integer average values for the average number of samples classified to each class in its confusion matrix. This observation on the KDDCup dataset which suffers from high imbalance ratio is the main achievement of the proposed framework on this dataset. However, as can be concluded from these experiments the recall rate of the proposed approach is higher than the DSLR method especially on minority classes (i.e., R2L and U2R).

Table 4-6. The average confusion matrix of the 10-fold cross validation using DLSR approach on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 28).

Class name	DOS	Normal	Probe	R2L	U2R	Accuracy
DOS	391.1	0.6	0	0.2	0.1	0.997704
Normal	0.2	96.5	0	0.3	0	0.994845
Probe	0.1	1.2	2.7	0	0	0.675
R2L	0.1	0.4	0	0.4	0.1	0.4
U2R	0.1	1.1	0	0.4	3.4	0.68

Table 4-7. The average confusion matrix of the 10-fold cross validation using the proposed PCA+DSLR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 10).

Class name	DOS	Normal	Probe	R2L	U2R	Accuracy
DOS	392	0	0	0	0	1
Normal	1	95	0	1	0	0.979381
Probe	0	0	4	0	0	1
R2L	0	0	0	1	0	1
U2R	0	1	0	0	4	0.8

Table 4-8. The average confusion matrix of the 10-fold cross validation using the proposed LDA+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 28).

Class name	DOS	Normal	Probe	R2L	U2R	Accuracy
DOS	392	0	0	0	0	1
Normal	1	95	0	1	0	0.979381
Probe	0	0	4	0	0	1
R2L	0	0	0	1	0	1
U2R	0	1	0	0	4	0.8

Table 4-9. The average confusion matrix of the 10-fold cross validation using the proposed MDS+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1).

Class name	DOS	Normal	Probe	R2L	U2R	Accuracy
DOS	392	0	0	0	0	1
Normal	1	95	0	1	0	0.979381
Probe	0	0	4	0	0	1
R2L	0	0	0	1	0	1
U2R	0	1	0	0	4	0.8

Table 4-10. The average confusion matrix of the 10-fold cross validation using the proposed Isomap+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1).

Class name	DOS	Normal	Probe	R2L	U2R	Accuracy
DOS	392	0	0	0	0	1
Normal	1	95	0	1	0	0.979381
Probe	0	0	4	0	0	1
R2L	0	0	0	1	0	1
U2R	0	1	0	0	4	0.8

Table 4-11 shows the average confusion matrix of the 10-fold cross validation using the proposed LLE+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1).

Table 4-11. The average confusion matrix of the 10-fold cross validation using the proposed LLE+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1).

Class name	DOS	Normal	Probe	R2L	U2R	Accuracy
DOS	392	0	0	0	0	1

Normal	1	95	0	1	0	0.979381
Probe	0	0	4	0	0	1
R2L	0	0	0	1	0	1
U2R	0	1	0	0	4	0.8

Table 4-12. The average confusion matrix of the 10-fold cross validation using the proposed KPCA+DLSR dimension reduction on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1).

Class name	DOS	Normal	Probe	R2L	U2R	Accuracy
DOS	392	0	0	0	0	1
Normal	1	95	0	1	0	0.979381
Probe	0	0	4	0	0	1
R2L	0	0	0	1	0	1
U2R	0	1	0	0	4	0.8

Table 4-13. The average confusion matrix of the 10-fold cross validation using the proposed Autoencoder+DLSR approach on KDD dataset using 7-NN classifier for the best latent dimensionality (i.e. 1).

Class name	DOS	Normal	Probe	R2L	U2R	Accuracy
DOS	392	0	0	0	0	1
Normal	1	95	0	1	0	0.979381
Probe	0	0	4	0	0	1
R2L	0	0	0	1	0	1
U2R	0	1	0	0	4	0.8

Also, Table 4-14 shows the best results of the proposed approach which is achieved using SVM classifier and Autoencoder as the manifold learning approach. This experiment is also performed using 10-fold cross validation. The same results as above are observed in the following table while the classification accuracy of the approach in different classes (even the minority classes) is considerably high.

Table 4-14. The average confusion matrix of the 10-fold cross validation using the proposed Autoencoder+DLSR approach on KDD dataset using the SVM classifier for the best latent dimensionality (i.e. 9).

Class name	DOS	Normal	Probe	R2L	U2R	Accuracy
DOS	32	0	0	0	0	1
Normal	0	97	0	0	0	1
Probe	0	0	4	0	0	1
R2L	0	0	0	1	0	1
U2R	0	1	0	0	4	0.8

To have a better representation of the achievements of the proposed approach on KDD dataset compared with some recent and the state of the art approaches, we also had a class-wise comparison between the best accuracy results of the proposed method and the TVCPSO [69] and CANN [70] approaches on the KDD dataset. The TVCSPO [69] approach is designed to propose a framework for the intrusion detection using an adaptive, robust with precise optimization novel approach called the Time-varying chaos particle swarm optimization which is used for concurrent parameter setting and feature selection for the multiple criteria linear programming (MCLP) and the SVM classification. In this approach a weighted objective function is used which handles a tradeoff between the detection rate maximization and false alarm rate minimization, by considering the number of features. Furthermore, in this approach, in order to make the particle swarm optimization faster in finding the global optimal point and avoid the local optima, the chaos is concept is adopted in the PSO and the time varying inertia weight and the time varying acceleration coefficient is introduced.

CANN [70], proposes an approach called the cluster center and nearest neighbor. In this approach, two distances are measured and aggregated, the first one is based on the distance between each data sample and its cluster center, and the second one is based on the distance between each data point and its nearest neighbor form the same cluster. This

new and one-dimensional representation of data points is used for the intrusion detection by a KNN classifier.

Table 4-15 shows the results of the mentioned approaches in comparison with the proposed approach. As seen in Table 4-15, the proposed method as is specified in Table 14, performs considerably better in terms of accuracy in comparison with the recently proposed methods [69], [70] on the KDD dataset. Other than an improvement on majority classes such as DOS and Normal, the proposed approach is highly efficient in identifying the minority classes such as R2L and U2R which is the main drawback of previous approaches on these datasets.

Table 4-15. A comparison between the accuracy of the proposed method and some other recent works on different classes of the KDD based on the 10-fold cross validation.

Class name	Proposed Method using AE and SVM	CANN [70]	TVCPSO-MCLP [69]	TVCPSO-SVM [69]
DOS	1	0.9968	۰,۹۸۶۴	۰,۹۸۸۴
Normal	1	0.9704	۰,۹۷۵۹	۰,۹۹۱۳
Probe	1	0.8761	۰,۸۷۹۰	۰,۸۹۲۹
R2L	1	0.5702	۰,۷۵۰۸	۰,۶۷۸۴
U2R	0.8	0.385	۰,۵۹۶۲	۰,۴۰۳۸
Average	0.96	0.7597	0.83766	0.79096

Chapter 5: Conclusion and Future Work

5.1 Introduction

In this research a novel method for distance metric learning with the aim of preserving the local neighborhoods between similar data points and also covering data imbalance problem has been proposed and the implementation steps and its experimental results in comparison with other distance metric learning and dimensionality reduction algorithms has been evaluated. In the proposed method, it has been tried to first learn the neighborhoods between the data points based on their neighborhood relations on the manifold. For each data point, two neighborhoods with same number of members consisting of the similar and dissimilar data points to the given point are created. Consequently, distance metric learning is performed with the goal of making the similar points nearer to the given data point and to push back the dissimilar data away from it. Finally, thanks to learned transformation matrix, data are mapped to the similarity space and then the classification is preformed using k-NN and SVM classifiers. The evaluations are performed on 12 datasets with different sizes and imbalance ratio specially the KDD, which resulted in significant results based on the three criteria of accuracy, sensitivity and specificity.

5.2 Future work

In future we would like to have a study on different approaches of data sampling specially the graph-based prototype selection approaches which preserve the local structures of the data. Besides, as for the graph prototype selection, we need to calculate the appropriate distance between different graphs in order to select the most expressive ones. Therefore, another area that we could invest on in the future is to study on the effect of using different graph editing distances on the graph-based prototype selection.

An analysis on the selection of the most appropriate manifold learning approaches (as different manifold learning approaches result in differently manifolds) could have an

extensive impact on the improvement of the learned distance metric. To do this, we would like to have a study on the effect of using the deep neural networks e.g., convolutional neural networks and generative adversarial networks as the learning approaches and analyze their results in comparison with the existing manifold learning approaches.

References

[1] A. Bellet, A. Habrard, and M. Sebban, "A Survey on Metric Learning for Feature Vectors and Structured Data," *arXiv Prepr. arXiv1306.6709*, p. 57, 2013, doi: 10.1073/pnas.0809777106.

[2] C. Domeniconi, J. Peng, and D. Gunopulos, "Locally adaptive metric nearest-neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1281–1285, 2002, doi: 10.1109/TPAMI.2002.1033219.

[3] L. Yang, R. Jin, R. Sukthankar, and Y. Liu, "An efficient algorithm for local distance metric learning," in *Proceedings of the National Conference on Artificial Intelligence*, 2006, vol. 1, pp. 543–548.

[4] C. Domeniconi and D. Gunopulos, "Adaptive nearest neighbor classification using support vector machines," 2002.

[5] Z. Zhang, J. Kwok, and D. Yeung, "Parametric distance metric learning with label information," *Proc. IJCAI*.

[6] P. Schneider, K. Bunte, H. Stiekema, B. Hammer, T. Villmann, and M. Biehl, "Regularization in matrix relevance learning," *IEEE Trans. Neural Networks*, vol. 21, no. 5, pp. 831–840, 2010, doi: 10.1109/TNN.2010.2042729.

[7] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, 2007, doi: 10.1109/TPAMI.2007.1096.

[8] K. Q. Weinberger and L. K. Saul, "Fast solvers and efficient implementations for distance metric learning," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 1160–1167.

[9] Y. Mu, W. Ding, D. Tao, and T. F. Stepinski, "Biologically inspired model for crater detection," in *Proceedings of the International Joint Conference on Neural Networks*, 2011, pp. 2487–2494, doi: 10.1109/IJCNN.2011.6033542.

[10] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1299–1313, 2009, doi:

10.1109/TKDE.2008.212.

[11] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, “Distance metric learning, with application to clustering with side-information,” 2002.

[12] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, “Neighbourhood components analysis,” 2005.

[13] M. Sugiyama, “Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis,” *J. Mach. Learn. Res.*, vol. 8, pp. 1027–1061, 2007.

[14] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009.

[15] T. Hastie and R. Tibshirani, “Discriminant adaptive nearest neighbor classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, 1996, doi: 10.1109/34.506411.

[16] T. M. Cover and P. E. Hart, “Nearest Neighbor Pattern Classification,” *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967, doi: 10.1109/TIT.1967.1053964.

[17] J. Winn, A. Criminisi, and T. Minka, “Object categorization by learned universal visual dictionary,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, vol. II, pp. 1800–1807, doi: 10.1109/ICCV.2005.171.

[18] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, 2000, doi: 10.1109/34.895972.

[19] Z. Lai, W. K. Wong, Y. Xu, J. Yang, and D. Zhang, “Approximate Orthogonal Sparse Embedding for Dimensionality Reduction,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 27, no. 4, pp. 723–735, Apr. 2016, doi: 10.1109/TNNLS.2015.2422994.

[20] M. Turk and A. Pentland, “Eigenfaces for Recognition,” *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, Jan. 1991, doi: 10.1162/jocn.1991.3.1.71.

[21] M. Kirby and L. Sirovich, “Application of the Karhunen-Loeve procedure for the characterization of human faces,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 103–108, 1990, doi: 10.1109/34.41390.

[22] L. Sirovich and M. Kirby, “Low-dimensional procedure for the characterization of

human faces.," *J. Opt. Soc. Am. A.*, vol. 4, no. 3, pp. 519–524, 1987, doi: 10.1364/JOSAA.4.000519.

[23] Q. Liu, H. Lu, and S. Ma, "Improving Kernel Fisher Discriminant Analysis for Face Recognition," *IEEE Trans. Circuits Syst.*, vol. 14, no. 1, pp. 42–49, 2004, doi: 10.1109/TCSVT.2003.818352.

[24] C. Fraley and a E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *J. Am. Stat. Assoc.*, vol. 97, no. 458, pp. 611–631, 2002, doi: 10.1198/016214502760047131.

[25] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces Vs. Fisherfaces: Recognition Using Class Specific Linear Projection," vol. 19, no. 7, pp. 711–720, 1997, doi: 10.1007/BFb0015522.

[26] Jun Li and Dacheng Tao, "Simple Exponential Family PCA," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 3, pp. 485–497, Mar. 2013, doi: 10.1109/TNNLS.2012.2234134.

[27] D. Tao, X. Li, S. Member, X. Wu, and S. Member, "Geometric Mean for Subspace Selection," *TIANJIN Univ. DOWNLOADED DECEMBER 8, 2009 0433 FROM IEEE XPLORE. Restrict. APPLY. YUAN AL. Bin. SPARSE NONNEGATIVE MATRIX FACTORIZATION 777*, vol. 31, no. 2, pp. 260–274, 2009, doi: 10.1109/TPAMI.2008.70.

[28] J. Li and D. Tao, "On preserving original variables in Bayesian PCA with application to image analysis," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4830–4843, 2012, doi: 10.1109/TIP.2012.2211372.

[29] S. T. Roweis, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science (80-.).*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000, doi: 10.1126/science.290.5500.2323.

[30] J. B. Tenenbaum, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science (80-.).*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000, doi: 10.1126/science.290.5500.2319.

[31] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003, doi:

10.1162/089976603321780317.

[32] Xiaofei He, Deng Cai, Shuicheng Yan, and Hong-Jiang Zhang, "Neighborhood preserving embedding," *Tenth IEEE Int. Conf. Comput. Vis. Vol. 1*, vol. 2, pp. 1208-1213 Vol. 2, 2005, doi: 10.1109/ICCV.2005.167.

[33] E. Kokiopoulou and Y. Saad, "Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2143–2156, 2007, doi: 10.1109/TPAMI.2007.1131.

[34] X. He and P. Niyogi, "Locality preserving projections," *Neural Inf. Process. Syst.*, vol. 16, p. 153, 2004, doi: 10.1.1.19.9400.

[35] "Yin Zhou's Home Page @ University of Delaware." https://www.eecis.udel.edu/~zhou/Research.html (accessed Oct. 08, 2017).

[36] Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition*. Elsevier, 1990.

[37] R. Short and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 622–627, 1981.

[38] P. C. Mahalanobis, "On the generalized distance in statistics," *Proc. Natl. Inst. Sci.*, vol. 2, pp. 49–55, 1936.

[39] X. He and P. Niyogi, "Locality preserving projections," 2004.

[40] I. T. Jolliffe, "Principal component analysis and factor analysis," *Princ. Compon. Anal.*, pp. 150–166, 2002.

[41] F. Wang, B. Zhao, and C. Zhang, "Unsupervised large margin discriminative projection," *IEEE Trans. Neural Networks*, vol. 22, no. 9, pp. 1446–1456, 2011, doi: 10.1109/TNN.2011.2161772.

[42] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Adv. Neural Inf. Process. Syst.*, vol. 14, pp. 585–591, 2002.

[43] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science (80-.).*, vol. 290, no. 5500, 2000, doi: 10.1126/science.290.5500.2319.

[44] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding. In: Advances in neural information processing systems (NIPS)," *pp 833–840*, 2002.

[45] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[46] J. Handl, E. Hart, P. R. L. M. López-ibáñez, G. Ochoa, B. Paechter, and D. Hutchison, *Parallel Problem Solving from Nature – PPSN XIV*. 2016.

[47] L. Meng, S. Ding, and Y. Xue, "Research on denoising sparse autoencoder," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 5, pp. 1719–1729, 2017, doi: 10.1007/s13042-016-0550-y.

[48] M. A. Cox, Trevor F; Cox, *Multidimensional scaling*. CRC press, 2000.

[49] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," *Neighb. Components Anal.*, pp. 513–520, 2004.

[50] F. Wang and C. Zhang, "Feature extraction by maximizing the average neighborhood margin," 2007, doi: 10.1109/CVPR.2007.383124.

[51] K. Q. Weinberger and J. Blitzer, "Distance metric learning for large margin nearest neighbor classification," *Adv. Neural Inf. Process. Syst.*, 2005.

[52] K. Fukunaga, "Statistical Pattern Stas-tical Pattern Recognition," *Pattern Recognit.*, vol. 22, no. 7, pp. 833–834, 1990, doi: 10.1016/0098-3004(96)00017-9.

[53] J. V Davis, B. Kulis, P. Jain, S. Suvrit, and I. S. Dhillon, "Information-theoretic metric learning. In: International conference on machine learning (ICML)," *pp 209–216*, 2007.

[54] A. Kocsor, K. Kovács, and C. Szepesvári, "Margin maximizing discriminant analysis," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2004, vol. 3201, pp. 227–238.

[55] N. Shental, T. Hertz, D. Weinshall, and M. Pavel, "Adjustment learning and relevant component analysis. In: Proceedings of European conference on computer vision," *pp 776–790*, 2002.

[56] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K. R. Müllers, "Fisher discriminant analysis with kernels. In: Neural networks for signal processing IX, 1999. proceedings of the 1999 IEEE signal processing society workshop," *pp 41–48*, 1999.

[57] I. W. Tsang, P. M. Cheung, and J. T. Kwok, "Kernel relevant component analysis for distance metric learning. In: In IEEE International joint conference on neural networks

(IJCNN)," *pp 954–959*, 2005.

[58] Y.-F. Guo, S.-J. Li, J.-Y. Yang, T.-T. Shu, and L.-D. Wu, "A generalized Foley-Sammon transform based on generalized fisher discriminant criterion and its application to face recognition," *Pattern Recognit. Lett.*, vol. 24, no. 1–3, pp. 147–158, 2003, doi: 10.1016/S0167-8655(02)00207-6.

[59] Y. Jia, F. Nie, and C. Zhang, "Trace ratio problem revisited," *IEEE Trans. Neural Networks*, vol. 20, no. 4, pp. 729–735, 2009, doi: 10.1109/TNN.2009.2015760.

[60] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 23, no. 11, pp. 1738–1754, 2012, doi: 10.1109/TNNLS.2012.2212721.

[61] S. C. H. Hoi, W. Liu, and S.-F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval," 2008, doi: 10.1109/CVPR.2008.4587351.

[62] F. Wang, S. Chen, C. Zhang, and T. Li, "Semi-supervised metric learning by maximizing constraint margin. In: Proceedings of the 17th ACM conference on information and knowledge management," *pp 1457–1458*, 2008.

[63] X. Yang, H. Fu, H. Zha, and J. Barlow, "Semi-supervised nonlinear dimensionality reduction. In: 23rd International conference on machine learning," *pp 1065–1072*, 2006.

[64] M. Bilenko and S. Basu, "Mooney RJ (2004) Integrating constraints and metric learning in semi-supervised clustering," *Proc. twenty-first Int. Conf. Mach. Learn.*, pp. 11–18, 2004.

[65] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering," 2004.

[66] F. Wang and J. Sun, *Survey on distance metric learning and dimensionality reduction in data mining*, vol. 29, no. 2. 2014.

[67] I. Davidson, K. L. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4213 LNAI, pp. 115–126, 2006.

[68] A. Singh, R. D. Nowak, and X. Zhu, “Unlabeled data: now it helps, now it doesn’t. In: Advances in neural information processing systems,” *pp 1513–1520,* 2008.

[69] S. M. Hosseini Bamakan, H. Wang, T. Yingjie, and Y. Shi, “An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization,” *Neurocomputing*, vol. 199, pp. 90–102, 2016, doi: 10.1016/j.neucom.2016.03.031.

[70] W. C. Lin, S. W. Ke, and C. F. Tsai, “CANN: An intrusion detection system based on combining cluster centers and nearest neighbors,” *Knowledge-Based Syst.*, vol. 78, no. 1, pp. 13–21, 2015, doi: 10.1016/j.knosys.2015.01.009.

Appendix

Results of the SVM (the remainder)

Sensitivity

Table 0-1. Comparison of the results for the sensitivity criterion using the SVM classifier.

Dataset	Dimensionality Reduction				Feature Selection		Proposed Method						
	PCA	LLE	Kernel PCA	Auto-endoder	Fisher	Gini	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	0.5(d=1)	0.8 (d=5)	0(d=1)	0.65 (d=1)	0.65 (d=1)	0.3 (d=1)	0.9 (d=5)	0.95 (d=13)	0.9 (d=5)	0.9 (d=5)	0.95 (d=5)	0.9 (d=1)	0.95 (d=9)
Bupa	0.066667 (d=3)	0.066667 (d=1)	0(d=1)	0(d=1)	0.166667 (d=1)	0.733333 (d=1)	0.466667 (d=1)	0.4 (d=1)	0.466667 (d=1)	0.466667 (d=1)	0.4 (d=1)	0.333333 (d=5)	0.4 (d=3)
Glass	0.857143 (d=1)	1 (d=1)	0.857143 (d=9)	1(d=9)	0.857143 (d=7)	0.857143 (d=7)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=3)	1(d=1)	1(d=1)
Ionosphere	0.913043 (d=15)	1(d=1)	0.869565 (d=15)	1(d=9)	1(d=1)	1(d=1)	0.956522 (d=1)	1(d=1)	0.956522 (d=1)	0.913043 (d=1)	1(d=15)	0.956522 (d=1)	0.956522 (d=1)
Iris	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	0.8 (d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
KDD	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=10)	1(d=19)	1(d=1)	0.997449 (d=1)	1(d=1)	0.997449 (d=1)	1(d=1)	0.997449 (d=1)	0.997449 (d=1)
Monks	0.666667 (d=1)	1(d=1)	0.833333 (d=3)	0.666667 (d=1)	0.833333 (d=5)	0.833333 (d=5)	1(d=1)	0.666667 (d=3)	1(d=5)	1(d=5)	1(d=3)	1(d=5)	1(d=3)
New-thyroid	1(d=3)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Pima	0.296296 (d=1)	0.037037 (d=7)	0(d=7)	0(d=7)	0.296296 (d=1)	0.296296 (d=1)	0.407407 (d=3)	0.444444 (d=1)	0.407407 (d=3)	0.407407 (d=3)	0.407407 (d=3)	0.407407 (d=3)	0.407407 (d=3)
WDBC	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=8)	1(d=8)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Wine	0.333333 (d=1)	1(d=4)	0.388889 (d=7)	1(d=7)	0.666667 (d=1)	0.333333 (d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Wholesale	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=3)	0.966667 (d=1)	1(d=3)	1(d=3)	1(d=3)	0.966667 (d=1)	0.966667 (d=1)

Table 0-2. Ranking of the results for the sensitivity criterion based on the average of the 10-fold cross-validation.

Dataset	Dimensionality Reduction				Feature Selection		Proposed Method k-NN						
	PCA	LLE	Kernel PCA	Auto-endoder	Fisher	Gini	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	11	8	13	9	9	12	4	1	4	4	1	4	1
Bupa	10	10	12	12	9	1	2	5	2	2	5	8	5
Glass	10	1	10	1	10	10	1	1	1	1	1	1	1
Ionosphere	11	1	13	1	1	1	7	1	7	11	1	7	7
Iris	1	1	1	1	1	1	1	13	1	1	1	1	1
KDD	1	1	1	1	1	1	1	10	1	10	1	10	10
Monks	11	1	8	11	8	8	1	11	1	1	1	1	1
New-thyroid	1	1	1	1	1	1	1	1	1	1	1	1	1
Pima	8	11	12	12	8	8	2	1	2	2	2	2	2
WDBC	1	1	1	1	1	1	1	1	1	1	1	1	1
Wine	12	1	11	1	10	12	1	1	1	1	1	1	1
Wholesale	1	1	1	1	1	1	1	11	1	1	1	11	11
Average Rank	6.5	3.166666667	7	4.333333333	5	4.75	1.916666667	4.75	1.916666667	3	1.416666667	4	3.5

Specificity

Table 0-3. Comparison of the results for the specificity criterion using the SVM classifier.

Dataset	Dimensionality Reduction				Feature Selection		Proposed Method						
	PCA	LLE	Kernel PCA	Auto-endoder	Fisher	Gini	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	1(d=5)	0.784615 (d=1)	0(d=1)	0.65 (d=1)	1(d=9)	1(d=5)	0.984615 (d=1)	1(d=1)	0.984615 (d=1)	0.984615 (d=1)	0.984615 (d=1)	0.984615 (d=1)	0.984615 (d=1)
Bupa	1(d=3)	0.95 (d=1)	1(d=1)	1(d=1)	1(d=5)	1(d=3)	0.9 (d=1)	0.9 (d=5)	0.9 (d=1)	0.9 (d=1)	0.9 (d=1)	0.9 (d=1)	0.9 (d=1)
Glass	0.666667 (d=7)	0.933333 (d=1)	0.666667 (d=1)	1(d=3)	0.733333 (d=5)	0.733333 (d=5)	0.666667 (d=5)	0.666667 (d=5)	0.666667 (d=5)	0.666667 (d=1)	0.666667 (d=7)	0.666667 (d=3)	0.666667 (d=1)
Ionosphere	1(d=15)	0.153846 (d=8)	1(d=1)	0(d=1)	1(d=15)	1(d=22)	1(d=8)	0.923077 (d=1)	1(d=8)	0.923077 (d=8)	0.923077 (d=1)	0.923077 (d=1)	1(d=8)
Iris	1(d=1)	1(d=2)	1(d=1)	1(d=2)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
KDD	0.96729 (d=1)	0.943925 (d=19)	0.009346 (d=10)	0.925234 (d=19)	1(d=19)	1(d=10)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Monks	1(d=1)	0(d=1)	0.833333 (d=1)	0.833333 (d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	0.833333 (d=1)	1(d=1)	1(d=3)
New-thyroid	0.571429 (d=1)	0.571429 (d=5)	0(d=1)	0.428571 (d=5)	0.857143 (d=3)	0.857143 (d=3)	0.857143 (d=5)	0.714286 (d=5)	0.857143 (d=5)	0.857143 (d=5)	0.857143 (d=1)	0.857143 (d=1)	0.857143 (d=3)
Pima	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=3)	1(d=3)	0.94 (d=1)	0.96 (d=7)	0.94 (d=1)	0.94 (d=1)	0.94 (d=1)	0.94 (d=1)	0.94 (d=1)
WDBC	0.047619 (d=1)	0.714286 (d=29)	0(d=1)	0(d=1)	0.714286 (d=1)	0.761905 (d=1)	0.904762 (d=1)	0.904762 (d=1)	0.904762 (d=1)	0.904762 (d=1)	1(d=15)	0.952381 (d=1)	0.952381 (d=1)
Wine	1(d=4)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Wholesale	0(d=1)	0.357143 (d=5)	0(d=1)	0(d=1)	0 (d=1)	0 (d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)

Table 0-4. Ranking of the results for the specificity criterion based on the average of the 10-fold cross-validation.

Dataset	Dimensionality Reduction				Feature Selection		Proposed Method k-NN						
	PCA	LLE	Kernel PCA	Auto-endoder	Fisher	Gini	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	1	11	13	12	1	1	5	1	5	5	5	5	5
Bupa	1	6	1	1	1	1	7	7	7	7	7	7	7
Glass	5	2	5	1	3	3	5	5	5	5	5	5	5
Ionosphere	1	12	1	13	1	1	1	8	1	8	8	8	1
Iris	1	1	1	1	1	1	1	1	1	1	1	1	1
KDD	10	11	13	12	1	1	1	1	1	1	1	1	1
Monks	1	13	10	10	1	1	1	1	1	1	10	1	1
New-thyroid	10	10	13	12	1	1	1	9	1	1	1	1	1
Pima	1	1	1	1	1	1	8	7	8	8	8	8	8
WDBC	11	9	12	12	9	8	4	4	4	4	1	2	2
Wine	1	1	1	1	1	1	1	1	1	1	1	1	1
Wholesale	9	8	9	9	9	9	1	1	1	1	1	1	1
Average Rank	4.333333	7.083333	6.666667	7.083333	2.5	2.416667	3	3.833333	3	3.583333	4.083333	3.416667	2.833333

results of the k-NN+S

Accuracy

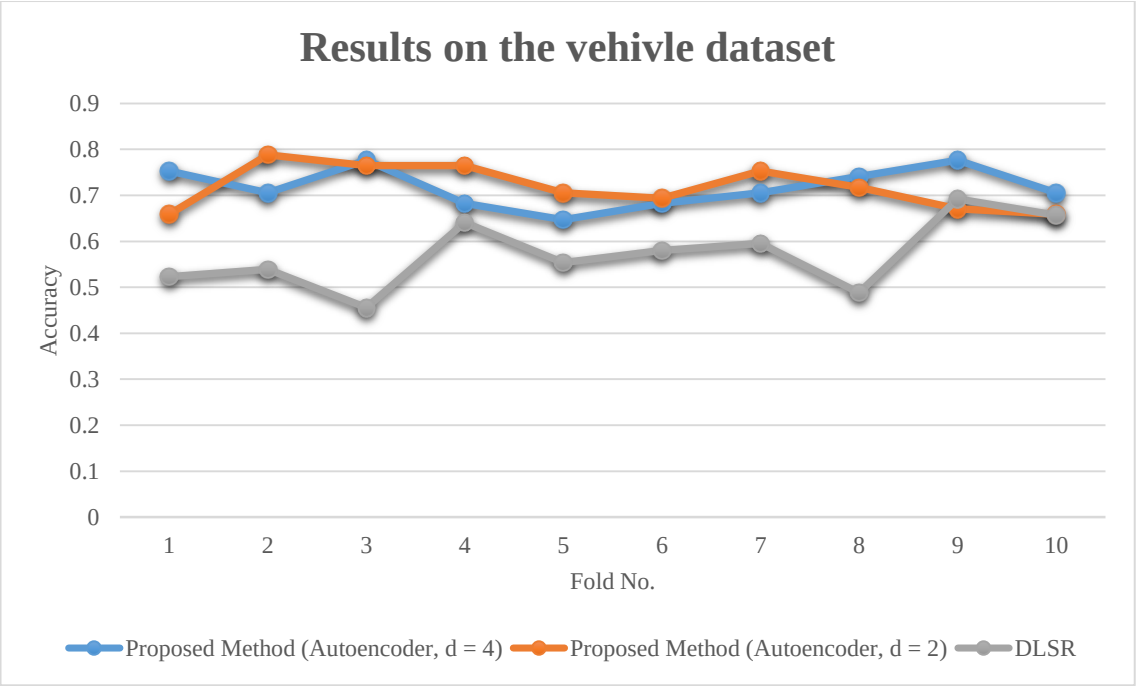


Figure 0-1. Comparison of the averagre accuracy of the proposed method on 10-fold settings based on the neighborhoods on the manifold learnt by autoenocders and DLSR on the vehicle dataset using the k-NN+S classifier.

Table 0-5. Comparison of the results for the accuracy criterion using the 7-NN classifier.

Dataset	Dimensionality Reduction				Proposed Method k-NN+S						
	PCA	LLE	Kernel PCA	Auto-endoder	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	0.682353 (d=13)	0.611765 (d=17)	0.258824 (d=9)	0.529412 (d=5)	0.8 (d=1)	0.776471 (d=1)	0.8 (d=1)	0.811765 (d=5)	0.811765 (d=9)	0.764706 (d=13)	0.788235 (d=17)
Bupa	0.571429 (d=1)	0.657143 (d=3)	0.428571 (d=5)	0.571429 (d=3)	0.685714 (d=1)	0.742857 (d=3)	0.685714 (d=1)	0.685714 (d=1)	0.685714 (d=1)	0.714286 (d=1)	0.714286 (d=5)
Glass	0.545455 (d=1)	0.545455 (d=5)	0.5 (d=7)	0.454545 (d=1)	0.681818 (d=3)	0.636364 (d=1)	0.681818 (d=3)	0.681818 (d=3)	0.681818 (d=7)	0.727273 (d=1)	0.727273 (d=1)
Ionosphere	0.888889 (d=8)	0.805556 (d=22)	0.916667 (d=8)	0.888889 (d=15)	0.861111 (d=8)	0.916667 (d=1)	0.888889 (d=22)	0.861111 (d=1)	0.916667 (d=8)	0.916667 (d=1)	0.888889 (d=1)
Iris	1 (d=1)	1 (d=2)	1 (d=2)	1 (d=1)	0.8 (d=2)	0.666667 (d=1)	0.8 (d=2)	0.8 (d=3)	0.866667 (d=3)	0.8 (d=1)	0.733333 (d=2)
KDD	0.987976 (d=1)	0.983968 (d=28)	0.791583 (d=10)	0.981964 (d=37)	0.785571 (d=1)	0.785571 (d=1)	0.785571 (d=1)	0.785571 (d=1)	0.785571 (d=1)	0.785571 (d=1)	0.785571 (d=1)
Monks	0.833333 (d=5)	0.916667 (d=5)	0.833333 (d=3)	0.5 (d=1)	0.916667 (d=5)	0.75 (d=75)	0.916667 (d=5)	0.666667 (d=1)	0.833333 (d=3)	0.916667 (d=3)	0.916667 (d=3)
New-thyroid	0.954545 (d=5)	0.909091 (d=3)	0.590909 (d=1)	0.909091 (d=3)	0.909091 (d=1)	0.909091 (d=1)	0.909091 (d=1)	0.909091 (d=1)	0.909091 (d=1)	0.909091 (d=1)	0.909091 (d=1)
Pima	0.753247 (d=3)	0.714286 (d=7)	0.649351 (d=3)	0.688312 (d=7)	0.753247 (d=1)	0.766234 (d=1)	0.753247 (d=1)	0.74026 (d=1)	0.753247 (d=1)	0.753247 (d=1)	0.753247 (d=3)
WDBC	0.929825 (d=7)	0.912281 (d=8)	0.631579 (d=22)	0.859649 (d=8)	0.982456 (d=1)	0.982456 (d=1)	0.982456 (d=1)	0.982456 (d=1)	0.982456 (d=1)	0.929825 (d=8)	0.929825 (d=1)
Wine	0.777778 (d=4)	0.944444 (d=13)	0.388889 (d=1)	0.722222 (d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Wholesale	1(d=3)	0.954545 (d=3)	0.681818 (d=3)	0.704545 (d=5)	0.863636 (d=1)	0.863636 (d=1)	0.886364 (d=1)	0.863636 (d=1)	0.909091 (d=1)	0.886364 (d=1)	0.931818 (d=1)

Table 0-6. Ranking of the results for the accuracy criterion based on the average of the 10-fold cross-validation using the 7-NN classifier.

Dataset	Dimensionality Reduction				Proposed Method k-NN+S						
	PCA	LLE	Kernel PCA	Auto-endoder	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	8	9	11	10	3	6	3	1	1	7	5
Bupa	9	8	11	9	4	1	4	4	4	2	2
Glass	8	8	10	11	3	7	3	3	3	1	1
Ionosphere	5	11	1	5	9	1	5	9	1	1	5
Iris	1	1	1	1	6	11	6	6	5	6	10
KDD	1	2	4	3	5	5	5	5	5	5	5
Monks	6	1	6	11	1	9	1	10	6	1	1
Newthyroid	1	2	11	2	2	2	2	2	2	2	2
Pima	2	9	11	10	2	1	2	8	2	2	2
WDBC	6	9	11	10	1	1	1	1	1	6	6
Wine	9	8	11	10	1	1	1	1	1	1	1
Wholesale	1	2	11	10	7	7	5	7	4	5	3
Average Rank	4.75	5.833333	8.25	7.666667	3.666667	4.333333	3.166667	4.75	2.916667	3.25	3.583333

Sensitivity

Table 0-7. Comparison of the results for the sensitivity criterion using the 7-NN classifier.

Dataset	Dimensionality Reduction				Proposed Method k-NN+S						
	PCA	LLE	Kernel PCA	Auto-endoder	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	0.95 (d=13)	0.75 (d=4)	1(d=1)	0.95 (d=9)	0.9 (d=1)	0.95 (d=5)	0.9 (d=1)	0.9 (d=1)	1(d=1)	1(d=5)	1(d=9)
Bupa	0.466667 (d=3)	0.666667 (d=3)	1 (d=1)	0.533333 (d=5)	0.4 (d=1)	0.4 (d=3)	0.4 (d=1)	0.4 (d=1)	0.4 (d=1)	0.466667 (d=1)	0.466667 (d=5)
Glass	0.857143 (d=1)	0.857143 (d=1)	0.5 (d=7)	0.714286 (d=1)	0.571429 (d=3)	0.428571 (d=1)	0.571429 (d=3)	0.571429 (d=3)	0.571429 (d=5)	1(d=3)	0.857143 (d=1)
Ionosphere	0.956522 (d=8)	0.956522 (d=22)	0.913043 (d=1)	0.956522 (d=15)	0.913043 (d=1)	0.913043 (d=1)	0.913043 (d=1)	0.913043 (d=1)	0.956522 (d=1)	0.956522 (d=1)	0.913043 (d=1)
Iris	1(d=1)	1(d=1)	1 (d=1)	1(d=1)	1(d=2)	0.8(d=1)	1(d=2)	1(d=3)	1(d=2)	1(d=1)	1(d=2)
KDD	1 (d=10)	1 (d=10)	1 (d=10)	1 (d=19)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Monks	1(d=5)	1(d=1)	0.833333 (d=3)	1(d=3)	1(d=5)	0.833333 (d=3)	1(d=5)	1(d=5)	0.833333 (d=1)	1(d=3)	1(d=3)
New-thyroid	1(d=5)	1(d=1)	0.866667 (d=1)	1(d=5)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Pima	0.444444 (d=3)	0.444444 (d=3)	1(d=1)	0.925926 (d=3)	0.407407 (d=1)	0.407407 (d=1)	0.407407 (d=1)	0.37037 (d=1)	0.407407 (d=1)	0.407407 (d=1)	0.407407 (d=3)
WDBC	1(d=8)	1(d=8)	1(d=22)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Wine	0.85 (d=1)	0.833333 (d=4)	1(d=4)	1(d=4)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Wholesale	1(d=3)	0.966667 (d=3)	1(d=3)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=5)

Table 0-8. Ranking of the results for the sensitivity criterion based on the average of the 10-fold cross-validation using the 7-NN classifier.

Dataset	Dimensionality Reduction				Proposed Method k-NN+S						
	PCA	LLE	Kernel PCA	Auto-endoder	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	5	11	1	5	8	5	8	8	1	1	1
Bupa	4	2	1	3	7	7	7	7	7	4	4
Glass	2	2	10	5	6	11	6	6	6	1	2
Ionosphere	1	1	6	1	6	6	6	6	1	1	6
Iris	1	1	1	1	1	11	1	1	1	1	1
KDD	1	1	1	1	1	1	1	1	1	1	1
Monks	1	1	9	1	1	9	1	1	9	1	1
New-thyroid	1	1	11	1	1	1	1	1	1	1	1
Pima	3	3	1	2	5	5	5	11	5	5	5
WDBC	1	1	1	1	1	1	1	1	1	1	1
Wine	10	11	1	1	1	1	1	1	1	1	1
Wholesale	1	11	1	1	1	1	1	1	1	1	1
Average Rank	2.583333	3.833333	3.666667	1.916667	3.25	4.916667	3.25	3.75	2.916667	1.583333	2.083333

Specificity

Table 0-9. Comparison of the results for the specificity criterion using the 7-NN classifier.

Dataset	Dimensionality Reduction				Proposed Method k-NN+S						
	PCA	LLE	Kernel PCA	Auto-endoder	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	0.969231 (d=13)	0.892308 (d=17)	1 (d=5)	0.830769 (d=1)	0.984615 (d=1)	0.984615 (d=1)	0.984615 (d=1)	1(d=5)	0.938462 (d=1)	0.984615 (d=1)	0.984615 (d=17)
Bupa	0.7 (d=1)	0.65 (d=3)	0 (d=1)	0.7 (d=1)	0.9(d=1)	0.9(d=1)	0.9(d=1)	0.9(d=1)	1(d=3)	0.9(d=1)	0.9(d=3)
Glass	0.666667 (d=7)	0.733333 (d=3)	0.8 (d=1)	0.8 (d=5)	0.933333 (d=1)	0.933333 (d=1)	0.933333 (d=1)	0.933333 (d=1)	0.933333 (d=3)	0.933333 (d=1)	0.933333 (d=9)
Ionosphere	0.769231 (d=8)	0.615385 (d=15)	1 (d=8)	0.769231 (d=15)	0.846154 (d=22)	0.923077 (d=1)	0.846154 (d=22)	0.769231 (d=1)	0.846154 (d=8)	0.846154 (d=1)	0.846154 (d=1)
Iris	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	0.9(d=1)
KDD	0.990654 (d=10)	0.943925 (d=10)	0.037383 (d=10)	1 (d=37)	0(d=1)	0(d=1)	0(d=1)	0(d=1)	0(d=1)	0(d=1)	0(d=1)
Monks	0.833333 (d=3)	0(d=1)	0.833333 (d=1)	1(d=3)	0.833333 (d=3)	0.833333 (d=1)	0.833333 (d=3)	0.666667 (d=3)	0.833333 (d=3)	0.833333 (d=3)	0.833333 (d=3)
New-thyroid	0.857143 (d=3)	0.714286 (d=3)	0.866667 (d=1)	0.857143 (d=5)	0.714286 (d=1)	0.714286 (d=1)	0.714286 (d=1)	0.714286 (d=1)	0.714286 (d=1)	0.714286 (d=1)	0.714286 (d=1)
Pima	0.92 (d=5)	0.9 (d=7)	1(d=3)	0.82 (d=7)	0.94 (d=1)	0.96 (d=1)	0.94 (d=1)	0.94 (d=1)	0.94 (d=1)	0.94 (d=1)	0.94 (d=1)
WDBC	1(d=1)	0.761905 (d=1)	1(d=1)	0.761905 (d=8)	0.952381 (d=1)	0.952381 (d=1)	0.952381 (d=1)	0.952381 (d=1)	0.952381 (d=1)	0.809524 (d=8)	0.809524 (d=1)
Wine	1(d=4)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)	1(d=1)
Wholesale	1(d=3)	0.928571 (d=3)	1(d=1)	0.071429 (d=5)	0.642857 (d=3)	0.571429 (d=3)	0.642857 (d=3)	0.571429 (d=1)	0.714286 (d=1)	0.714286 (d=1)	0.857143 (d=1)

Table 0-10. Ranking of the results for the specificity criterion based on the average of the 10-fold cross-validation using the 7-NN classifier.

Dataset	Dimensionality Reduction				Proposed Method k-NN+S						
	PCA	LLE	Kernel PCA	Auto-endoder	PCA	LDA	MDS	Isomap	LLE	Kernel PCA	Auto-encoder
Vehicle	8	10	1	11	3	3	3	1	9	3	3
Bupa	8	10	11	8	2	2	2	2	1	2	2
Glass	11	10	8	8	1	1	1	1	1	1	1
Ionosphere	8	11	1	8	3	2	3	8	3	3	3
Iris	1	1	1	1	1	1	1	1	1	1	11
KDD	2	3	4	1	5	5	5	5	5	5	5
Monks	2	11	2	1	2	2	2	10	2	2	2
New-thyroid	2	4	1	2	4	4	4	4	4	4	4
Pima	9	10	1	11	3	2	3	3	3	3	3
WDBC	1	10	1	10	3	3	3	3	3	8	8
Wine	1	1	1	1	1	1	1	1	1	1	1
Wholesale	1	3	1	11	7	9	7	9	5	5	4
Average Rank	4.5	7	2.75	6.083333	2.916667	2.916667	2.916667	4	3.166667	3.166667	3.916667

Comparison of the average runtime of the k-NN, k-NN+S, and SVM

Table 0-11. Comparison of the average execution time per test data on 10-fold over different datasets between the three classification methods k-NN, k-NN + S, and SVM in the proposed similarity space.

Classifier	KNN							k-NN+S							SVM						
Dataset	PCA	LDA	MDS	Isomap	LLE	KernelPCA	Autoencoder	PCA	LDA	MDS	Isomap	LLE	KernelPCA	Autoencoder	PCA	LDA	MDS	Isomap	LLE	KernelPCA	Autoencoder
Vehicle	0.007142	0.007202	0.007263	0.007288	0.007352	0.007603	0.007559	**3.82E-05**	**3.95E-05**	**4.00E-05**	**3.88E-05**	**4.38E-05**	**4.07E-05**	**3.93E-05**	0.003549	0.003792	0.003724	0.003992	0.003493	0.003273	0.00312
KDD	0.006896	0.006315	0.006676	0.006304	0.006317	0.006573	0.006661	**3.78E-05**	**3.70E-05**	**3.82E-05**	**3.71E-05**	**3.79E-05**	**3.81E-05**	**3.70E-05**	0.001875	0.001711	0.001869	0.001754	0.001998	0.001701	0.001733
Bupa	0.000677	0.00066	0.000659	0.000658	0.000662	0.000661	0.00075	**4.07E-05**	**3.89E-05**	**3.85E-05**	**3.86E-05**	**3.98E-05**	**3.84E-05**	**3.88E-05**	0.001035	0.001061	0.001032	0.001015	0.001062	0.001076	0.001075
Glass	0.000376	0.000392	0.000374	0.000375	0.000374	0.000408	0.000378	**3.92E-05**	**4.13E-05**	**3.97E-05**	**3.98E-05**	**3.89E-05**	**4.23E-05**	**4.00E-05**	0.002931	0.003098	0.002948	0.002922	0.002885	0.002979	0.00288
Ionosphere	0.000711	0.0008	0.000696	0.000701	0.000777	0.000776	0.000817	**3.95E-05**	**4.40E-05**	**3.90E-05**	**3.87E-05**	**4.17E-05**	**4.27E-05**	**4.53E-05**	0.001065	0.000937	0.001026	0.000997	0.000893	0.000999	0.001116
Monks	0.000268	0.00023	0.000255	0.000249	0.000238	0.000245	0.000242	**4.54E-05**	**4.06E-05**	**3.98E-05**	**4.11E-05**	**3.90E-05**	**4.03E-05**	**4.01E-05**	0.00183	0.001669	0.001746	0.001726	0.001683	0.001684	0.001697
New-thyroid	0.000387	0.000381	0.000382	0.00038	0.000376	0.000384	0.000375	**3.91E-05**	**3.98E-05**	**3.94E-05**	**3.99E-05**	**3.96E-05**	**3.99E-05**	**3.95E-05**	0.001494	0.001522	0.001509	0.001526	0.001549	0.001451	0.001488
Pima	0.005826	0.005707	0.005737	0.005712	0.005725	0.005727	0.00572	**3.89E-05**	**3.77E-05**	**3.86E-05**	**3.75E-05**	**3.83E-05**	**3.83E-05**	**3.81E-05**	0.001899	0.001906	0.001848	0.001906	0.001868	0.001958	0.00184
WDBC	0.002872	0.003063	0.002775	0.002812	0.002804	0.002903	0.00304	**3.84E-05**	**3.92E-05**	**3.84E-05**	**3.80E-05**	**3.84E-05**	**3.96E-05**	**4.02E-05**	0.001002	0.000902	0.000955	0.000971	0.00097	0.000725	0.001083
Iris	0.000434	0.000374	0.000349	0.000308	0.000345	0.000313	0.00027	**4.45E-05**	**5.21E-05**	**4.53E-05**	**4.42E-05**	**4.69E-05**	**4.63E-05**	**3.98E-05**	0.002175	0.002122	0.002181	0.0021	0.002135	0.002168	0.001809
Wine	0.000331	0.000333	0.00035	0.000324	0.000333	0.000314	0.000376	**4.01E-05**	**4.25E-05**	**4.29E-05**	**3.87E-05**	**4.13E-05**	**4.01E-05**	**4.61E-05**	0.001768	0.001845	0.0018	0.001745	0.001775	0.001538	0.001749
Wholesale	0.000974	0.000984	0.001051	0.000999	0.000995	0.000993	0.001058	**3.87E-05**	**3.91E-05**	**3.96E-05**	**3.92E-05**	**3.92E-05**	**3.91E-05**	**3.91E-05**	0.000846	0.000832	0.000854	0.000851	0.000848	0.0008	0.00085
Average runtime	0.002241	0.002203	0.002214	0.002176	0.002192	0.002242	0.002271	**4E-05**	**4.1E-05**	**3.99E-05**	**3.93E-05**	**4.04E-05**	**4.05E-05**	4.03E-05	0.001789	0.001783	0.001791	0.001792	0.001763	0.001696	0.001703

As you can see in Table 0-11, the classification using the k-NN + S method by the proposed methods is significantly faster than the k-NN and SVM methods, because the k- method Instead of calculating the distance between each point and each other point, which is especially time consuming, especially on large data sets, NN + S uses the similarity matrix obtained by learning the distance criterion method. Uses a suggestion. Thus, another achievement of the proposed method in this research could be to provide a similarity space, which can make us needless to calculate the distance between points, and in cases where the classification speed is based on a large volume of Data is of great importance and can be very useful and efficient.