SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I6P110

Volume 11 Issue 6, 91-100, June 2024
©2024 Seventh Sense Research Group®

Original Article

Multi-Dimensional Machine Intelligence Technique on High Computational Data for Bigdata Analytics

K. Kishore Raju[1], Ch.S.V.V.S.N. Murty[2], Suresh Kumar Kanaparthi[3], Amdewar Godavari[4], Kayam Saikumar[5]

[1]Department of Information Technology, SRKR Engineering College (A), Andhra Pradesh, India.
[2]Dept of CSE, Aditya College of Engineering and Technology, Andhra Pradesh, India.
[3]School of Computer Science and Artificial Intelligence, SR University Warangal, Telangana, India.
[4]Department of Computer Science and Engineering (Networks), Kakatiya Institute of Technology and Science, Telangana, India.
[5]koneru Lakshmaiah Education Foundation, Andhra Pradesh, India.

[1]Corresponding Author : kkishoreraju78@gmail.com

Received: 05 April 2024 Revised: 07 May 2024 Accepted: 05 June 2024 Published: 29 June 2024

***Abstract** - In the current digital environment, copious amounts of data are generated across diverse sectors like healthcare, content creation, the internet, and businesses. ML algorithms are pivotal in analyzing this data to unveil significant ways to make decisions. However, not all features within these datasets are relevant for constructing robust machine learning models. Some features may be insignificant or have minimal impact on the prediction outcomes. By filtering out these irrelevant features, the computational burden on machine learning algorithms is reduced. Using the freely available MINIST dataset, this study explores the application of t-SNE, LDA, and Principal Component Analysis (PCA) alongside several prominent ML techniques like Naive Bayes, SVM classifiers, and K-NN classifications employed. Experimental outcomes illustrate the effectiveness of ML algorithms in this context. Furthermore, the experiments demonstrate that employing PCA with machine learning algorithms leads to improved outcomes, particularly when dealing with high-dimensional datasets. Performance measures like Accuracy 98.34%, Sensitivity 98.76%, Recall 98.45% and Throughput 98.65% have been attained, which was a good improvement.*

***Keywords** - Dimensionality reduction, KNN, ML, NB, PCA, LDA, t-SNE, SVM.*

1. Introduction

In the realm of machine intelligence, the recognition of handwritten digits stands as a pivotal challenge with profound implications across various domains [1]. Over the past two decades, significant strides have been made in this field, fueled by extensive research into handwritten digit recognition methodologies [2].

However, the productivity of machine learning algorithms in this context is often hindered by the curse of dimensionality inherent in high-dimensional data [3]. As datasets grow larger, the precision of machine learning-based classification diminishes, necessitating the adoption of Dimensionality Reduction (DR) techniques to enhance accuracy [4].

This paper delves into the intersection of dimensionality reduction methods and machine learning algorithms, focusing particularly on their application in handwritten digit recognition tasks. With handwritten numerals serving as a ubiquitous form of communication in everyday life, the accurate identification of such digits has garnered widespread attention globally [5].

Leveraging insights from the literature, we explore various approaches to dimensionality reduction and their integration with machine learning frameworks to improve the overall performance and accuracy of digit recognition systems [6] through an examination of seminal works, including those by Md. Golam Sarowar et al., G. Thippa Reddy, Hany Yan, and others highlight the efficacy of dimensionality reduction techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Singular Value Decomposition (SVD) in enhancing classifier performance [7].

Additionally, we discuss the potential pitfalls of employing multiple DR techniques simultaneously, as noted by Gustavo et al., and underscore the importance of selecting appropriate DR methods tailored to specific datasets and classification tasks [8].

By synthesizing insights from diverse research endeavors, this paper aims to contribute to the ongoing discourse on optimizing machine intelligence techniques for high-dimensional data analytics, with a focus on handwritten digit recognition [9, 10].

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

2. Literature Survey

Over the past two decades, extensive research has delved into handwritten digit recognition, a crucial aspect of recognition methodologies [11]. The productivity of creating ML algorithms is affected by high-dimensionality data. With more measurement data, the precision of ML-based grouping falls. To improve accuracy, we must reduce high dimensionality to low dimensionality. In the realm of computer vision, the task of recognizing handwritten digits is crucial for various applications.

This process involves intricate procedures of Dimensionality Reduction (DR) methods and machine learning algorithms. Numerals written by hand are one of the main uses. Due to its use in everyday life, the identification of handwritten numbers has grown significantly in popularity globally [12].

Literature devotes much effort to digit recognition, explicit framework use, and becoming familiar with mathematics. There are many research areas with experience combining and creating different framework learning approaches, which will improve the overall presentation and accuracy in recognizing digits [13].

When using a dataset with a high dimension size, the DR method is crucial. The goal of DR technology is to make raw records less dimensional. It is essential for learning about frameworks and data mining. This could enhance the overall performance of the classifier while also decreasing computational complexity by cutting down the variety of data highlights [14]. Linear and nonlinear techniques for dimensionality reduction were found [15].

The MNIST dataset and the unique accuracy data from many classifiers were explained in [16]. The highest accuracy, 80.84%, was determined using a PCA-based CNN with ACO. G. Thippa Reddy and others [6] noted that using the Cardiotocography (CTG) dataset, classifiers with PCA outperformed those with LDA in terms of overall performance.

However, it also issued a warning that these DR strategies may also be evaluated on high-dimensional data such as text, images, and other types of statistics. Hany Yan et al. [5] recognized that applying an appropriate DR that appears before training can effectively increase class accuracy. Additionally, DR lowers the required garage this reduces the computational complexity involved in digit recognition [7].

Adiwijaya et al. [1] warned against using the discount approach PCA and compared SVM [16] and LMBP algorithms in a cancer detection scheme that is based only on microarray statistics. Pitoyo Hartono [3] worked on the same dataset MNIST and was able to acquire the class ability while embedding the elegant data in its low-dimensional depiction of rRBF. Additionally worked on other dimensionality discount techniques, including PCA, NCA, and t-SNE. According to Gustavo et al. [2], using multiple DR techniques can have worse effects than using only one. The World Development Indicator (WDI) dataset was utilized by Abbas et al. [8].

According to M.Ramakrishna Murty et al. [14], Singular Value Decomposition (SVD) was utilized for dimensionality reduction. as noted by Rizgar et al. [9], who explored various feature selection and feature extraction methods. Ramakrishna et al. introduced Least Square SVM and Singular Value Decomposition were utilized to cluster text data, with the objective of predicting the optimal number of clusters [17].

Tausif et al. introduced a lightweight CNN model for the MNIST dataset was developed, with a focus on optimizing execution time [18, 19]. Additionally, [20] compares different models utilizing various machine learning classifier techniques on high-dimensional data.

3. Machine Learning Techniques

Different ML classification methods are explained and listed clearly for deep understanding.

3.1. Naive Bayes Algorithm

Depending on the independence of characteristics and the Bayes theorem, the Naive Bayes method is a straightforward yet effective classifying method. It finds extensive application in many domains as medical diagnosis, spam filtering, and text categorization, due to its efficiency and effectiveness, especially for large datasets.

$$.P(c|x) = \frac{P(X|C)*P(c)}{P(x)} \quad (1)$$

$$P(c|x) = P(x1|c) \times P(x2|c) \times \ldots. \times P(xn|c) \times P(c) \quad (2)$$

- P(c|x) represents the posterior probability of the given predictor (x, characteristics) for the class (c , target).
- P(c) is the prior probability of the class.
- P(x|c) is the likelihood, which is the probability of the predictor given the class.
- P(x) is the prior probability of the predictor.

The above algorithm is more effective in extracting features; compared to past models, working models attained more improvement. The model training and testing can be possible with a GPU of 8 GB supported by Nvidia. The system configuration of the Linux 20.01 version with Python 10 and above versions is imported for this research. The training time taking 200 seconds, epochs of 20 and batch size of 8 have been fixed for this work. Finally, after training got one weight file like the best P^th, which is an efficient file to test real-time samples.

3.2. Support Vector Machine (SVM)

SVM is used to determine the optimal line or decision boundary that can effectively separate classes in an n-dimensional space, ensuring precise classification of new data points. This boundary, known as a hyperplane, is optimal as it maximizes the margin between classes in the n-dimensional space. SVM accomplishes this by selecting crucial points/vectors called support vectors to define the hyperplane. The resulting algorithm, known as SVM, belongs to the category of supervised learning algorithms.

3.3. K-Nearest Neighbour (KNN) Algorithm

K-Nearest Neighbor [18] is a method used in supervised learning. The KNN algorithm is versatile and can handle both regression and classification tasks, although its primary use is for classification problems. It is often referred to as an instance-based or lazy learner algorithm.

This similarity is typically measured using the Euclidean distance between data points, which reflects the proximity between two points and is essential for K Nearest Neighbors. Only continuous variables are applicable for three distance metrics (Euclidean, Manhattan, and Minkowski distance). Minkowski distance recommends using the Hamming distance for handling categorical variables.

Distance: It is more confusing than other measures. The Manhattan distance and the Euclidean distance are both possible hypotheses. There are three conditions necessary for this operation.

Zero Vector: While all other vectors have positive lengths, The zero vector has a magnitude of zero. For instance, if we travel from one location to another starting at one, the distance will always be known. In any event, that distance is 0 if we move from one place to another.

Scalar Factor: The length of the vector is changed while maintaining its direction when you multiply it by a positive amount. The course stays the same; for instance, if we go a certain distance in one direction and add a comparable distance.

Triangle inequality: A straight line connects two focuses with the shortest possible distance.

$$D(x,y) = (\sum_{i=1}^{k}(|x_i - y_i|)^q)^{\frac{1}{q}} \tag{3}$$

3.4. t-Distributed Stochastic Neighbor Embedding (tSNE)

t-SNE, an unsupervised dimensionality reduction technique, and its modified version, t-SNE [15], both aim to minimize the discrepancy among t-SNE, an unsupervised technique for reducing dimensionality, the distribution in a corresponding low-dimensional embedding that evaluates pairwise similarity given two things the two xi and xj. The two's pairwise similarities are listed. Based on the conditional likelihood that i would take an item, as indicated by the equation, item j is its neighbor.

$$P_{j|_i} = \frac{\exp\left(-d(x_i,x_j)^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-d(x_i,x_j)^2/2\sigma_i^2\right)} \tag{4}$$

By symmetrizing the pairwise distance between the two items, as listed below,

$$P_{i|j} = \frac{P_{i|_j}+P_{j|_i}}{2N} \tag{5}$$

3.5. Principal Component Analysis (PCA)

One popular dimensionality reduction method in statistics and machine learning is Principal Component Analysis (PCA). Its goal is to convert a set of potentially linked variables into principal components or a collection of linearly uncorrelated elements. Here are some key points about PCA.

This process makes smaller datasets easier to analyze and interpret, as well as being faster and simpler to analyze for ML classification, we need to minimize the dimensionality. This involves the following 5 steps:

3.5.1. Standardization

Prior to PCA, standardization must be completed. For each value of each variable, solve Equation (4) by dividing by the standard deviation after taking the mean out.

$$.x_j^i = \frac{x_j^i - \overline{x_J}}{\sigma_j} \quad \forall j \tag{6}$$

3.5.2. Covariance Matrix Computation

The same scale will be applied to each variable. To find the correlations, we must compute the covariance matrix.

$$\Sigma = \frac{1}{m}\sum_i^m (x^i)(x^i)^T, \Sigma \in R^{n*n} \tag{7}$$

3.5.3. Computation of Eigenvectors

To determine the principal component, it is crucial to calculate the eigenvalues and eigenvectors of the covariance matrix.

$$u^T \sum = \lambda\mu$$

$$U = \begin{bmatrix} | & | & | \\ u_1 & u_2 & u_3 \\ | & | & | \end{bmatrix}, u_i \in R^n \tag{8}$$

3.5.4. Feature Vector

The n-dimensional data must be projected onto a k-dimensional subspace. For this, we select the top k eigenvectors.

$$x_i^{new} = \begin{bmatrix} u_1^T x^i \\ u_2^T x^i \\ \vdots \\ \vdots \\ u_k^T x^i \end{bmatrix} \in R^k \tag{9}$$

3.5.5. Recast

Project the data onto the axes defined by projecting onto the principal components. Throughout the process of selecting principal components to construct the feature vector in the preceding steps, the input dataset consistently retains its representation in terms of the original axes.

$$Final\ dataset = Feature\ vector * Standardize\ original\ datase \tag{10}$$

3.6. Linear Discriminant Analysis (LDA)

To reduce dimensionality, LDA was utilized in this context. LDA retains all discriminative information while reducing dimensionality. Moreover, it projects data points onto a line to ensure the maintenance of distinct clusters, with each cluster having a centroid that is relatively close. Its objective is to identify boundaries near the class clusters.

Two methods are commonly used for dimensionality reduction: LDA utilizes feature extraction, as opposed to feature selection, as the method for dimensionality reduction. By obtaining fresh independent variables, LDA divides the majority of the dependent variable's classes.

If we take two classes into account and use µ1 and µ2 as their means, sample feature extraction can be mathematically expressed as,

$$\omega = S_\omega^{-1}(\mu_1 - \mu_2) \tag{11}$$

Where ω is the eigenvector of $S_\omega^{-1} S_b$that corresponds to the biggest eigenvalue.

Here $S_\omega = S_1 + S_2$

S1 and S2 are the scatter matrices of class 1 and class 2, and the mathematical formula for Sb is shown in Equation (12).

$$S_b = \frac{1}{c}\sum_{i=1}^{c}(\mu_i - \mu)(\mu_i - \mu)^T \tag{12}$$

Here, T is the Threshold.

4. Proposed Model

The suggested approach, as illustrated in Figure 1, aims to assess the performance of this particular model by following the steps outlined below.

Step 1 : Gathering of data sets.
Step 2 : Preprocess or normalize the dataset.
Step 3 : Train and test the specified ML algorithms, then assess their performance.
Step 4 : Utilize PCA, LDA, and t-SNE techniques the Normalize the data, followed by training and testing the ML algorithm using the reduced dataset.
Step 5 : Compare the outcomes obtained in step 3 and step 4 using parameters such as accuracy, precision, recall, and F1-score.

The MNIST dataset comprises 42,000 annotated grayscale images of handwritten numbers 0-9, each with dimensions of 28 x 28 pixels, along with an additional 28,000 unlabeled test images. This dataset is widely used to benchmark various machine learning categorization approaches to ensure accurate classification of the digits.

By leveraging techniques such as Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), and k-Nearest Neighbors (k-NN), we can achieve high accuracy in digit recognition. Additionally, the dataset's simplicity and diversity make it an excellent starting point for developing and testing new algorithms in the field of image recognition.

Data Normalization: Normalization involves transforming data to make them dimensionless or to align their distributions. This normalization procedure is also referred to as standardization, feature scaling, etc. In any machine learning application, including model fitting and data pre-processing, normalization is a critical step. Here, we employ the standard score normalization procedure to normalize the input dataset.

$$z=(x-\mu)/\sigma \tag{13}$$

Where,
z : standard score,
σ : standard deviation,
µ : population mean.

Using machine learning methods like Naive Bayes, SVM, and K-NN, the normalized data is tested. The effectiveness of the classifiers is then assessed using a variety of measures, including Precision, Recall, F1-Score, and Accuracy. LDA, PCA, and t-SNE are utilized. After normalization of the data, the reduced dataset undergoes testing using Naive Bayes, SVM, and K-NN machine learning methods. The outcomes are then evaluated again based on precision, recall, F1-Score, and accuracy metrics.

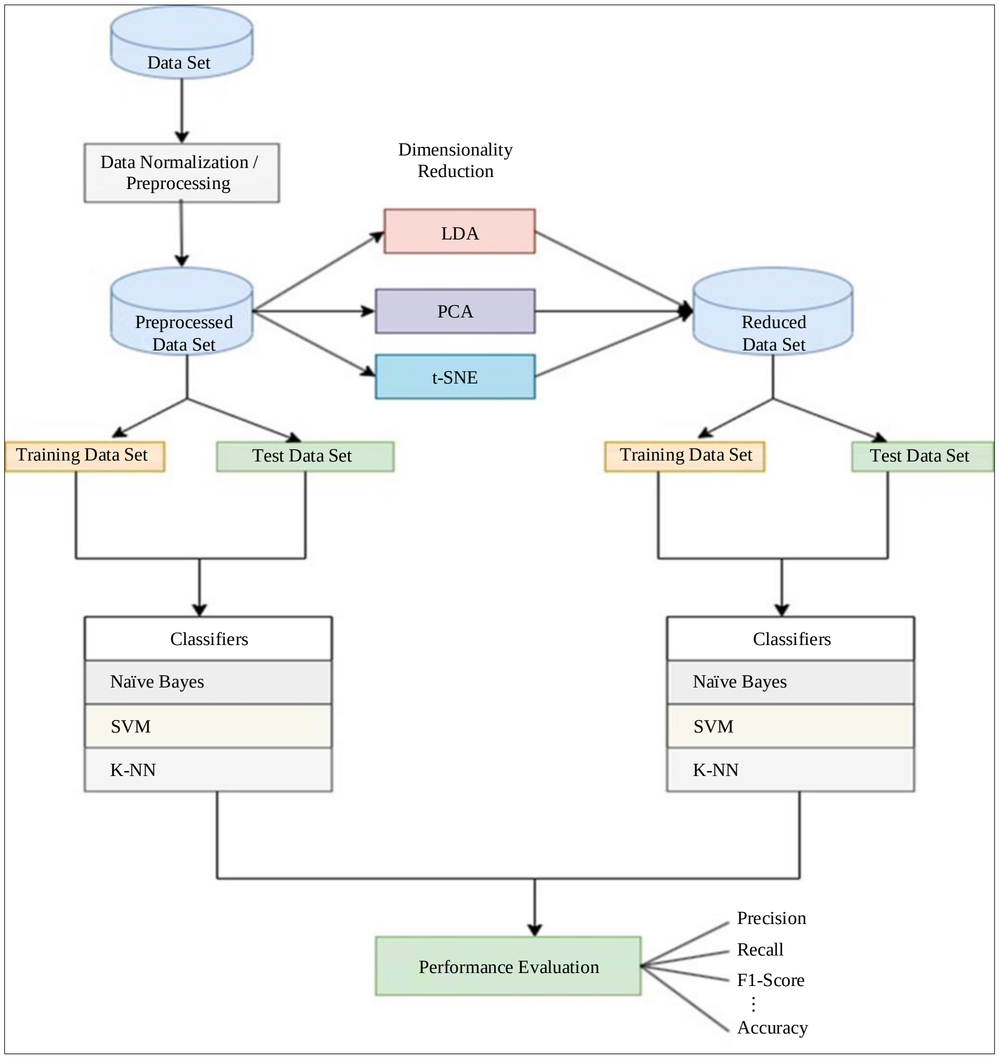

Fig. 1 The proposed model, which integrates dimensionality reduction and classification techniques

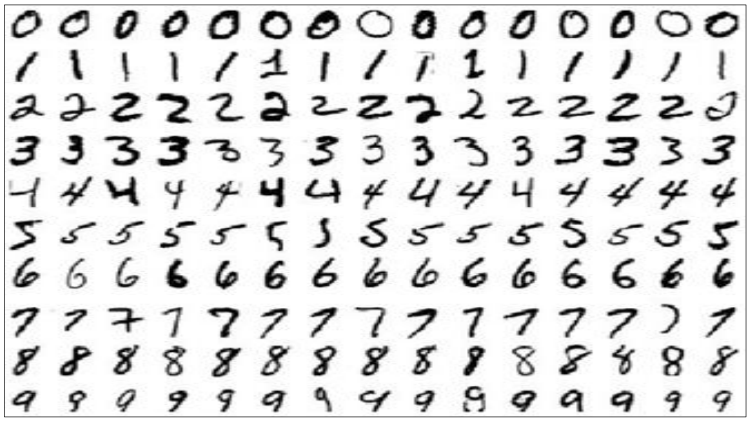

Fig. 2 A sample image of the MNIST dataset

5. Performance Evaluation Metrics

In this analysis, Metrics such as precision, accuracy, recall, and F1-score are utilized to assess the performance of this method. We delve into a discussion of these metrics here.

Accuracy: Accuracy denotes the total number of correct predictions made.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{14}$$

Precision: Precision is the ratio of correctly identified positive examples to the total number of positive examples (TP) to the total number of examples that were anticipated to be positive (TP+FP). It demonstrates that a favorable prediction was accurate.

$$Precision = \frac{TP}{(TP+FP)} \tag{15}$$

Recall: Recall measures the proportion of all correctly classified positive (TP) cases to all possible positive predictions.

$$Recall = \frac{TP}{(TP+FN)} \tag{16}$$

F1 score: A weighted average of sensitivity and precision is the F1 score. The F1 score may be a suitable option for achieving a balance between precision and recall.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{17}$$

6. Result Analysis

The efficiency of DL and ML classifications is determined by making use of the previously mentioned metrics: F1-score, accuracy, precision, and recall. Equation (14) is used, as shown in Table 1, to compare the accuracy of different classifications with as well as without DR.

The kNN classification algorithm, for example, achieves 94% accuracy without DR when k=3. However, after implementing PCA, LDA, and t-SNE, which reduce the number of dimensions to 70, the kNN classifier achieves an accuracy of 97.5%. Additionally, LDA combined with Naive Bayes (LDA+NB) yields 88% accuracy, while kNN with t-SNE achieves 92.5% accuracy.

Table 1. Accuracy comparison of classification with and without Dimensionality Reduction (DR)

Accuracy				
	Without Dimensionality Reduction	**With Dimensionality Reduction**		
		t-SNE	**LDA**	**PCA (n=70)**
SVM	0.9171	0.8514	0.8975	0.9381
NB	0.5447	0.8363	0.8848	0.8754
kNN (k=3)	0.9400	0.9255	0.9108	0.9750

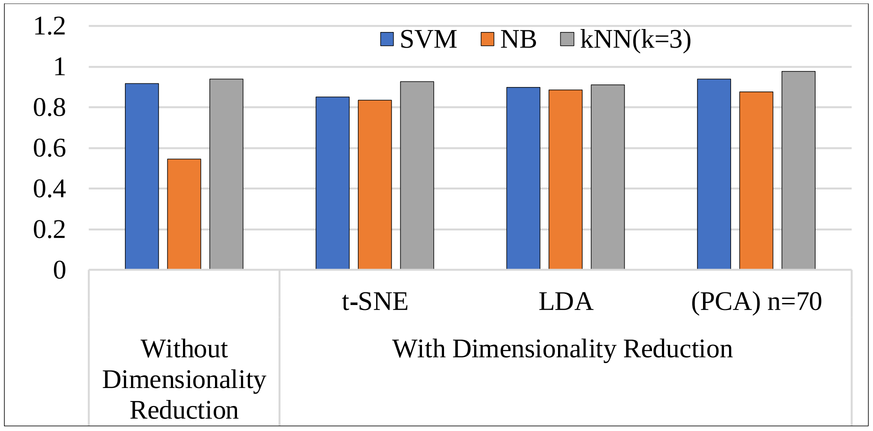

Fig. 3 Accuracy of without and with DR

Equation (15), used to display the precision of several classifiers with and without DR, is shown in Table 2. The following are the different algorithms' predicted precisions for each digit (0 to 9). When we solely used classification techniques, precision levels were low. When classification and PCA are used together, precision improves. Equation (16), which displays recall for several classifications both with and without DR, is found in Table 3. Following are the recalls for each digit (0 to 9) from memory. Recall values for NB were extremely low. When we combine categorization with PCA, LDA and t-SNE, recall climbed. Figures 4 and 5 clearly explain that the performance measures explanation in this proposed method attain more improvement compared to other methods like NB, SVM, SNE-KNN, KNN, LDA-NB, PCA-SVM and PCA-KNN. Moreover, in terms of performance and real-time efficiency also been improved. The proposed method with deep learning can give average Recall of 96%, and average F1 score is 96%, which is a good improvement.

Table 2. Precision of classification both with and without a DR

Precision												
	NB	**SVM**	**KNN**	**t-SNE +NB**	**t-SNE +SVM**	**t-SNE +KNN**	**LDA +NB**	**LDA +SVM**	**LDA +KNN**	**PCA +NB**	**PCA +SVM**	**PCA +KNN**
0	0.68	0.94	0.94	0.93	0.92	0.95	0.94	0.94	0.94	0.96	0.96	**0.98**
1	0.79	0.96	0.95	0.92	0.94	0.96	0.96	0.93	0.92	**0.98**	0.97	**0.98**
2	0.87	0.89	0.95	0.91	0.89	0.93	0.88	0.88	0.90	0.81	0.92	**0.98**
3	0.66	0.87	0.92	0.73	0.83	0.91	0.87	0.88	0.88	0.83	0.92	**0.97**
4	0.84	0.89	0.93	0.80	0.82	0.90	0.88	0.88	0.88	0.86	0.91	**0.97**
5	0.48	0.88	0.93	0.81	0.80	0.89	0.82	0.85	0.87	0.78	0.91	**0.97**
6	0.68	0.96	0.96	0.93	0.93	0.95	0.92	0.93	0.95	0.93	0.95	**0.98**
7	0.92	0.93	0.93	0.72	0.77	0.91	0.93	0.92	0.96	0.93	0.96	**0.97**
8	0.28	0.92	0.97	0.87	0.93	0.94	0.79	0.87	0.90	0.85	0.94	**0.98**
9	0.41	0.92	0.91	0.76	0.69	0.90	0.85	0.89	0.90	0.84	0.94	**0.96**

Table 3. Recall of classification with and without DR

Recall												
	NB	**SVM**	**KNN**	**t-SNE +NB**	**t-SNE +SVM**	**t-SNE +KNN**	**LDA +NB**	**LDA +SVM**	**LDA +KNN**	**PCA +NB**	**PCA +SVM**	**PCA+ KNN**
0	0.91	0.97	0.99	0.98	0.98	0.98	0.95	0.96	0.98	0.94	0.98	**1.00**
1	0.95	0.98	**0.99**	0.96	0.98	0.98	0.93	0.96	0.97	0.93	0.98	**0.99**
2	0.20	0.90	0.92	0.86	0.84	0.93	0.87	0.87	0.91	0.86	0.92	**0.97**
3	0.33	0.89	0.94	0.84	0.88	0.91	0.85	0.87	0.88	0.85	0.92	**0.97**
4	0.08	0.95	0.94	0.90	0.87	0.93	0.92	0.94	0.95	0.86	0.97	**0.98**
5	0.03	0.85	0.91	0.72	0.83	0.89	0.82	0.84	0.84	0.85	0.90	**0.96**
6	0.92	0.95	0.97	0.94	0.90	0.96	0.92	0.93	0.94	0.91	0.95	**0.99**
7	0.26	0.93	0.93	0.79	0.75	0.91	0.87	0.90	0.92	0.86	0.94	**0.98**
8	0.72	0.86	0.89	0.81	0.81	0.88	0.85	0.84	0.83	0.87	0.91	**0.95**
9	0.94	0.87	0.91	0.57	0.66	0.87	0.86	0.86	0.89	0.83	0.90	**0.96**

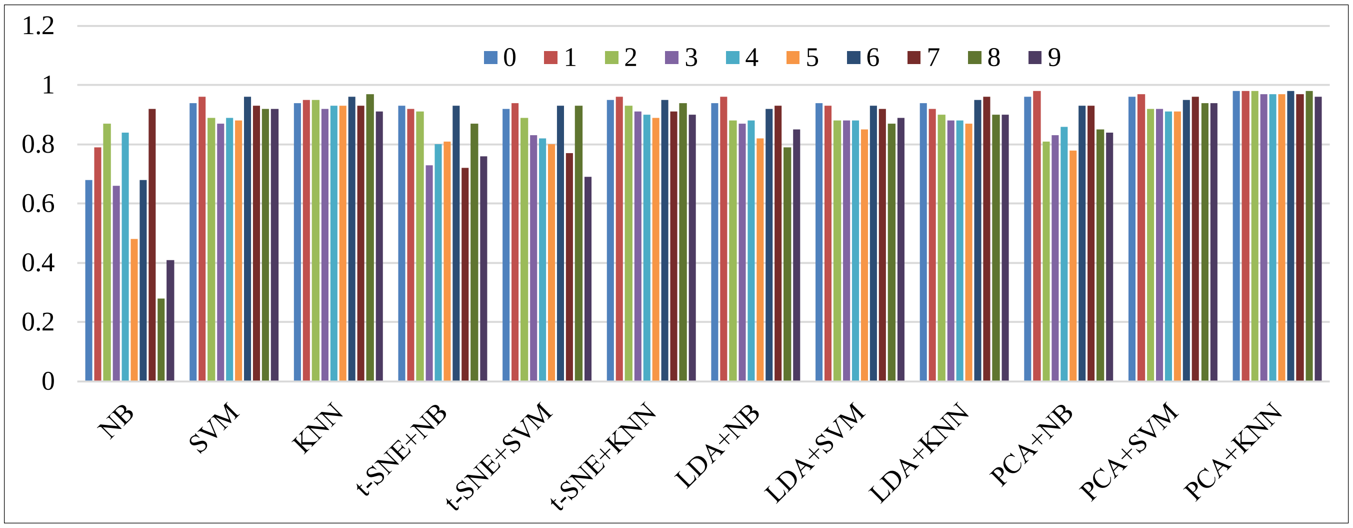

Fig. 4 Precision classifiers' precipitation with and without DR

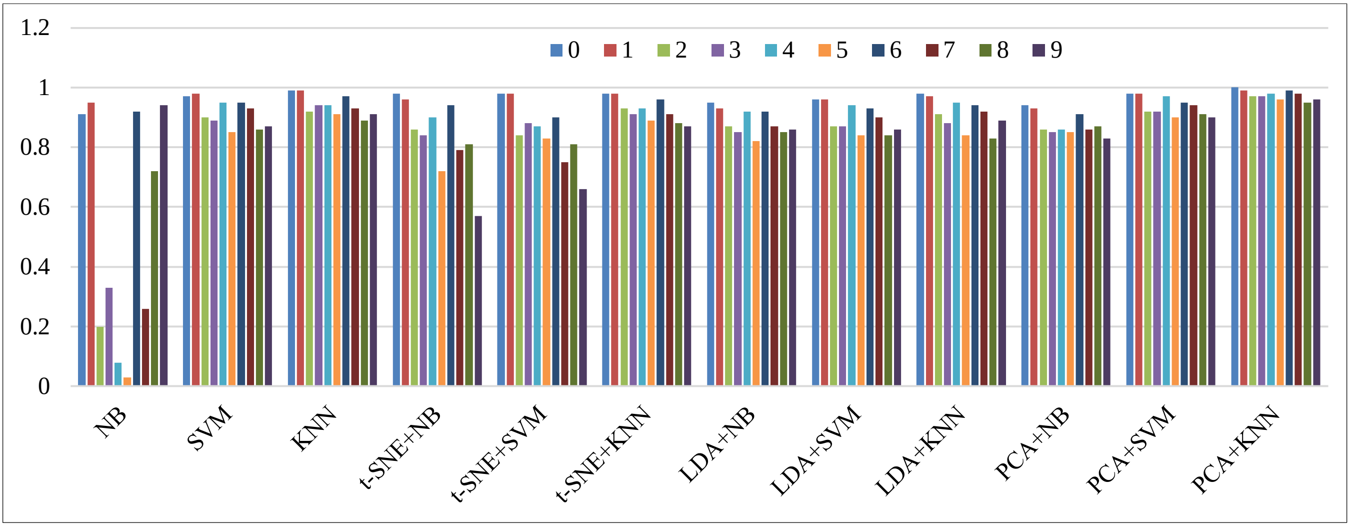

Fig. 5 Recall of classifiers with and without DR

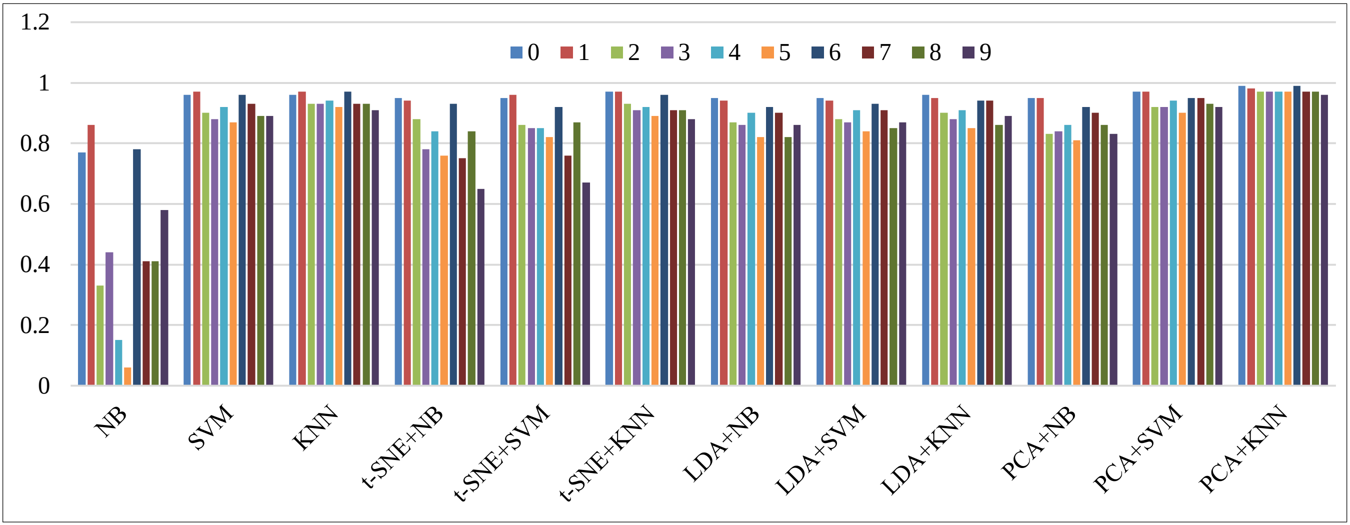

Fig. 6 F1-score of classifiers with and without DR

Figure 6 explains about F1-score of classifiers with and without DR, in this proposed method attains more improvement. Using Equation (17), the F1-score of several classifications with and without DR is displayed. The following are the F1 scores for accurately predicting each digit (0 to 9). When we solely use categorization, our F1 score is really low. When we combine classification with PCA, LDA and t-SNE, we can see that these values have increased are shown in Table 4.

Table 4. F1-score of classification with and without DR

	F1-Score											
	NB	**SVM**	**KNN**	**t-SNE +NB**	**t-SNE +SVM**	**t-SNE +KNN**	**LDA +NB**	**LDA +SVM**	**LDA +KNN**	**PCA +NB**	**PCA +SVM**	**PCA+ KNN**
0	0.77	0.96	0.96	0.95	0.95	0.97	0.95	0.95	0.96	0.95	0.97	**0.99**
1	0.86	0.97	0.97	0.94	0.96	0.97	0.94	0.94	0.95	0.95	0.97	**0.98**
2	0.33	0.90	0.93	0.88	0.86	0.93	0.87	0.88	0.90	0.83	0.92	**0.97**
3	0.44	0.88	0.93	0.78	0.85	0.91	0.86	0.87	0.88	0.84	0.92	**0.97**
4	0.15	0.92	0.94	0.84	0.85	0.92	0.90	0.91	0.91	0.86	0.94	**0.97**
5	0.06	0.87	0.92	0.76	0.82	0.89	0.82	0.84	0.85	0.81	0.90	**0.97**
6	0.78	0.96	0.97	0.93	0.92	0.96	0.92	0.93	0.94	0.92	0.95	**0.99**
7	0.41	0.93	0.93	0.75	0.76	0.91	0.90	0.91	0.94	0.90	0.95	**0.97**
8	0.41	0.89	0.93	0.84	0.87	0.91	0.82	0.85	0.86	0.86	0.93	**0.97**
9	0.58	0.89	0.91	0.65	0.67	0.88	0.86	0.87	0.89	0.83	0.92	**0.96**

7. Conclusion

This study investigates the influence of dimensionality reduction-DR techniques like PCA, LDA, and t-SNE on machine learning classification algorithms. The MNIST contains 784 features totaling 42,000 annotated grayscale images (28 × 28 pixels). Results have been identified using ML Classification applied to raw and reduced datasets.

Combining the dimensionality reduction and ML classification methods improves performance. Future applications of the DR technique's potency can include text and image datasets (both of which have large dimensionality). In future researchers can test more classification techniques in order to improve performance.

Acknowledgements

Thanks to Indian Institute of Technology Hyderabad (IITH) and TiHAN which is providing supporting software tools to do smooth research. Special thanks to Dr Kayam Saikumar, post-doctoral fellowship in India (PostDoc fellow from IITH), who is giving support in model training and testing scenarios.

References

[1] Adiwijaya et al., "Dimensionality Reduction Using Principal Component Analysis for Cancer Detection based on Microarray Data Classification," *Journal of Computer Science*, vol. 14, no. 11, pp. 1521-1530, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[2] Gustavo Eloi de Rodrigues, Wilson Marcílio, and Danilo Eler, "Data Classification: Dimensionality Reduction Using Combined and Non-Combined Multidimensional," *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, Sao Paulo, Brazil, pp. 402-407, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Pitoyo Hartono, "Classification and Dimensional Reduction Using Restricted Radial Basis Function Networks," *Natural Computing Applications*, vol. 30, pp. 905-915, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] Md. Golam Saroware et al., "Performance Evaluation of Feature Extraction and Dimensionality Reduction Techniques on Various Machine Learning Classifiers," *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, Tiruchirappalli, India, pp. 19-24, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Hany Yan, and Hu Tianyu, "Unsupervised Dimensionality Reduction for High-Dimensional Data Classification," *Machine Learning Research*, vol. 2, no. 4, pp. 125-132, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] G. Thippa Reddy et al., "Analysis of Dimensionality Reduction Techniques on Big Data," *IEEE Access*, vol. 8, pp. 54776-54788, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Areej Alsaafin, and Ashraf Elnagar, "A Minimal Subset of Features Using Feature Selection for Handwritten Digit Recognition," *Journal of Intelligent Learning Systems and Applications*, vol. 9, no. 4, pp. 55-68, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] Abbas Mardani et al., "A Multi-Stage Method to Predict Carbon Dioxide Emissions Using Dimensionality Reduction, Clustering, and Machine Learning Techniques," *Journal of Cleaner Production*, vol. 275, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Rizgar 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. 1, pp. 56-70, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Drishti Beohar, and Akhtar Rasool, "Handwritten Digit Recognition of MNIST Dataset Using Deep Learning State-of-the-Art Artificial Neural Network (ANN) and Convolutional Neural Network (CNN)," *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, Pune, India, pp. 542-548, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Tausifa Jan Saleem, and Mohammad Ahsan Chishti, "Assessing the Efficacy of Machine Learning Techniques for Handwritten Digit Recognition," *International Journal of Computing and Digital Systems*, vol. 9, no. 2, pp. 299-308, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Cheng-Lung Huang, and Jian-Fan Dun, "A Distributed PSO–SVM Hybrid System with Feature Selection and Parameter Optimization," *Applied Soft Computing*, vol. 8, no. 4, pp. 1381-1391, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[13] Y. Lecun et al., "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[14] M. RamakrishnaMurty, J.V.R. Murthy, and Prasad Reddy P.V.G.D., "Text Document Classification Based on a LeastSquare Support Vector Machines with Singular Value Decomposition," *International Journal of Computer Application (IJCA)*, vol. 27, no. 7, pp. 21-26, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[15] Lingyun Wang et al., "Short-Term Power Load Forecasting Model Based on t-SNE Dimension Reduction Visualization Analysis, VMD and LSSVM Improved with Chaotic Sparrow Search Algorithm Optimization," *Journal of Electrical Engineering & Technology*, vol. 17, pp. 2675-2691, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] I.T. Jolliffe, *Generalizations and Adaptations of Principal Component Analysis*, Principal Component Analysis, New York, pp. 223-234, 1986. [CrossRef] [Google Scholar] [Publisher Link]

[17] Olivier Chapelle et al., "Choosing Multiple Parameters for Support Vector Machines," *Machine Learning*, vol. 46, pp. 131-159, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[18] T. Cover, and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, 1967. [CrossRef] [Google Scholar] [Publisher Link]

[19] L. Breiman, "Bagging Forests [J]", *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001. [Google Scholar]

[20] D. Padmaja Usharani et al., "Classification of High-Dimensionality Data Using Machine Learning Techniques," *Intelligent System Design*, vol. 494, pp. 227-237, 2022. [CrossRef] [Google Scholar] [Publisher Link]