Al-Rafidain Journal of Engineering Sciences

Journal homepage https://rjes.iq/index.php/rjes

ISSN 3005-3153 (Online)

Dimensionality Reduction in Data Mining Application: A Systematic Review

Nahlah M.A.M. Najm[1], Nada Adnan Taher[1], Hazem Noori Abdulrazzak[1], Aya Ayad Hussein[2]

[1]Department of Computer Techniques Engineering, College of Engineering and Technical Engineering, , Al-Rafidain University, Baghdad, Iraq
[2]Department of Artificial Intelligence and Robotics Engineering, Faculty of Engineering, Al-Iraqia University, Baghdad, Iraq

ARTICLE INFO

Article history:
Received 31 March 2026
Revised 31 March 2026,
Accepted 20 April 2026,
Available online 23 April 2026

Keywords:
Dimensionality,
Data Mining, Classification,
ML,
AI.

ABSTRACT

The rapid growth of high-dimensional data introduces redundancy and noise that hinder data mining and machine learning, a challenge commonly known as the curse of dimensionality. Dimensionality Reduction (DR) techniques, including feature selection and feature extraction, alleviate these issues by reducing computational complexity while maintaining informative structures and enhancing predictive accuracy. This review systematically examines classical DR methods such as Principal Component Analysis (PCA) and its variants, alongside recent advances in nonlinear and manifold learning approaches, including Isomap and Locally Linear Embedding (LLE). Emerging probabilistic, soft computing, automated pipeline, and semi-supervised DR techniques are also discussed. Applications across bioinformatics, recommender systems, trajectory mining, and disease diagnosis are reviewed, with comparative analyses highlighting the strengths and limitations of each approach.

1. Introduction

In the modern era of digitalization, rapid expansion of digital data varying from multiple fields, including bioinformatics and sensor data, to e-commerce and social media, it is quite ordinary to encounter datasets with thousands or even tens of thousands of features per observation. Moreover, it brings a potential challenge of dealing with complex phenomena and high dimensionality, which further contribute to increased storage requirements, longer training times, overfitting, and difficulties in data visualization and interpretation.[1- 3] such as:

1. Redundancy and Noise: A lot of features frequently repeat data or don't help us grasp the underlying patterns.
2. Computational Overhead: Processing speeds and real-time applications may be hampered by the substantial computational resources needed for high-dimensional data.
3. The Curse of Dimensionality: Traditional algorithms may perform badly as data becomes sparse as dimensionality rises.

Corresponding author E-mail address: hazem.n@ruc.edu.iq
https://doi.org/10.61268/jnhmdq69

Dimensionality Reduction play a pivotal role in overcoming these challenges by reducing the number of features while preserving the significant information, which subsequently not only makes data analysis more efficient but also improves the accuracy of machine learning models [4]. On the other hand, conventional techniques such as PCA and Factor Analysis (FA) were applied to reduce dimensionality [5]. However, as machine learning has advanced, many DR techniques have been developed, such as probabilistic and soft computing techniques (e.g., copula-based approaches, sparse PCA, and genetic algorithms), nonlinear manifold learning techniques (e.g., Isomap, Locally Linear Embedding, Laplacian Eigen maps), and integrated automated pipelines that combine DR with feature engineering [4], [6] .Furthermore, dimensionality reduction offers a number of benefits, including the removal of redundant and unnecessary patterns from the dataset, which lowers the processing time and memory usage. Reducing the dataset will further improve data quality, increase algorithm efficiency and accuracy, and make pattern construction and analysis easier [7].

The main abbreviations used in this paper are illustrated in Table 1.

Table 1: Abbreviation List

No.	Abbreviations	Definition
1	DR	Dimensionality Reduction
2	FA	Factor Analysis
3	GA	Genetic Algorithms
4	HGAPSO	Hybrid Genetic Algorithm and Particle Swarm Optimization
5	KNN	K-Nearest Neighbors
6	LDA	Linear Discriminant Analysis
7	LLE	Locally Linear Embedding
8	MRMR	Minimum Redundancy Maximum Relevance
9	NMF	Non-negative Matrix Factorization
10	PCA	Principal Component Analysis
11	PCA	Principal Components
12	PSO	Particle Swarm Optimization
13	RFE	Recursive Feature Elimination
14	SFS	Sequential Forward Selection
15	SLR	Systematic Literature Review
16	SPCA	Sparse PCA
17	SPSA	Simultaneous Perturbation Stochastic Approximation
18	SSDR	Semi-Supervised Dimensionality Reduction
19	SVD	Singular Value Decomposition
20	t-SNE	t-distributed Stochastic Neighbor Embedding
21	UMAP	Uniform Manifold Approximation and Projection

This review consolidates findings from seminal and recent studies covering a broad range of dimensionality reduction techniques, including conventional methods such as PCA and Factor Analysis, as well as advanced nonlinear and hybrid approaches, including automated pipelines and semi-supervised methods. Specialized applications in domains such as healthcare and trajectory data mining are also examined. In addition, complementary surveys, including the foundational work of Guyon and Elisseeff on variable and feature selection, are discussed. [8] and the comparative review by van der Maaten et al. [9]

2. Background

2.1. High-Dimensional Data and the Curse of Dimensionality

A large number of variables in a dataset characterized as high-dimensional data. Despite the frequent benefits of having more features, many features are either unnecessary or redundant. This redundancy further contributes in computational complexity and noise, which frequently results in learning algorithms performing worse or contributing in overfitting trend. The term "curse of dimensionality" describes a number of issues that come up while examining data in high-dimensional environments, such as important signals may be obscured by the correlation or duplication of many features. The learning process is made more difficult by irrelevant factors, which increase noise. The cost of computing and storage rises with the number of features. [1, 2] Examples of the curse of dimensionality include the absence of the data points, similarity in differing regions, and the space used in data increases rapidly. To address these issues, non-linear approaches like Isomap and Locally Linear Embedding [10], [11] have been developed which reveal the low-dimensional form hidden in high-dimensional structures.

2.2. Defining Dimensionality Reduction

Dimensionality reduction includes transforming high-dimensional ($X \in R^{n \times D}$) data into lower dimensional ($Y \in R^{n \times d}$) with ($d \ll D$) in order to preserve as much as pertinent information as feasible[8], [12]. Dimensionality Reduction methods are broadly divided into two categories:

1. Feature Selection: Selecting a subset of the original features that are most pertinent to the prediction task is known as feature selection. Filter techniques (such as mutual information and correlation), wrapper techniques (including evolutionary algorithms and particle swarm optimization), and embedding techniques (like LASSO and ridge regression) are a few examples [13].
2. Feature Extraction: Feature Extraction involves, process of creating a new collection of features from the existing feature space, frequently by using nonlinear mappings or linear combinations. PCA and FA are examples of classical techniques; nonlinear techniques such as Isomap, Laplacian Eigen maps, locally linear embedding (LLE), and manifold learning are examples of contemporary techniques [1], [6].

Both the approaches have been studied frequently and analysis shows that both approaches bring its own advantages and trade-offs regarding computational complexity, interpretability and in term of accuracy. Furthermore, a number of hybrid techniques incorporate elements of feature extraction and selection, and current developments include automated Dimensionality Reduction combined with deep learning pipelines. The Role and Impact of Dimensionality Reduction in Modern Data Mining:

An effective Dimensionality Reduction not just important for increasing data visualization interpretability, decreasing storage, computing expenses and improving predictive model performance but also it spans over variety of fields, application includes:

1. Bioinformatics and Gene Expression Analysis: Reducing millions of gene characteristics to a crucial subset that subsequently increases the classification accuracy, which is a goal of bioinformatics and gene expression analysis [3], [8], [13]
2. Image and Signal Processing: Using auto encoders, kernel PCA, and PCA to compress image data for medical imaging or facial identification [14].
3. Big Data Analytics: Applying DR to simplify large-scale datasets in fields like web data and sensor networks [15, 16]
4. Recommender Systems: Singular Value Decomposition (SVD) and latent semantic

indexing are used in recommender systems to provide effective recommendations [17, 18].

5. Trajectory Data Mining: Using several learning approaches for traffic analysis and urban mobility is known as "trajectory datamining" [19, 20].
6. Chronic Disease Diagnosis: Combining feature selection and Dimensionality Reduction to enhance diagnostic models and eliminate duplicate clinical features is known as chronic disease diagnosis [13].

❖ **Impact of Dimensionality Reduction on Data Mining [21]**

Effective Dimensionality Reduction had significance advantages including:

- Speed: Training and inference times signific-ally reduced by reducing size.
- Accuracy: Dimensionality Reduction also enhance classifier and regression model performance by removing noise and duplication.
- Visualization: Improved data visualization and comprehension are made possible by lower-dimensional representations.

3. Machine Learning and Principal Component Analysis:

3.1. Principal Component Analysis

PCA, or principal component analysis, is one of the most popular Dimensionality Reduction methods. It converts a collection of potentially correlated variables into a collection of uncorrelated elements arranged according to the variation they are able to capture [14], [22]. In mathematics, PCA resolves the eigenvalue issue given a zero-mean data matrix X.

$$\sum w = \lambda w \quad (1)$$

Where $\Sigma = \frac{1}{n} X^T X$ [14] **.** The principal components (PCs) are given by si=Xɷi and the cumulative explained by the first components is k

$$\sum_{i}^{k} = 1\ \lambda_i / trace(\Sigma) \quad (2)$$

PCA has been widely used for numerous domains including image recognition, financial forecasting, and intrusion detection [8], [14]. As with the consistent increase in data complexities scholars have defended that PCA by itself may not be sufficient for advanced levels of data intricacy because of the development of Kernel PCA, Sparse PCA, and Probabilistic PCA frameworks [23, 24]

Table 2: Key Characteristics of PCA

Characteristic	Description	Typical Applications
Transformation Type	High-dimensional data with linear mapping	Image recognition, gene expression
Basic Determination	Eigen-decomposition for the matrix of covariance	Financial forecasting, anomaly detection
Variance Preservation	Maximizes the overall variation in the smaller area.	Intrusion detection, signal processing
Applications	Gene expression analysis, intrusion detection, and image processing	Complex, non-Gaussian datasets

3.2. Principal Component Analysis (PCA) for Dimensionality Reduction

Principal Component Analysis (PCA) is one of the most well-known methods for linear dimensionality reduction and has been widely proven and applied to diverse data collection techniques for redundancy reduction, efficient use of computing resources, and data visualization in

lower dimensions [25-27]. The initial features that are correlated are transformed into principal components (PCs), which are newer set of uncorrelated variables that are ranked in the order of the extracted variance from the dataset [28] Each subsequent principal component (PC2, PC3, etc.), which is orthogonal to the previous one, captures a lesser degree of variation compared to the first principal component (PC1) [29].

Due to PCA's efficacy in feature extraction as well as particularly in noise reduction, it has been used in diverse set of domains including text analytics, bioinformatics, and image processing [30] . As with any other data analysis approach, PCA has its own set of limitations. Standardization and normalization are necessary preprocessing techniques to guarantee equivalent contribution of every feature to the analysis especially when dimensions differ in magnitude [33]. An explained variance ratio, often displayed with a scree plot, is then used to determine the ideal value for the main components. Components with values below a threshold level of variance are removed [34, 35].

The following Figure (1) represents the streamline flow of variance distribution among component's along with pre-processing steps. It further shows how PCA simplifies a dataset while retaining the most salient features. Besides making the data easier to understand, these visuals also assist a machine learning pipeline in making insightful decisions. While particularly if we see Fig, it shows the left side of the image which illustrates the preprocessing pipeline that is done prior to PCA, such as normalization and scaling. The corresponding scree plot on the right displays the percentage of variance explained by each principal component (PC1, PC2, PC3...). The first five components reveal the astonishing capability of PCA to minimize dimensionality while keeping a huge amount of data variability, and thus, information intact.

The Steps in PCA will be as:

Step 1: Compute Covariance Matrix

$$\sum = \frac{1}{n} (X - \hat{X})^T (X - \hat{X}) \quad (3)$$

This measures how the variables vary together.

Step 2: Eigen Decomposition

$$\sum V_i = \lambda_i V_i \quad (4)$$

Find the eigenvalues (λ_i) and eigenvectors (v_i) of the covariance matrix.

Step 3: Select Top k Principal Components

Choose the k eigenvectors with the largest eigenvalues these represent the directions with the most variance.

Step 4: Project Data onto Principal Components

$$Z = X W_k \quad (5)$$

Transform the original data X into the new coordinate system defined by the top k principal components W_k.

Fig.1. High-order Overview of the Principal Component Analysis (PCA)

3.3. Extensions of PCA and Techniques

PCA has proven effective in many applications and several extensions has been proposed including; Kernel PCA which is useful for capturing nonlinear interactions because it uses a kernel function to project data into a higher-dimensional space before performing PCA [36, 37]. Subsequently, in order to generate more interpretable components, Sparse PCA (SPCA) incorporates sparsity limitations, which include the use of LASSO. A probabilistic approach that is more effective and resilient to outliers and missing data is offered by probabilistic PCA [36]. These variations enable PCA to be applied to a diverse variety of issues, guaranteeing that the most useful and important features are still recorded even in cases where the data are sparse or nonlinear.

Table 3: Performance Comparison of Techniques

Ref.	Technique	Accuracy (%)	Execution Time (ms)	Dataset Used	Key Strengths
[2]	Non-negative Matrix Factorization (NMF)	86	0.88	Healthcare and bioinformatics datasets	Handles non-negative data well.
[36]	PCA	92.7	0.77	Weather and soil datasets	High accuracy, low execution time.
[38]	K-Nearest Neighbors (KNN)	78.9	0.75	Agricultural datasets	Simple to implement, good for small datasets.

3.4. Critical Review of PCA-based Approaches

Recent studies have shown that PCA performs best when data shows substantial nonlinearities, but it is computationally efficient and effective for linear data. To overcome these problems, hybrid strategies that combine PCA with feature selection or nonlinear mapping have been put forward[13], [36]. One of the recent examples includes applying an employ manifold learning algorithms to capture the underlying structure after initially using PCA to decrease noise [36]. These studies are an example of their effectiveness when combined with other techniques. Hybrid strategies have demonstrated encouraging outcomes, such as using PCA for early noise reduction and then manifold learning. This particular section emphasizes the necessity for techniques that can capture the intrinsic complexity of contemporary datasets while simultaneously being computationally efficient.

3.5. Review for Principal Component Analysis Algorithm:

As the foundational method of dimensionality reduction for many years, PCA offers a linear approach to identifying the most effective directions for projecting high-dimensional data. Many researchers have revisited PCA over time to examine its limitations, how it processes information, and its usefulness. Some studies in our review focus on PCA as one of the baseline techniques, while simultaneously proposing extensions and hybrid methods to address some of its inherent shortcomings. A comprehensive review [39] presents various DR techniques and highlights PCA's main strengths: its simplicity and speed. However, it also becomes clear that when dealing with complex, nonlinear data distributions, the assumption of linearity in PCA is its most significant flaw. Kumar and Abdulazeez [40] provide an in-depth review of PCA, including the mathematical derivation, eigen-decomposition, applications across different fields, and its limitations. Their

findings suggest that while PCA effectively increases variance with low computational cost, it performs poorly in the presence of noise or outliers, or when the data structure is nonlinear. Despite its simplicity and effectiveness, PCA remains a widely used method, although recent research points out several notable drawbacks. Additionally, methods such as Laplacian Eigenmaps [41] and Diffusion Maps [42] build upon the idea of preserving local structures, which PCA often overlooks. While Han Liao et al. [43] demonstrate that sophisticated techniques are necessary for complex applications like intrusion detection in IoT, Velliangiri et al. [44] note that PCA's effectiveness on nonlinear datasets is limited by its linear assumptions. Moreover, K. Thangavel and A. Pethalakshmi [45] review rough set theory approaches, which, when combined with PCA, better handle redundancy and nonlinear patterns, and Hanan E. Abdelkader et al. [45, 46] show how integrating soft computing techniques can enhance PCA performance in noisy environments. Collectively, these findings underscore the importance of hybrid approaches that extend PCA's potential.

To evaluate more advanced techniques, some studies have used PCA as a baseline. As a case, Houari et al.[47] applied a copula-based DR technique and included PCA as a benchmark, noting that while PCA is relatively good in terms of conserving overall variance, it does not necessarily preserve the geometrical structure of the data. Similarly, Radhika and Masood [36] investigated soft computing methods for DR and set the value of non-linear processing stages with PCA as a baseline. Finally, the comparative assessment by van der Maaten et al.[9] offers a more elaborate account and justification of why, after all, PCA has shortcomings, and assesses its value in comparison to non-linear techniques. The rest of the literature suggest that there is no question that PCA is still a requirement for most of the cases because it is efficient and simple to use. At the same time, more complex contemporary applications have led to the creation of newer forms of PCA such as Kernel, Sparse, and probabilistic PCA. Further investigation must be made into the development of hybrid models that merge PCA with nonlinear approaches, thus blending PCA's effectiveness with techniques capable of more accurately representing complex data forms. These results provide a glimpse of PCA's relevance, strengths, and shortcomings as reviewed from several literature, which is consolidated in table below.

Table 4: Review of Principal Component Analysis (PCA) Algorithm

Ref.	Objective/Problem Statement	Key Findings on PCA	Advantages of PCA	Limitations of PCA	Applications
[1]	To review PCA and other DR techniques for feature selection and extraction.	PCA is effective for reducing dimensions while preserving variance in high-dimensional data.	Captures maximum variance, reduces redundancy, and improves computational efficiency.	Struggles with non-linear data; may lose interpretability of transformed features.	General-purpose dimensionality reduction for feature selection and extraction.
[2]	To review feature extraction and selection methods for high-dimensional data analysis.	PCA is effective for linear datasets, capturing maximum variance with minimal information loss.	Reduces noise, is computationally efficient, and captures maximum variance.	Assumes linearity; may lose non-linear relationships.	Healthcare, bioinformatics, and gene expression datasets.
[9]	Comparative review of	PCA remains a	Captures maximum	PCA's simplicity	Agricultural

	linear and nonlinear DR methods	valuable tool for its computational simplicity, despite nonlinear methods often outperforming it in complex scenarios	variance, reduces redundancy, and improves computational efficiency.	comes at the cost of not capturing nonlinear structures	datasets for crop yield prediction.
[36]	To propose PCA and a decision tree for crop yield prediction using dimensionality reduction.	PCA achieved 92.7% accuracy in reducing dimensions for crop yield prediction.	High accuracy, low execution time, and effective for dynamic datasets.	Limited to linear datasets; may not handle non-linear relationships effectively.	Agricultural datasets for crop yield prediction.
[38]	To survey dimensionality reduction schemes for high-dimensional datasets in data mining.	PCA is widely used for linear dimensionality reduction and is effective for preprocessing.	Low computational cost, reduces noise, and simplifies data visualization.	Sensitive to scaling; may not capture non-linear structures.	Bioinformatics, agricultural datasets, and protein expression data.

3.6. Integration of Dimensionality Reduction Techniques in Machine Learning Pipeline Architecture

Creating diverse data processing pipelines which effectively handle large, diverse datasets has been facilitated by recent developments in big data analytics and machine learning [48]. Data collection, feature selection, dimensionality reduction, feature extraction, and simulation-based decision-making are all steps in a typical machine learning pipeline. Starting with data collection from big data and data warehouse systems, the architecture shown in [14], [49] exhibits a complete flow. This is followed by feature selection based on domain-specific factors, including area, subject of matter, product category, and status. To eliminate data duplication and to optimize computational resources usage efficiency, dimensionality reduction, which is frequently accomplished by Principal Component Analysis, or PCA is important [50]. Feature selection can be performed interactively or automatically where factors internal to the systems like the area, subject matter, product category, and status are used. In the process of feature extraction, K-nearest neighbors, Support vector machines, and Factor analysis are useful but not as greatly as the heuristics of order of magnitude. These steps are crucial in achieving robust regression model with dummy variables as they deal with the outlier problem and missing variable problem [51, 52]. These methods guarantee the adequate data quality, then predictive and descriptive models can be developed and put into operation to uncover meaningful patterns and construct forecasting models and play an important role in resolving issues with outliers and missing variables, which can have a big impact on model performance [53- 55]. Descriptive and predictive analytics assure data quality and play a significant role in gaining insight in the model as well as building forecasting models. Predictive modeling makes use of correlation analysis and linear regression in order to determine causality and patterns in the dataset [56, 57]. This pipeline also includes the assessment of squared error, accuracy, likelihood, and cost/utility analysis as metrics within training set preparation and optimization of the fitness function [58]. Furthermore, the design focuses on the integration of novel approaches such as neural networks, gradient descent optimization, and techniques using simulation-based modeling including discrete event simulation, agent-based models, and symbolic artificial intelligence [59, 60]. As this structured approach provide an insight view of the machine learning lifecycle from pre-processing to model deployment [61, 62]

Fig. 2 adapted from [63] amplifies the standard structure of a machine learning pipeline, paying close attention to the role of dimensionality reduction methods. Dimensionality reduction techniques are most suitable for data preprocessing and to reduce feature space with concern to important patterns. This step also helps to enhance model performance, reduce computing cost, and improve learning efficiency. The integration of DR enables the performance of sophisticated data mining and knowledge discovery from high dimensional datasets.

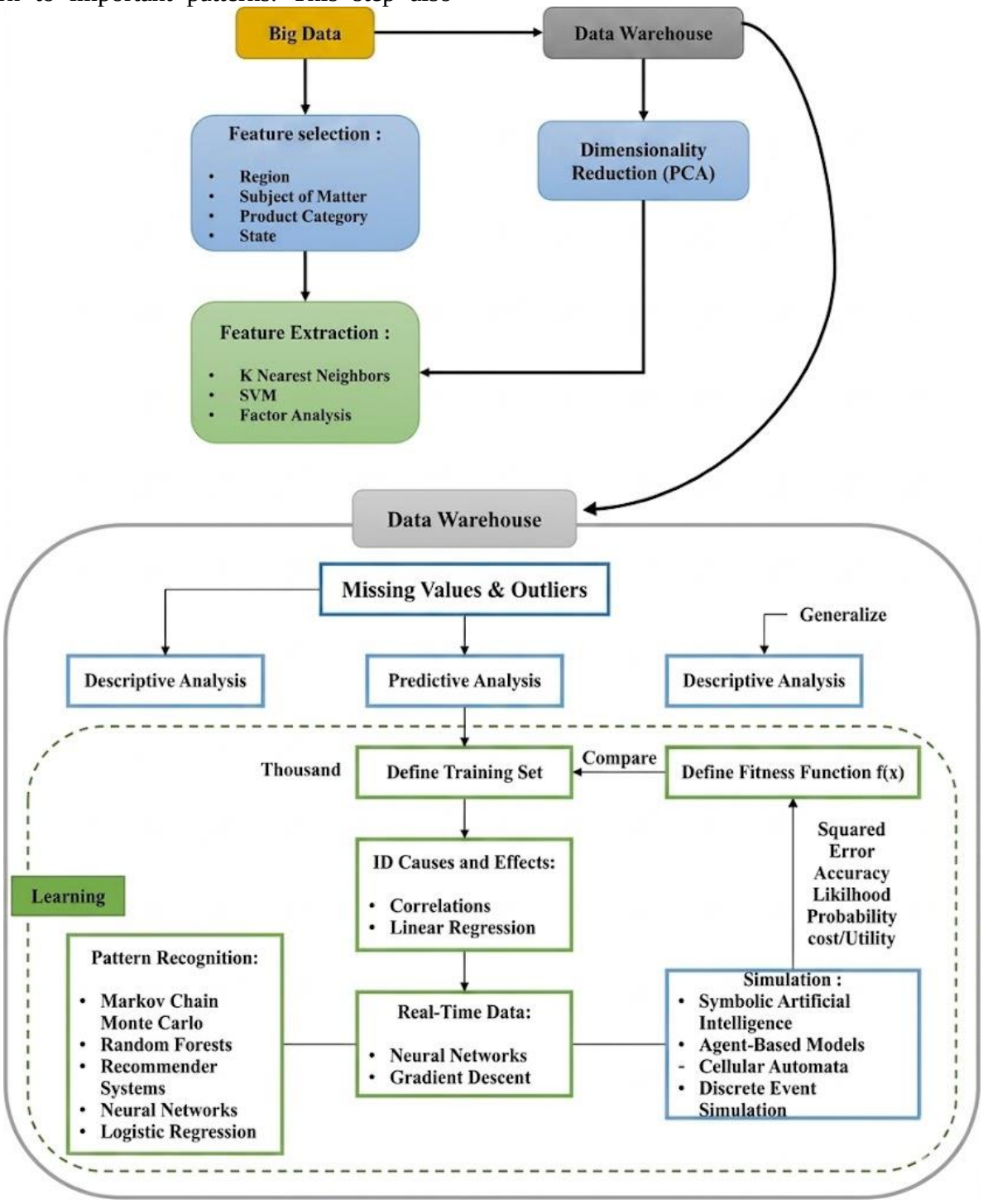

Fig.2. Integration of Dimensionality Reduction Techniques in a Machine Learning Pipeline

4. Feature Selection Methods and Techniques:

The ultimate goal of feature selection is to extract a subset of the most valuable information or features from the original dataset [64]. This process has significantly improved and improved prediction accuracy while also lowering computational costs. Also fig. 3 illustrate the abstract flow of selecting features;

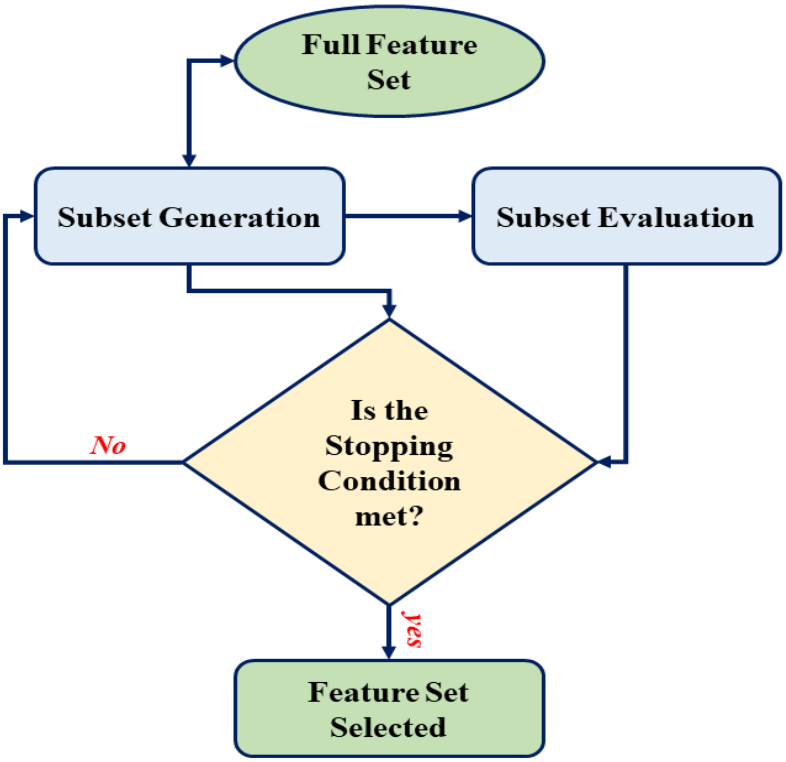

Fig.3. Process of Feature Selection

Flow Containing Feature Selection Technique is illustrated in Fig. 4, which is adapted from [65, 66]:

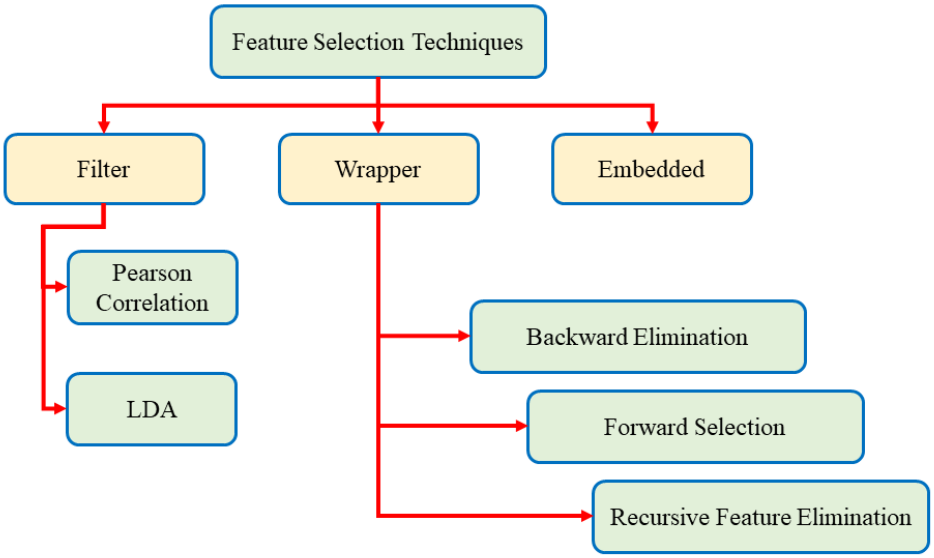

Fig.4. Feature Selection Techniques [65]

4.1. *Filter methods*

Independent of any classifier, filter methods assess each feature according to statistical standards (such as correlation, mutual information, and chi-squared tests). Measuring linear relationships between characteristics and goal variables is done via correlation coefficients. Mutual Information evaluate how dependent one variable is on another. Assessing the importance of features for classification tasks using chi-squared tests. For example, variable ranking and selection based on correlation and information theory are covered in Guyon and Elisseeff's groundbreaking work [8]. Although these techniques are quick and scalable, feature interactions could be overlooked [67, 68].

4.2. Wrapper Methods

Wrapper approaches measure classifier performance (e.g., accuracy, ROC area) by incorporating a particular learning algorithm to assess feature subsets. Numerous research has used methods including Particle Swarm Optimization (PSO), genetic algorithms, and Sequential Forward Selection (SFS) [2]. In SFS, add features iteratively to boost efficiency, while Genetic Algorithms (GA), use evolutionary techniques to investigate feature subset space and PSO further, Optimize feature selection through a swarm-based methodology. Wrappers typically offer superior prediction accuracy despite being computationally more costly [69, 70]

4.3. Embedded Methods

Embedded approaches incorporate Feature selection directly into the model training procedure. Common examples of regularization techniques are ridge regression (L2 penalty) and LASSO (L1 penalty). LASSO Regression penalizes less significant features using L1 regularization. On the other hand, L2 regularization is used in edge regression to lessen the influence of unimportant factors. These techniques nevertheless take feature interactions into account and are typically more effective than wrappers [71].

Table 5: Comparison of Feature Selection Approaches

Approach	Classifier Dependency	Computational Cost	Strengths	Weaknesses
Filter Methods	Independent	Low	Fast, scalable; good for baseline ranking	Might disregard feature interactions
Wrapper Methods	Dependent	High	Tailored to specific models; high accuracy	Computationally expensive
Embedded Methods	Dependent	Moderate	Integrates selection in training; balances cost	May be model-specific

Table 6: Applications of Feature Selection Techniques

Technique	Applications	Key Findings	Advantages	Limitations
MRMR [71]	High-dimensional datasets, redundant features.	Balances relevance and redundancy effectively.	Effective for high-dimensional data; improves classification accuracy.	Computationally expensive for large datasets.
ReliefF [72]	Multi-class datasets, noisy data.	Handles multi-class data; robust to noise.	Simple to implement; effective for noisy data.	Sensitive to feature scaling; may miss redundant features.
Recursive Feature Elimination (**RFE)** [73]	Supervised learning, feature selection.	Recursively removes less important features.	Effective for feature selection; improves model performance.	Computationally intensive; requires a classifier.
Hybrid Genetic Algorithm and Particle Swarm Optimization **(HGAPSO)** [74]	High-dimensional datasets, complex feature interactions.	Combines GA and PSO for optimal feature selection.	Effective for complex feature interactions; improves model accuracy.	High computational cost; complex implementation.
Simultaneous Perturbation Stochastic Approximation **(SPSA)** [75]	High-dimensional datasets, noisy features.	Efficient for high-dimensional data; robust to noise.	Effective for noisy data; improves model performance.	Sensitive to parameter tuning; may converge to local optima.

4.4. Comparative Overview of Feature Selection Methods

This section provides a comparative analysis of feature selections method, these methods include ranking multiple features based on their importance evaluated through chi-square tests, mutual information, and correlation coefficients, as illustrated in Fig.5 Adapted from [35]

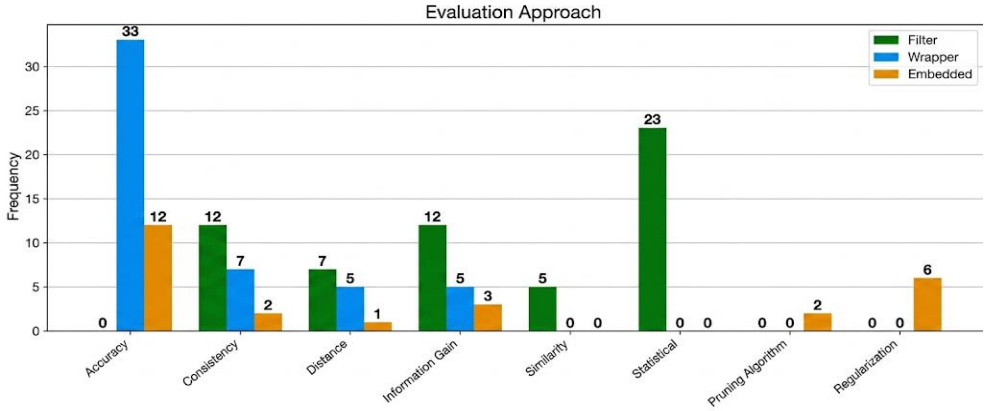

Fig.5. Evaluation Approaches [35]

Fig.6, adapted from [35] offers a comprehensive overview of the trade-off between the computational cost and the effectiveness of the prediction by grouping methods into supervised and unsupervised classes. Approaches that rely on labeled data are referred to as supervised while those that do not are termed unsupervised, emphasizing the trade-offs between computational efficiency and predictive performance.

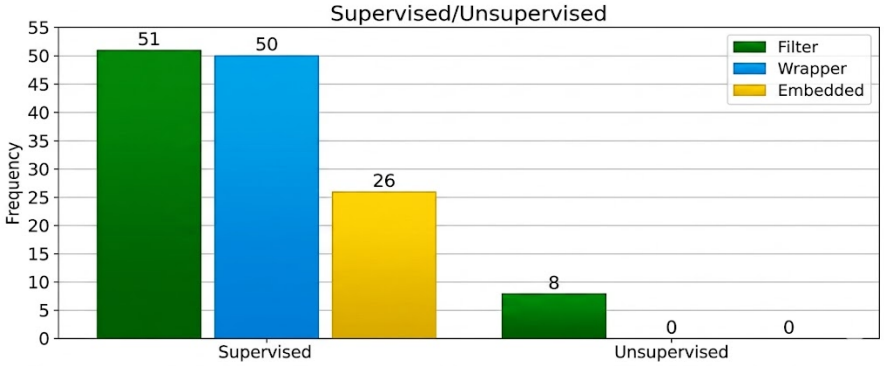

Fig.6. Supervised vs. Unsupervised Methods [35]

The techniques applied to retrieve the most appropriate feature subsets are illustrated in Fig. 7 Adapted from [35]. These subsets are created using exhaustive, heuristic, and metaheuristic approaches.

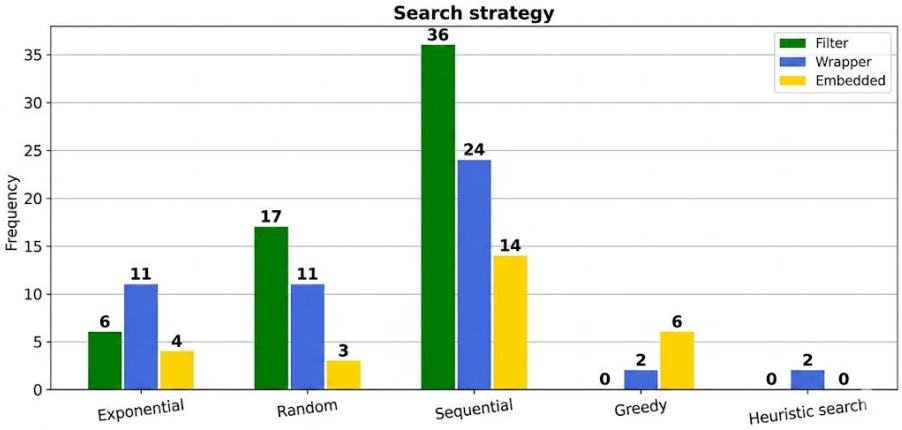

Fig.7. Search Strategies [35]

5. Feature Extraction Techniques

Feature Extractions convert original high-dimensional features into a new collection of variables, also known as latent features, which ideally capture the fundamental structure of the data [1] **.** Below Fig. 8 provides an overview of the classification of feature extraction techniques.

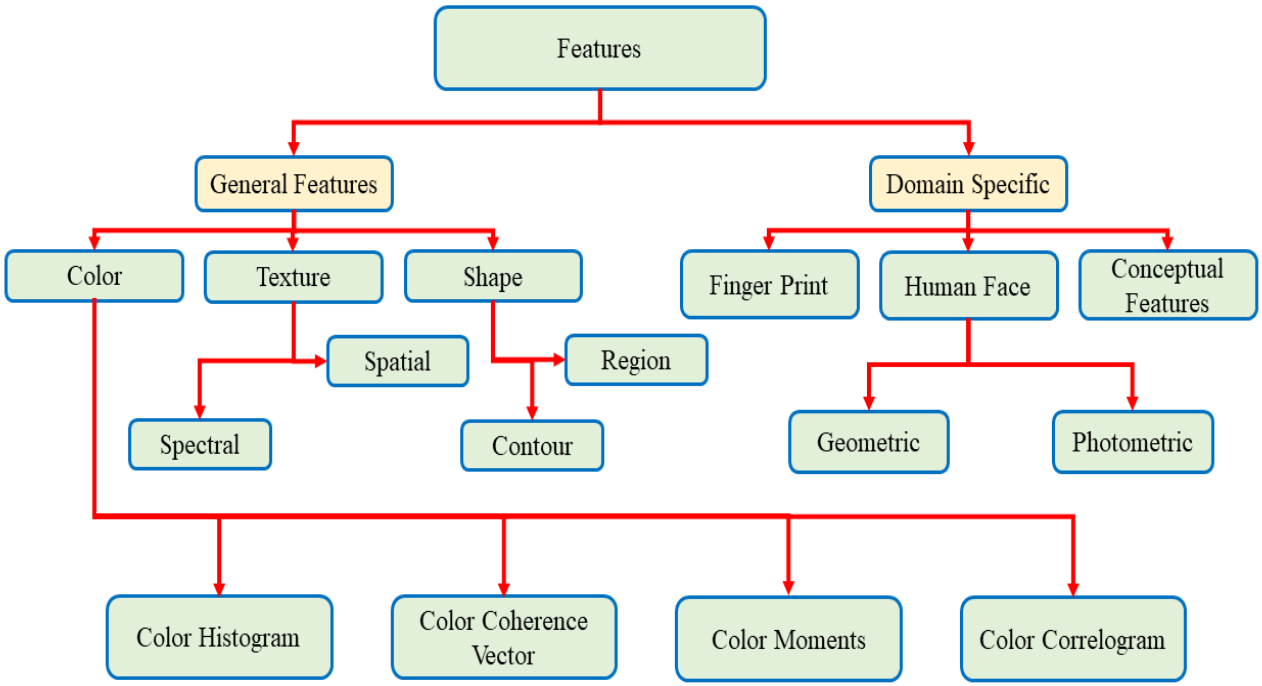

Fig.8. Overview of Feature Extraction Techniques [76]

5.1. Linear Methods

The most common linear technique is PCA. Another technique that makes the assumption that underlying latent components account for the observed correlations between variables is factor analysis (FA). When the objective is to reveal hidden structures in data, like in gene expression analysis or psychological testing, FA is especially helpful [8].

The latest approaches are particularly based on extraction by conventional and neural networks, which termed as autoencoders and hybrids. Always wonderful and very quiet with figures from Kingma and Welling [77]

and Vincent et al. [78] particularly demonstrate how autoencoder designs transform the representation of data, inputted in high dimensions, into lower dimensional depiction.

5.2. Nonlinear Methods

Non-Linear Feature Extraction Method play a vital role for datasets that don't follow a linear connection. Among the widely used methods are: Isomap, which preserves geodesic distances along a manifold to capture the intrinsic geometry of data. Locally Linear Embedding (LLE), This method produces a low-dimensional representation by preserving local neighborhood relationships. Laplacian Eigenmaps and Diffusion Maps, Emphasis on using graph-based representations to preserve local geometry [79]. Neural networks that learn compressed representations without supervision are known as autoencoders.

5.3. Soft Computing and Hybrid Methods

Soft computing methods (such as PSO and evolutionary algorithms) have been used in recent developments for Dimensionality Reduction. In order to provide "low loss" representations with the least amount of computing time, these techniques frequently integrate feature extraction and feature selection inside an optimization framework [36]. According to Houari et al. [47] copula-based techniques use LU-decomposition to eliminate duplicate features and model dependencies probabilistically. Copula-Based Methods, often used in conjunction with LU-decomposition, these methods use probabilistic models to capture multivariate dependencies and remove superfluous features.

Table7: Summary of Feature Extraction Methods

Method Type	Key Idea	Suitable for	References
PCA	Linear projection maximizing variance	Linear data	[8], [14]
Factor Analysis	Modeling latent factors	Psychological, bio data	[8]
Isomap	Preserving geodesic distances	Manifold-structured data	[2]
Autoencoders	Neural network-based encoding	Complex nonlinear data	[36]
Copula-based Methods	Modeling multivariate dependencies	High-dimensional, redundant data	[47]

Table 8: Applications of Feature Extraction Techniques

Technique	Applications	Key Findings	Advantages	Limitations
PCA	Healthcare,	Reduces dimensions	Computationally	Assumes linearity; may

	bioinformatics, agriculture, gene expression datasets.	while preserving variance; effective for linear datasets.	efficient; captures maximum variance.	lose non-linear relationships.
Linear Discriminant Analysis (LDA)	Classification tasks, supervised learning, and face recognition.	Maximizes class separability; good for labelled datasets.	Effective for supervised learning; improves classification accuracy.	Requires labeled data; sensitive to outliers.
NMF	Image processing, text mining, pattern recognition.	Handles non-negative data; useful for image and text data.	Preserves non-negativity; good for interpretability.	Sensitive to initialization; computationally expensive.
Isomap	Gene expression, protein data, non-linear manifolds.	Preserves geodesic distances; good for non-linear datasets.	Captures non-linear structures; effective for complex datasets.	Computationally intensive; sensitive to noise.
LLE	Non-linear datasets, local structure preservation.	Preserves local relationships; good for non-linear data.	Captures local structures; effective for small datasets.	Struggles with sparse data; computationally expensive.
t-distributed Stochastic Neighbor Embedding (t-SNE)	Visualization of high-dimensional data, clustering.	Excellent for visualization; preserves local structures.	Effective for data visualization; good for clustering.	Computationally expensive; not suitable for high-dimensional data reduction.

6. Different Techniques of Dimensionality Reduction

Several Dimensionality Reduction approaches discussed in the literature are integrated in this part and contrasted according to their application fields, computing efficiency, and underlying methodology. Moreover, for instance, t-SNE [34] and Uniform Manifold Approximation and Projection (UMAP) [80] serve an important role in data visualization as they provide more appealing visuals, particularly in two-dimensional projections of multidimensional data. This is especially important in areas of bioinformatics and urban traffic analysis.

6.1. Linear Dimensionality Reduction Techniques

In the linear method Data are assumed to lie on or close to a linear subspace by linear algorithms, PCA and FA are the most widely used methods. Although they might not be able to capture complex nonlinear interactions, they are extensively utilized due to their simplicity and ease of comprehension [81].

6.2. Nonlinear Dimensionality Reduction Techniques

Nonlinear DR techniques include a variety of learning strategies that can reveal intricate patterns in data, such as diffusion maps, Laplacian Eigen maps, LLE, and Isomap. Isomap preserves geodesic distances along a manifold to capture the intrinsic geometry of data. LLE, this method produces a low-dimensional representation by preserving local neighbourhood relationships. Laplacian Eigen maps and Diffusion Maps, Emphasis on using graph-based representations to preserve local geometry. Although these methods' computational complexity and sensitivity to sample density continue to be issues, they have demonstrated greater performance on synthetic datasets with nonlinear patterns [82].

6.3. Hybrid and Automated Methods

Dimensionality Reduction has recently been included into frameworks for soft computing and automated data processing. For instance, in order to automatically optimize feature engineering, Auto-ML pipelines now include

DR as a preprocessing step [83]. In order to balance prediction accuracy and computational cost, hybrid approaches that combine wrapper and filter techniques have also been developed [2].

6.4. Domain-Specific Applications

Dimensionality Reduction methods have been widely used for various domains, such as Big Data Analytics. Large datasets are preprocessed using methods like PCA and LDA to ensure effective classification [84]. Recommender Systems, to provide real-time suggestions, Singular Value Decomposition (SVD) and latent semantic indexing minimize user-item matrices [17]. Trajectory Data Mining, Urban traffic study uses manifold learning to capture spatiotemporal patterns [19]. Diagnosis of Chronic Disease, Dimensionality Reduction and feature selection improve the predictive capabilities of diagnostic models in the medical field [13]. Semi-Supervised Learning, for robust Dimensionality Reduction, SSDR methods integrate paired constraints with both labeled and unlabeled data.

Table 9: Comparative Overview of DR Techniques by Application

Application Domain	Common Techniques	Key Benefits	References
Big Data Analytics	PCA, LDA, SVD	Reduced training time, noise reduction	[15], [85]
Recommender Systems	SVD, Latent Semantic Indexing	Improved scalability, faster response	[17] , [33]
Trajectory Data Mining	Isomap, LLE, Laplacian Eigenmaps	Preserves spatio-temporal structure	[19]
Chronic Disease Diagnosis	Wrapper feature selection, DR	Enhanced diagnostic accuracy	[13]
Semi-Supervised DR	Semi-Supervised Dimensionality Reduction (SSDR)	Utilizes unlabeled data and constraints	[86]

7. Methodology

We systematically examined the results from multiple benchmark and recent studies on dimensionality reduction and put together a report. The current study conducts a systematic literature review (SLR) to analyze the current state of the art on dimensionality reduction techniques within data mining while ensuring accuracy and repeatability [87, 88].A combination of appropriate keywords “dimensionality reduction”, “feature extraction”, “feature selection”, “high-dimensional data”, “data mining”, and “PCA” were utilized to conduct a search on multiple digital libraries such as Google Scholar, Elsevier, IEEE Xplore, ACM, ScienceDirect, and SpringerLink [89] . Our strategy began from reformulating the primary study issue, and that is to understand the evolution and benefits as well as the shortcomings and applications of various dimensionality reduction techniques in contemporary data mining and machine learning. As a matter of preventing selection bias, the PRISMA method was used at the identification, screening, eligibility, and inclusion stages of the systematic review process [90, 91]. The literature was then analyzed in depth on the basis of their techniques, algorithmic approaches, data sources, dimensionality, assessment metrics and fusion with machine learning pipelines [92] and [93]. The review not only captures the state of the art but goes further in illustrating the intertwining of

methods and the comparative analysis within the emerging challenges in the area of deep dimensionality reduction, making it a useful reference for researchers and practitioners [94]. We utilized specific keywords such as "PCA," "feature selection," "manifold learning," and "automated dimensionality reduction" and conducted searches across multiple academic databases to identify relevant articles. After the relevant literature was sourced, it was divided into three major categories: non-linear methods such as Isomap, Locally Linear Embedded, and Laplacian Eigen maps, hybrid or automated methods that substitute advanced proprietary software, and classical linear methods, which include Principal components analysis and Factor Analysis. Each bucket was assigned a set of key metrics, such as variance retention, speed, and applicability of other fields, and comparison matrices were created to demonstrate the similarities and differences of the methods used. Where applicable, we conducted a complete synthesis of quantitative aspects with qualitative analysis. We studied the balance of accuracy and computing cost, checked claimed performance metrics, and discovered issues like sensitivity to noise and linear assumption constraints. These findings helped us in forming the aggregate evaluation and provided a clear explanation of how and what the evolution of multi-dimensional scaling describable strategies has encountered. The need to deliver a comprehensive yet concise account justified our methodological choices. Because the synthesis is based on our self-contained review, we intentionally translated the methodology section with the least number of direct citations. This approach has enabled us to identify shortcomings in the prior research and analyze the so-called advanced methods comprehensively and critically. Our examination suggests that subsequent studies ought to focus on the development of truly non-linear approaches that are both scalable and efficient for complex, high-dimensional data. In addition, applying these approaches to automated processes could enhance their value further. For seeing approaches to research suggests that with modern integrated techniques, diaphragm leveling can be approached both theoretically and practically with greater emphasis on performance and strategy.

8. Conclusion and Future Scope

Dimensionality reduction has become a fundamental approach for managing high-dimensional data in data mining and machine learning. Evidence from the reviewed literature indicates that both classical and modern techniques linear, nonlinear, hybrid, and semi-supervised play a critical role in improving computational efficiency and predictive performance. Linear methods such as PCA and Factor Analysis remain attractive due to their simplicity and scalability, whereas nonlinear manifold-based techniques are more effective in capturing complex data structures. Feature selection enables rapid dimensionality reduction by removing redundant attributes, while feature extraction uncovers latent representations. Recent studies emphasize automated and scalable DR frameworks integrating machine learning. Future research should focus on interpretable and scalable nonlinear methods, as well as hybrid frameworks that combine feature selection, advanced extraction, and deep or soft computing models to address real-world applications.

References

[1] R. R. Zebari, A. M. Abdulazeez, D. Q. Zeebaree, D. A. Zebari, and J. N. Saeed, “A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction,” *Journal of Applied Science and Technology Trends*, pp. 56–70, 2020.

[2] P. Ray, S. S. Reddy, and T. Banerjee, “Various dimension reduction techniques for high dimensional data analysis: A review,” 2021.

[3] C. O. S. Sorzano, J. Vargas, and A. Pascual-Montano, “A survey of dimensionality reduction techniques,” *arXiv preprint*, arXiv:1403.2877, 2014.

[4] D. A. Zebari *et al*., “A comprehensive review of dimensionality reduction techniques for efficient computation,” *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[5] H. Xie, J. Li, and H. Xue, “A survey of dimensionality reduction techniques based on random projection,” *arXiv preprint*, arXiv:1706.04371, 2017.

[6] W. Jia, M. Sun, J. Lian, and S. Hou, “Feature dimensionality reduction: A review,” *Complex & Intelligent Systems*, vol. 8, pp. 2663–2693, 2022.

[7] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: A review," *Complex & Intelligent Systems*, vol. 8, pp. 2663–2693, 2022.
[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1150–1183, 2003.
[9] L. van der Maaten, E. Postma, and H. J. van der Herik, "Dimensionality reduction: A comparative review," *Journal of Machine Learning Research*, vol. 8, pp. 2663–2693, 2007.
[10] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[11] G. T. Reddy *et al*., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, pp. 1–1, 2020.
[12] M. Espadoto *et al*., "Towards a quantitative survey of dimension reduction techniques," *IEEE Transactions on Visualization and Computer Graphics*, 2019.
[13] A. M. Alhassan and W. M. N. W. Zainon, "Review of feature selection, dimensionality reduction and classification for chronic disease diagnosis," in *Proc. IEEE*, 2021.
[14] B. M. S. Hasan and A. M. Abdulazeez, "A review of principal component analysis algorithm for dimensionality reduction," *Journal of Soft Computing and Data Mining*, vol. 2, no. 1, pp. 20–30, 2021.
[15] G. T. Reddy et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, 2020.
[16] V. Svetnik, "Comprehensive review of dimensionality reduction algorithms: challenges, solutions, and future directions," *PeerJ Computer Science*, 2025.
[17] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system – A case study," Univ. of Minnesota, 2000.
[18] S. Ayesha, M. K. Hanif, and R. Talib, "Overview and comparative study of dimensionality reduction techniques for high dimensional data," *Information Fusion*, vol. 59, pp. 44–58, 2020.
[19] Senzhang Wang , Jiannong Cas, Philip S.Yu,"Deep learning for Spatio-temporal Data Mining:A Survey",IEEE Transactions on Knowledg,Vol 34 Issue 8.
[20] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.
[21] M. Espadoto *et al*., "Toward a quantitative survey of dimension reduction techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 3, pp. 2153–2173, Mar. 2021.
[22] S. Marukatat, "Tutorial on PCA and approximate kernel PCA," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5445–5477, 2023.
[23] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer, 2002.
[24] H. Abdi and L. J. Williams, "Principal component analysis," *WIREs Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.
[25] I. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Phil. Trans. R. Soc. A*, vol. 379, 2021.
[26] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, pp. 37–52, 1987.
[27] J. Shlens, "A tutorial on principal component analysis," arXiv:1404.1100, 2014.
[28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed., Springer, 2009.
[29] A. Sharma and K. Paliwal, "Fast principal component analysis using fixed-point algorithm," *Pattern Recognition Letters*, vol. 28, pp. 1151–1155, 2007.
[30] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
[31] D. J. Hand, "Classifier technology and the illusion of progress," *Statistical Science*, vol. 21, pp. 1–14, 2006.
[32] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, pp. 411–430, 2000.
[33] D. L. Donoho and C. Grimes, "Hessian eigenmaps...," *PNAS*, vol. 100, pp. 5591–5596, 2003.
[34] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
[35] D. Theng and K. K. Bhoyar, "Feature selection techniques for machine learning: A survey...," *Knowledge and Information Systems*, vol. 66, pp. 1575–1637, 2024.
[36] A. Radhika and M. S. Masood, "Effective dimensionality reduction by using soft computing method in data mining techniques," *Springer-Verlag*, 2021.
[37] S. Londhe and M. Patil, "Dimensional reduction techniques for huge volume of data," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 10, no. 3, pp. 125–138, 2022.
[38] N. Sharma and K. Saroha, "Study of dimension reduction methodologies in data mining," in *Proc. ICCCA*, pp. 133–136, 2015.
[39] R. R. Zebari *et al*., "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[40] L. Qu and Y. Pei, "A comprehensive review on discriminant analysis for addressing challenges of class-level limitations, small sample size, and robustness," *Processes*, vol. 12, no. 7, p. 1382, 2024.

[41] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[42] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[43] H. Liao et al., "A survey of deep learning technologies for intrusion detection in Internet of Things," *IEEE Access*, 2024.

[44] S. Velliangiri, S. Alagumuthukrishnan, and S. I. T. Joseph, "A review of dimensionality reduction techniques for efficient computation," *Procedia Computer Science*, vol. 165, pp. 104–111, 2019.

[45] K. Thangavel and A. Pethalakshmi, "Dimensionality reduction based on rough set theory: A review," *Applied Soft Computing*, vol. 9, pp. 1–12, 2009.

[46] H. E. Abdelkader, A. G. Gad, A. A. Abohany, and S. E. Sorour, "An efficient data mining technique for assessing satisfaction level with online learning...," *IEEE Access*, vol. 10, 2022.

[47] R. Houari, A. Bounceur, M.-T. Kechadi, A.-K. Tari, and R. Euler, "Dimensionality reduction in data mining: A Copula approach," 2016.

[48] M. M. Ahsan *et al*., "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, p. 52, 2021.

[49] M. Ali, R. Borgo, and M. W. Jones, "Concurrent time-series selections using deep learning and dimension reduction," *Knowledge-Based Systems*, vol. 233, p. 107507, 2021.

[50] I. Jolliffe and J. Cadima, "Principal component analysis...," *Phil. Trans. R. Soc. A*, vol. 379, no. 2191, 2021.

[51] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE TPAMI*, vol. 35, no. 8, pp. 1798–1828, 2013.

[52] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[53] H. Garcia-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: A review," *Neural Computing and Applications*, vol. 19, no. 2, pp. 263–282, 2010.

[54] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *Proc. MIPRO*, 2015, pp. 1200–1205.

[55] V. Borisov *et al*., "Deep neural networks and tabular data: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 6, pp. 1–21, 2022.

[56] D. A. Freedman, *Statistical Models: Theory and Practice*, Cambridge Univ. Press, 2009.

[57] M. M. Ahsan *et al*., "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, p. 52, 2021.

[58] S. Raschka and V. Mirjalili, *Python Machine Learning*, 3rd ed., Packt, 2019.

[59] E. Bonabeau, "Agent-based modeling: Methods and techniques...," *PNAS*, vol. 99, pp. 7280–7287, 2002.

[60] C. M. Macal and M. J. North, "Tutorial on agent-based modeling...," *Journal of Simulation*, vol. 4, no. 3, pp. 151–162, 2010.

[61] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[62] R. Pugliese, S. Regondi, and R. Marini, "Machine learning-based approach: Global trends, research directions, and regulatory standpoints," *Data Science and Management*, vol. 4, pp. 19–29, 2021.

[63] S. Ayesha, M. K. Hanif, and R. Talib, "Overview and comparative study of dimensionality reduction techniques for high dimensional data," *Information Fusion*, vol. 59, pp. 44–58, 2020.

[64] F. Ros and R. Riad, *Feature and Dimensionality Reduction for Clustering with Deep Learning*. 2024.

[65] S. Velliangiri, S. Alagumuthukrishnan, and S. I. Thankumar Joseph, "A review of dimensionality reduction techniques for efficient computation," *Procedia Computer Science*, vol. 165, pp. 104–111, 2019.

[66] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. Springer, 2013.

[67] A. K. Saxena, R. Pandey, and N. K. Singh, "Dimension reduction techniques for optimization," in *Proc. Int. Conf. Computing, Sciences and Communications (ICCSC)*, IEEE, 2024.

[68] C. K. J. Hou and K. Behdinan, "Dimensionality reduction in surrogate modeling: A review," *Data Science and Engineering*, vol. 7, pp. 402–427, 2022.

[69] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: A review," *Complex & Intelligent Systems*, vol. 8, no. 3, pp. 2663–2693, 2022.

[70] D. Patel, A. Saxena, and J. Wang, "A machine learning-based wrapper method for feature selection," *International Journal of Data Warehousing and Mining*, vol. 20, no. 1, Jan. 2024.

[71] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[72] R. J. Urbanowicz *et al*., "Relief-based feature selection: Introduction and review," *Journal of Biomedical Informatics*, vol. 85, pp. 189–203, 2018.

[73] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-relevance and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[74] D. Sharma, P. Gaur, and A. P. Mittal, "Comparative analysis of hybrid GAPSO optimization technique with GA and PSO methods for cost optimization of an off-grid hybrid energy system," *Energy Technology & Policy*, vol. 1, no. 1, 2014.

[75] M. Qurashi, T. Ma, and E. Chaniotakis, "PC–SPSA: Employing dimensionality reduction to limit SPSA search noise in DTA model calibration," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, 2020.

[76] A. O. Salau and S. Jain, "Feature extraction: A survey of the types, techniques, applications," in *Proc. Int. Conf. Signal Processing and Communication (ICSC)*, 2019.

[77] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," arXiv:1312.6114, 2013.

[78] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, pp. 1096–1103, 2008.

[79] S. Lespinats, B. Colnge, and D. Dutykh, *Nonlinear Dimensionality Reduction Techniques: A Data Structure Preservation Approach*. 2022.

[80] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," arXiv:1802.03426, 2018.

[81] L. Viale, A. P. Daga, A. Fasana, and L. Garibaldi, "Dimensionality reduction methods of a clustered dataset for the diagnosis of a SCADA-equipped complex machine," *Machines*, vol. 11, no. 1, p. 36, 2023.

[82] S. Kumari, V. Panchal, and G. Kaur, "Methods for dimension reduction," *International Journal for Technological Research in Engineering*, vol. 9, no. 9, May 2022.

[83] A. Mumuni and F. Mumuni, "Automated data processing and feature engineering for deep learning and big data applications: A survey," *Journal of Information and Intelligence*, 2024.

[84] G. T. Reddy et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, 2020.

[85] R. Rani *et al*., "Big data dimensionality reduction techniques in IoT: Review and taxonomy," *Cluster Computing*, vol. 25, no. 8, pp. 1–23, 2022.

[86] D. Zhang, Z.-H. Zhou, and S. Chen, "Semi-supervised dimensionality reduction," 2020.

[87] B. A. Kitchenham and S. Charters, *Guidelines for Performing Systematic Literature Reviews in Software Engineering*, EBSE-2007-01, 2007.

[88] M. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and P. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, 2007.

[89] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Springer, 1998.

[90] D. Moher *et al*., "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement," *PLoS Medicine*, vol. 6, no. 7, p. e1000097, 2009.

[91] P. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. Wiley-Blackwell, 2006.

[92] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint*, arXiv:1404.1100, 2014.

[93] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. Springer, 2007.

[94] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1–4, pp. 131–156, 1997.