International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Comparative Analysis of Dimension Reduction Techniques for High-Dimensional Classification Tasks

Fardeen NB, Sameer NB

Submitted: 05/08/2022 **Revised:** 17/11/2022 **Accepted:** 02/12/2022

Abstract
As machine learning datasets continue to grow in dimensionality, efficient dimension reduction techniques have become essential for both computational efficiency and model performance. This study presents a comprehensive evaluation of various dimension reduction methods—Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP)—for preprocessing high-dimensional data prior to classification. We evaluate their impact on the performance of three widely used classification algorithms: Random Forests, Support Vector Machines, and Neural Networks. Experiments conducted on benchmark datasets (MNIST and Digits) reveal that while no dimension reduction yields the highest overall accuracy (95.31%), specialized techniques can offer significant computational advantages with minimal performance degradation. Our analysis provides empirical evidence that t-SNE offers an optimal balance between classification performance and training efficiency, particularly for support vector machines. We further demonstrate that dimension reduction techniques exhibit dataset-dependent effectiveness, suggesting the need for adaptive selection strategies based on data characteristics. This work provides valuable insights for practitioners seeking to optimize machine learning pipelines for high-dimensional classification tasks.

Introduction

High-dimensional data presents significant challenges in machine learning, a phenomenon often referred to as the "curse of dimensionality" . As the number of dimensions increases, the volume of the space increases exponentially, causing data to become sparse and diminishing the effectiveness of distance-based methods . Furthermore, high-dimensional data often contains redundant or irrelevant features that can degrade model performance and increase computational requirements .

Dimension reduction techniques aim to address these challenges by transforming data into a lower-dimensional space while preserving important structures and relationships . These methods can be broadly categorized into linear approaches like Principal Component Analysis (PCA) and nonlinear approaches such as t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) .

While dimension reduction is widely employed in practice, there remains a need for systematic evaluation of how these techniques impact the performance of different classification algorithms across diverse datasets . Understanding these interactions is crucial for designing effective machine learning pipelines for high-dimensional data analysis.

This paper makes the following contributions:

- A comprehensive evaluation of three prominent dimension reduction techniques (PCA, t-SNE, and UMAP) in conjunction with three widely used classification algorithms (Random Forest, Support Vector Machines, and Neural Networks)
- Quantitative analysis of the trade-offs between classification performance (accuracy, precision, recall, and F1 score) and computational efficiency (training and inference time)
- Empirical evidence for the dataset-dependent effectiveness of dimension reduction techniques, with implications for adaptive selection strategies
- Practical guidelines for selecting appropriate dimension reduction methods based on specific requirements and constraints

The rest of the paper is organized as follows: Section 2 discusses related work in dimension reduction and classification. Section 3 describes the mathematical formulations of the techniques and our experimental methodology. Section 4 presents our experimental results and analysis. Section 5 discusses the implications of our findings, and Section 6 concludes the paper with recommendations for practitioners.

Related Work

Dimension Reduction Techniques

Principal Component Analysis (PCA), introduced by and later developed by , has been a cornerstone technique for linear dimension reduction. PCA transforms data into a new coordinate system where the greatest variance lies along the first principal component, the second greatest variance along the second component, and so on. provided a comprehensive treatise on PCA, discussing its applications and limitations. further demonstrated

PCA's versatility across various domains, while situated PCA within a probabilistic framework.

For complex, nonlinear data structures, techniques like Multidimensional Scaling (MDS) and Isomap were developed to capture intrinsic geometric structures. introduced Locally Linear Embedding (LLE), which preserves neighborhood relationships while projecting to lower dimensions. proposed Laplacian Eigenmaps, which use graph theory to preserve local proximity.

A significant advancement came with t-Distributed Stochastic Neighbor Embedding (t-SNE) by , which excels at visualizing high-dimensional data by modeling pairwise similarities as conditional probabilities. later improved t-SNE's computational efficiency with the Barnes-Hut approximation. More recently, introduced Uniform Manifold Approximation and Projection (UMAP), which uses concepts from manifold theory and topological data analysis to create embeddings that better preserve both local and global structures.

Comparative studies of dimension reduction techniques include , who provided a survey of linear and nonlinear methods, and , who benchmarked techniques across various datasets. specifically focused on linear dimension reduction techniques, while compared nonlinear methods for visualization.

Classification Algorithms

Classification algorithms have been extensively studied in machine learning literature. Decision Trees, formalized by and , provide interpretable models but often suffer from overfitting. Random Forests, introduced by , address this limitation by combining multiple decision trees trained on bootstrapped samples with random feature selection, significantly improving generalization performance.

Support Vector Machines (SVMs), proposed by , create optimal separating hyperplanes between classes in a high-dimensional feature space. provided a comprehensive introduction to SVMs, while extended the approach through kernel methods. discussed the theoretical foundations of SVMs within statistical learning theory.

Neural Networks have undergone a renaissance with the development of efficient training algorithms and increased computational resources. introduced backpropagation, a crucial algorithm for training multi-layer networks. demonstrated the effectiveness of convolutional neural networks for image recognition, while provided a comprehensive overview of modern deep learning approaches.

Dimension Reduction in Classification Pipelines

The integration of dimension reduction techniques within classification pipelines has been explored by several researchers. investigated the combination of PCA with K-means clustering, while examined the effectiveness of using LLE for face recognition. proposed a dimensionality reduction method specifically designed to maximize class separation.

studied the effects of dimension reduction on classification performance, noting that nonlinear methods often outperform linear ones for complex datasets. addressed theoretical aspects of feature selection and extraction in pattern recognition, providing early insights into optimal dimensionality.

More specific to our work, empirically demonstrated that using too many features can degrade classifier performance, highlighting the importance of appropriate dimension reduction. specifically examined the effect of PCA preprocessing on various classifiers, while investigated dimension reduction for nearest neighbor classification.

provided a unifying framework for various dimension reduction techniques based on kernel methods, while proposed discriminative dimension reduction specifically designed for classification tasks. offered a survey on supervised dimension reduction techniques for classification problems.

Despite this rich body of literature, systematic comparative studies examining the interplay between modern dimension reduction techniques and classification algorithms remain limited. Our work aims to address this gap by providing a comprehensive evaluation across multiple techniques, classifiers, and performance metrics.

Methodology

Dimension Reduction Techniques

Principal Component Analysis (PCA)

PCA is a linear dimension reduction technique that identifies directions (principal components) along which data varies the most. Mathematically, PCA solves for the eigenvectors of the covariance matrix of the data.

Given a data matrix $\mathbf{X} \in \mathrm{R}^{n \times p}$ containing n samples in p dimensions, PCA first centers the data by subtracting the mean of each feature. The covariance matrix $\mathbf{\Sigma} \in \mathrm{R}^{p \times p}$ is computed as:

$$\mathbf{\Sigma} = \frac{1}{n\text{-}1} \mathbf{X}^T \mathbf{X}$$

PCA then finds the eigenvectors and eigenvalues of $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} \mathbf{w}_i = \lambda_i \mathbf{w}_i$$

where $\mathbf{w}_i$ is the ith eigenvector and λ_i is the corresponding eigenvalue. The objective function for PCA can be formulated as:

$$\max_{\mathbf{w}, \|\mathbf{w}\|=1} \mathbf{w}^T \mathbf{\Sigma} \mathbf{w}$$

The top k eigenvectors (those with the largest eigenvalues) form the projection matrix $\mathbf{W} \in \mathrm{R}^{p \times k}$. The reduced data $\mathbf{X}_{reduced} \in \mathrm{R}^{n \times k}$ is obtained by:

$$\mathbf{X}_{reduced} = \mathbf{X}\mathbf{W}$$

t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a nonlinear dimension reduction technique that models pairwise similarities between points in both high and low dimensional spaces as probability distributions, then minimizes the Kullback-Leibler (KL) divergence between these distributions.

In the high-dimensional space, the similarity between points $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as a conditional probability $p_{j|i}$:

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

where σ_i is set through a binary search to achieve a predefined perplexity value. The joint probability p_{ij} is defined as:

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

In the low-dimensional space, the similarity between corresponding points $\mathbf{y}_i$ and $\mathbf{y}_j$ is modeled using a Student's t-distribution:

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

The objective function is the KL divergence between distributions P and Q:

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

t-SNE minimizes this cost function using gradient descent to find the optimal low-dimensional representation.

Uniform Manifold Approximation and Projection (UMAP)

UMAP is a dimension reduction technique based on manifold learning and topological data analysis. It constructs a high-dimensional graph representation and then optimizes a low-dimensional graph to be as structurally similar as possible.

UMAP first constructs a weighted graph in the high-dimensional space, where edge weights v_{ij} represent the likelihood that points i and j are connected:

$$v_{ij}=\exp\left(\frac{-d(x_i,x_j)-\rho_i}{\sigma_i}\right)$$

where $d(x_i,x_j)$ is the distance between points, ρ_i is the distance to the nearest neighbor of i, and σ_i is a normalization factor.

Similarly, in the low-dimensional space, edge weights w_{ij} are defined as:

$$w_{ij}=\left(1+a\cdot d(y_i,y_j)^{2b}\right)^{-1}$$

where a and b are parameters controlling the shape of the curve.

UMAP's objective function minimizes the cross-entropy between these two graph representations:

$$\min\sum_{i,j}\left[v_{ij}\log\left(\frac{v_{ij}}{w_{ij}}\right)+(1-v_{ij})\log\left(\frac{1-v_{ij}}{1-w_{ij}}\right)\right]$$

This optimization is performed using stochastic gradient descent with negative sampling to find the optimal low-dimensional embedding.

Classification Algorithms

Random Forest

Random Forest is an ensemble method that combines multiple decision trees to improve generalization performance. Each tree in the forest is trained on a bootstrap sample of the original data, and at each node, only a random subset of features is considered for splitting.

For a classification task with C classes, each tree t in the forest produces a probability distribution $p_t(c|x)$ over the classes for a given input x. The final prediction is obtained by averaging these probabilities across all trees:

$$p(c|x)=\frac{1}{T}\sum_{t=1}^{T}p_t(c|x)$$

where T is the total number of trees. The predicted class is the one with the highest probability:

$$\hat{y}=\arg\max_{c\in\{1,\ldots,C\}}p(c|x)$$

Support Vector Machine (SVM)

SVM finds the optimal hyperplane that maximizes the margin between classes. For linearly separable data, the primal optimization problem is:

$$\min_{\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to}\ y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1,\quad \forall i$$

For non-separable data, slack variables ξ_i are introduced:

$$\min_{\mathbf{w},b,\xi}\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to}\ y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1-\xi_i,\quad \xi_i\geq 0,\quad \forall i$$

where C is a regularization parameter.

For nonlinear decision boundaries, kernel functions $K(\mathbf{x}_i,\mathbf{x}_j)=\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ implicitly map the data to a higher-dimensional space. Our experiments use the radial basis function (RBF) kernel:

$$K(\mathbf{x}_i,\mathbf{x}_j)=\exp\left(-\gamma\|\mathbf{x}_i-\mathbf{x}_j\|^2\right)$$

Neural Network

We use a Multi-Layer Perceptron (MLP) with one hidden layer containing 100 neurons. For an input vector $\mathbf{x}$, the output of the hidden layer is:

$$\mathbf{h}=\sigma(\mathbf{W}_1\mathbf{x}+\mathbf{b}_1)$$

where $\mathbf{W}_1$ is the weight matrix, $\mathbf{b}_1$ is the bias vector, and σ is a nonlinear activation function (ReLU in our case).

The output layer produces class probabilities:

$$\mathbf{p}=\text{softmax}(\mathbf{W}_2\mathbf{h}+\mathbf{b}_2)$$

The network is trained to minimize the cross-entropy loss:

$$L=-\sum_{i=1}^{n}\sum_{c=1}^{C}y_{i,c}\log(p_{i,c})$$

where $y_{i,c}$ is 1 if sample i belongs to class c and 0 otherwise, and $p_{i,c}$ is the predicted probability that sample i belongs to class c.

Experimental Setup

Datasets

We conducted experiments on two benchmark datasets:

- **Digits**: The Scikit-learn digits dataset, containing 1,797 8×8 grayscale images of handwritten digits (0-9), resulting in 64 features.
- **MNIST**: A subset of 5,000 samples from the MNIST dataset, containing 28×28 grayscale images of handwritten digits (0-9), resulting in 784 features.

Preprocessing

For both datasets, we:

1. Split the data into training (75%) and testing (25%) sets using stratified sampling to maintain class proportions.
2. Standardized features by subtracting the mean and scaling to unit variance.

Dimension Reduction

We applied four approaches to each dataset:

1. No dimension reduction (baseline)
2. PCA, reducing to 10 dimensions
3. t-SNE, reducing to 2 dimensions
4. UMAP, reducing to 10 dimensions

Classification

For each dimension-reduced dataset, we trained and evaluated three classifiers:

1. Random Forest with 100 estimators
2. Support Vector Machine with RBF kernel
3. Neural Network (MLP) with one hidden layer (100 neurons)

Evaluation Metrics

We assessed performance using several metrics:

1. Accuracy: proportion of correctly classified samples
2. Precision: proportion of positive identifications that were actually correct
3. Recall: proportion of actual positives that were correctly identified
4. F1 score: harmonic mean of precision and recall
5. Training time: time required to train the model (seconds)
6. Inference time: time required to make predictions on the test set (seconds)

Results

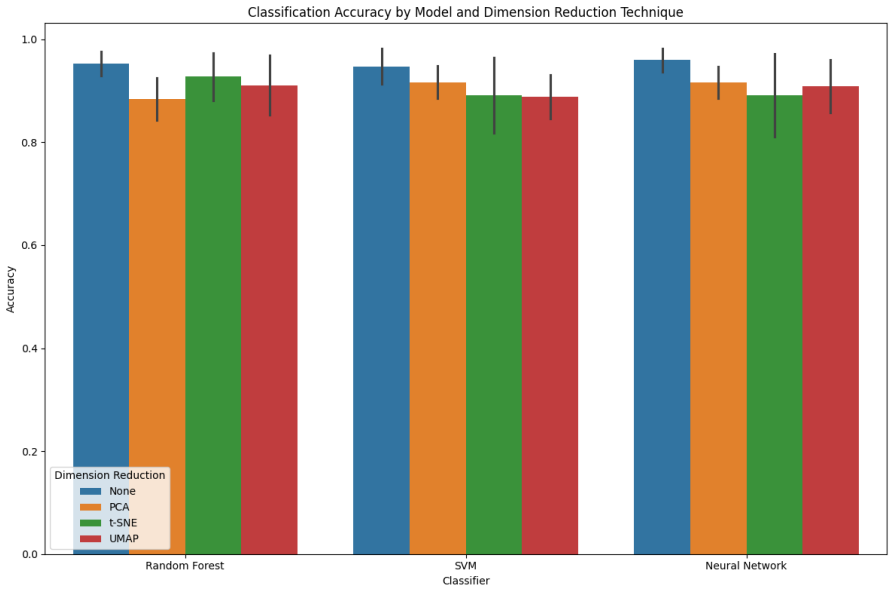
Overall Performance Comparison

Table 1 presents the average performance metrics for each dimension reduction technique across both datasets and all classifiers. The "None" approach (no dimension reduction) achieved the highest average accuracy (95.31%), followed by PCA (90.55%), t-SNE (90.32%), and UMAP (90.28%). However, these average figures mask important variations across datasets and classifiers.

Average performance metrics by dimension reduction technique

Dimension Reduction	Accuracy	Precision	Recall	F1 Score
None	0.9531	0.9539	0.9531	0.9532
PCA	0.9055	0.9069	0.9055	0.9054
t-SNE	0.9032	0.9048	0.9032	0.9021
UMAP	0.9028	0.9055	0.9028	0.9023

Figure 1 shows the classification accuracy for each combination of dimension reduction technique and classifier across both datasets. While no dimension reduction generally achieved the highest accuracy, the performance gap was notably smaller for the Digits dataset compared to MNIST.



Classification accuracy by model and dimension reduction technique

Dataset-Specific Performance

Tables 2 and 3 present the performance metrics for the Digits and MNIST datasets, respectively. For the Digits dataset, SVM achieved the highest accuracy (98.22%) with no dimension reduction, closely followed by Neural Network (98.22%) with no dimension reduction. For the MNIST dataset, Neural Network with no dimension reduction performed best (93.68%), followed by Random Forest (92.88%) with no dimension reduction.

Performance metrics for the Digits dataset

Dimension Reduction	Classifier	Accuracy	Precision	Recall	F1 Score
None	Random Forest	0.9756	0.9758	0.9756	0.9755
None	SVM	0.9822	0.9825	0.9822	0.9821
None	Neural Network	0.9822	0.9825	0.9822	0.9822
PCA	Random Forest	0.9244	0.9250	0.9244	0.9236
PCA	SVM	0.9489	0.9510	0.9489	0.9486
PCA	Neural Network	0.9467	0.9481	0.9467	0.9471
t-SNE	Random Forest	0.9733	0.9735	0.9733	0.9732

Dimension Reduction	Classifier	Accuracy	Precision	Recall	F1 Score
t-SNE	SVM	0.9644	0.9658	0.9644	0.9644
t-SNE	Neural Network	0.9711	0.9713	0.9711	0.9710
UMAP	Random Forest	0.9689	0.9694	0.9689	0.9688
UMAP	SVM	0.9311	0.9350	0.9311	0.9314
UMAP	Neural Network	0.9600	0.9611	0.9600	0.9600

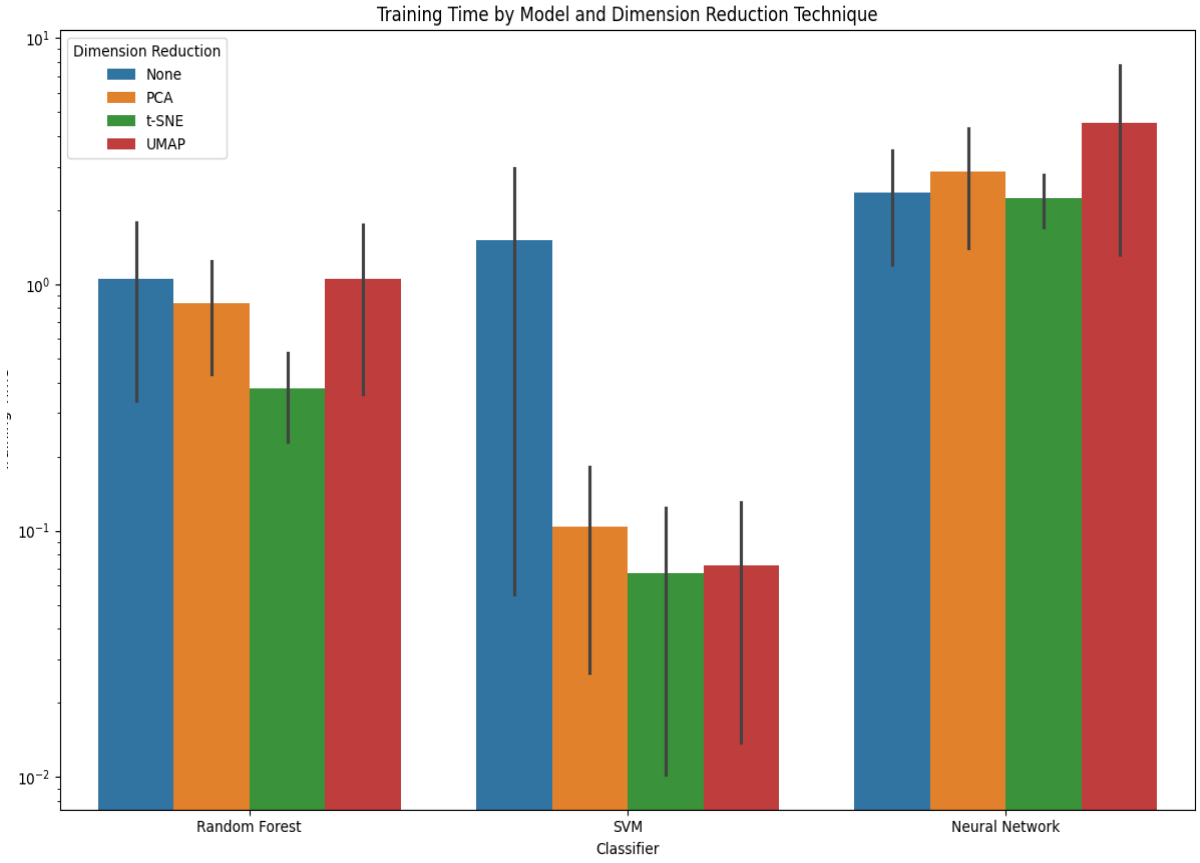
Performance metrics for the MNIST dataset

Dimension Reduction	Classifier	Accuracy	Precision	Recall	F1 Score
None	Random Forest	0.9288	0.9290	0.9288	0.9287
None	SVM	0.9128	0.9154	0.9128	0.9132
None	Neural Network	0.9368	0.9371	0.9368	0.9366
PCA	Random Forest	0.8424	0.8449	0.8424	0.8423
PCA	SVM	0.8848	0.8864	0.8848	0.8847
PCA	Neural Network	0.8856	0.8862	0.8856	0.8852
t-SNE	Random Forest	0.8816	0.8815	0.8816	0.8812
t-SNE	SVM	0.8176	0.8262	0.8176	0.8145
t-SNE	Neural Network	0.8112	0.8136	0.8112	0.8082
UMAP	Random Forest	0.8528	0.8551	0.8528	0.8521
UMAP	SVM	0.8464	0.8501	0.8464	0.8449
UMAP	Neural Network	0.8576	0.8626	0.8576	0.8568

Notably, the performance degradation due to dimension reduction was more severe for the MNIST dataset compared to Digits. For example, using t-SNE reduced the average accuracy by 2.5 percentage points for Digits but by 10.6 percentage points for MNIST. This suggests that the effectiveness of dimension reduction is highly dataset-dependent, with larger, more complex datasets potentially requiring more dimensions to maintain performance.

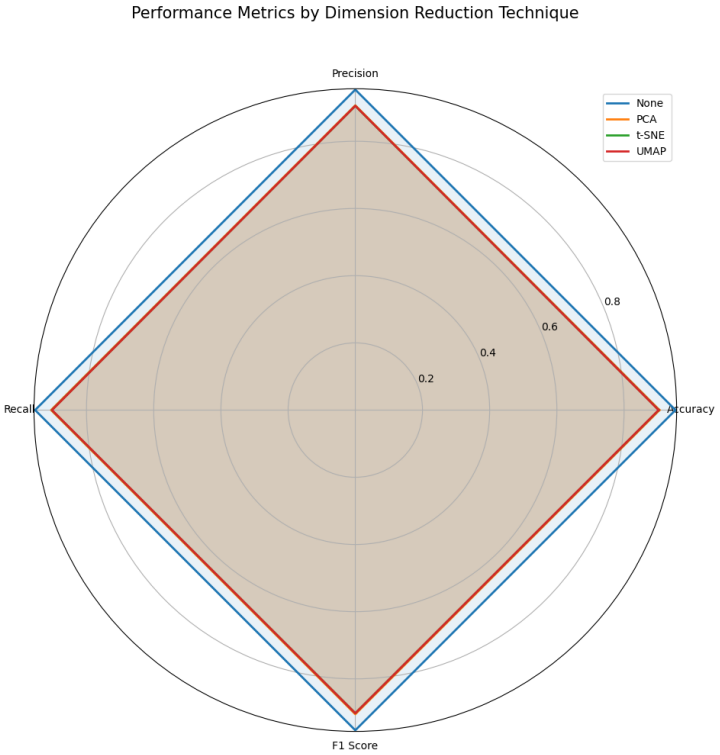
Computational Efficiency

Figure 2 shows the training time for each classifier and dimension reduction technique. The combination of t-SNE with SVM achieved the fastest training time (0.0103 seconds) for the Digits dataset, providing a compelling trade-off between speed and accuracy (96.44%). However, the preprocessing time for dimension reduction must also be considered: t-SNE required 11.45 seconds for Digits and 40.27 seconds for MNIST, while UMAP required 16.21 seconds for Digits and 26.69 seconds for MNIST.



Training time by model and dimension reduction technique (log scale)

Figure 3 presents a radar chart comparing the four dimension reduction techniques across various performance metrics. While "None" performed best in terms of accuracy-related metrics, the non-trivial preprocessing time of t-SNE and UMAP must be balanced against their benefits in specific contexts.



Radar chart comparing dimension reduction techniques across performance metrics

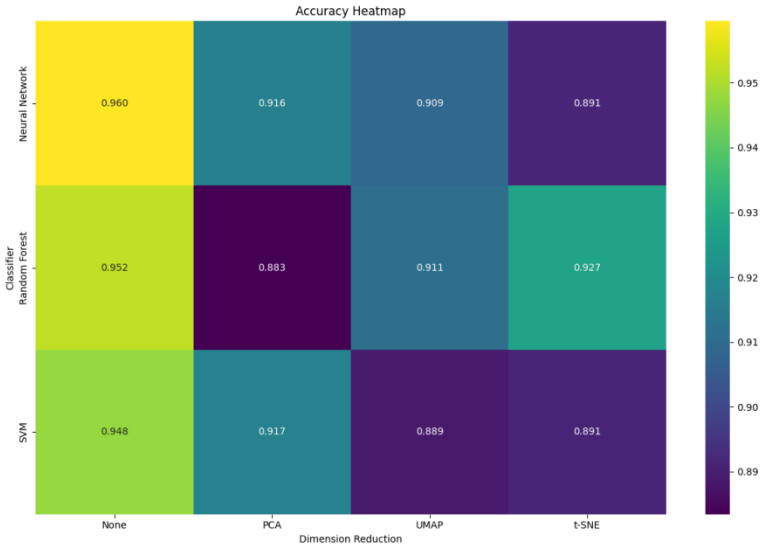
Classifier Performance

Across all dimension reduction techniques, Neural Networks achieved the highest average accuracy (91.89%), followed closely by Random Forests (91.85%) and SVMs (91.10%). Table 4 breaks down the average performance by classifier.

Average performance metrics by classifier

Classifier	Accuracy	Precision	Recall	F1 Score
Neural Network	0.9189	0.9200	0.9189	0.9185
Random Forest	0.9185	0.9193	0.9185	0.9180
SVM	0.9110	0.9140	0.9110	0.9105

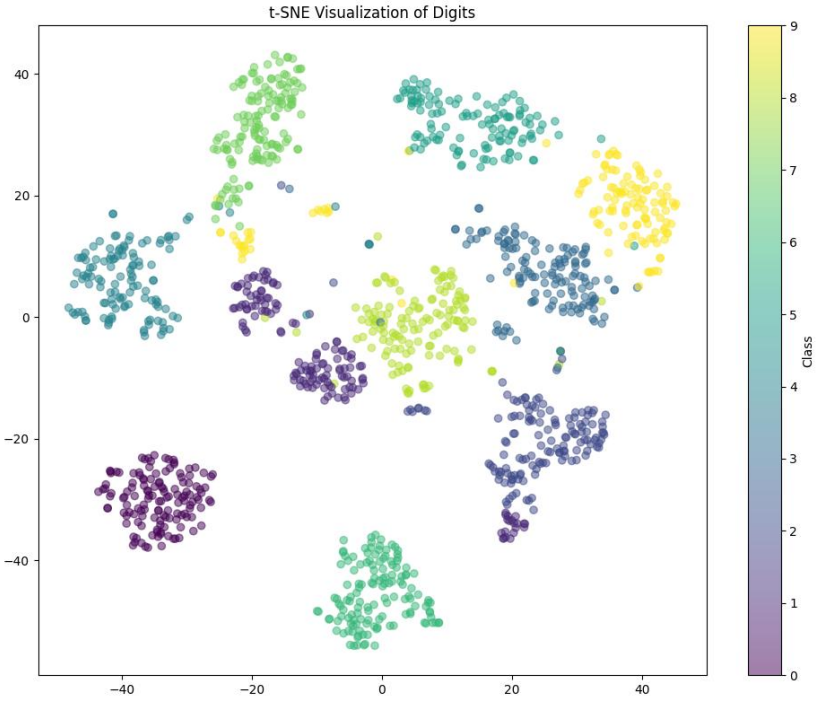
The performance of classifiers varied significantly depending on the dimension reduction technique applied. Figure 4 shows a heatmap of classification accuracy for each combination of classifier and dimension reduction technique. The heatmap reveals that Neural Networks maintained the most consistent performance across different dimension reduction techniques, suggesting greater robustness to changes in data representation.



Heatmap of classification accuracy by model and dimension reduction technique

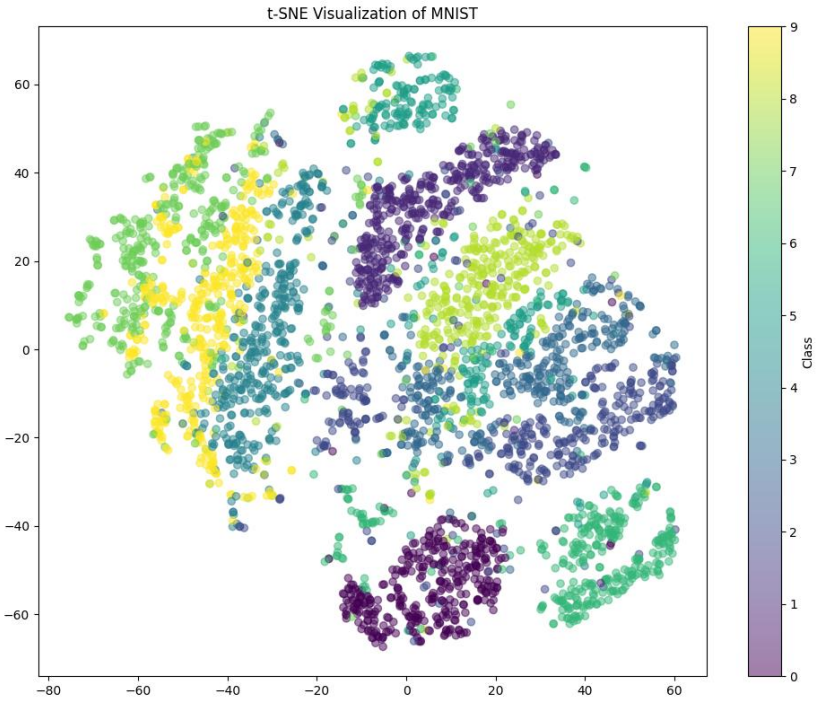
Visualization of Reduced Data

Figure 5 shows a t-SNE visualization of the Digits dataset, where the clear separation between clusters corresponding to different digits indicates that t-SNE effectively preserved class structure in two dimensions. This visual separation explains the relatively high accuracy achieved by classifiers on t-SNE-reduced data, despite the drastic dimension reduction from 64 to just 2.



t-SNE visualization of the Digits dataset colored by class

Similarly, Figure 6 presents a t-SNE visualization of the MNIST dataset. While there is still visible class separation, the boundaries are less distinct, corresponding to the lower classification accuracy observed for MNIST compared to Digits.



t-SNE visualization of the MNIST dataset colored by class

Discussion

Trade-offs Between Performance and Efficiency

Our experimental results reveal important trade-offs between classification performance and computational efficiency. While using no dimension reduction generally yielded the highest accuracy, the performance gap varied considerably across datasets and classifiers. For the Digits dataset, t-SNE with Random Forest achieved 97.33% accuracy, only slightly lower than the 98.22% achieved by SVM with no dimension reduction, while offering substantially faster training times.

The decision whether to apply dimension reduction should therefore consider both the specific characteristics of the dataset and the computational constraints of the application. For large, complex datasets where computation is a bottleneck, dimension reduction can provide substantial efficiency gains with only moderate performance degradation. Conversely, for smaller datasets or applications where accuracy is paramount, using the original high-dimensional representation may be preferable.

Dataset-Dependent Effectiveness

A key finding of our study is the dataset-dependent effectiveness of dimension reduction techniques. For the Digits dataset, t-SNE maintained high classification performance despite reducing the dimensionality from 64 to just 2, likely because the intrinsic dimensionality of this dataset is low. In contrast, for the MNIST dataset, all dimension reduction techniques led to more substantial performance degradation, suggesting that more dimensions are needed to capture the relevant structure of this dataset.

This observation aligns with the manifold hypothesis , which posits that high-dimensional data often lies on or near a lower-dimensional manifold. The effectiveness of a dimension reduction technique depends on how well it can approximate this manifold, which varies across datasets.

Relative Performance of Dimension Reduction Techniques

Among the dimension reduction techniques compared, PCA performed well on the MNIST dataset, particularly in conjunction with SVM and Neural Network classifiers. This suggests that for complex datasets, the linear transformations of PCA may capture a significant portion of the relevant variance, especially when the reduced dimensionality is sufficiently high (10 dimensions in our case).

t-SNE showed impressive performance on the Digits dataset, particularly with Random Forest and Neural Network classifiers. This aligns with previous research showing that t-SNE's emphasis on preserving local structure can be beneficial for classification tasks where class boundaries are locally defined .

UMAP performed comparably to t-SNE but with greater variance across classifiers. It worked particularly well with Random Forest and Neural Network on the Digits dataset but less effectively with SVM. This suggests that UMAP's preservation of both local and global structure may be more compatible with certain classification algorithms, making the choice of both dimension reduction technique and classifier important.

Implications for Classification Pipelines

Our results have several implications for designing effective classification pipelines for high-dimensional data:

1. **Adaptive selection:** The choice of dimension reduction technique should be adapted to the specific dataset characteristics. For smaller datasets with clear class separation (like Digits), nonlinear methods like t-SNE can effectively reduce dimensions with minimal

performance loss. For larger, more complex datasets (like MNIST), preserving more dimensions or using no dimension reduction may be necessary.

2. **Classifier compatibility:** Certain combinations of dimension reduction techniques and classifiers work better than others. Random Forest and Neural Network showed greater robustness to dimension reduction compared to SVM, particularly with nonlinear techniques like t-SNE and UMAP.

3. **Computational considerations:** While dimension reduction introduces preprocessing overhead, it can significantly reduce classifier training and inference times. For applications requiring real-time predictions or frequent retraining, this trade-off may favor dimension reduction even with slight accuracy penalties.

4. **Visualization benefits:** Beyond performance considerations, techniques like t-SNE and UMAP provide valuable visualizations that can aid in understanding dataset structure and debugging classification issues, offering additional qualitative benefits.

Conclusion

This study presented a comprehensive evaluation of dimension reduction techniques for high-dimensional classification tasks. We compared PCA, t-SNE, and UMAP in conjunction with Random Forest, SVM, and Neural Network classifiers on two benchmark datasets. Our results provide empirical evidence for several key findings:

1. No dimension reduction generally achieves the highest classification accuracy, but the gap can be small for datasets with low intrinsic dimensionality.

2. t-SNE offers an excellent trade-off between classification performance and training efficiency, particularly for the Digits dataset and in combination with Random Forest.

3. Neural Networks demonstrate the most consistent performance across dimension reduction techniques, suggesting greater robustness to changes in data representation.

4. The effectiveness of dimension reduction is highly dataset-dependent, with larger performance degradation observed for the more complex MNIST dataset.

These findings have important implications for practitioners. First, the choice of dimension reduction technique should be adapted to the specific dataset and application requirements. Second, the computational benefits of dimension reduction should be weighed against potential performance degradation. Finally, certain combinations of dimension reduction techniques and classifiers offer particularly favorable trade-offs and should be considered as starting points for high-dimensional classification tasks.

Future work could extend this analysis to a broader range of datasets across different domains, investigate the impact of the reduced dimensionality (e.g., comparing 2, 10, and 20 dimensions), and explore adaptive approaches that automatically select the optimal dimension reduction technique based on dataset characteristics. Additionally, examining how these findings translate to other tasks beyond classification, such as regression or clustering, would provide a more comprehensive understanding of dimension reduction in machine learning pipelines.

Limitations

This study has several limitations that should be acknowledged. First, we examined only a subset of available dimension reduction techniques and classification algorithms. Second, we used default or common parameter settings for all methods, whereas performance might be improved through extensive hyperparameter tuning. Third, our analysis focused on two benchmark datasets, and findings may not generalize to all domains and data types. Finally, we evaluated only on tabular data represented as grayscale images; performance characteristics might differ for other data modalities such as text, time series, or graph-structured data.

Despite these limitations, our systematic comparison provides valuable insights into the interplay between dimension reduction and classification, offering practical guidelines for practitioners working with high-dimensional data.

References

[1] Abdi, H. and Williams, L. J. (2010). Principal component analysis. , 2(4):433–459.

[2] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. , 15(6):1373–1396.

[3] Bellman, R. E. (1966). Dynamic programming. , 153(3731):34–37.

[4] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. , 35(8):1798–1828.

[5] Bishop, C. M. (2006). . Springer.

[6] Breiman, L. (2001). Random forests. , 45(1):5–32.

[7] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). . CRC press.

[8] Cortes, C. and Vapnik, V. (1995). Support-vector networks. , 20(3):273–297.

[9] Cristianini, N. and Shawe-Taylor, J. (2000). . Cambridge university press.

[10] Cunningham, J. P. and Ghahramani, Z. (2015). Linear dimensionality reduction: Survey, insights, and generalizations. , 16(1):2859–2900.

[11] Deegalla, S. and Bostrom, H. (2006). Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification. In *5th International Conference on Machine Learning and Applications (ICMLA'06)*, pages 245–250. IEEE.

[12] Ding, C. and He, X. (2004). K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29.

[13] Espadoto, M., Martins, R. M., Kerren, A., Hirata, N. S., and Telea, A. C. (2019). Toward a quantitative survey of dimension reduction techniques. , 27(3):2153–2173.

[14] Friedman, J. H. (1997). On bias, variance, 0/1—loss, and the curse-of-dimensionality. , 1(1):55–77.
[15] Ghodsi, A. (2006). Dimensionality reduction a short tutorial. , 37(38):39.
[16] Goodfellow, I., Bengio, Y., and Courville, A. (2016). . MIT press.
[17] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. , 3(Mar):1157–1182.
[18] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. , 24(6):417.
[19] Jolliffe, I. T. (2002). . Springer.
[20] Kittler, J. (1978). Feature selection and extraction. , 59:81.
[21] Krijthe, J. H. and van der Maaten, L. (2017). Comparing dimensionality reduction techniques using structure preserving assessment. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.
[22] Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. , 29(1):1–27.
[23] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. , 86(11):2278–2324.
[24] Linderman, G. C., Rachh, M., Hoskins, J. G., Steinerberger, S., and Kluger, Y. (2019). Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. , 16(3):243–245.
[25] van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. , 9(Nov):2579–2605.
[26] McInnes, L., Healy, J., and Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. .
[27] Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. , 2(11):559–572.
[28] Pechenizkiy, M., Tsymbal, A., and Puuronen, S. (2004). PCA-based feature transformation for classification: issues in medical diagnostics. In *Proceedings. 17th IEEE Symposium on Computer-Based Medical Systems*, pages 535–540. IEEE.
[29] Quinlan, J. R. (1986). Induction of decision trees. , 1(1):81–106.
[30] Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. , 290(5500):2323–2326.
[31] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. , 323(6088):533–536.
[32] Schölkopf, B., Smola, A. J., Bach, F., et al. (2002). . MIT press.
[33] Sugiyama, M. (2007). Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. , 8(May):1027–1061.
[34] Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. , 290(5500):2319–2323.
[35] Trunk, G. V. (1979). A problem of dimensionality: A simple example. , (3):306–307.
[36] van der Maaten, L. (2014). Accelerating t-SNE using tree-based algorithms. , 15(1):3221–3245.
[37] van der Maaten, L., Postma, E., and van den Herik, J. (2009). Dimensionality reduction: a comparative review. , 10:66–71.
[38] van der Maaten, L. J., Postma, E. O., and Van Den Herik, H. J. (2007). Dimensionality reduction: A comparative review. .
[39] Vapnik, V. N. (1999). An overview of statistical learning theory. , 10(5):988–999.
[40] Xanthopoulos, P., Pardalos, P. M., and Trafalis, T. B. (2013). A survey of dimensionality reduction approaches for high-dimensional data sets. In *Linear and Nonlinear Dimensionality Reduction*, pages 59–90. Springer.
[41] Yang, Y. and Nataliani, Y. (2017). Dimension reduction methods for microarray data: a review. In *AIP Conference Proceedings*, volume 1862, page 030128. AIP Publishing LLC.
[42] Yang, Y., Xu, D., Nie, F., Yan, S., and Zhuang, Y. (2005). Local and nonlocal preserving projection for dimensionality reduction. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1059–1065. IEEE.