Visualization Research Center
University of Stuttgart
Allmandring 19
D–70569 Stuttgart

Masterarbeit

Reproducing, Extending and Updating Dimensionality Reductions

Munmun Debnath

Course of Study:	Information Technology (INFOTECH)
Examiner:	Prof. Dr. Daniel Weiskopf
Supervisor:	M.Sc. David Hägele
Commenced:	May 8, 2021
Completed:	November 8, 2021

Abstract

Dimensionality reduction techniques play a key role in data visualization and analysis, as these techniques project high-dimensional data in low-dimensional space by preserving critical information about the data in low-dimensional space. Dimensionality reduction techniques may suffer from various drawbacks, e.g., many dimensionality reduction techniques are missing a natural out-of-sample extension, i.e., the ability to insert additional data points into an existing projection. Therefore when a data set grows and new data points are introduced, the projection has to be recalculated, which often cannot be well related to the previous projection. This thesis proposes a technique based on kernel PCA to reproduce and update the result of dimensionality reduction techniques to overcome the stated problems with better run-time performance. The proposed technique uses an initial projection provided by an arbitrary dimensionality reduction technique as a template of the embedding space. A corresponding kernel matrix is then approximated to project out-of-sample instances. The approach is evaluated on several datasets for reproduction of projections of different dimensionality reduction techniques. It is shown that the proposed technique provides a coherent projection for out-of-sample data, and has a better run-time performance than several other dimensionality reduction techniques.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

1.1 Motivation

In the current data driven world, the amount of data is ever increasing. It is quite natural because the traditional systems are moving more and more to digital systems to manage their data. Data represents an individual piece of information and it can be described by a set of features e.g., colour, shape, and size are features of fruits, and each feature is considered as a dimension of the data. Normally in tabular representation of the data, dimensions are represented as columns of a table and each data refers to a row of the table. Typically, there is no restriction on the number of dimensions data can have, but human eyes are incapable of perceiving high-dimensional data. There are several traditional techniques to visualize high-dimensional data e.g., parallel coordinates plots, scatter plot matrices, heat-maps. But they work efficiently with small and medium sizes of data. As a result, dimensionality reduction (DR) comes into play to project and visualize the high-dimensional data into low dimensions, usually in 2D to be viewed on a computer screen.

The necessity of dimensionality reduction in the field of visualization, data science and machine learning etc. is ever increasing, and choosing the appropriate dimensionality reduction technique for different purposes and measuring differences among several techniques are very important. Different DR techniques have different working principles. But one thing is common in them, that these techniques aim to preserve essential information of the high-dimensional data in low dimensions according to their working principles or algorithms. Each DR technique has its own advantages and disadvantages, and its performance even depend on the input data. There is no single DR technique which can ensure best low-dimensional result on any given dataset. A key problem of many DR techniques related to performance is: The DR technique calculates a projection for whole data set each time whenever new data arrive, it cannot use previous knowledge of the projection which is time consuming. Some DR techniques (e.g., Multidimensional scaling (MDS) [Kru64; Tor58]) are not suitable for dataset having a lot of instances.

There can be several possibilities to overcome these problems of the DR techniques. A novel approach discussed in this thesis is, to use existing DR techniques and use the learnt knowledge to overcome the problem of missing an out-of-sample extension and performance bottleneck.

1.2 Challenges

There are several dimensionality reduction techniques found in the literature, but each of them has their own limitation. Some of these limitations are described below.

- Many of the dimensionality reduction techniques cannot project new unseen data in low-dimension space, using previous knowledge for low-dimensional projection. It can be very expensive when a dimensionality reduction technique takes a long time to calculate and project data points in low-dimensional space. E.g., in the case of MDS, every time a new data point is introduced, MDS needs to be run on whole dataset for projection.
- Performance of dimensionality reduction techniques depends on the number of dimensions or number of instances. If the size of the data set increases, time required by a dimension reduction technique to project the data points in low-dimensional space also increases rapidly or quadratically or polynomially. Many conventional dimensionalily reduction techniques are not able to project the new unseen data points (large size of data) in low-dimensional space using previous knowledge for projection, learnt from a small sample of dataset (during training).
- Some dimensionality reduction technique, e.g., t-distributed Stochastic Neighbor Embedding (t-SNE), Uniform Manifold Approximation and Projection (UMAP) do not possess any projection function i.e., instead of using any projection function, they use a loss function to map data points in low-dimensional space.

1.3 Objective

As mentioned above existing dimensionality reduction techniques suffer from several critical issues, e.g., time requirement to project new data points, performance bottleneck due to the huge volume of data etc. These problems can have serious impact on low-dimensional projection. In order to alleviate these concerns and achieve desired low-dimensional projection, a reasonable solution is required.

The main objective of this master thesis is to develop such a solution using Kernel-PCA which reproduces the original projection and provides an out-of-sample extension of different DR techniques, provides good run-time performance for projection of large data set, as well as also furnishes a coherent low-dimensional result comparable with the original dimensionality reduction technique. Additionally, the proposed solution is to be generic so that it can provide projections of different dimensionality reduction techniques e.g., t-SNE, UMAP, ISOMAP etc. The Proposed solution should meet following requirements: this solution first reproduces the projection of a dimensionality reduction technique on training data points; and secondly, it calculates the projection of unseen or out-of-sample data points in low-dimensional space based on the training result of the dimensionality reduction technique; and lastly, when new data points are introduced, it updates the low-dimensional projection on unseen data smoothly with new data points.

Another key objective is to ensure that the proposed solution meets the quality requirements by performing a detailed evaluation on the solution with different dimensionality reduction techniques, a large number of datasets and several evaluation metrics, and comparing the result of the proposed solution with the results of the original DR techniques.

1.4 Outline

The contents of this thesis are divided into several chapters. It starts with an introduction of the thesis which contains motivation, challenges and objective of the thesis and outline of this document. The related work or similar work of the thesis is described in Chapter 2. Following that, Chapter 3 demonstrates the prerequisites i.e., dimensionality reductions and clustering. In Chapter 4, detail descriptions of the proposed algorithm are illustrated along with the implementation of the proposed algorithm. Chapter 5 describes the datasets and detail evaluation of the algorithm with help of evaluation metrics, figures and time performance. In the end, a conclusion with summary, discussion and future work are described in Chapter 6.

2 Related Work

This chapter provides an overview of both available techniques and research performed in the area of dimensionality reduction that are related to this thesis. The chapter includes representation of different DR techniques' results with Kernel-PCA and out-of-sample projection of different DR techniques.

2.1 Representation of different DR techniques' results with Kernel-PCA

Kernel-PCA can be used to represent the result of different dimensionality reduction techniques. Ham et al. [HLMS04] showed that different DR techniques such as Isomap, LLE and graph Laplacian eigenmap can be expressed in terms of kernel methods only for training samples. In order to achieve this, Gram matrices of the result of DR techniques need to be calculated, and these Gram matrices will be used as a kernel matrix for KPCA.
This encourages our approach of computing a kernel matrix for any given projection. In our proposed algorithm, a kernel matrix is calculated for out-of-sample data points with the help of knowledge acquired during training and low-dimensional projection of out-of-sample data points using KPCA. Afterwards, KPCA uses this calculated kernel matrix to project out-of-sample data points in low-dimensional projection.

2.2 Out-of-sample projection of different DR techniques

The ability to project new high-dimensional data points which are not part of training is a desirable property of a DR technique. Some DR techniques have the ability to project out-of-sample data points e.g., PCA, KPCA [Wil02]. But we can extend the out-of-sample projection of other DR techniques in various ways, e.g., nearest neighbour approach [LCS05], using Kernel PCA to approximate the eigenvectors of out-of-sample data [BPV+03].

Li et al.[LCS05] showed that out-of-sample projection can be achieved by using nearest neighbour information. In this approach, it is assumed that there exists a low-dimensional projection of high-dimensional data points from where new out-of-sample high-dimensional data points are introduced. Calculation of out-of-sample projection by this approach is briefly discussed below.
The first step is to calculate the nearest neighbour for each of the new out-of-sample data points in high-dimensional space which already has a previously calculated low-dimensional projection. Then, find the low-dimensional points of the nearest neighbour. Afterwards, project the new out-of-sample data in low-dimensional space close to the low-dimensional data point of nearest neighbour of out-of-sample data points.

The proposed algorithm is influenced by the nearest neighbour approach to incorporate the information or characteristics of the low-dimensional projection of original DR techniques (for training data points) to project out-of-sample data points.
In the proposed algorithm, instead of directly placing the out-of-sample data points near to the nearest neighbour, the lower dimensional projection of nearest neighbor determined by DR techniques for training data points are used here for further calculation.

Kernel t-SNE [GSH15] also projects out-of-sample data points for DR techniques using Gaussian kernel. In this approach, a reference matrix is calculated during the training phase. Afterwards, the reference matrix is used to approximate the out-of-sample projection by multiplying this reference matrix with the matrix calculated by the Gaussian kernel for out-of-sample data points. The calculation of the reference matrix is performed for each data point individually using the Gaussian kernel which results in slow performance.
In our approach, corresponding low-dimensional data points of the nearest neighbours are used as feature vectors, and the kernel matrix is computed in combination of low-dimensional projection of KPCA on out-of-sample data points and these feature vectors. Here, in the case of the proposed algorithm, out-of-sample projection is performed by KPCA. Here, original data points are used for calculating the low-dimensional projection of a DR technique and low-dimensional projection of KPCA for all cases, i.e., training and out-of-sample projection. It is not shown in the kernel t-SNE approach, the process of calculating low-dimensional projection for iteratively coming out-of-sample data points without calculating a reference matrix for whole out-of-sample data points. In our approach, new out-of-sample data points are projected without performing the whole process again for new and previously introduced out-of-sample data points.

3 Prerequisites

In this section, some prerequisites of the thesis i.e., dimensionality reduction and clustering are discussed to understand the background and techniques used for the proposed algorithm.

3.1 Dimensionality Reduction

Dimensionality of a dataset is referring to the number of attributes or features the data consists of. Dimensionality reduction is a technique to reduce the dimensionality of a data by preserving the essential information which is required to represent the data meaningfully. Eigen values and Eigenvectors of covariance matrix, K nearest neighbour etc. are most commonly used to compute the essential information of the data [Fuk13; VPV+09]. Some information of data can be lost during dimensionality reduction. Despite of losing information, dimensionality reduction provides several benefits e.g., reducing computational complexity as number of dimensions are reduced, minimizing memory consumption as a smaller volume of information needs to be stored, getting rid of curse of dimensionality [XLX17], making data visualization easy for human eyes as human eyes are capable of perceiving data up to three dimensions.

Dimensionality Reduction Techniques Dimensionality reduction techniques can be classified as linear and non-linear depending on the type of the data set (linear or non-linear) it works effectively. There are several popular non-linear dimensionality reduction techniques e.g., Kernel Principal Component Analysis (KPCA) [SSM98], Isometric Feature Mapping (ISOMAP) [TDL00], Locally Linear Embedding (LLE) [RS00] , t- Stochastic Neighbor Embedding (t-SNE) [VH08], Uniform Manifold Approximation and Projection (UMAP) [MHM20] etc.

Principal Component Analysis (PCA) PCA is a linear dimensionality reduction technique. The main objective of this technique [Hot33; JC16] is to calculate the principal components which determine the uncorrelated features of data point.

Principal components are the eigen vectors of the covariance matrix of high-dimensional data, and, it represents in which direction variance of the data is maximal. Eigen decomposition, Singular value decomposition techniques can be used to determine the principal components, which include finding eigen values and corresponding eigen vector of the covariance matrix of the high-dimensional dataset. Square root of an eigen value represents the length of the data distribution, and the eigen vector represents the direction of the data distribution of high-dimensional data set. In the field of signal processing, it is known as Karhunen-Loéve transform. PCA works efficiently in linear datasets.

The mathematical description of PCA is described below. Lets consider a sample dataset $\{x_k\}_{k=1}^{l}$ and $x_k \in R^N$, where x_k has N number of features in high-dimensional space with mean

$$\mu = \frac{1}{l}\sum_{k=1}^{l} x_k = 0$$

and covariance matrix

$$C = \frac{1}{l}\sum_{j=1}^{l} x_j x_j^T$$

C can be expressed as

$$C = U\Lambda U^T$$

where U is an orthogonal matrix and Λ is a diagonal matrix, and the diagonal matrix Λ contains the eigenvalues of the covariance matrix [SSM97]. Eigen value λ and eigen vector v of the covariance matrix are determined using the mathematical formula $\lambda v = Cv$. While projecting the data in low M-dimensional space (where $M < N$), eigenvectors corresponding to largest M eigenvalues are chosen. Low-dimensional projection (Y) can be expressed as

$$Y_M = XU_M$$

Advantages and Disadvantages: PCA has following advantages and disadvantages.
This dimensionality reduction technique determines the uncorrelated features of high-dimensional input data points. As a result, this DR technique removes the correlated features or redundant information. It is fast to project high-dimensional data points in low-dimensional space as it performs linear transformation.
In order to apply PCA as a DR technique, input data needs to be standardized. Without proper data standardization, wrong principle components can be calculated which may lead to error in low-dimensional projection. PCA works efficiently in linear dataset.

Kernel Principal Component Analysis (KPCA) Kernel PCA is a non-linear dimensionality reduction technique based on the linear dimensionality reduction technique PCA [SSM98].

In order to introduce non-linearity in PCA, a kernel concept is included in it. It is achieved by projecting the data points in high-dimensional feature space with the help of kernel function. The kernel describes the dot product of data points in an implied feature space. Afterwards, principal component analysis is applied on these high-dimensional data points to reduce the dimension of the original data points.
Lets consider a sample dataset $\{x_k\}_{k=1}^{l}$ and $x_k \in R^N$. Φ is a function which maps the data points in high-dimensional space $\Phi(x_1), \ldots, \Phi(x_l) \in R^D$ and $D >> N$. The mean of the transformed data points is calculated as

$$\hat{\mu} = \frac{1}{l}\sum_{k=1}^{l} \Phi(x_k)$$

and the covariance matrix of the newly transformed data points is calculated as

$$\hat{C} = \frac{1}{l}\sum_{j=1}^{l}\Phi(\mathrm{x}_j)\Phi(\mathrm{x}_j)^T$$

A $l \times l$ kernel matrix (K) is defined as $K_{ij} =< \Phi(\mathrm{x}_i), \Phi(\mathrm{x}_j) >$ and here instead of calculating the covariance matrix like PCA, kernel matrix is calculated. In order to perform PCA on the kernel matrix, the mean of the kernel matrix should be set to zero to make the first principle component in the direction of maximal variance and to set the mean to zero for the kernel matrix, double centering on the kernel matrix is performed.
The equation to make the kernel matrix double centred is mentioned below.

$$\tilde{K_{ij}} = K - 1_l K - K 1_l + 1_l K 1_l$$

where $(1_l)_{ij} = \frac{1}{l}$ and 1_l is a $l \times l$ matrix. Eigenvalues and corresponding eigenvectors of kernel matrix are determined using the formula $\lambda v = \hat{C} v$.
The low-dimensional projection is determined with the below equation.

$$(V^k\Phi(\mathrm{x})) = \sum_{i=1}^{l}\alpha_i^k(\Phi(\mathrm{x}_i)^T\Phi(\mathrm{x})) = \sum \alpha_i k(\mathrm{x}, \mathrm{x}_i)$$

where α_j^k is the coefficient of k^{th} eigen vector.
RBF (Radial Basis Function), Polynomial, cosine and sigmoid are some examples of popular kernel function.
RBF or Gaussian kernel function is defined as

$$\mathrm{k}(\mathrm{x}, \mathrm{y}) = exp(-||\mathrm{x} - \mathrm{y}||^2/2\sigma^2)$$

It is a very powerful kernel as this kernel allows mapping to infinite number of dimensions in the feature space. Here, $||\mathrm{x} - \mathrm{y}||$ is the euclidean distance between x, y, and σ is the variance.
Polynomial kernel function is defined as

$$\mathrm{k}(\mathrm{x}, \mathrm{y}) = (\mathrm{x}^T\mathrm{y} + a)^d$$

where d is degree and a is a coefficient.
Cosine kernel function also known as cosine similarity and it is defined as

$$\mathrm{k}(\mathrm{x}, \mathrm{y}) = \cos\theta = \langle\frac{\mathrm{x}}{||\mathrm{x}||}, \frac{\mathrm{y}}{||\mathrm{y}||}\rangle$$

where θ is the angle between x and y

Advantages and Disadvantages: The advantages and disadvantages of KPCA are described below.
KPCA preserves the global structure of high-dimensional dataset. In case of this technique, low-dimensional projection does not depend on the local structure of high-dimensional data points e.g., nearest neighbour of high-dimensional manifold.
Choice of the appropriate kernel function for KPCA is difficult. It is a trial-and-error process. KPCA has a high computational complexity and memory consumption issue as size of the kernel matrix depends on the number of data points of input data i.e., square of the input data points. When N is the number of data points, $N \times N$ is the size of the kernel matrix.

Multidimensional Scaling (MDS): Classical multidimensional scaling [Kru64; Tor58] is a linear dimensionality reduction technique that preserves the similarity between the data points i.e., the closeness between the data points. It has non-linear versions also, depending on the stress functions that are used. MDS operates on pairwise distance matrix of the input dataset. Hence, in first step, it calculates a pairwise distance matrix of the given dataset. Lets the dataset has n number of data points, here a distance matrix D of size $n \times n$ is determined.
Afterwards, the eigenvalue decomposition technique is used i.e., finding the eigen values and the eigen vectors of the matrix $\tau(D)$, where $\tau(D)$ is determined by calculating the square of the distance matrix D and followed by double centering the resultant square matrix. $\tau(D)$ is described as

$$\tau(D) = \frac{-HSH}{2}$$

where S is the squared distances matrix of D, $\mathrm{s}_{ij} = \mathrm{D}_{ij}^2$, and H is the "centering matrix" $\mathrm{H}_{ij} = \delta_{ij} - \frac{1}{N}$, where δ_{ij} is an identity matrix.

In order to minimize the error between high and low-dimensional space, the pairwise distances between $\tau(D)$ and $\tau(D_Y)$ is reduced using the loss function. Equation of the loss function is mentioned below.

$$E = \|\tau(D) - \tau(D_Y)\|_{L^2} = \sqrt{\sum_{i,j} \left(\tau(D) - \tau(D_Y)\right)^2}$$

Where $D = d(i,j) = \|\mathrm{x}_i - \mathrm{x}_j\|$, is the pairwise distance matrix of high-dimensional space and $D_Y = d_Y(i,j) = \|\mathrm{y}_i - \mathrm{y}_j\|$, is the pairwise distance matrix of low-dimensional space.

Advantages and Disadvantages: MDS has several advantages and disadvantages, and they are described below.
It represents the similarity or closeness of the information and maintains the global structure of the input data points.
It has performance bottleneck on large amount of data points as it has a high computational and memory complexity and it is not suitable to preserve the local structure of the input data points.

Isometric Feature Mapping (ISOMAP): ISOMAP [TDL00] is a non-linear dimensionality reduction technique which uses the basic properties of MDS (Multidimensional Scaling). Instead of calculating euclidean distance between the data points like MDS, it focuses on calculating geodesic (geometric) distance between the nearest data points to preserve the original geometric non-linear structure of high-dimensional data points.
ISOMAP algorithm consists of three steps. These steps are described below.

First step is finding the K nearest neighbours or all neighbours within radius r for each of the data points $\mathrm{x}_i \in \mathrm{X}$ and building the neighbourhood weighted graph G based on the pairwise distance $d_x(i,j)$ between the data points in original high-dimensional input space. If the data points x_i is nearest neighbour of x_j then there exists an edge between x_i and x_j, and the length of the edge of the weighted graph G is same as the distance between x_i and x_j.

Second step is computing the shortest paths and corresponding shortest paths matrix or geodesic distance matrix (D_G) for x_i and $\mathrm{x}_j \in X$ where $D_G = d_x(i,j)$. There are several algorithms to

find the shortest path of a graph e.g., Floyd-Warshall algorithm, Dijkstra's algorithm with Fibonacci heap [QC15].

In the third step, classical MDS is used to find an embedding for the dissimilarities by the shortest paths.

Advantages and Disadvantages: ISOMAP has several advantages and disadvantages. They are described below.
Compare to a linear dimensionality reduction technique, ISOMAP is capable of identifying the non-linear original geometric structure of high-dimensional data. It is a non-iterative algorithm as this algorithm does not include repetition of steps.
ISOMAP suffers from the short-circuiting problem when higher number of nearest neighbours are chosen compare to the original nearest neighbours reside on high-dimensional space and if noises are present in the input dataset. This problem leads to the wrong data in the geodesic distance matrix as well as a wrong embedding in low-dimensional space.
It also suffers to project the input data in low-dimensional space when holes are presents in high-dimensional manifold e.g., when shape of the manifold is like cylinder [Ten+98].

t-distributed Stochastic Neighbor Embedding (t-SNE) t-SNE [VH08] is a non-linear and non-parametric dimensionality reduction and visualization technique to visualize high-dimensional data into low-dimensional space i.e., in two or three dimensions. It is a variant of the stochastic neighbor embedding (SNE).

It preserves non-linear property and local structure of high-dimensional data by keeping similar nearest neighbour of high-dimensional data points close together in low-dimensional space. This is achieved by minimizing the probabilistic loss value between high-dimensional data point and corresponding low-dimensional data point. It is also capable of retaining global structure to some extent of high-dimensional data set by preserving the distances between clusters and choosing higher number of the nearest neighbours. It does not have any specific projection model to embed the data points in low-dimensional space. It uses loss function to project data points.

It works as follows. Firstly, it finds the pairwise similarity of high-dimensional data points by calculating the pairwise distances of high-dimensional data points using Gaussian kernel and then converts the distance to the conditional probability by normalizing the value of the Gaussian kernel. Nearest data points will have higher conditional probability compare with data points that are far apart. The equation of conditional probability of high-dimensional data points are mentioned below.

$$p_{j|i} = \frac{exp(-||\mathrm{x}_i - \mathrm{x}_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||\mathrm{x}_i - \mathrm{x}_k||^2/2\sigma_i^2)}$$

where $\|\mathrm{x}_i - \mathrm{x}_j\|^2$ is the pairwise distance of data points between x_i and x_j and σ is the variance of the data points.

Then this condition probability will be symmetrized using the formula mentioned below in order to avoid the negative impact of the outliers of high-dimensional data points in cost function.

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

where n is the number of data points.
Secondly, it finds the pairwise similarity of the low-dimensional data points by using a heavy-tailed student t-distribution function in terms of conditional probability. The equation is mentioned here.

$$q_{ij} = \frac{(1 + \|\mathrm{y}_i - \mathrm{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathrm{y}_k - \mathrm{y}_l\|^2)^{-1}}$$

where $\|\mathrm{y}_i - \mathrm{y}_j\|^2$ is the pairwise distance between y_i and y_j
Thirdly, it uses kullback-Leibler divergence as cost function. To minimize the difference or loss between the two conditional probabilities p_{ij} and q_{ij}, gradient of the cost function is computed. The equation of the cost function and corresponding gradient are mentioned below.
Cost function is described as

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Gradient of the cost function is described as

$$\frac{\delta C}{\delta \mathrm{y_i}} = 4 \sum_j (p_{ij} - q_{ij})(\mathrm{y_i} - \mathrm{y_j})(1 + \|\mathrm{y}_i - \mathrm{y}_j\|^2)^{-1}$$

Advantages and Disadvantages: Compare to other non-parametric dimensionality reduction technique, t-SNE has few advantages and disadvantages.
t-SNE preserves the local structure of high-dimensional input data points very well. It also preserves global structure of high-dimensional input data points to some extent when a large number of nearest neighbours are chosen.
But it suffers from several shortcomings too. As the cost function of t-SNE is not convex, it may not converge to a globally optimal solution. It is not scaled well when projecting to more dimensions other than two or three dimensions. It has the perplexity hyper parameter i.e., numbers of nearest neighbour that has to be optimized to produce good results.

Uniform Manifold Approximation and Projection (UMAP): UMAP [MHM20] is a non-linear dimensionality reduction technique with improved run-time performance compares to t-SNE and it preserves the local structure of the high-dimensional input data points very well.

UMAP works as follows. Firstly, it constructs a weighted graph based on k-neighbour to learn the manifold structure or local structure of high-dimensional data points. This step is related to the Riemannian metric calculation i.e., calculating the distance between each of the data points and to its neighbour, and fuzzy simplicial set construction i.e, constructing a weighted k-neighbour graph.

Lets consider a input data set $X = \{x_1 ... x_N\}$. The distances between each of the data points x_i

with its k-nearest neighbour are computed to calculate the ρ_i and σ_i, where ρ_i is expressed as the minimum distance between each of the data point and its corresponding k nearest neighbour. The mathematical description of ρ_i is mentioned below.

$$\rho_i = min\{d(\mathrm{x_i}, \mathrm{x_{i_j}})|1 \leq j \leq, d(\mathrm{x_i}, \mathrm{x_{i_j}}) > 0\}$$

σ_i represents the normalization factor of the calculated distances. The mathematical description of σ_i is mentioned below.

$$\sigma_i = \sum_{j=1}^{k} exp(\frac{-max(0, d(\mathrm{x_i}, \mathrm{x_{i_j}}) - \rho_i)}{\sigma_i}) = \log_2(k)$$

ρ_i represents the fuzzy simplicial set i.e., local connectivity of each data point x_i in high-dimensional space. σ_i is a normalization factor of the above calculated distance and it represents the Riemannian metric i.e., the distance between x_i and its k^{th} nearest neighbour.
After that a weighted adjacency matrix A of the weighted directed graph $\hat{G}$ is computed to represent the local structure of the input data points in terms of the global structure. A weighted directed graph $\hat{G}$ is defined as

$$\hat{G} = (V, E, w)$$

where V are the vertices and it represents the dataset X, E are the directed edges between data points and its corresponding nearest neighbour e.g., an edge from x_i to x_j . Equation of E is as follows

$$E = \{(\mathrm{x_i}, \mathrm{x_{i_j}})|1 \leq j \leq K, 1 \leq i \leq N, \}$$

and weights are defined as

$$w(\mathrm{x_i}, \mathrm{x_{i_j}}) = exp(\frac{-max(0, d(\mathrm{x_i}, \mathrm{x_{i_j}}) - \rho_i)}{\sigma_i})$$

A symmetric matrix B is computed from the weighted adjacency matrix A. The equation for the symmetric matrix B of the undirected graph G is mention below and it provides the probabilistic information about the presence of minimum one directed edge between x_i and x_j. This undirected edge can be from x_i to x_j or from x_j to x_i. The symmetric matrix B can be described as

$$B = A + A^T - A \circ A^T$$

Secondly, it projects the input data points initially in low-dimensional space using the spectral embedding as it converges faster compare with random initialization. A weighted graph H is constructed in low-dimensional space using the points $\{Y_i\}_{i=1,..,N}$. In order to make low-dimensional weighted graph H equivalent to high-dimensional weighted graph G, cross-entropy (CE) cost function is used.
Initially total edge-wise cross-entropic losses are measured between the weighted graph G and H then attractive force along the edges and repulsive force among the vertices are applied to minimize the loss. This attractive force and repulsive force function are derived by computing the gradient of the cross-entropy cost function.

Attractive force is defined by the equation

$$\mathrm{F}_{attractive} = \frac{-2ab\|\mathrm{y}_i - \mathrm{y}_j\|_2^{2(b-1)}}{(1 + \|\mathrm{y}_i - \mathrm{y}_j\|_2^2)}(w(\mathrm{x_i}, \mathrm{x}_j))(\mathrm{y}_i - \mathrm{y}_j)$$

and repulsive force is defined by

$$\mathrm{F}_{repulsive} = \frac{2b}{(\epsilon + \|\mathrm{y}_i - \mathrm{y}_j\|_2^2)(1 + a\|\mathrm{y}_i - \mathrm{y}_j\|_2^{2b})}(1 - w(\mathrm{x_i}, \mathrm{x}_j))(\mathrm{y}_i - \mathrm{y}_j)$$

where a and b are hyper parameter.

In this way by utilizing the attractive force and the repulsive force to minimize the cross-entropic loss between high and low-dimensional weighted graph, low-dimensional projection matches the topological structure of high-dimensional input data points.

Advantages and Disadvantages: Advantages and disadvantages of UMAP are described below. Run-time performance is far better compare to t-SNE. It preserves the topological structure of the input dataset and more global structure compare to t-SNE. It is a more deterministic and more stable algorithm compare to t-SNE. In order to be deterministic, it uses the normalized Procrustes distance to minimize the squared error between high-dimensional input data points and rotated low-dimensional embedding data points.
Like other dimensionality reduction algorithms, UMAP has also disadvantage. It does not preserve the global structure of high-dimensional input data points as good as MDS.

3.2 Clustering

Clustering is a process of grouping similar data points or objects in the same group and dissimilar data points or objects in the different group. It can be used to label unlabelled data points. There are several example of clustering algorithm in the field of the data mining and machine learning e.g., K-means, mean shift, Gaussian Mixture Model. In comparison to other clustering techniques, Gaussian Mixture Model (GMM) has several advantages e.g., it uses probabilistic information to group the data points in the similar group instead of only using the distance of data points from the mean like K-means algorithm. GMM is used here to find class labels of time series datasets, and during low-dimension projection different classes are visualized with different colours.

Gaussian Mixture Model (GMM): Gaussian mixture model [Ras+99; SB07] is a probabilistic model and it can be used as a clustering algorithm for unlabelled data points. In this model, it is assumed that the data are distributed as Gaussian having more than one mode or Gaussian density. Mixture model of Gaussian defines linear combination of Gaussians and expressed as follows:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

where $\mathcal{N}(x|\mu_k, \Sigma_k)$ is called mixture component of Gaussian distribution or density. μ_k is mean, Σ_k is covariance and π_k is mixing coefficient. Where mixing coefficient is defined as

$$\sum_{k=1}^{K} \pi_k = 1$$

Parameters i.e., mean, covariance and mixing coefficient of GMM are calculated by the maximum likelihood, specifically using expectation-maximization (EM) algorithm.
Here, a binary random variable z is introduced having k dimensions. Where, $z_k = 1$ for a particular element and $z_k = 0$ for others. The marginal distribution of z in terms of the mixing coefficient is defined as below.

$$p(z_k) = \pi_k$$

Here, the conditional distribution of the input x with respect to z is define as

$$p(x|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k)$$

and the marginal distribution for the input x is calculated as

$$p(x) = \sum_{z} p(z)p(x|z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

Lets input dataset $\{x_1, ..x_N\}$ will be grouped or clustered by GMM algorithm. The log likelihood of GMM is defined as below

$$lnp(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} ln\Big\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \Big\}$$

Here, X is a matrix of size $N \times D$.
Now, the aim is to maximize the log likelihood function. For that, first initialize the parameters i.e., mean, covariance and mixing coefficient randomly. After that, in expectation step, we calculate the posterior probabilities or responsibilities by the below formula.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

In maximization step, parameters are updated by posterior probability values. After that, log likelihood function is evaluated if log likelihood function or parameters are converged. If it is not converged, then again expectation and maximization steps are performed.

4 Implementation

In this section, proposed algorithm is discussed elaborately. Apart from that, implementation and used techniques are also presented here.

4.1 Proposed Algorithm

The first phase of the proposed algorithm is training, and the training phase is executed only once. During training low-dimensional projection of a given train dataset is obtained by applying given dimensionality reduction technique on the train dataset. This proposed algorithm uses nearest neighbour information of out-of-sample data points to map out-of-sample data points in low-dimensional points. In order to do that, first the algorithm calculates the nearest neighbours of the out-of-sample data points in the training data points. Secondly, it calculates the low-dimensional projection of the corresponding nearest neighbour [FH89]. This low-dimensional projection of the corresponding nearest neighbour are considered as feature vectors of the intermediate matrix. Afterwards, low-dimensional projection of corresponding nearest neighbour is combined with low-dimensional embedding of Kernel-PCA for the out-of-sample data points to compute the intermediate matrix and to incorporate the characteristics of the out-of-sample data points as well as to overcome the problem of placing all out-of-sample data points to a specific point when out-of-sample data points have same nearest neighbours. Next, it computes the kernel matrix by calculating matrix inner product of intermediate matrix which contains nearest neighbour and kernel-PCA embedding.
Here, numbers of nearest neighbour are restricted to two to optimally use the nearest neighbour information as with the information of two nearest neighbours, a desirable out-of-sample projection of the original DR technique's result can be obtained.

Lets $DR_{technique}$ be a dimensionality reduction technique, and the main objective of the proposed algorithm is to reproduce the projection of the $DR_{technique}$ and to extend it for the out-of-sample data points. The first phase of the proposed algorithm is training, and the training phase is executed only once. During training low-dimensional projection $X_{trainLowDimension}$ of a given train data set X_{train} is obtained by applying given dimensionality reduction technique $DR_{technique}$ on the train data set. X_{train} and $X_{trainLowDimension}$ are stored and used in later phases for out-of-sample projection. Let X_{train} has n_{train} number of data points, and $DR_{technique}$ projects high-dimensional data points in d dimensions, then $X_{trainLowDimension}$ is a matrix of size $n_{train} \times d$.

Algorithm 4.1 Training phase

1: **function** TRAIN(X_{train}, $DR_{technique}$)
2: $X_{trainLowDimension} \leftarrow DR_{technique}(X_{train})$
3: **return** $X_{trainLowDimension}$

In the second phase of this algorithm, the projection of the out-of-sample data points (unseen) are performed. This phase is divided into two cases.

- out-of-sample data points projection for the first time
- out-of-sample data points projection after the first projection

Here, we first describe the functionality of the algorithm 4.2 `outOfSampleProjection`, and then we elaborate on the above-mentioned cases.

Inputs to the out-of-sample projection function are training dataset X_{train} (decided before training), low-dimensional projection corresponding to train data $X_{trainLowDimension}$ (calculated during training), unseen data points X_{test}, number of neighbour k, kernel function ϕ and a matrix M. The matrix M helps in updating the projection of the out-of-sample data points, it is empty before the first out-of-sample projection, or it contains values of earlier out-of-sample projections.

Algorithm 4.2 Out-of-sample projection

1: **function** OUTOFSAMPLEPROJECTION(X_{train}, $X_{trainLowDimension}$, X_{test}, k, ϕ, M)
2: $X_{testKPCA} \leftarrow$ kernelPCA (X_{test}, ϕ)
3: $P \leftarrow$ findNearestNeighbours(X_{test}, X_{train}, k)
4: **for** each data point X_i in X_{test} **do**
5: $\gamma \leftarrow$ findLowDimensionalProjOfNearestNeighbours(P_i, $X_{trainLowDimension}$)
6: $[M_{test}]_i \leftarrow (\gamma, X_{testKPCA[i]})$
7: **end for**
8: $M \leftarrow$ appendRow(M, M_{test})
9: $A \leftarrow (MM^T)$
10: $X_{testDRprojection} \leftarrow kernelPCA\ (A)$
11: **return** $X_{testDRprojection}$, M

Out-of-sample projection works as follows.
First a low-dimensional projection $X_{testKPCA}$ for the input dataset X_{test} is calculated using kernelPCA with kernel function ϕ. Let X_{test} have n_{test} number of data points, and $X_{testKPCA}$ projects high-dimensional data points in the d_{kpca} dimensions, then $X_{testKPCA}$ has size $n_{test} \times d_{kpca}$. Here, in the experiment, value of d_{kpca} is two as we are projecting out-of-sample data points in 2D.

Afterwards, algorithm 4.3 `findNearestNeighbours` determines a matrix P which contains indices of nearest neighbour of X_{test}. Inputs to `findNearestNeighbours` function are out-of-sample data X_{test}, training data X_{train} and the number of nearest neighbour k. The `calcPairwiseDist` function determines a matrix $D_{testTrain}$ of size $n_{test} \times n_{train}$ containing pairwise distances between all data points from X_{test} and all data points from X_{train}.

Here, the i^{th} row d_i of the distance matrix $D_{testTrain}$ represents the distance between the i^{th} data point in X_{test} and all data from X_{train}. Now, the `findIndicesOfSmallestDistances` function finds the indices of k smallest distances in the distance record d_i, and the result is stored in the i^{th} row of `nearestNeighboursIndices` matrix, and the size of this matrix is $n_{test} \times k$. The i^{th} record in `nearestNeighboursIndices` contains indices of k nearest neighbours from X_{train} of i^{th} data in X_{test}.

Algorithm 4.3 Find nearest neighbour

1: **function** FINDNEARESTNEIGHBOURS(X_{test}, X_{train}, k)
2: $D_{testTrain} \leftarrow$ calcPairwiseDist(X_{test}, X_{train})
3: **for** each row d_i in $D_{testTrain}$ **do**
4: $[nearestNeighboursIndices]_i \leftarrow$ findIndicesOfSmallestDistances(d_i, k)
5: **end for**
6: **return** $nearestNeighboursIndices$

After that, for each data point X_i in X_{test} `findLowDimensionalProjectionOfNearestNeighbours` function of algorithm 4.2 finds the low-dimensional projection of nearest neighbour. Inputs to the `findLowDimensionalProjectionOfNearestNeighbours` are P_i and $X_{trainLowDimension}$, where P_i is the record of P at i^{th} position, which contains indices of k nearest neighbours from X_{train} of X_i and $X_{trainLowDimension}$ is low-dimensional projection matrix. It appends $X_{trainLowDimension}[j]$ to $result$, where $X_{trainLowDimension}[j]$ is the value at i^{th} position of $X_{trainLowDimension}$, and $result$ is stored in an intermediate variable γ.

Algorithm 4.4 Find low-dimensional projection of nearest neighbours

1: **function** FINDLOWDIMENSIONALPROJOFNEARESTNEIGHBOURS(P_i, $X_{trainLowDimension}$)
2: $result$ = [] /* empty array */
3: **for** each j in P_i **do**
4: $result \leftarrow$ append($X_{trainLowDimension}[j]$)
5: **end for**
6: **return** $result$

γ is an array of size (k d) where k is the number of neighbours and d is the number of dimension of each data in $X_{trainLowDimension}$ (determined in training). After that, algorithm 4.2 appends γ and corresponding i^{th} data points in $X_{testKPCA}$, and produces an array of size ((k d) + d_{kpca}) , and stores the record in i^{th} row of the matrix M_{test}.

The matrix M_{test} has size $n_{test} \times ((k\ d) + d_{kpca})$. In next step, M (input to the out-of-sample projection function) is extended by appending M_{test} into M row-wise. Let, before extension, M has n number of rows, then after the extension M has $n + n_{test}$ number of rows. Note, when M is empty, then number of rows n = 0 before extension. After that, inner product of M is performed and a squared matrix A is produced of size $(n + n_{test}) \times (n + n_{test})$. This squared matrix A works a precomputed kernel to kernel-PCA, and Kernel-PCA calculates final low-dimensional projection $X_{testDRprojection}$. Finally, `outofSampleProjection` fuction returns $X_{testDRprojection}$ and M.

In case of out-of-sample data points projection for the first time, the input matrix M is empty.

During the first projection of out-of-sample data points, data are stored in matrix M for the first time. Note, this matrix M helps updating the projection of out-of-sample data in later projections.

In case of out-of-sample data points projection after the first projection, matrix M is not empty. It contains the data calculated in earlier projections. This matrix M is extended and used in current projection as mentioned above, and also stored data for future use for updating the projection of out-of-sample data points.

Hence, this matrix M helps in updating the projection of out-of-sample data smoothly. When new out-of-sample data arrives, all the steps mentioned in the algorithm are executed only for this new data points, because the matrix M serves required information of earlier projections.

4.2 Technical Detail

In this subsection, we provide a brief technical detail on our implementation. We have implemented the proposed algorithm in python programming language, and used several python libraries for different purpose, e.g., sklearn [PVG+11], NumPy [HMW+20], sciPy [VGO+20], matplotlib [Hun07], csv and umap-learn [MHM18].

sklearn library is used to import pairwise-distance function to calculate the pairwise distance between out-of-sample and training data points. Different DR methods e.g., t-SNE, Isomap, MDS and KPCA and clustering algorithm GMM are imported from this library.
For better visualization of the data points, different colours are used in 2D plot for different labels. But as time series data set such as Air quality does not have labels, GMM is used to label the data points.
In our solution, NumPy library is used for several purposes e.g., to find nearest neighbours, to process input data sets and to calculate matrix operations e.g., matrix multiplication, initializing matrix dimensions.
Scipy is introduced in this implementation to calculate one of the evaluation metrics i.e., the Spearman rank correlation metric.
Here, Matplotlib is used to visualize the low-dimensional data points in 2D.

5 Evaluation

In this chapter, used datasets and detail evaluation of the proposed algorithm are demonstrated.

5.1 Dataset Description

In order to evaluate the robustness of the algorithm, evaluating the algorithm on variety of datasets are important. Eight datasets have been used in evaluation, based on the several literature review [EMK+19; ZWG+21]. In this literature, datasets are described in terms of the number of instances, type of the instances (image, text, multivariate i.e., more than one Gaussian mode, time series etc.), number of dimensions or features and number of categories or classes.

MNIST: Modified National Institute of Standards and Technology database or MNIST[1][2] is a dataset of grey scale images of handwritten digits. It consists of 70000 instances. It has 784 dimensions and 10 classes [LCB10].

Fashion MNIST: Fashion MNIST[3][4] is an image dataset. It consists of 70000 images of clothes, shoes and bags. This dataset has 784 dimensions and 10 classes [XRV17].

COIL 20: Columbia university image library[5][6] is a data set of gray scale images of 20 objects. It consists of 1440 instances, 400 dimensions and 20 classes [EMK+19; NNM+96].

BBC-News: BBC-News[7] is a text-based dataset consists of 2225 instances, 9635 dimensions and 5 classes. Here, instance represents number of articles or documents, dimension represents number of distinct words. Class labels for this dataset are business, entertainment, politics, sport and technology [GC06].

[1] http://yann.lecun.com/exdb/mnist/
[2] https://www.tensorflow.org/datasets/catalog/mnist
[3] https://github.com/zalandoresearch/fashion-mnist
[4] https://www.tensorflow.org/datasets/catalog/fashion_mnist
[5] https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[6] https://mespadoto.github.io/proj-quant-eval/post/datasets/
[7] http://mlg.ucd.ie/datasets/bbc.html

Spambase: Spambase[8] is a multivariate, text-based dataset, can be used to detect spam in a mail. It consists of 4601 instances which represent number of e-mails, 57 dimensions which represent number of distinct words and 2 class labels. Class labels are spam or non-spam [HRFS99].

Airquality: Airquality[9] is a multivariate, time-series dataset of air quality over time. Data is collected from air quality sensor device on hourly basis. It consists of 9358 instances and 15 dimensions. Dimensions of the dataset represent the concentrations of different chemical present in the air, temperatures, humidity etc. [DMP+08].

Simulated data for survival modelling (Survival Data): Survival Data[10] is a multivariate, time-series dataset consisting of 120000 instances and 24 dimensions. It can be used to train and test a model to learn strategy of survival.

IRIS: IRIS[11][12] is a multivariate dataset consists of 150 instances, 4 dimensions and 3 classes. These are 3 flower samples and the dimensions are size measures of different parts of flowers [Fis36].

Table 5.1: Characteristic of Datasets

	Number of Instances	Type of Instances	Dimensions	Number of classes
MNIST	70000	*Image*	784	10
Fashion MNIST	70000	*Image*	784	10
COIL 20	1440	*Image*	400	20
BBC-News	2225	*Text*	9635	5
Spambase	4601	*Text*	57	2
Airquality	9358	*Timeseries*	15	–
Survival Data	12000	*Timeseries*	24	–
IRIS	150	*Tabular*	4	3

5.2 Metric Definition

Dimensionality reduction evaluation metrics are used to quantify the projection quality and to choose the appropriate technique for a particular task. Computing the quantitative and qualitative difference among different DR techniques are based on several criteria e.g., numbers of false

[8] https://archive.ics.uci.edu/ml/datasets/Spambase
[9] https://archive.ics.uci.edu/ml/datasets/Air+Quality
[10] https://archive.ics.uci.edu/ml/datasets/Simulated+data+for+survival+modelling
[11] https://archive.ics.uci.edu/ml/datasets/iris
[12] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html

neighbour or missing neighbour in low-dimensional projection, preserving the pairwise distances or all-point-pair distances, run-time performance, difference between true and predicted label etc. There are number of evaluation metrics find in the literature, and some of them are described below that are used in this thesis based on the several literature review [EMK+19; ZWG+21].

Trustworthiness: Trustworthiness [EMK+19; VK06] measures the proportion of the wrong or false neighbour that are present in the low-dimensional space compare to the high-dimensional space. This metric is used to measure the quality of the local structure maintained in low-dimensional space. The equation to calculate the trustworthiness is mentioned below.

$$1 - \frac{2}{NK(2n - 3K - 1)} \sum_{i=1}^{N} \sum_{j \in U_i^{(K)}} (r(i, j) - K)$$

$U_i^{(K)}$ is the number of k neighbours that are present in low-dimensional projection but not present in the high-dimensional input space, N is the number of samples, n is the number of dimensions of input data set, $r(i, j)$ is the rank of low-dimensional projection . The value of trustworthiness lies between 0 to 1. Trustworthiness is highest when number of false neighbours present in the low-dimensional projection are zero and then value of this metric is one. On the other hand, trustworthiness is lowest when all neighbours that are present in low-dimensional projection are false and in that case it has value as zero.

Continuity: Continuity [EMK+19] measures the proportion of missing neighbour in the low-dimensional projection which is presents in the high-dimensional input data set. Like trustworthiness, this metric is also used to measure the quality of local structure maintained in the low-dimensional space. The mathematical equation of continuity is below.

$$1 - \frac{2}{NK(2n - 3K - 1)} \sum_{i=1}^{N} \sum_{j \in V_i^{(K)}} (\hat{r}(i, j) - K)$$

where $v_i^{(K)}$ represents the k nearest neighbours present in the high-dimensional space but not present in low-dimensional space, N represents the number of samples, n defines the number of dimensions of input data set. The value of Continuity lies between 0 to 1. It provides highest value as one when number of missing neighbours is zero in the low-dimensional projection and it provides lowest value as zero when all neighbours present in the high-dimensional space are missing in the low-dimensional space.

Normalized Stress: Normalized stress [EMK+19] defines the normalized value of the difference of the pair wise distance of the high-dimensional data points and the low-dimensional data points. It measures quality of the global structure of input dataset maintained in low-dimensional projection. The value of this metric lies between 0 to 1. Zero metric value represents the difference of the pairwise distance of the high-dimensional points and the pairwise distance of the low-dimensional points is zero, and low-dimensional projection preserves the global structure. One metric value

represents high and corresponding low-dimensional points are far apart and the global structure of the input data points are not maintained. The equation of the normalized stress is below.

$$\frac{\sum_{ij}(\Delta^n(\mathrm{x_i},\mathrm{x_j}) - \Delta^q(P(\mathrm{x_i}),P(\mathrm{x_j})))^2}{\sum_{ij}\Delta^n(\mathrm{x_i},\mathrm{x_j})^2}$$

Δ^n and Δ^q denotes the pairwise distance metric e.g., Euclidean, Manhattan, cosine etc. Choice of distance metric depends on application or other numerous factors.

Spearman Rank Correlation: Spearman rank correlation coefficient [ZK99] provides an intuition on how the high-dimensional data points and the low-dimensional data points are correlated to each other. The value of this coefficient ranges from +1 to −1. +1 and −1 coefficient value depict that rank of the data points are perfectly correlated to each other and 0 refers that rank of the data points are not correlated at all. The mathematical equation of Spearman rank correlation is given below.

$$r_s = \frac{n\sum_{i=1}^n u_i v_i - \left(\sum_{i=1}^n u_i\right)\left(\sum_{i=1}^n v_i\right)}{\sqrt{[n(\sum_{i=1}^n u_i^2) - (\sum_{i=1}^n u_i)^2][n(\sum_{i=1}^n v_i^2) - (\sum_{i=1}^n v_i)^2]}} = 1 - \frac{n\sum_{i=1}^n d_i^2}{n(n^2-1)}$$

Where $d_i = u_i - v_i$
u_i is rank of the i^{th} data points of high-dimensional dataset, v_i is rank of i^{th} data point of low-dimensional points and n is the total number of data points.

Shepard Diagram: Shepard diagram represents the relation between pairwise distance of the low-dimensional data points and pairwise distance of the high-dimensional data points in a graphical way by using scatterplot [De 11]. It provides a concrete concept of how original and projected data points are far apart. All pair wise distance of the original data points will be along the X -axis and the pairwise distance of the projected data points will be along the Y-axis in the scatterplot. When data points before and after applying dimensionality reduction technique are close to each other, the plot is along the diagonal line.

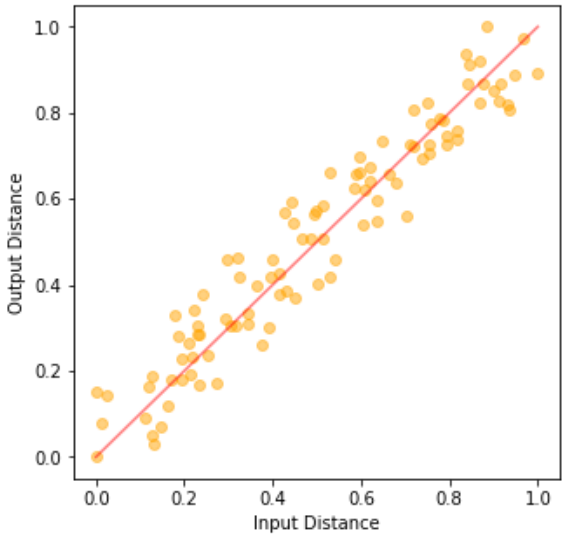


Figure 5.1: Shepard Diagram

5.3 Experimental Setup

In this experiment, we have used eight datasets. Apart from MNIST and Fashion-MNIST, different size of training, out-of-sample and additional out-of-sample datasets are considered, and they are listed in table 5.2. Training data is used to train the proposed model. Out-of-sample data are used to project the unseen data points in low-dimensional space by the proposed algorithm with the training knowledge for the first time. The experiment is also performed on extended out-of-sample data. Extended out-of-sample data includes previous out-of-sample data points and additional out-of-sample data points which are introduced newly for the projection. E.g., in case of IRIS, proposed algorithm is trained with 80 data points, and it performs projection for the first time with 50 out-of-sample data points. Afterwards, out-of-sample data is extended as additional 20 out-of-sample data are included for projection, and proposed algorithm smoothly update the projection for extended out-of-sample data points of size 70 (50 + 20).
In this experiment, two nearest neighbours of out-of-sample data points are considered but in case of BBC-News three nearest neighbours are considered as BBC-News has higher number of dimensions compare to the data instances, cosine similarity kernel function is used for KPCA when embedding of out-of-sample data points are calculated to compute intermediate matrix, and proposed algorithm projects high-dimensional data points in two dimensions (2D). In case of determining trustworthiness and continuity, the number of nearest neighbours are chosen based on the size of data set. 10 nearest neighbours are used when number of instances are large (>300). Otherwise 5 nearest neighbours are considered.

Table 5.2: Datasets used for this experiment

	Training data	Out-of-sample data	Additional out-of-sample data
MNIST	5000	5000	5000
Fashion MNIST	5000	5000	5000
COIL 20	700	600	140
BBC-News	1100	800	325
Spambase	2000	1500	1100
Airquality	4500	3500	1200
Survival Data	4000	3000	2000
IRIS	80	50	20

5.4 Experiment Result

In this section, the experimental results of the proposed algorithms are discussed in details. To evaluate the proposed algorithm, three DR techniques, eight datasets and five evaluation metrics are used. Three DR techniques are UMAP, t-SNE and Isomap. Five evaluation metrics are trustworthiness, continuity, normalized stress, Spearman correlation and Shepard diagram. Eight datasets are MNIST, Fashion-MNIST, Coil20, BBC-News, Spambased, Air Quality, Survival Data and IRIS. Experimental results will be represented as evaluation metrics, figures and time-based performance.

5.4.1 Evaluation Metrics-Based Comparison

In table 5.3, 5.5 and 5.7, we have compared the result of the evaluation metrics between different DR techniques and the newly proposed algorithm on out-of-sample data points for different datasets. E.g., in table 5.3, it can be observed that the results of evaluation metric compared between UMAP and the newly proposed algorithm. The first row of the table provides the information of trustworthiness, continuity, normalized stress and Spearman correlation of UMAP and the proposed algorithm for MNIST dataset. The trustworthiness of UMAP is 0.96 and for the newly proposed algorithm is 0.92.
In table 5.4, 5.6 and 5.8, we have compared the values of the evaluation metrics for the extended out-of-sample data points.

Here, we have performed further study to identify the exceptions where the results are not comparable in some extent. In our study, we have considered a threshold value of 0.04 to identify such cases where results are noticeably different. If for a particular scenario absolute difference between the evaluation result of an original DR technique and the evaluation result of the proposed algorithm is more than 0.04, then that particular record is highlighted in the table - with colour green where result of the proposed algorithm is noticeably better than result of an original DR technique, and with colour orange where result of an original DR technique is noticeably better than that of the proposed algorithm.
Apart from normalized stress, higher value of the evaluation metric refers better projection quality.

Table 5.3: UMAP vs. Proposed algorithm for out-of-sample data

	Trustworthiness		Continuity		Normalized stress		Spearman correlation	
Dataset	**UMAP**	**Proposed**	**UMAP**	**Proposed**	**UMAP**	**Proposed**	**UMAP**	**Proposed**
MNIST	0.96	0.92	0.96	0.95	0.90	0.90	0.34	0.39
Fashion-MNIST	0.97	0.96	0.98	0.97	0.91	0.90	0.60	0.63
Coil20	0.99	0.96	0.99	0.98	0.88	0.88	0.39	0.43
BBC-News	0.66	0.60	0.79	0.72	0.97	0.97	0.14	0.19
Spambased	0.99	0.98	0.99	0.99	0.99	0.99	0.47	0.63
Air Quality	0.99	0.98	0.99	0.98	0.99	0.99	0.54	0.55
Survival Data	0.99	0.98	0.99	0.98	0.96	0.96	0.84	0.86
IRIS	0.97	0.96	0.97	0.94	0.59	0.63	0.81	0.84

Table 5.4: UMAP vs. Proposed algorithm for extended out-of-sample data

	Trustworthiness		Continuity		Normalized stress		Spearman correlation	
Dataset	**UMAP**	**Proposed**	**UMAP**	**Proposed**	**UMAP**	**Proposed**	**UMAP**	**Proposed**
MNIST	0.96	0.91	0.97	0.95	0.90	0.90	0.35	0.38
Fashion-MNIST	0.97	0.96	0.98	0.97	0.90	0.90	0.60	0.63
Coil20	0.99	0.97	0.99	0.98	0.88	0.88	0.26	0.43
BBC-News	0.67	0.59	0.83	0.71	0.97	0.98	0.15	0.19
Spambased	0.99	0.99	0.99	0.99	0.99	0.99	0.49	0.65
Air Quality	0.99	0.98	0.99	0.98	0.98	0.99	0.53	0.56
Survival Data	0.99	0.98	0.99	0.98	0.96	0.96	0.87	0.86
IRIS	0.97	0.97	0.97	0.96	0.61	0.6	0.77	0.9

Table 5.5: t-SNE vs. Proposed algorithm for out-of-sample data

	Trustworthiness		Continuity		Normalized stress		Spearman correlation	
Dataset	**t-SNE**	**Proposed**	**t-SNE**	**Proposed**	**t-SNE**	**Proposed**	**t-SNE**	**Proposed**
MNIST	0.98	0.90	0.96	0.96	0.91	0.91	0.45	0.45
Fashion-MNIST	0.98	0.96	0.98	0.97	0.92	0.92	0.68	0.71
Coil20	0.99	0.96	0.98	0.97	0.87	0.86	0.61	0.66
BBC-News	0.66	0.60	0.82	0.78	0.97	0.98	0.34	0.49
Spambased	0.99	0.99	0.99	0.99	0.99	0.99	0.63	0.57
Air Quality	0.99	0.98	0.99	0.99	0.99	0.99	0.49	0.54
Survival Data	0.99	0.98	0.98	0.98	0.97	0.97	0.58	0.60
IRIS	0.95	0.97	0.94	0.98	0.9	0.63	0.57	0.9

Table 5.6: t-SNE vs. Proposed algorithm for extended out-of-sample data

	Trustworthiness		Continuity		Normalized stress		Spearman correlation	
Dataset	**t-SNE**	**Proposed**	**t-SNE**	**Proposed**	**t-SNE**	**Proposed**	**t-SNE**	**Proposed**
MNIST	0.98	0.91	0.97	0.96	0.91	0.91	0.37	0.45
Fashion-MNIST	0.99	0.96	0.98	0.98	0.92	0.92	0.68	0.71
Coil20	0.99	0.96	0.99	0.97	0.89	0.86	0.56	0.66
BBC-News	0.65	0.61	0.78	0.78	0.98	0.98	0.25	0.45
Spambased	0.99	0.99	0.99	0.99	0.99	0.99	0.50	0.57
Air Quality	0.99	0.99	0.99	0.99	0.90	0.99	0.51	0.54
Survival Data	0.99	0.98	0.98	0.98	0.97	0.96	0.61	0.60
IRIS	0.97	0.96	0.96	0.97	0.81	0.63	0.67	0.9

Table 5.7: Isomap vs. Proposed algorithm for out-of-sample data

	Trustworthiness		Continuity		Normalized stress		Spearman correlation	
Datasets	**Isomap**	**Proposed**	**Isomap**	**Proposed**	**Isomap**	**Proposed**	**Isomap**	**Proposed**
MNIST	0.76	0.79	0.95	0.94	0.92	0.92	0.50	0.51
Fashion-MNIST	0.91	0.92	0.97	0.97	0.93	0.92	0.75	0.75
Coil20	0.90	0.91	0.97	0.96	0.89	0.90	0.38	0.40
BBC-News	0.57	0.57	0.59	0.71	0.99	0.99	0.52	0.25
Spambased	0.97	0.97	0.98	0.98	0.99	0.99	0.95	0.95
Air Quality	0.92	0.97	0.98	0.98	0.99	0.99	0.55	0.49
Survival Data	0.98	0.98	0.98	0.98	0.97	0.97	0.99	0.99
IRIS	0.93	0.91	0.91	0.89	0.69	0.7	0.9	0.83

Table 5.8: Isomap vs. Proposed algorithm for extended out-of-sample data

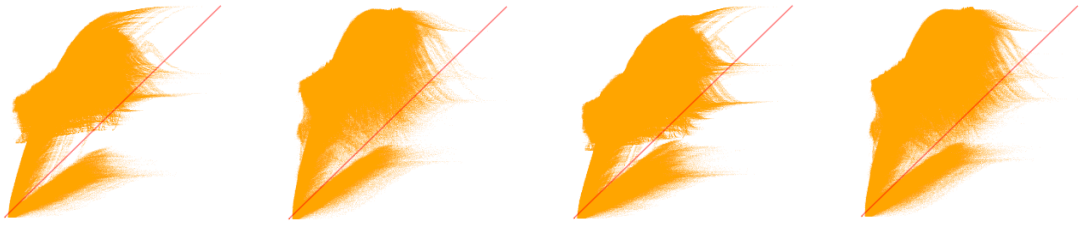
	Trustworthiness		Continuity		Normalized stress		Spearman correlation	
Datasets	**Isomap**	**Proposed**	**Isomap**	**Proposed**	**Isomap**	**Proposed**	**Isomap**	**Proposed**
MNIST	0.76	0.79	0.96	0.94	0.92	0.92	0.55	0.52
Fashion-MNIST	0.91	0.92	0.97	0.97	0.93	0.92	0.75	0.74
Coil20	0.92	0.92	0.98	0.97	0.90	0.90	0.32	0.39
BBC-News	0.92	0.92	0.98	0.97	0.90	0.90	0.32	0.39
Spambased	0.98	0.97	0.99	0.98	0.99	0.99	0.95	0.95
Air Quality	0.94	0.97	0.98	0.95	0.99	0.99	0.54	0.49
Survival Data	0.98	0.98	0.98	0.98	0.97	0.97	0.99	0.99
IRIS	0.95	0.93	0.95	0.91	0.69	0.7	0.88	0.82

The proposed algorithm uses knowledge acquired during training to project the out-of-sample data points. As out-of-sample projection by the proposed algorithm is greatly influenced by the training knowledge, discrepancies between original projection by the DR technique and estimated projection by the proposed algorithm could be observed for some cases e.g., in table 5.5, it can be observed that trustworthiness for t-SNE is better than proposed algorithm for MNIST dataset that depicts the proposed algorithm introduces more false neighbours than t-SNE. In the same table, the proposed algorithm is better than t-SNE for Spearman correlation for Coil20, BBC-News, Air Quality and IRIS dataset which depicts linear relationship between data points are maintained more in case of the proposed algorithm than t-SNE. In table 5.6, it is observed that the value of normalized stress is better for proposed algorithm than t-SNE. In table 5.7, continuity for the proposed algorithm is better than Isomap for BBC-News dataset which demonstrates that number of missing neighbours are less for proposed algorithm compare to Isomap.

The proposed algorithm maps out-of-sample data points as per training knowledge and DR technique projects freshly for that out-of-sample data points. We can observe from above tables that in case of BBC-News dataset, overall performance of UMAP, t-SNE and the proposed algorithm are not good as it has less number of data points compare with the number of dimensions, which could be the reason behind such performance.

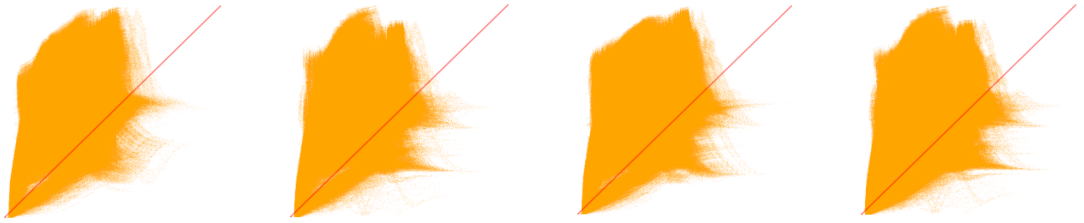
Based on the results mentioned above, we can conclude that our proposed algorithm provides good quality outputs as the overall evaluation results of the proposed algorithm are comparable to the evaluation results of corresponding original DR technique.

Shepard diagram-based comparison Shepard diagram gives the clear idea about the closeness of high and low-dimensional data points in graphical representational way. In Shepard diagram-based comparison, we are comparing shape of the Shepard diagrams of the proposed algorithm and corresponding original DR technique. Matching shape of the Shepard diagrams of the proposed algorithm and corresponding original DR technique ensure that the proposed algorithm projects the out-of-sample data points same as an original DR technique.
We can see from the given figures that the newly proposed algorithm preserves the closeness of high and low-dimensional data points as it is preserved by the original DR techniques as the shape of the Shepard diagram is maintained.



(a) UMAP **(b)** Proposed **(c)** UMAP **(d)** Proposed

Figure 5.2: Shepard diagram of UMAP and the proposed algorithm on Survival dataset for out-of-sample, extended out-of-sample data points



(a) t-SNE **(b)** Proposed **(c)** t-SNE **(d)** Proposed

Figure 5.3: Shepard diagram of t-SNE and proposed algorithm on Survival dataset for out-of-sample, extended out-of-sample data points

(a) Isomap **(b)** Proposed **(c)** Isomap **(d)** Proposed

Figure 5.4: Shepard diagram of Isomap and proposed algorithm on Survival dataset for out-of-sample, and extended out-of-sample data points

Figure 5.2a represents the Shepard diagram of UMAP for out-of-sample survival dataset, 5.2b represents the Shepard diagram of the proposed algorithm for out-of-sample survival dataset, 5.2c which is denser than 5.2a, represents the Shepard diagram of UMAP for extended out-of-sample data points and 5.2d which is denser than 5.2b, represents the Shepard diagram of the proposed algorithm for extended out-of-sample data points.
While comparing 5.2a and 5.2b, we can see that the shape of the diagrams is almost same but the upper part of the figure 5.2b is wider than figure 5.2a, and there is a spike in figure 5.2a which is missing in figure 5.2b. In the lower part of the same figures, we can observe that the lower part of the figures below the diagonal line for figures 5.2b is wider than 5.2a which depicts that proposed algorithm introduced some error in low-dimensional projection.
When data points are above the diagonal line, pair-wise distances between high-dimensional points and low-dimensional points are over-estimated and when below the diagonal line, pairwise distances are under-estimated. Though the Shepard diagram of both of the proposed algorithm and corresponding original DR technique contain over and under-estimated distances, in case of proposed algorithm, there are more data points which are above and below the diagonal line which depicts error insertion while projecting out-of-sample data points. But overall shape of the Shepard diagrams are same.

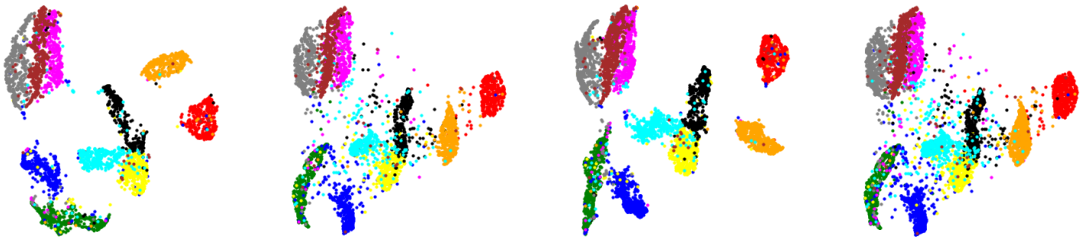
In figures 5.4a, 5.4b, 5.4c and 5.4d, we can observe that the data points are almost along the diagonal line for both, Isomap and the proposed algorithm but with little shift of the data points from the diagonal line for proposed algorithm compare to Isomap.

5.4.2 Figure-based Comparison

Here, we show the comparison of figures between three DR techniques and the newly proposed algorithm for out-of-sample and extended out-of-sample data points for eight datasets.

Comparison of experiment results between UMAP and the proposed algorithm In this sub section, figures of experimental result of UMAP and the proposed algorithm are shown. E.g., in the figure 5.5, figure 5.5a represents the projection of UMAP for out-of-sample data points for MNIST dataset, figure 5.5b represents the projection of proposed algorithm for out-of-sample data points, figure 5.5c which is denser than figure 5.5a, represents the projection of UMAP for

extended out-of-sample data points and figure 5.5d which is denser than figure 5.5b, represents the projection of the proposed algorithm for extended out-of-sample data points.



(a) UMAP **(b)** Proposed **(c)** UMAP **(d)** Proposed

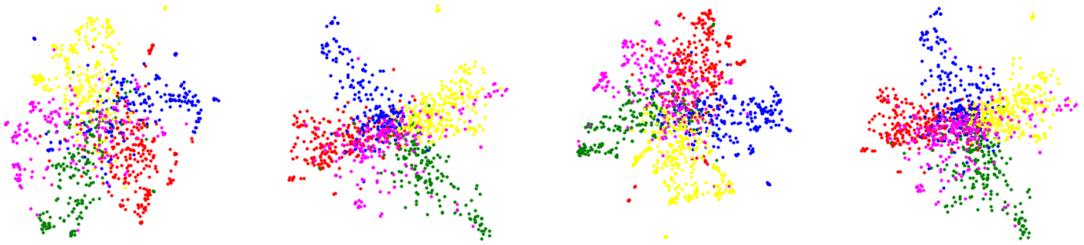
Figure 5.5: UMAP and proposed algorithm on MNIST dataset

(a) UMAP **(b)** Proposed **(c)** UMAP **(d)** Proposed

Figure 5.6: UMAP and proposed algorithm on Fashion-MNIST dataset

(a) UMAP **(b)** Proposed **(c)** UMAP **(d)** Proposed

Figure 5.7: UMAP and proposed algorithm on Coil20 dataset



(a) UMAP **(b)** Proposed **(c)** UMAP **(d)** Proposed

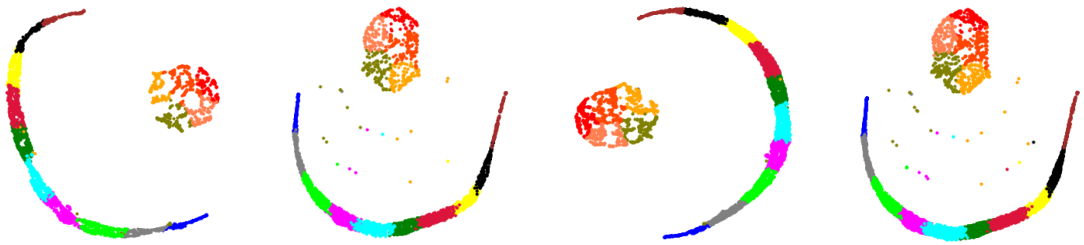
Figure 5.8: UMAP and proposed algorithm on BBC-News dataset

(a) UMAP **(b)** Proposed **(c)** UMAP **(d)** Proposed

Figure 5.9: UMAP and proposed algorithm on Spambased dataset

(a) UMAP **(b)** Proposed **(c)** UMAP **(d)** Proposed

Figure 5.10: UMAP and proposed algorithm on Air Quality dataset



(a) UMAP **(b)** Proposed **(c)** UMAP **(d)** Proposed

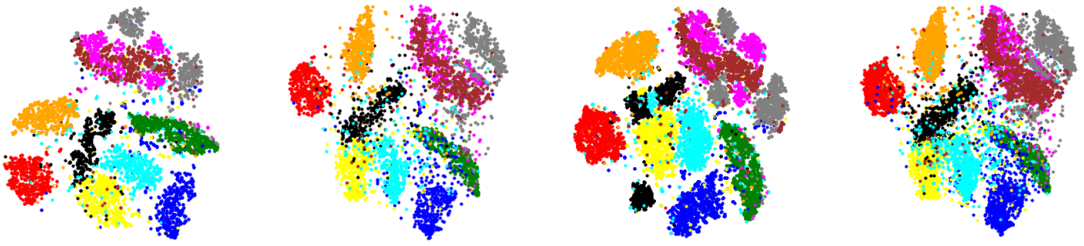
Figure 5.11: UMAP and proposed algorithm on Survival dataset

(a) UMAP **(b)** Proposed **(c)** UMAP **(d)** Proposed

Figure 5.12: UMAP and proposed algorithm on IRIS dataset

We can observe in figures 5.5b and 5.5d that few data points of a specific color are more scattered than the original projections shown in figures 5.5a and 5.5c respectively. This depicts that proposed algorithm introduced some error while projecting the out-of-sample data points.
We can find the same observation for figures 5.6 and 5.11.
In figure 5.9, we can observe a mismatch between projection of UMAP and the proposed algorithm as the proposed algorithm is preserving the training knowledge while projecting out-of-sample data points, and where as projection of UMAP varies with every new samples.

Comparison of experiment results between t-SNE and proposed algorithm In this sub section, figures of experimental result of t-SNE and proposed algorithm are shown. E.g., in the figure 5.13, figure 5.13a represents the projection of t-SNE for out-of-sample data points for MNIST dataset, figure 5.13b represents the projection of the proposed algorithm for the out-of-sample data points, figure 5.13c which is denser than 5.13a, represents the projection of t-SNE for extended out-of-sample data points and figure 5.13d which is denser than 5.13b, represents the projection of the proposed algorithm for extended out-of-sample data points.

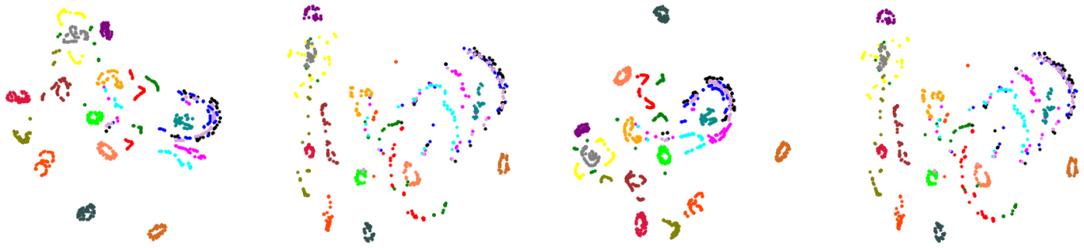


(a) t-SNE **(b)** Proposed **(c)** t-SNE **(d)** Proposed

Figure 5.13: t-SNE and proposed algorithm on MNIST dataset

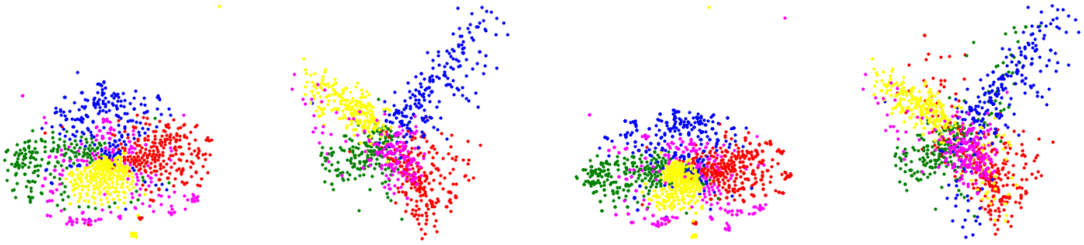
(a) t-SNE **(b)** Proposed **(c)** t-SNE **(d)** Proposed

Figure 5.14: t-SNE and proposed algorithm on Fashion-MNIST dataset



(a) t-SNE **(b)** Proposed **(c)** t-SNE **(d)** Proposed

Figure 5.15: t-SNE and proposed algorithm on Coil20 dataset



(a) t-SNE **(b)** Proposed **(c)** t-SNE **(d)** Proposed

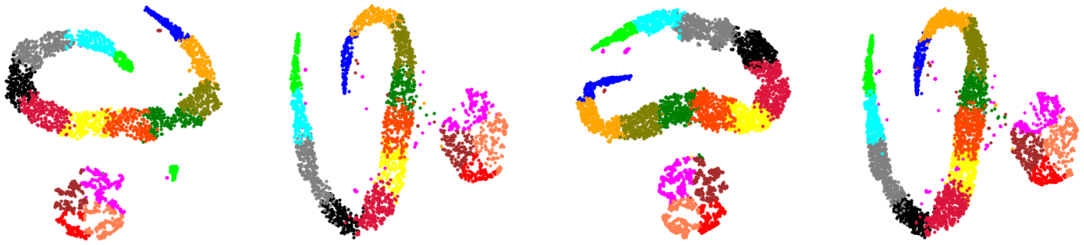
Figure 5.16: t-SNE and proposed algorithm on BBC-News dataset

(a) t-SNE **(b)** Proposed **(c)** t-SNE **(d)** Proposed

Figure 5.17: t-SNE and proposed algorithm on Spambase dataset

(a) t-SNE **(b)** Proposed **(c)** t-SNE **(d)** Proposed

Figure 5.18: t-SNE and proposed algorithm on Air Quality dataset



(a) t-SNE **(b)** Proposed **(c)** t-SNE **(d)** Proposed

Figure 5.19: t-SNE and proposed algorithm on Survival dataset

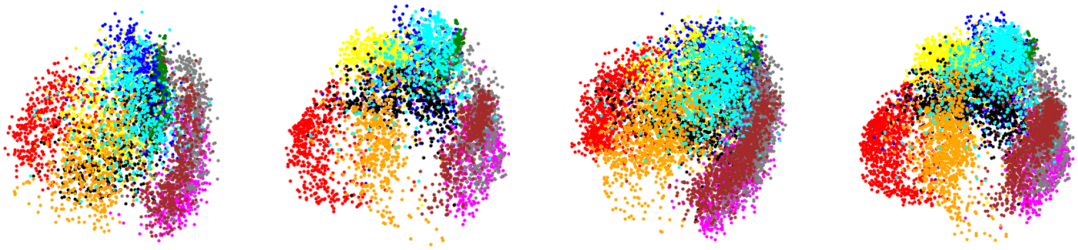
(a) t-SNE **(b)** Proposed **(c)** t-SNE **(d)** Proposed

Figure 5.20: t-SNE and proposed algorithm on IRIS dataset.

In figure 5.13b and 5.13d, we can observe that few data points of a specific color are more scattered than the original projections while comparing with the figure 5.13a and 5.13c which also supports the trustworthiness of t-SNE and the proposed algorithm for MNIST dataset from table 5.5 and 5.6 respectively.
In figure 5.19a, we can see that t-SNE loses the connection for green coloured out-of-sample data points where the proposed algorithm projects it intactly in 5.19b.

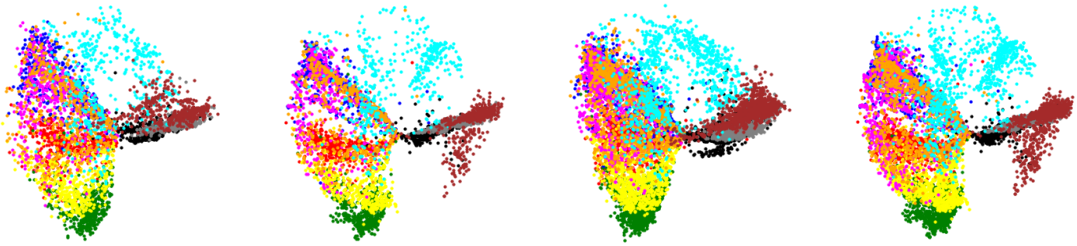
Comparison of experiment result between Isomap and proposed algorithm In this sub section, figures of experimental result of Isomap and the proposed algorithm are shown. E.g., in the figure 5.21, figure 5.21a represents the visualization of out-of-sample data points for MNIST dataset using Isomap, figure 5.21b represents the visualization of out-of-sample data points for MNIST dataset

using the proposed algorithm, figure 5.21c which is denser than 5.21a, represents the visualization of the extended out-of-sample data points for MNIST dataset using Isomap and figure 5.21d which is denser than 5.21b, represents the visualization of the extended out-of-sample data points for MNIST dataset using the proposed algorithm.



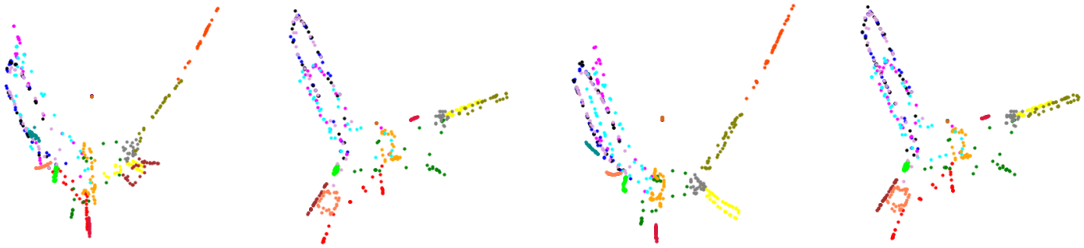
(a) Isomap **(b)** Proposed **(c)** Isomap **(d)** Proposed

Figure 5.21: Isomap and proposed algorithm on MNIST dataset



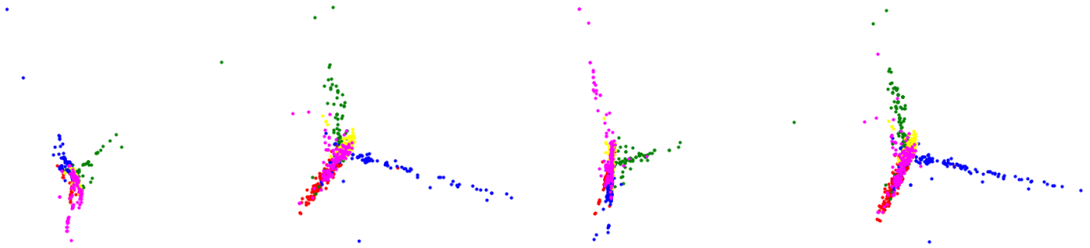
(a) Isomap **(b)** Proposed **(c)** Isomap **(d)** Proposed

Figure 5.22: Isomap and proposed algorithm on Fashion-MNIST dataset



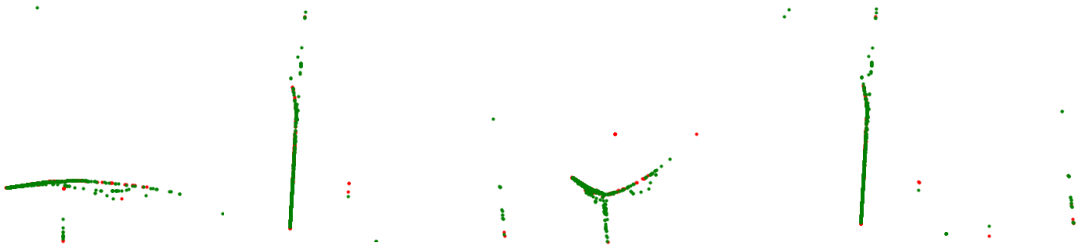
(a) Isomap **(b)** Proposed **(c)** Isomap **(d)** Proposed

Figure 5.23: Isomap and proposed algorithm on Coil20 dataset



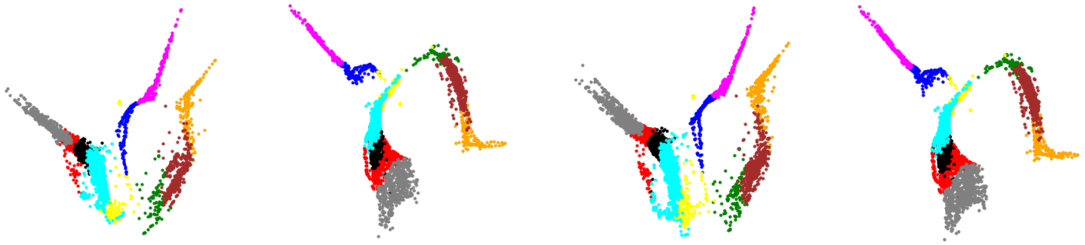
(a) Isomap **(b)** Proposed **(c)** Isomap **(d)** Proposed

Figure 5.24: Isomap and proposed algorithm on BBC-News dataset



(a) Isomap **(b)** Proposed **(c)** Isomap **(d)** Proposed

Figure 5.25: Isomap and proposed algorithm on Spambase dataset

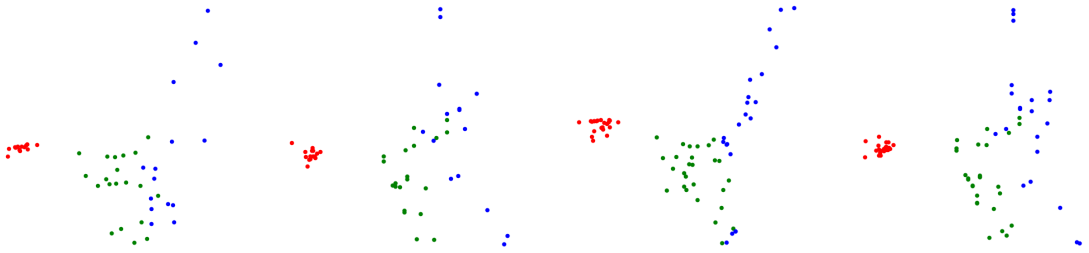


(a) Isomap **(b)** Proposed **(c)** Isomap **(d)** Proposed

Figure 5.26: Isomap and proposed algorithm on Air Quality dataset

(a) Isomap **(b)** Proposed **(c)** Isomap **(d)** Proposed

Figure 5.27: Isomap and proposed algorithm on Survival dataset



(a) Isomap **(b)** Proposed **(c)** Isomap **(d)** Proposed

Figure 5.28: Isomap and proposed algorithm on IRIS dataset.

From above comparison, we can observe that the figures produced by the original DR technique are comparable to the figures produced by the proposed algorithm and it depicts that the projections of the newly proposed algorithm for out-of-sample data points are comparable to the original DR techniques projection. The proposed algorithm is preserving and using the projection knowledge acquired during training. It causes few mismatch in figures while comparing the proposed algorithm to the corresponding DR technique. In some cases, there is a rotation in figures along the axis e.g., in figure 5.19.

5.4.3 Execution time-based comparison

In this subsection, we have shown comparison of execution time of three DR techniques i.e., UMAP, t-SNE and Isomap with proposed algorithm. The complete experiment is performed on jupyter server having AMD EPYC 7401 24-Core Processor with two CPUs and 2,000 max clock speed. MNIST digits dataset is used for this experiment. We have performed this experiment on different sizes of dataset e.g., 10,000 to 60,000 and captured the execution time in seconds for each DR technique and the newly proposed algorithm. Here, our proposed algorithm is trained with 5,000 samples and we have included the time required for the training in time measurements of the proposed algorithm.
For example, from the table 5.9, we can see that UMAP takes 20 seconds to project 10,000 samples, whereas proposed algorithm takes 28 seconds which is the total time required for the training with 5,000 samples and projection of new 10,000 out-of-sample data points.

Table 5.9: Time-based performance table: comparison of execution time (in seconds) of different DR techniques and Proposed algorithm

Dataset Size	UMAP	Proposed	t-SNE	Proposed	Isomap	Proposed
10000	20	28	72	55	127	51
20000	22	64	184	89	566	86
30000	27	112	434	143	1333	137
40000	33	203	553	231	2514	224
50000	40	316	671	343	4171	338
60000	49	470	793	504	6163	494

It can be clearly observed that time-performance of the newly proposed algorithm is better than t-SNE and Isomap. E.g., Isomap requires 6163 seconds, where the newly proposed algorithm requires only 494 seconds to project sixty thousand out-of-sample data points.

From figure 5.29, it can be observed that run-time performance of UMAP is better compare with the newly proposed algorithm as UMAP uses Numba [LPS15] library in python implementation to introduce parallelism in computational task e.g., finding nearest neighbour [NLR+20].

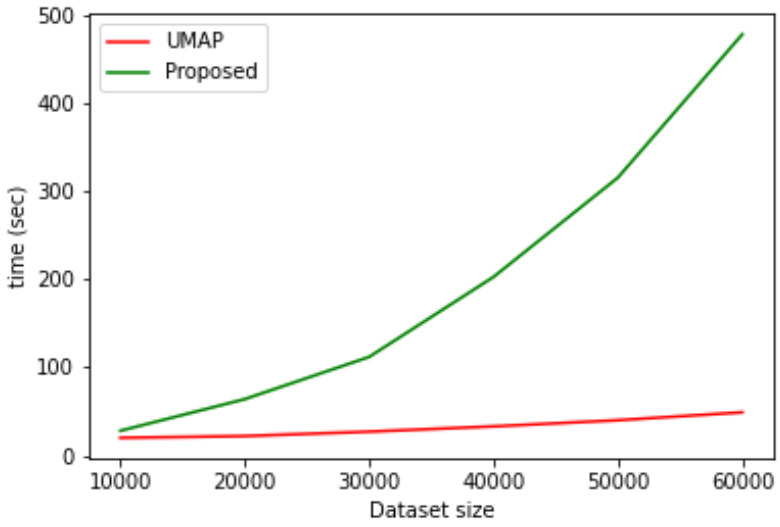


Figure 5.29: Time-performance curve: UMAP vs. Proposed

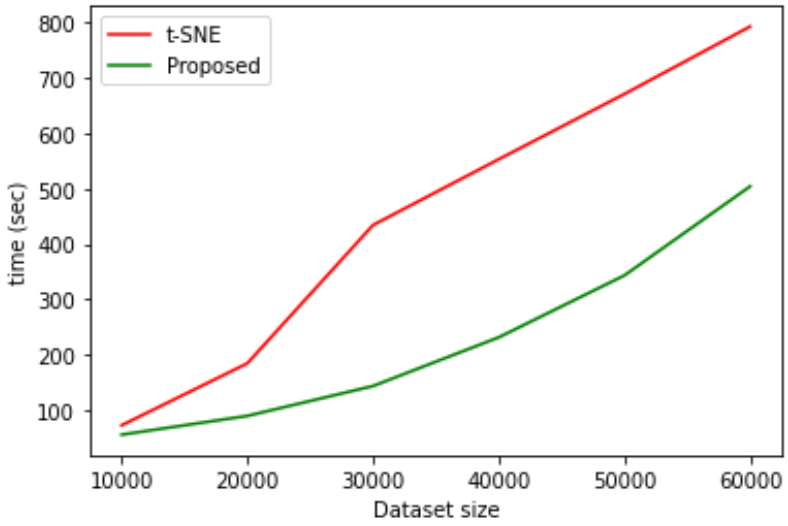


Figure 5.30: Time-performance curve: t-SNE vs. Proposed

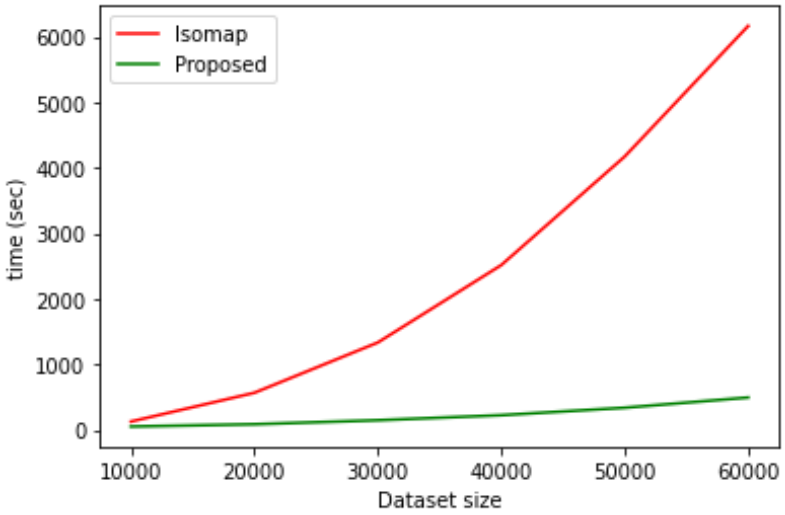
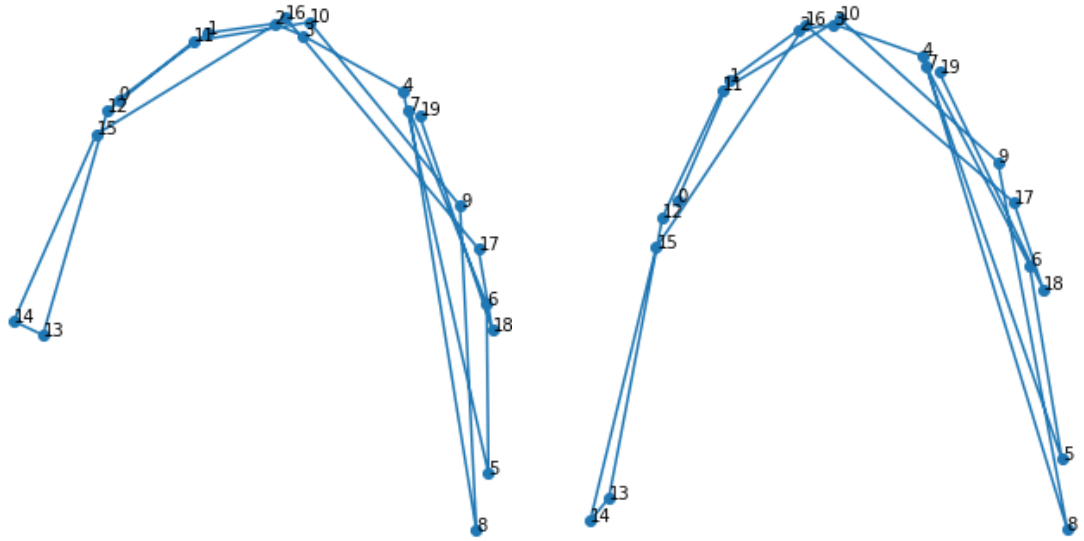


Figure 5.31: Time-performance curve: Isomap vs. Proposed

Above figures show clearly that our proposed algorithm provides better run-time performance compare to t-SNE and Isomap.

5.4.4 Comparison of time-series data projection

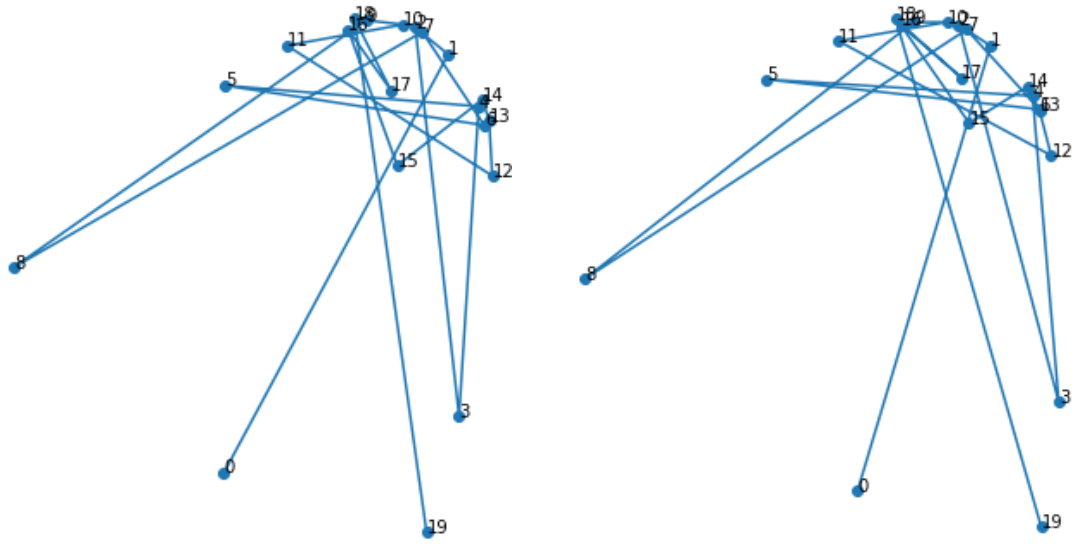
In time-series datasets, data is captured over a specific interval of time. To capture the evolving nature of the time-series data points, MDS is applied to project the high-dimensional data points in low-dimensional space as MDS preserves the global structure of the high-dimensional input dataset [BSH+15].
When we apply the newly proposed algorithm to reproduce the result of MDS, we can see that the newly proposed algorithm is capable to perfectly reproduce the result of MDS. In this experiment, we have used twenty data points to visualize the data points clearly.



(a) MDS **(b)** Proposed

Figure 5.32: MDS and newly proposed algorithm on Air quality dataset

Figure 5.32a and 5.32b show the low-dimensional projection of time-series data after applying MDS and the proposed algorithm respectively. In both figures, we can see points eight and five are close together, and eight and fourteen are far apart from each other.



(a) MDS **(b)** Proposed

Figure 5.33: MDS and newly proposed algorithm on Survival dataset

We have checked the newly proposed algorithm on another time-series dataset. It also shows a promising result. For example, data points five and eleven are close to each other, and five and eight data are far apart from each other in both of the figures 5.33a and 5.33b.

6 Conclusion

In this section, we provide the summary, discussion and future work of the thesis in details.

6.1 Summary

In this thesis a technique is introduced to project and visualize the result of an original dimensionality reduction technique for the out-of-sample data points, as well as modify the projection smoothly when new out-of-sample data points are introduced and visualize the projection. In this technique, we have used nearest neighbours and kernel PCA to map and project the high-dimensional data points in low-dimensional space. Related work is reviewed to provide similar work for projecting the out-of-sample data points and how KPCA can be used to project results of an original DR technique from the existing publications and to describe how these works influence this thesis and how the works differ from the proposed solution. Prerequisites provide basic and important concepts of dimensionality reduction. Eight datasets, three dimensionality reduction techniques and five evaluation metrics are used to evaluate the projection of the newly proposed technique. Statistics from evaluation metrics clearly show that the projections produced by the newly proposed technique are comparable to the projections of the original dimensionality reduction techniques. Apart from quantitative evaluation, we have performed an evaluation of the proposed technique in a graphical way. This shows a high similarity between projections by the new technique and the original dimensionality reduction techniques.

6.2 Discussion

The results from our quantitative evaluation via quality metrics and knowledge from the projection figures of the proposed algorithm provide evidence that the proposed algorithm is robust to project out-of-sample data points, and that is consistent with the initial projection. It produces convincing visualizations for out-of-sample data points.
The proposed technique provides better run-time performance compared to different original dimensionality techniques.
The presented solution loses the projection characteristics of original DR technique for out-of-sample data points. In case of very small dataset, though the projection of out-of-sample data points by the original DR techniques and proposed algorithm are quite similar, but using original DR technique to project data points in low-dimensional space is more meaning-full. The proposed algorithm will fail to project out-of-sample data points properly, when whole or majority of training data points are outliers as projection of out-of-sample data points uses knowledge acquired during training. It lacks to beat the run time performance of UMAP because UMAP uses parallelism in calculation.

Despite of these drawbacks, the proposed algorithm is worthful to use for projecting the out-of-sample data points in low-dimensional space and update the projection when new out-of-sample data points are introduced iteratively.

6.3 Future Work

In this section, we discuss the possibility to improve the proposed technique in various ways.
An extensive investigation which assesses the evaluation of the proposed algorithm for other dimensionality reduction techniques e.g., Local Linear Embedding (LLE) [(Roweis and Saul, 2000)], Laplacian Eigenmaps (Belkin and Niyogi, 2003), Multidimensional scaling (MDS) could be performed.
A detail evaluation of the proposed algorithm on variety and large number of datasets to check the robustness of the newly proposed algorithm would be worthful.
There is a significant scope for introducing a pre-processing phase for the training datasets to identify the outliers in the training dataset and to ensure that the training dataset well represents the whole dataset, as the out-of-sample projection highly depends on the knowledge acquired during training.
A comprehensive investigation in the field of the kernel function selection to evaluate the behaviour of the proposed algorithm would be worthful.
Develop a python pip package for the proposed solution to easily install and use the proposed solution in order to reproduce and update the out-of-sample projection for different datasets and different dimensionality reduction techniques.
Introduce a computational framework that will use parallelism in the computational task in order to improve the run-time performance as UMAP.

Bibliography

[BPV+03] Y. Bengio, J.-f. Paiement, P. Vincent, O. Delalleau, N. Roux, M. Ouimet. "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering". In: *Advances in neural information processing systems* 16 (2003), pp. 177–184 (cit. on p. 17).

[BSH+15] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, P. Dragicevic. "Time curves: Folding time to visualize patterns of temporal evolution in data". In: *IEEE transactions on visualization and computer graphics* 22.1 (2015), pp. 559–568 (cit. on p. 53).

[De 11] J. De Leeuw. "Shepard diagram". In: (2011) (cit. on p. 36).

[DMP+08] S. De Vito, E. Massera, M. Piga, L. Martinotto, G. Di Francia. "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario". In: *Sensors and Actuators B: Chemical* 129.2 (2008), pp. 750–757 (cit. on p. 34).

[EMK+19] M. Espadoto, R. M. Martins, A. Kerren, N. S. Hirata, A. C. Telea. "Toward a quantitative survey of dimension reduction techniques". In: *IEEE transactions on visualization and computer graphics* 27.3 (2019), pp. 2153–2173 (cit. on pp. 33, 35).

[FH89] E. Fix, J. L. Hodges. "Discriminatory analysis. Nonparametric discrimination: Consistency properties". In: *International Statistical Review/Revue Internationale de Statistique* 57.3 (1989), pp. 238–247 (cit. on p. 29).

[Fis36] R. A. Fisher. "The use of multiple measurements in taxonomic problems". In: *Annals of eugenics* 7.2 (1936), pp. 179–188 (cit. on p. 34).

[Fuk13] K. Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013 (cit. on p. 19).

[GC06] D. Greene, P. Cunningham. "Practical solutions to the problem of diagonal dominance in kernel document clustering". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 377–384 (cit. on p. 33).

[GSH15] A. Gisbrecht, A. Schulz, B. Hammer. "Parametric nonlinear dimensionality reduction using kernel t-SNE". In: *Neurocomputing* 147 (2015), pp. 71–82 (cit. on p. 18).

[HLMS04] J. Ham, D. D. Lee, S. Mika, B. Schölkopf. "A kernel view of the dimensionality reduction of manifolds". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 47 (cit. on p. 17).

[HMW+20] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362 (cit. on p. 32).

[Hot33] H. Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of educational psychology* 24.6 (1933), p. 417 (cit. on p. 19).

[HRFS99] M. Hopkins, E. Reeber, G. Forman, J. Suermondt. *SpamBase dataset. Hewlett-Packard Labs; 1501 Page Mill Rd.; Palo Alto; CA 94304*. 1999 (cit. on p. 34).

[Hun07] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95 (cit. on p. 32).

[JC16] I. T. Jolliffe, J. Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202 (cit. on p. 19).

[Kru64] J. B. Kruskal. "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis". In: *Psychometrika* 29.1 (1964), pp. 1–27 (cit. on pp. 13, 22).

[LCB10] Y. LeCun, C. Cortes, C. Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010) (cit. on p. 33).

[LCS05] H. Li, W. Chen, I.-F. Shen. "Supervised local tangent space alignment for classification". In: *IJCAI*. 2005, pp. 1620–1621 (cit. on p. 17).

[LPS15] S. K. Lam, A. Pitrou, S. Seibert. "Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. LLVM'15". In: (2015) (cit. on p. 52).

[MHM18] L. McInnes, J. Healy, J. Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018) (cit. on p. 32).

[MHM20] L. McInnes, J. Healy, J. Melville. "UMAP: uniform manifold approximation and projection for dimension reduction". In: (2020) (cit. on pp. 19, 24).

[NLR+20] C. J. Nolet, V. Lafargue, E. Raff, T. Nanditale, T. Oates, J. Zedlewski, J. Patterson. "Bringing UMAP Closer to the Speed of Light with GPU Acceleration". In: *arXiv preprint arXiv:2008.00325* (2020) (cit. on p. 52).

[NNM+96] S. A. Nene, S. K. Nayar, H. Murase, et al. "Columbia object image library (coil-100)". In: (1996) (cit. on p. 33).

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 32).

[QC15] T. Qu, Z. Cai. "A fast isomap algorithm based on Fibonacci heap". In: *International Conference in Swarm Intelligence*. Springer. 2015, pp. 225–231 (cit. on p. 23).

[Ras+99] C. E. Rasmussen et al. "The infinite Gaussian mixture model." In: *NIPS*. Vol. 12. Citeseer. 1999, pp. 554–560 (cit. on p. 26).

[RS00] S. T. Roweis, L. K. Saul. “Nonlinear dimensionality reduction by locally linear embedding”. In: *science* 290.5500 (2000), pp. 2323–2326 (cit. on p. 19).

[SB07] M. Svensén, C. M. Bishop. *Pattern recognition and machine learning*. 2007 (cit. on p. 26).

[SSM97] B. Schölkopf, A. Smola, K.-R. Müller. “Kernel principal component analysis”. In: *International conference on artificial neural networks*. Springer. 1997, pp. 583–588 (cit. on p. 20).

[SSM98] B. Schölkopf, A. Smola, K.-R. Müller. “Nonlinear component analysis as a kernel eigenvalue problem”. In: *Neural computation* 10.5 (1998), pp. 1299–1319 (cit. on pp. 19, 20).

[TDL00] J. B. Tenenbaum, V. De Silva, J. C. Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *science* 290.5500 (2000), pp. 2319–2323 (cit. on pp. 19, 22).

[Ten+98] J. B. Tenenbaum et al. “Mapping a manifold of perceptual observations”. In: *Advances in neural information processing systems* 10 (1998), pp. 682–688 (cit. on p. 23).

[Tor58] W. S. Torgerson. “Theory and methods of scaling.” In: (1958) (cit. on pp. 13, 22).

[VGO+20] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272 (cit. on p. 32).

[VH08] L. Van der Maaten, G. Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008) (cit. on pp. 19, 23).

[VK06] J. Venna, S. Kaski. “Visualizing gene interaction graphs with local multidimensional scaling.” In: *ESANN*. Vol. 6. Citeseer. 2006, pp. 557–562 (cit. on p. 35).

[VPV+09] L. Van Der Maaten, E. Postma, J. Van den Herik, et al. “Dimensionality reduction: a comparative”. In: *J Mach Learn Res* 10.66-71 (2009), p. 13 (cit. on p. 19).

[Wil02] C. K. Williams. “On a connection between kernel PCA and metric multidimensional scaling”. In: *Machine Learning* 46.1 (2002), pp. 11–19 (cit. on p. 17).

[XLX17] H. Xie, J. Li, H. Xue. “A survey of dimensionality reduction techniques based on random projection”. In: *arXiv preprint arXiv:1706.04371* (2017) (cit. on p. 19).

[XRV17] H. Xiao, K. Rasul, R. Vollgraf. “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms”. In: *CoRR* abs/1708.07747 (2017) (cit. on p. 33).

[ZK99] D. Zwillinger, S. Kokoska. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999 (cit. on p. 36).

[ZWG+21] H. Zhang, P. Wang, X. Gao, Y. Qi, H. Gao. “Out-of-sample data visualization using bi-kernel t-SNE”. In: *Information Visualization* 20.1 (2021), pp. 20–34 (cit. on pp. 33, 35).

All links were last followed on November 5, 2021.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature