*International Journal of Engineering Applied Sciences and Technology, 2020*
**Vol. 5, Issue 1, ISSN No. 2455-2143, Pages 290-296**
Published Online May 2020 in IJEAST (http://www.ijeast.com)

# AN EXPERIMENTAL REVIEW ON EFFECT OF PRINCIPAL COMPONENT ANALYSIS ON MACHINE LEARNING TECHNIQUES

Hari Krishna Modalavalasa,
ECE Dept., JNTUH-CEH,
JNT University-Hyderabad,
Hyderabad, Telangana, India.

Madhavi Latha Makkena
ECE Dept., JNTUH-CEH,
JNT University-Hyderabad,
Hyderabad, Telangana, India

*Abstract*— **Nowadays, huge amount of data is generating all around the world due to enhancements in the digital technology. Artificial Intelligent systems analyze the data, extract some useful hidden structures and create models for classification or regression of future data. Machine Learning is a subset of Artificial Intelligence, which is capable to design of intelligent models based on historical relationships or trends inside data. Data Mining and Machine Learning techniques playing very crucial role in analysis and management of huge profusion of data. Principal Component Analysis (PCA) reduces the complexity of data analysis by transforming the components as linear uncorrelated components, decreasing dimensions of data without effecting the accuracy of the model. This paper presents an experimental comparative analysis on the effect of PCA in 24 different machine learning models. Considered 24 Machine Learning techniques are implemented and trained with standard and multivariate Fisher's Iris data set. This dataset has 150 samples of three species of Iris (setosa, virginica and versicolor) and each sample has four features (lengths and the widths of the sepals and petals). A comparative analysis is made in terms of clustering accuracies and learning times at different levels of PCA.**

*Keywords*— **Clustering, Machine Learning, Principal Component Analysis**

## I. INTRODUCTION

The enhancements in technology leads the society towards big data. Specifically, the digital technology provides enormous real time sensors and processors at lower sizes and costs. Nowadays sensors are very common in all the digital devices and these sensors generating a profusion amount of data all around the world. Manual analysis of huge amount of data is practically impossible with rapid increment in data generation. To handle this problem, Machine Learning (ML) algorithms are adopted. Machine Learning algorithms gradually consumes infinite amount of data and generates models by analyzing the hidden structures in the data. A lot of research has been going in this area and many algorithms are developed. Learning from very huge databases is always a challenging issue in ML and Data Mining algorithms [1][2][3]. Machine Learning algorithms are broadly classified into the categories based on availability of input data and labels like Supervised Machine Learning, Unsupervised Machine Learning and Reinforcement Machine Learning. Supervised Machine Learning needs huge labelled data and used for regression and classification. Unsupervised Machine learning extracts the hidden structures and relationships present inside given unlabeled data and form clusters[4][5]. Reinforcement learning is only option if environment data is not available prior to application of algorithm. It directly interacts with the real time environment and searches for the solution with maximum reward. ML algorithms have their applications in diverge fields. ML algorithms drawn a huge amount of research in many fields like communication, controls, financial businesses, bioinformatics, medicine, marketing. These fields have extensive sets of raw data, which are stored. The advancements on cloud computing technologies also yielding enormous amounts of data [6]. Even though many ML algorithms developed for data mining, big data challenges may require the redesign of the existing algorithms. Simultaneously, the machine learning algorithms facing major challenges with dimensionality of the problem. In order to resolve this issue, Feature Selection and dimensionality reduction techniques can be employed for big data analysis, prior to the application of any data mining methods like clustering, regression, classification [2]. Principal Component Analysis (PCA) methods can be applied to reduce the complexity of the input data, which converts the data into Linear space [7][8].

PCA is reduction method which considers the input dataset as set of rows representing characteristics in a high dimensional space and all rows are put up to a direction which represents the best set of features. After this the PCA generates an axis that contains principle eigenvector where all the points of all observations of each feature are spread out. At this point PCA finds the maximized variance of data on this axis. After that for second eigenvector, PCA observes axis along which the variance of distance from first axis is greatest and so on. set of eigenvectors are represented by matrix of points, then to minimize the root mean square (RMS) error, it approximates the data for the given number of columns in the matrix consider. Finally, the original features of input data are approximated by PCA with fewer dimensions [1].

## II. METHODOLOGY

In this work, 24 different ML algorithms are implemented for data clustering [see in Table-1]. This work includes different versions of trees, KNN and SVM models [9][10][11]. Fisher's Iris data set is considered as problem data set[12]. This Iris data set is standard, multivariate and introduced by British statistician Ronald Fisher. This Dataset has 3 types of iris species like Setosa, Virginica and Versicolor and each iris type has 50 samples. In total 150 samples, each sample has 4 features as sepal length, sepal width, petal length and petal width. All the 24 machine learning algorithms are applied on considered data set and their performance is compared in terms of clustering accuracy and learning time. Principal Component Analysis is applied on data before applying actual algorithm by incorporating into all the algorithms. The PCA is applied at different levels. As the data has 4-dimensional feature set and required at least two features out of four, PCA is applied with maximum feature selection count 2 and 3. Finally a comprehensive analysis is made to identify the best and worst performers and to analyze the effect of PCA on the performance of each ML algorithm.

## III. RESULTS & DISCUSSIONS

All the 24 algorithms are developed in MATLAB r2019b software. Simulation setup uses a workstation laptop with Hexa-Core Intel Core-i7 [ 9th generation- 9750] processor with Nvidia GTX 1660-Ti GPU and 32GB RAM. All the timing calculations are with respect to MATLAB parallel Processing calculations with six Processors).

First all the algorithms are applied on Iris dataset without PCA with dimensionality 4 (Dim4) and accuracy and learning times are analyzed. After that PCA is applied on dataset to minimize the dimensionality to 3 and then all algorithms are applied and performances are analyzed. Again, PCA is applied on dataset to further reduce the dimensionality to 2 and performance of all algorithms are analyzed. Finally, a comparative analysis is made to analyze the effect of PCA on different algorithms at different dimensionality levels. Fig.1 and Fig.2 shows the actual Iris data set in sepal width vs sepal length plot and petal length vs sepal width plot respectively. In all figures in this work, Setosa spices are represented with Blue color, Virginica spices are represented with Red color and Versicolor spices are represented with Yellow color. In considered 24 algorithms, both Linear Discriminant algorithm and Quadratic SVM algorithm clustered the Iris data in better way with an accuracy of 98% without any PCA applied to Fisher Iris Data. The clustered Iris Data after the application of Linear Discriminant algorithm is shown in Fig.3. In this figure the Cross-marks on the circular dots represents the invalid or incorrect clustered data and the

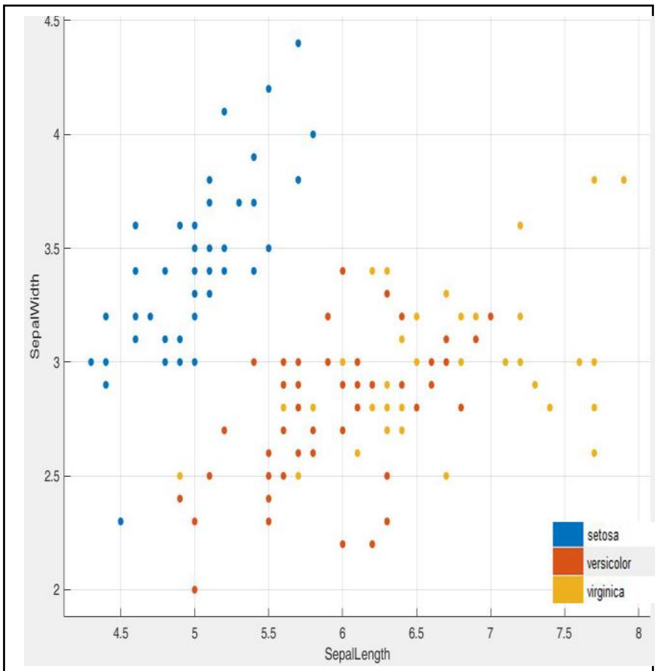


Fig.1. Original Fisher Iris Data on sepal-length vs sepal-width plot

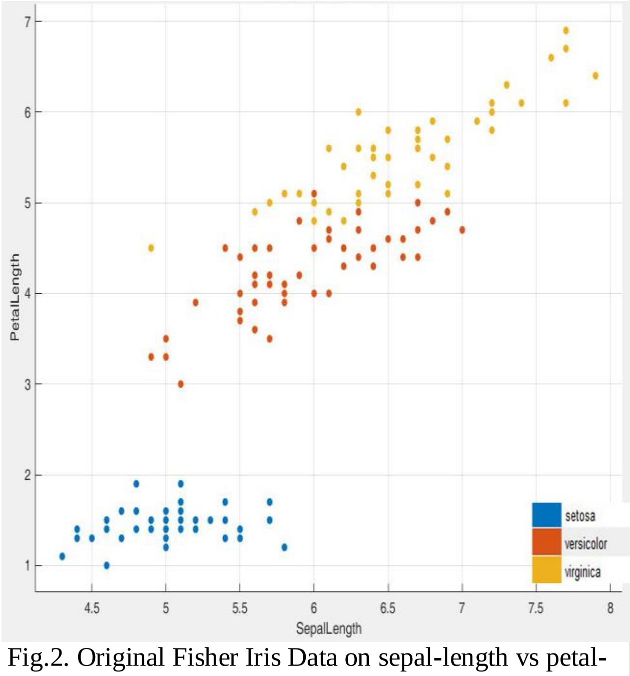


Fig.2. Original Fisher Iris Data on sepal-length vs petal-length plot

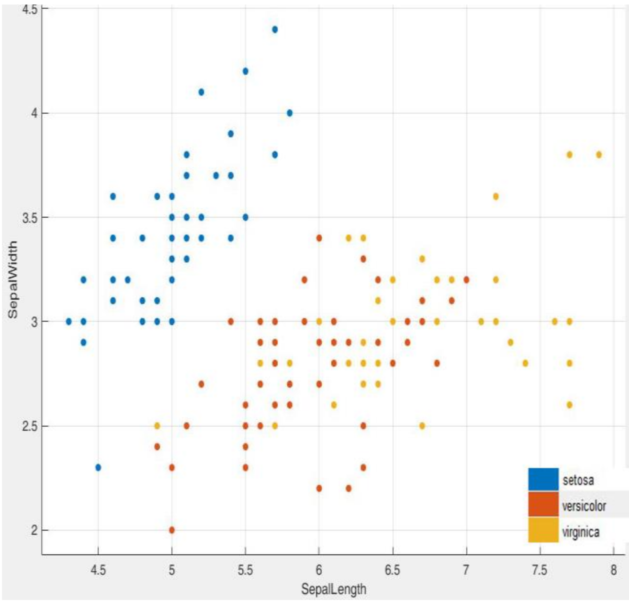


Fig.3. Clustered Fisher Iris Data from LDA without PCA on sepal-length vs sepal-width plot

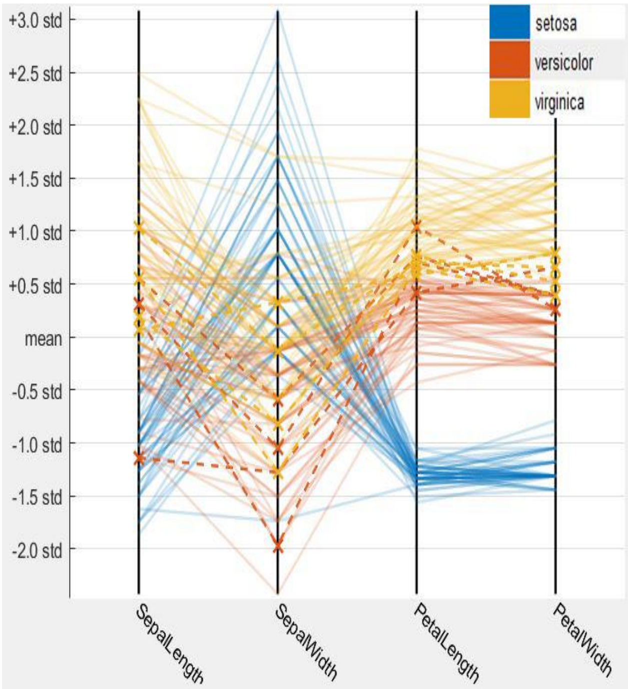


Fig.4. Clustered Fisher Iris Data from LDA without PCA on parallel analysis plot

circular dots represents accurately clustered data in their original category. The category representation is given by color of each dot. The error in the clustering procedure can be clearly visualized by the parallel plot as shown in Fig.4. In this parallel plot, four vertical plots represent the four features of the given Iris data set and connected lines between these feature lines represents each entry in the data set with specific color classification dedicated to their category. In this figure, the dotted lines represent the invalid classification of data using the Linear Discriminant algorithm. The confusion matrix shows the accuracy of any machine learning algorithm, here Fig.5 represents the confusion matrix of the Linear Discriminant algorithm which has the statistical analysis of the algorithm. Form the confusion matrix we can observe that, all 50 Setosa samples are classifies as Setosa but out of 50 Versicolor samples only 48 samples are truly classified and remaining 2 samples classified as Virginica samples and out of 50 Virginica samples 49 samples classified truly and one sample classified as Versicolor.

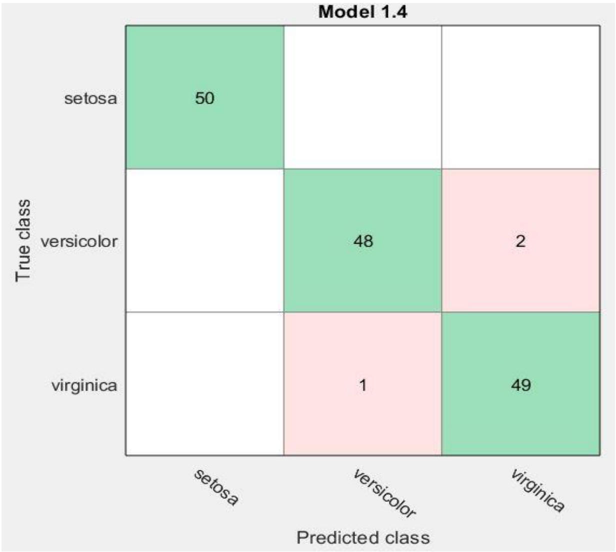


Fig.5. Confusion matrix of LDA without PCA

Out of considered 24 algorithms, Boosted Trees and RUS-Boosted Trees algorithms shows lesser accuracy in terms of classification statistics. In terms of resource utilization and computation time metrics, Coarse Tree gave best performance out of 24 algorithms with 12.06 seconds and Subspace KNN is the most time-consuming algorithm with 30.79 seconds of computation time.

After applying the PCA to reduce the feature dimensionality size from 4 to 3 (Dim3p), both Linear Discriminant Algorithm (LDA) and Quadratic SVM algorithm gave the best clustering accuracy out of 24 algorithms with 98.7% accuracy. The clustered data from this algorithm is shown in Fig.6 and parallel analysis plot is shown in Fig.7, which shows the invalid clustered data samples with dotted lines on vertical feature axes. The confusion matrix for this algorithm after applying PCA is shown in the Fig.8, which shows the classification metrices. Here all 50 samples of setosa are classified as Setosa samples and all 50 samples of Virginica samples are classified as Virginica Samples but out of 50 samples of Versicolor only 48 are classified truly and remaining 2 are classified as virginica. In terms of resource utilization and computation time metrics, Gaussian Navie Bayes algorithm gave best performance out of 24 algorithms with 15.69 seconds and subspace discriminant algorithm is the most time-consuming algorithm with 38.74 seconds of computation time.

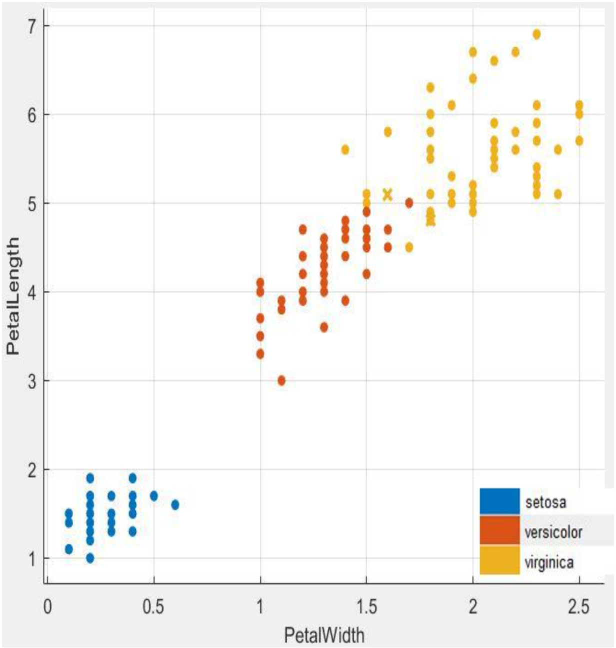


Fig.6. Clustered Fisher Iris Data from LDA with PCA and dimensionality 3 on sepal-length vs sepal-width plot

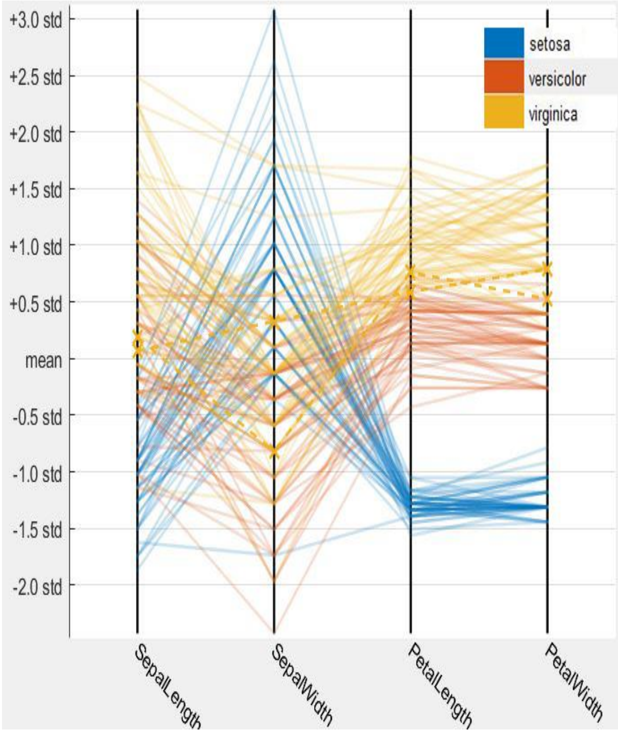
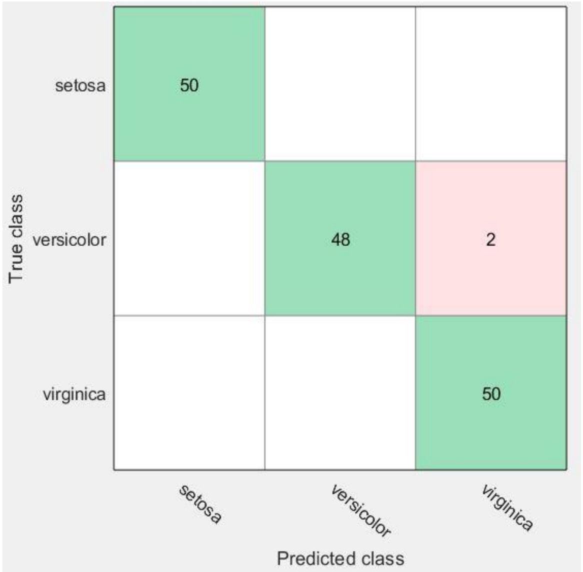


Fig.7. Clustered Fisher Iris Data from LDA with PCA and dimensionality 3 on parallel analysis plot


True class
Predicted class

| True class \ Predicted class | setosa | versicolor | virginica |
|---|---|---|---|
| setosa | 50 | | |
| versicolor | | 48 | 2 |
| virginica | | | 50 |

Fig.8. Confusion matrix of LDA with PCA and dimensionality 3

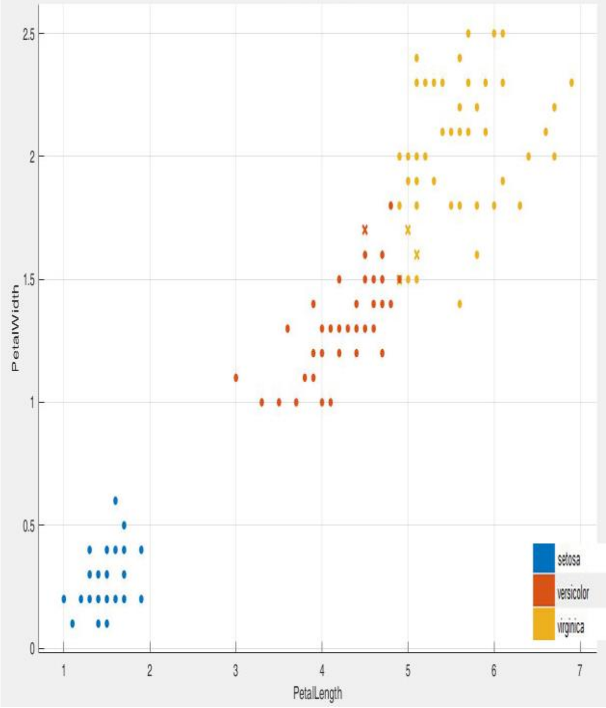


Fig.9. Clustered Fisher Iris Data from LDA with PCA and dimensionality 2 on sepal-length vs sepal-width plot

The dimensionality of the feature size is further reduced from 4 to 2 (Dim2p) by applying the PCA to the Fisher Iris data set. All the considered 24 algorithms are again applied on Fisher Iris Data by considering only 2 major influencing features out of 4 available features. Here Linear Discriminant and Quadratic Discriminant algorithms gave the best classification accuracy with 96.7% accuracy. Fig.9 shows the clustered data from Linear Discriminant algorithm with color-based labels. The parallel analysis plot is shown in the Fig.10 where dotted lines indicate the incorrectly clustered samples. The confusion matrix of this algorithm is shown in Fig.11 which shows the classification statistics. All the Setosa samples are classified correctly but Three Versicolor samples are classified incorrectly as Virginica samples and one Virginica sample classified incorrectly as Versicolor.

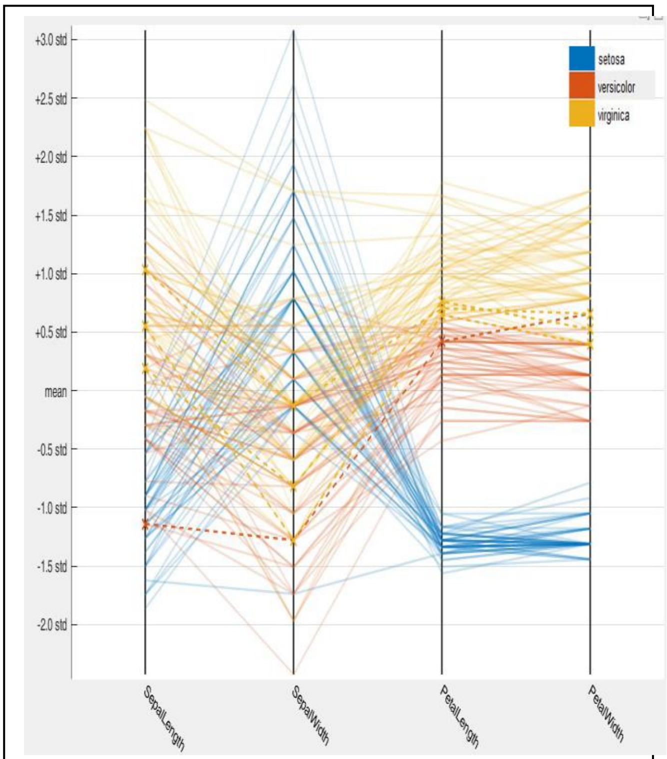


Fig.10. Clustered Fisher Iris Data from LDA with PCA and dimensionality 2 on parallel analysis plot

Here, the dimensionality reduction leads incorrect classifications and reduces the accuracy from 98% to 96.7%. In terms of resource utilization and computation time metrics, fine tree algorithm gave best performance out of 24 algorithms with 3.54 seconds and subspace KNN is the most time-consuming algorithm with 33.31 seconds of computation time.

Table-1 shows the overall performance statistics both in terms of classification accuracy and learning time for all considered 24 algorithms. This table includes the statistics of all three cases mentioned above (without applying PCA with feature dimensionality size 4, with application of PCA and feature size 3, with application of PCA and feature size 2). To ease understand the performance of each algorithm in all cases, the complete data is visualized in a Glyph plot as shown in Fig.12. In this glyph plot top side corners represent performance in terms of accuracy and bottom side corners of each shape represents performance in terms of learning time. So, the shape with higher upper area and low lower area are better suits for the considered application and dataset. In plot

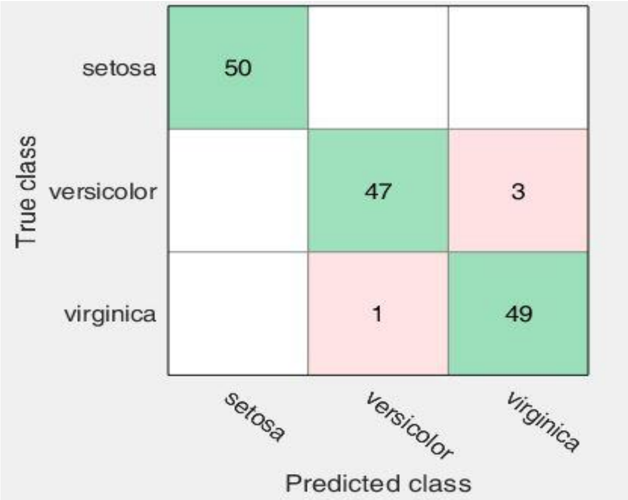


Fig.11. Confusion matrix of LDA with PCA and dimensionality 2.

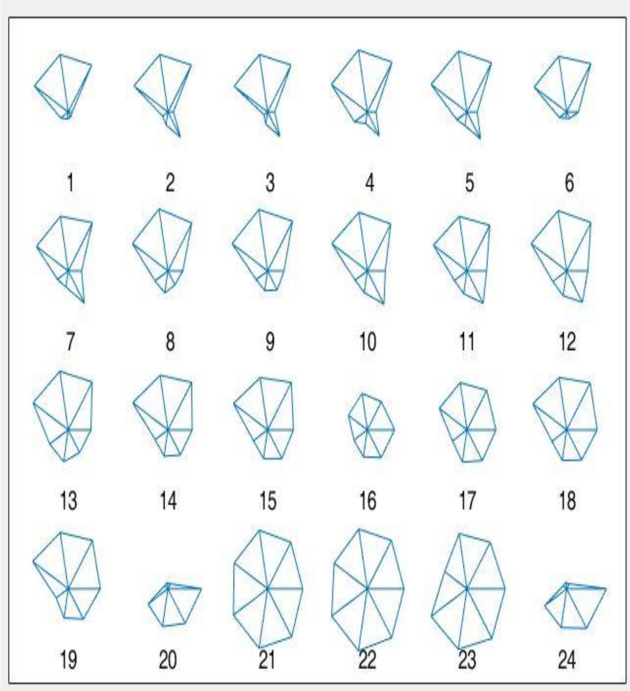


Fig.12. Glyph plot of all 24 algorithms with Accuracy and time statistics

shape 1 to shape 6 have best upper to lower are ration means these algorithms are highly accurate with lower learning time. Shape 20 and shape 24 has worst upper to lower area ratio means these algorithms have low accuracy with higher learning times. Shapes 21 and 22 have the upper to lower area ration near unity means these algorithms provides better accuracy but requires larger learning times. A triangle like structure in upper half of all the shapes clearly showing that the classification accuracy increases with the application of

TABLE I. CLUSTER ACCURACY AND LEARNING TIME STATISTICS

| AlgNo | Algorithm \ Performance | Accuracy (Percentage) | | | Learning Time (Seconds) | | |
|---|---|---|---|---|---|---|---|
| | | Dim4 | Dim3p | Dim2p | Dim4 | Dim3p | Dim2p |
| 1 | Fine Tree | 94.0 | 93.3 | 92.7 | 14.41 | 16.00 | 3.541 |
| 2 | Medium Tree | 94.0 | 93.3 | 92.7 | 13.07 | 18.49 | 17.10 |
| 3 | Coarse Tree | 94.0 | 93.3 | 92.0 | 12.06 | 18.19 | 16.71 |
| 4 | Linear Discriminant | 98.0 | 98.7 | 96.7 | 17.42 | 17.87 | 16.23 |
| 5 | Quadratic Discriminant | 97.3 | 97.3 | 96.7 | 15.87 | 18.74 | 18.21 |
| 6 | Gaussian Navie Bayes | 96.0 | 91.3 | 90.0 | 14.20 | 15.69 | 4.486 |
| 7 | Kernal Navie Bayes | 96.0 | 90.0 | 88.7 | 16.26 | 19.77 | 21.56 |
| 8 | Linear SVM | 95.3 | 98.7 | 96.0 | 17.48 | 22.19 | 9.519 |
| 9 | Quadratic SVM | 98.0 | 98.0 | 96.0 | 16.59 | 21.30 | 12.91 |
| 10 | Cubic SVM | 94.0 | 94.7 | 96.0 | 16.99 | 22.08 | 22.38 |
| 11 | Fine gaussian SVM | 92.0 | 90.0 | 92.0 | 17.25 | 22.53 | 21.82 |
| 12 | Medium gaussian SVM | 96.0 | 96.7 | 96.0 | 18.02 | 23.34 | 21.12 |
| 13 | Coarse gaussian SVM | 95.3 | 94.7 | 95.3 | 20.68 | 26.35 | 15.38 |
| 14 | Fine KNN | 94.7 | 90.0 | 95.3 | 17.38 | 24.03 | 16.93 |
| 15 | Medium KNN | 94.7 | 87.3 | 93.3 | 18.17 | 25.00 | 19.29 |
| 16 | Coarse KNN | 66.7 | 68.7 | 63.3 | 17.68 | 24.28 | 18.71 |
| 17 | Cosine KNN | 83.3 | 81.3 | 80.7 | 18.37 | 26.47 | 20.71 |
| 18 | Cubic KNN | 94.0 | 87.3 | 92.7 | 18.55 | 25.81 | 20.20 |
| 19 | weighted KNN | 96.0 | 92.0 | 96.0 | 16.95 | 25.18 | 20.38 |
| 20 | Boosted Trees | 33.3 | 33.3 | 33.3 | 21.50 | 28.73 | 23.24 |
| 21 | Bagged Trees | 93.3 | 94.7 | 92.0 | 29.82 | 37.74 | 32.25 |
| 22 | Subspace Discriminant | 96.0 | 96.0 | 91.3 | 30.62 | 38.74 | 32.77 |
| 23 | subspace KNN | 93.3 | 90.7 | 70.0 | 30.79 | 38.15 | 33.31 |
| 24 | RUS-Boosted Trees | 33.3 | 33.3 | 33.3 | 22.78 | 29.17 | 26.57 |

the PCA on the dataset up to some point but further reduction in dimensionality leads lesser accurate clusters due to losing of useful learning features. On the side, the learning time also following same manner for most of the algorithms. In all the cases the Linear Discriminant algorithm exhibits the best accuracy of clustering with average learning times.

## IV. CONCLUSION

An experimental analysis on different machine learning algorithms is performed by implementing 24 different machine learning algorithms and by applying on Fisher Iris data with 150 samples with 4 features each. The dataset is applied to Principal Component Analysis to reduce the dimensionality to different levels and at each level all 24 implemented algorithms are applied on dataset and accuracy & learning time statistics are analyzed. Out of 24 algorithms, Linear Discriminant Algorithm gave the best accuracy in clustering of Fisher Iris data. Before applying the PCA the LDA algorithm gave 98% accuracy with a learning time of 17.42 seconds. After application of PCA, with reduced dimensionality 3, the accuracy of the LDA increased to 98.7% but the learning time also slightly increased to 17.87 seconds. Further reduction in dimensionality with PCA effects the accuracy of LDA algorithm but improves the learning time. The learning time of some algorithms like Fine Tree algorithm rapidly decreasing with reduction in the dimensionality. For Fine Tree algorithm the learning time reduced nearly 75% with the reduction of dimensionality from 4 to 2 where the Accuracy is decreased by only 1.4%. This kind of integration of machine learning techniques with PCA is very useful in high speed low precision systems. With this analysis, it is evident that proper application of PCA improves the performance of the machine learning algorithms

## V. REFERENCE

[1] B. Corona, M. Nakano, H. Pérez, “Adaptive Watermarking Algorithm for Binary Image Watermarks”, *Lecture Notes in Computer Science, Springer, pp. 207-215, 2004.*

[2] A. A. Reddy and B. N. Chatterji, "A new wavelet based logo-watermarking scheme," Pattern Recognition Letters, vol. 26, pp. 1019-1027, 2005.

[3] P. S. Huang, C. S. Chiang, C. P. Chang, and T. M. Tu, "Robust spatial watermarking technique for colour images via direct saturation adjustment," Vision, Image and Signal Processing, IEE Proceedings -, vol. 152, pp. 561-574, 2005.

[4] F. Gonzalez and J. Hernandez, " A tutorial on Digital Watermarking ", In IEEE annual Carnahan conference on security technology, Spain, 1999.

[5] D. Kunder, "Multi-resolution Digital Watermarking Algorithms and Implications for Multimedia Signals", Ph.D. thesis, university of Toronto, Canada, 2001.

[6] J. Eggers, J. Su and B. Girod," Robustness of a Blind Image Watermarking Scheme", Proc. IEEE Int. Conf. on Image Proc., Vancouver, 2000.

[7] Barni M., Bartolini F., Piva A., Multichannel watermarking of color images, IEEE Transaction on Circuits and Systems of Video Technology 12(3) (2002) 142-156.

[8] Kundur D., Hatzinakos D., Towards robust logo watermarking using multiresolution image fusion, IEEE Transcations on Multimedia 6 (2004) 185-197.

[9] C.S. Lu, H.Y.M Liao, “Multipurpose watermarking for image authentication and protection,” *IEEE Transaction on Image Processing*, vol. 10, pp. 1579-1592, Oct. 2001.

[10] L. Ghouti, A. Bouridane, M.K. Ibrahim, and S. Boussakta, “Digital image watermarking using balanced multiwavelets”, *IEEE Trans. Signal Process*., 2006, Vol. 54, No. 4, pp. 1519-1536.

[11] P. Tay and J. Havlicek, "Image Watermarking Using Wavelets", in *Proceedings of the 2002 IEEE*, pp. II.258 – II.261, 2002.

[12] P. Kumswat, Ki. Attakitmongcol and A. Striaew, "A New Approach for Optimization in Image Watermarking by Using Genetic Algorithms", *IEEE Transactions on Signal Processing*, Vol. 53, No. 12, pp. 4707-4719, December, 2005.

[13] H. Daren, L. Jifuen,H. Jiwu, and L. Hongmei, "A DWT-Based Image Watermarking Algorithm", in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 429-432, 2001.

[14] C. Hsu and J. Wu, "Multi-resolution Watermarking for Digital Images", *IEEE Transactions on Circuits and Systems- II*, Vol. 45, No. 8, pp. 1097-1101, August 1998.

[15] R. Mehul, "Discrete Wavelet Transform Based Multiple Watermarking Scheme", in *Proceedings of the 2003 IEEE TENCON*, pp. 935-938, 2003.