Analysis of the Behavior of Reduced and Compressed Data with Various Learning Algorithms

Swati Kaur
M. Tech Student
Department of Computer Science & Engineering
Rungta College of Engineering & Technology Bhilai

S. M. Ghosh
Associate Professor
Department of Computer Science & Engineering
Rungta College of Engineering & Technology Bhilai

Abstract

In machine learning and statistics, classification is issue of recognizing to which of an arrangement of classes a new observation belongs, on the basis of training set of information containing perceptions (or cases) whose classification enrollment is known. Progresses in information collection and storage capabilities amid the previous decades have prompted a data burden in many scientific areas. In this paper analysis of the performance of various classification algorithm is done on reduced and compressed data and concluded which data is more effective for classifying the data. Low Rank matrices perform the best among the three and Huffman's coding is preferred when we want to save memory space and AANNs is efficient when it to reduce the dimensionality of the data. For classification Low Rank Matrices produces more accurate result than other two.

Keywords: Dimension reduction, Data mining, data compression, auto associative neural network, Huffman's coding, Low rank Matrices

I. INTRODUCTION

Data mining is the extraction of riveting (important, relevant, constructive, previously unexplored and substantially valuable) patterns or information from gigantic pile of data. In other words, it is the exploration of links, associations and general patterns that prevail in extensive databases however are hidden or unknown. At present, the process of obtaining valuable information and facts from data has become more an art than science. Indeed, even before the information is gathered and prepared, an assumption of the nature of the knowledge to be extracted from the information exists in the human personality, subsequently the human instinct remain irreplaceable. Various techniques were developed for the extraction of data, each of them specially made for the specific set of information. Classification is a data mining function that allocates elements in a collection to target categories or classes. The goal of classification is to precisely foresee the objective class for each case in the data. For example, a classification model could be utilized to recognize loan candidates as low, medium, or high credit risks.

To interpret the data covered up in multidimensional data can be considered as trying furthermore, confused assignment. Ordinarily, dimension reduction or data compression is considered as the initial step to data analysis and investigation of multidimensional data. Statically and machine learning strategies confront a considerable issue when managing such high-dimensional data, and typically the quantity of data variables is decreased before an data mining algorithm can be effectively applied. The dimensionality reduction can be made in two unique courses: by just keeping the most relevant variables from the original dataset (this procedure is called feature selection) or by abusing the redundancy of the dataset and by finding a littler arrangement of new variables, each being a mix of the information variables, containing essentially the same data as the information variables (this system is called dimensionality reduction).

Here, Auto-Associative Neural Networks (AANNs) technique for data reduction. AANNs have the ability to deal with linear and nonlinear correlation among elements. This system often referred to as nonlinear Principal Component Analysis (NLPCA) or could likewise be known as Bottleneck Neural Networks (BNNs) due to their specific structures that consist of a blend of compression and decompression n/ws. The trained AANNs can deduce high dimensional data into lower dimensional information by compressing them on its bottleneck layer that later can be utilized for information representation. [1].

For data compression two techniques are being used-Huffman coding and Low Rank matrix. Huffman coding is a standout amongst the most prevalent procedure for removing coding redundancy. It has been used in various compression applications, including picture compression. It is a simple, yet elegant, compression technique that can supplement other compression algorithms. Huffman coding is a lossless information compression method. It is based on the frequency of occurrence of a data item i.e. pixel in images. The procedure is to utilize a lower number of bits to encode the data in to binary codes that happens all the more much of the time[2][3].

Another method used for data compression is Low Rank Matrix Approximation. In a matrix low rank approximation, given a matrix $R \in R$ $n\times m$, and a lower rank<rank(R), we find two matrices $P \in R$ $n\times k$ and $Q \in R$ $k\times m$ such that R is well approximated by PQ,i.e.,$R \approx PQ$. Low rank approximations vary depending on the constraints forced on the variables and in addition the measure

for the distinction between R and PQ. Low rank approximation draws huge interest in data mining and machine learning group, for its ability to address many foundational challenges in this territory. A couple of prominent systems of machine discovering that use low rank approximation are principal component analysis, element examination, inert semantic investigation, non-negative matrix factorization, etc In this research, AANNs have been used to perform reduction and two compression techniques has been applied to the data, Huffman coding and Low Rank Matrix approximation. Various Learning Algorithms has been applied to these data and performance analysis has been for separate learning algorithms. They have been redeveloped using MATLAB, applied and analyzed by computing it on multidimensional data of a well-known Iris flowers dataset.

II. Related work

Dimension reduction or attribute reduction of substantial data sets has dependably been a search area. From the perspective of classification, it is crucial to hold only those attributes that maximize the classification effectiveness. Data reduction manages not just reducing the number of attributes but also reducing the instances as well. The major of data reduction relies on attributes reduction. Consequently when the pre-processing is done for reducing attributes, the most important aspect is delivering the reduce with the same viability as the original data set.

Patil and Sane [5] performed a relative study for effective classification after data reduction. This study examined in brief the strategies of reduction with performing a correlation of accuracy after dimension reduction. They utilized the weka mining tool for implementing built in filters and fuzzy rough methodology. The outcomes demonstrated that fuzzy rough feature selection enhances the accuracy of artificial neural system classifiers.

Karegowda et al. [6] presented an examination between gain ratio and correlation feature selection. The study tended to the impact of those feature reduction methods for Pima Indian diabetic data set classification. Ali Anaissi[9] presented a system for a high dimensional data reduction based on three technologies: feature selection, linear dimensionality reduction and non-linear dimensionality reduction. Feature selection based on mutual information. Ibrahim M. El-Hasnony[20] proposed a method of more than one feature reduction technique from more than one previous study and hold comparison among 5 data reduction strategies to give general perspective about the effectiveness of each technique in enhancing the classification efficiency.

Zalhan Mohd Zin [7] proposed AANNs for being used to perform compression, clustering and visualization of multidimensional data. They have been developed using high level computer language, applied and analyzed by computing it on multidimensional data of a well-known Iris flowers and Italian olive oils datasets

Data compression plays a vital part in data mining in assessing the minability of data and a modality of evaluating similarities between complex objects. Compression algorithms make use of structure in data to achieve optimal compression I.e. the more structure; the simpler to compress that is patterns are used to compress data.

Aarti[4] presented an analysis of Huffman's coding in compressing the data. Huffman coding and decoding is used for scan testing to reduce test data volume, test data compression and decompression time. Hence we conclude that Huffman coding is efficient strategy for image compression and decompression to a few extents. The results also reveal that the original image used for coding is almost close to the decoded output image. Pankaj Kumar [8] has used double Huffman coding for compression. Using Double Huffman Coding we decrease the redundancy of the data and the efficiency of the system is increase also. Since we realize that space allocation is costly so utilizing this double Huffman coding we save the space and increase the performance of the System. Mamta Sharma [10] have analysed Huffman algorithm and compare it with other common compression algorithms like Arithmetic LZW and Run Length Encoding.

Ramakrishnan Kannan, Mariya Ishteva and Haesun Park [11] proposed an algorithm which is scalable for large matrices with missing elements on multi core frameworks with low memory. In this paper, we propose a new type of low rank approximation where the values of low rank matrix are restricted – that is, its elements are within a given range which we call as Bounded Matrix Low Rank Approximation (BMA).

Dimitris Achlioptas and Frank McSherry[12] technique amounts to independently sampling and/or quantizing the entries of A, thus speeding up computation by reducing the number of non-zero entries and/or the length of their representation. Yunhui Shi, He Li, Jin Wang, Wenpeng Ding, Baocai Yin[15] proposed a new method of inter prediction based on low-rank matrix completion. By collection and rearrangement, image regions with high correlations can be used to generate a low-rank or approximately low-rank matrix. We view prediction values as the missing part in an incomplete low-rank matrix, and obtain the prediction by recovering the generated low-rank matrix. .

Hossein Mobahi, Ce Liu, William T. Freeman [13] presented a simple compositional model of color, shape, and appearance to approximate image sets. The model is regularized by having shape and appearance representations be on a low-dimensional subspace, and having color be a global shift and rotation.

III. Methods

A. Auto Associative Neural Network:

AANNs have frequently been viewed as a distinct option for PCA for unsupervised learning, clustering and outlier detection [19] [21]. As depicted in [22], it is moreover known as Non Linear Principal Component Analysis (NLPCA) or can be otherwise called Bottleneck Neural Systems (BNN) [32]. AANNs are feed forward network and are trained to map Approximations of input vectors to their corresponding outputs. As described in [31], they can be viewed as circuits of highly interconnected units with adjustable interconnection weights. They can also be considered as a particular class of neural networks in which the objective result patterns is identical to its input patterns.

The structure of AANNs generally comprises of several layers: input layer, map layer, bottleneck layer, de-map layer and output layer as appeared in Figure 1. In this figure as well, the structure has three nodes in input and output layers, five nodes in map and de-map layers, and two nodes in bottleneck layer. This structure could also be seen as a combination of two varient networks, compression network and de-compression network, as mentioned in [17] [6] [31].

These two networks “join” in the middle at bottleneck layer. In compression network, the input is known but the output is not known while the opposite situation happens in de-compression network. During training of AANNs, the dataset is compressed to couple of potential variables, number of which relates to the number of nodes in the bottleneck layer. This number must be lesser than the number of nodes at input or output layers. Then the output of the bottleneck is decompressed at the de-mapping layer. Hence, the bottleneck outputs of AANNs represent a type of nonlinear principal components, which are recurrently more suitable than PCA for examining nonlinear, real-world datasets.

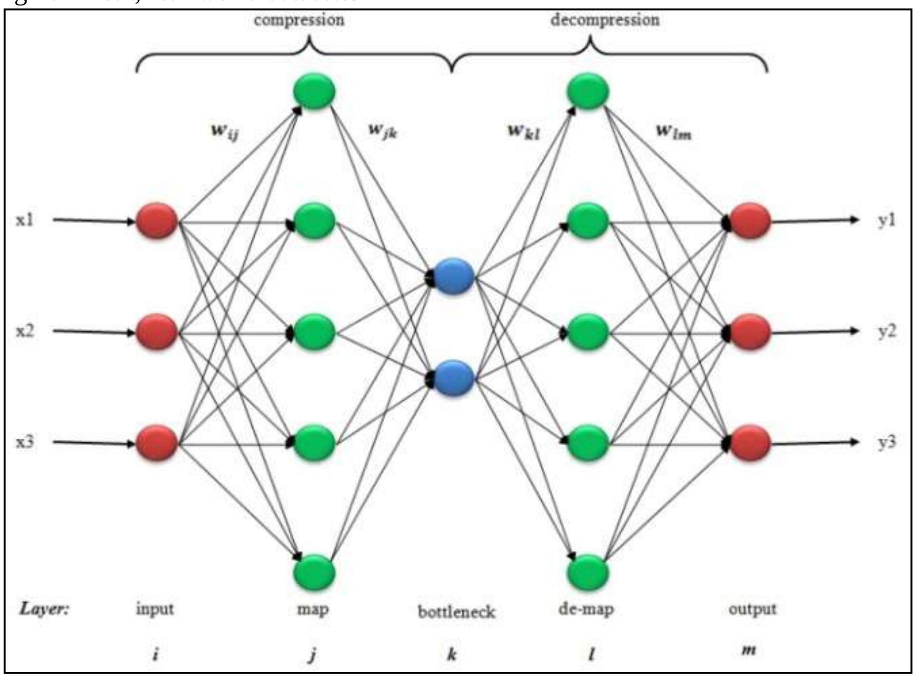

Fig. 1: Structure of AANN consists of 3 nodes at input and output layers, 5 nodes at map and de-maps layers and 2nodes at bottleneck layer.

Assuming $M_{(i,j)}$ function for this operation at map
Layer, it can be defined as:

$$M\,(i,j) = \sigma \sum_{i=1}^{nbmp} \sum_{j=1}^{nbinput} xjw\ ji \qquad (1)$$

With nbmp and nbinput are number of nodes in map and input layers respectively. The same processes continue at bottleneck, de-map and output layers. The error signal at node m in output layer is calculated as following:

$$e_m = y_{d,m}\,(p) - y_m(p) \qquad (2)$$

Where $y_{d,m}$ (p) is the desired output of node *m* in output layer at iteration *p*. These errors are back-propagated from output layer back to input layer. The rule used to update weights between output layer and de-map layer is:

$$w_{lm}(p+1) = w_{lm}(p) + \Delta w_{lm}(p) \qquad (3)$$

where *l* represent the nodes at de-map layer and weight correction $\Delta w_{lm}(p)=\alpha\times y_l(p)\times\rho_l(p)$ where $\rho_l(p)$ was the gradient error at neuron *m* in the output layer at iteration *p* and α is the learning rate.Mean Square Error is calculated as follows-

$$MSE = 1/n \sum_{i=1, m=1}^{nbinput, nboutput} (x_i - x_m)^2 \qquad (4)$$

Process until a varied MSE value. Generally it is considered till the value of MSE value reaches lower than 0.001.

B. Huffman Coding:

Huffman coding is a lossless information compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the doled out codes depend on the frequencies of corresponding characters. The most frequent character gets the littlest code and the slightest regular character gets the largest code.

The variable-length codes allotted to information characters are Prefix Codes, implies the codes (bit sequences) are appointed in a manner that the code doled out to one character is not prefix of code doled out to some another character. This is the means by which Huffman Coding ensures that there is no uncertainty when decoding the generated bit stream. There are basically two main parts in Huffman's Coding

1) Build a Huffman Tree from input characters.
2) Traverse the Huffman Tree and assign codes to characters.

C. Steps to construct Huffman Tree:

Input is array of distinctive characters along with their frequency of occurrences and output is Huffman Tree.

1) Make a leaf node for each distinctive character and build a min heap of all leaf nodes (Min Heap is used as a priority queue. The value of recurrence field is utilized to compare two nodes in min heap. Initially, the least frequent character is at root)
2) Separate two nodes with the least recurrence from the min heap.
3) Create a new internal node with frequency equal to the total of the two nodes frequencies. Make the first extracted node as its left child and the other selected node as its right child. Join this node in the min heap.
4) Repeat steps#2 and #3 up till the heap comprises only one node. The nodes which are left is the root node and the tree is complete.

1) Steps to print codes from Huffman Tree:

Examine the tree shaped beginning from the root. Keep up an auxiliary array. In the mean while moving to the left child, write 0 to the array. While shifting to the right child, write 1 to the array. Print the array following a leaf node is confronted.

D. Singular Value Decomposition:

In linear Algebra, the singular value decomposition is a factorization of a real or complex matrix. It is the observation of the Eigen decomposition of a positive semi absolute normal matrix (for example, a symmetric matrix with positive Eigen values) to any matrix through an addition of polar decomposition. It has many useful applications in signal processing and statistics.
Generally, the singular value decomposition of an $m \times n$ real or complex matrix M U Σ V *, where U is an $m \times m$ real or complex unitary matrix, Σ is a $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V is an $n \times n$ real or complex one matrix. The diagonal entries σ i of Σ am known as the singular values of M. The columns of U and the columns of V are called the left-singular vectors and right-singular vectors of M, respectively is a factorization of the form
The singular value decomposition can be figured utilizing the following observations:

1) The left-singular vectors of M are an arrangement of orthonormal eigenvectors of MM^*.
2) The right-singular vectors of M are an arrangement of orthonormal eigenvectors of M^*M.
3) The non-zero singular values of M (stand on the diagonal elements of Σ) are the square roots of the non-zero Eigen values of both M^*M and MM^*.

SVD look for a low-rank matrix X=U V', where U ∈Rn×c and V∈Rm×c , that minimizes the sum-squared distance to the fully observed target Matrix Y. The answer is given by the main singular vectors of Y. In the collaborative ltering domain, most of the values in Y will be absent, so the sum-squared distance is minimized with respect to the partially observed entries of the target matrix Y. Imperceptibly values of Y are then predicted using the corresponding entries of X. Let X= U V', where UЄRN×C and VЄRM×C denote the low-rank approximation to the partially observed target matrix YЄRN×M. Matrices U and V are introduced with little arbitrary qualities inspected from a zero-mean normal distribution with standard deviation 0.01. We lessen the subsequent objective function:

$$f=\sum_{i=1}^{N}\sum_{j=1}^{M} Iij(uivj' - Yij)2 + \lambda \sum_{ij} I_{ij}(\| u_i \|^2_{Fro})$$

Where (|| . ||2Fro) represents the Frobenius norm, and Iij is the indicator function, taking on value 1 if user I rated movie j, and 0 generally. We then carry out gradient descent in U and V to minimize the objective function of Eq.15.

We have utilized AANNs for the reduction of data and for compression two methods are used Huffman's coding and Low Rank Matrix Approximation. Thus it produced 3 sets of data one reduced and two compressed. On these data we have applied six different classifiers. From the result produced we come to the conclusion that Low Rank Matrices produces the best result. Although Huffman's coding is a better solution when it comes to saving the space in saving the data but its classification is not up to the mark. Reduced data (AANN) reduces the number of dimension of dataset and produces average result so it can be adopted for classification.

IV. EXPERIMENTAL FRAMEWORK AND RESULT

We have utilized AANNs for the reduction of data and for compression two methods are used Huffman's coding and Low Rank Matrix Approximation (SVD). Thus it produced 3 sets of data one reduced and two compressed. On these data we have applied 6 different classifiers which are Decision tree, nearest neighbor, Support Vector Machines, Boosted trees, Bagged trees and Subspace Discriminate classifier.

The dataset used for this study is iris flowers dataset. It consists of 150 flowers, divided equally into three different classes: Setosa, Versicolor and Virginica. Each of them has 50 samples of iris flowers. All these characteristic structure, equations and principles of AANN have been created and written using MATLAB platform.

A. Behaviour of Reduced and Compressed Data Obtained using Various Classifiers:

On passing datasets through various classifiers on looking at the graphs we can conclude that on applying low rank matrix reduction method we can get the best result. It generates the best percentage of accuracy. The accuracy obtained by applying Low Rank matrices is nearest to the original Data result.

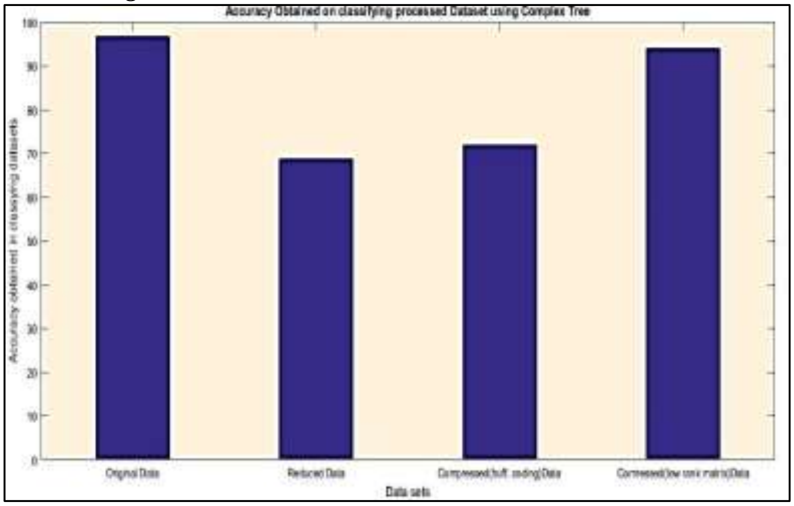

Fig. 4.1: Behaviour of reduced and compressed data obtained by using Decision Tree classifiers

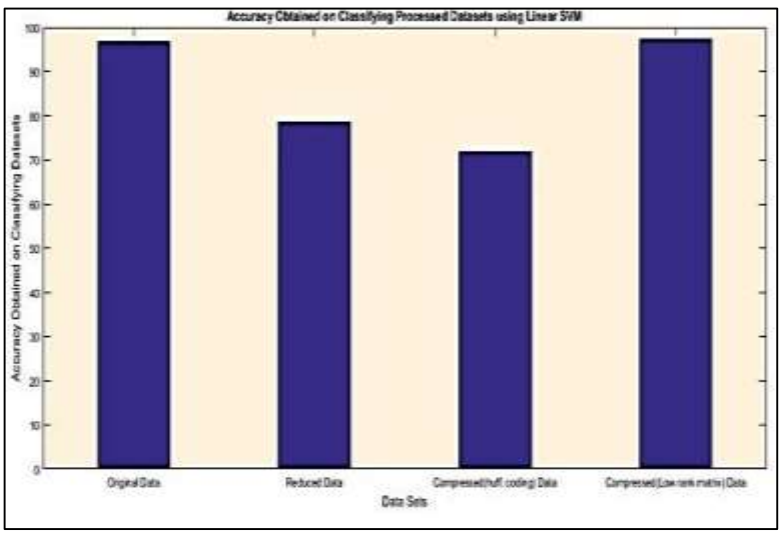

Fig. 4.2: Behaviour of reduced and compressed data obtained by using Linear SVM

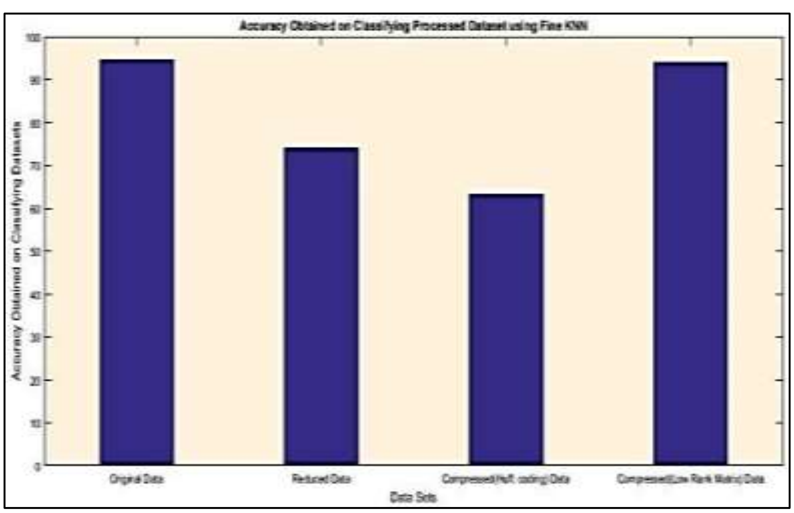

Fig. 4.3: Behaviour of reduced and compressed data obtained by using Fine KNN classifiers

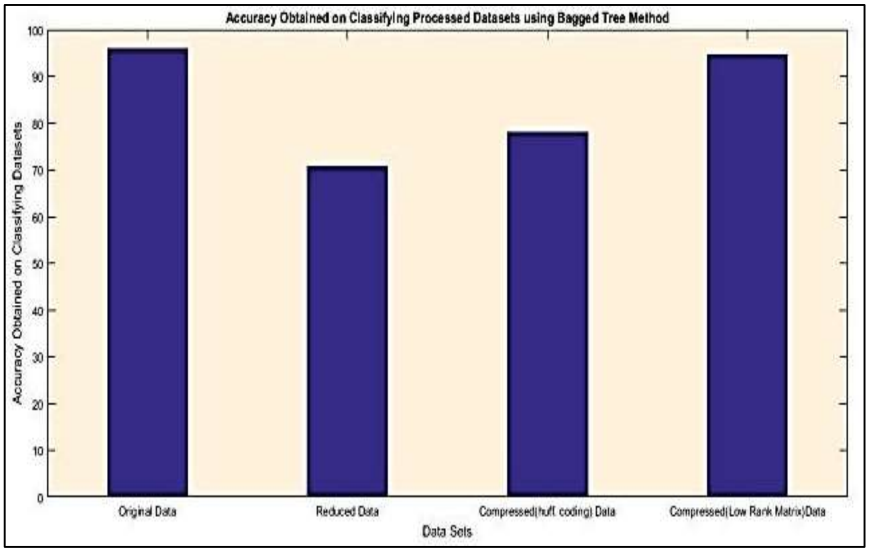

Fig. 4.4: Behaviour of reduced and compressed data obtained by using Boosted tree classifiers

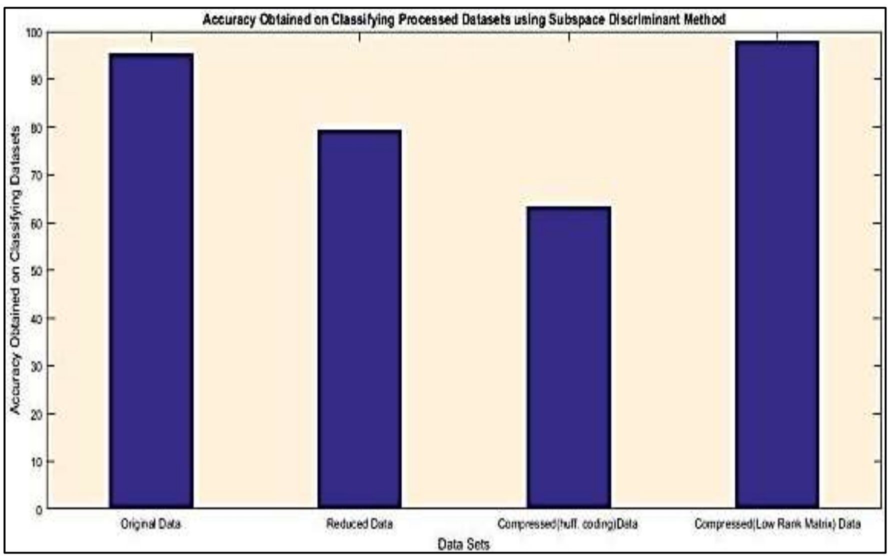

Fig. 4.5: Behaviour of reduced and compressed data obtained by using Bagged tree classifiers

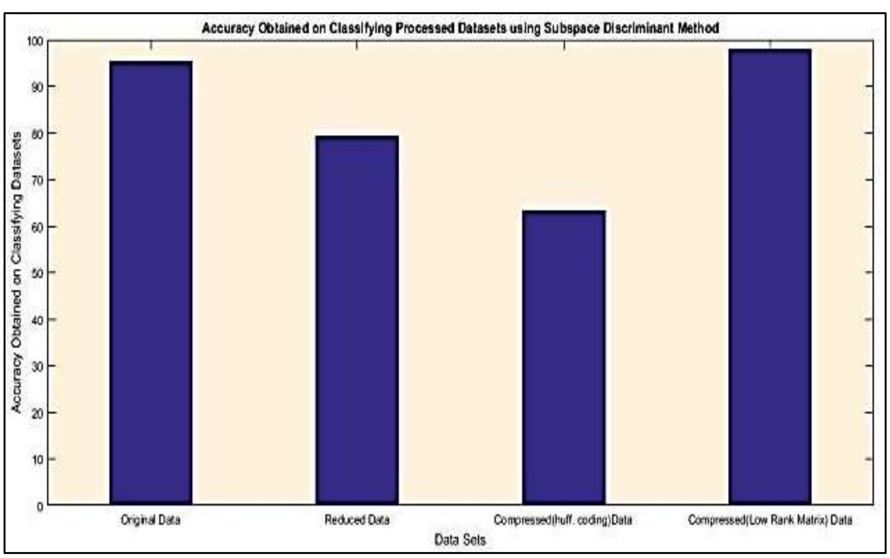

Fig. 4.6: Behaviour of reduced and compressed data obtained by using Subspace discriminate classifiers

Table – 1
Accuracy (%) obtained using various classifiers on reduced and compressed data

Data Types	*Decision tree*	*Linear SVM*	*Fine KNN*	*Boosted tree*	*Bagged tree*	*Subspace Discriminate*
AANN	*68.7*	*78.7*	*74.0*	*70.0*	*70.7*	*79.3*
Huffman's coding	*72.0*	*63.6*	*63.3*	*68.7*	*78.0*	*63.3*
Low Rank Matrix	*94.0*	*97.3*	*94.0*	*95.3*	*94.7*	*98.0*

Through the experimental results of reduced (AANN) and compressed (Huffman's coding and Low Rank Matrices) datasets, Low Rank Matrices produces the best result and overall bagged algorithm have been able to classify and visualize all three datasets into their appropriate classes most accurately. From these results too, the natureof the data clusters was found similar to the characteristics of the datasets themselves. However, it can be observed that the Huffman's coding reduces the size of the dataset to a greater percentage but performance is reduced in classifying the data. This might due to the fact that more complexity exists in the defining the classes.

V. CONCLUSION

In this work we presented a simple comparison of reduced and compressed with using various Learning Classifiers. The AANNs can be considered as a very powerful tool in exploratory data analysis with its ability to deal with linear and nonlinear correlation among variables. In this study we conclude that, AANNs can perform dimension reduction and

Data visualization usually using its two bottleneck activation nodes. Huffamn's coding technique is efficient when we want to reduce the data size and overall Low Rank Matrices produces the results nearest to the original dataset. In this paper, all the techniques have been developed using MATLAB platform. Our developed AANNs, Huffman's coding, Low Rank Matrices algorithm has been later used to perform compression, classification and visualization of multidimensional data using various classifiers on Iris flowers dataset. The experimental results have shown that Low Rank Matrices algorithm has been able to classify this high-dimensionality data appropriately according to their respective classes in most accurate way. This could give us the way to explore the potentiality to enhance this technique by integrating it with another technique for eg. SOM with the objective to serve solutions to data classification of very complex multidimensional data ie: gene expression data with multidimensional variability [24].

REFERENCES

[1] Pearson, K. On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine, 1901, 2, 559-572179-188
[2] Fodor, I. K. A survey of dimension reduction techniques Lawrence Livermore Natl. Laboratory, 2002
[3] Carreira-Perpiñán, M. A. A review of dimension reduction techniques Dept. Computer Science, Univ. Sheffield, 1997
[4] Aarti, "Performance Analysis of Huffman's coding", International Journal of Advanced Research in Computer Science and Software Engg. Publications, Vol.3, Issue 5, May 2013.
[5] Mohini D Patil, Dr. Shirish S. Sane," Effective Classification after Dimension Reduction: A Comparative Study", International Journal of Scientific and Research Publications, Vol.4, No7, 2014, pp.: 1-4.
[6] Asha Gowda Karegowda, A. S. Manjunath& M.A.Jayaram introduced ," comparative study of attribute selection using gain ratio and correlation based feature selection", International Journal of Information Technology and Knowledge Management, Vol.2, No. 2,2010, pp.: 271-277.
[7] Zalhan Mohd Zin,"Using Auto-Associative Neural Networks to Compress and Visualize Multidimensional Data",The 11th International Conference on Ubiquitous Robots and Ambient Intelligence,URAI, 2014Nov. 12 - 15, 2014
[8] Pankaj Kumar and Ankur Kumar Varshney, "Double Huffman Coding", International Journal of Advanced Research in Computer Science and Software Engg. Publications, Vol.2, Issue 8, August 2012.
[9] F. V. N. T. M. Jaisheel Mistry, "Missing Data Estimation Using Principle Component Analysis and Autoassociative Neural Networks," Journal of Systemics, Cybernatics and Informatics, vol. 7, no. 3,pp. 72-79, 2009.
[10] Mamta Sharma, "Compression using Huffman coding", International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010
[11] Ramakrishnan Kannan, Mariya Ishteva, Haesun Park, "Bounded Matrix Low Rank Approximation" 2012 IEEE 12th International Conference on Data Mining
[12] Dimitris Achlioptas, Frank McSherry,"Fast Computation of Low Rank Matrix Approximation", *STOC'01,* July 6-8, 2001, Hersonissos, Crete, Greece.Copyright 2001 ACM 1-58113-349-9/01/0007
[13] Joonseok Lee, Seungyeon Kim, Guy Lebanon,"Local Low Rank Matrix Approximation", International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.
[14] E. A. Juha Vesanto, "Clustering of the Self Organizing Maps," IEEE Transaction on Neural Networks, vol. 11, no. 3, pp. 586-600, 2000.
[15] Yunhui Shi, He Li, Jin Wang, Wenpeng Ding, Baocai Yin,"*Inter Prediction Based on Low Rank Matrix Completion",* 2012 IEEE International Conference on Multimedia and Expo Workshops.
[16] A. HyvÄarinen. Survey on independent component analysis. Neural Computing Surveys, 2:94{128, 1999.
[17] S. Y. B. C. M. A. Abidi, "Image Compression Using Hybrid Neural Networks Combining The Auto-Associative Multilayer Perceptron and The Self Organizing Feature Map," IEEE Transaction on Consumer Electronics, vol. 40, no. 4, pp. 796-811,1994.
[18] W. N. Venables and B. D. Ripley. Modern Applied Statistics with S-Plus. Springer, 1996.
[19] I.T. Jolliffe. Discarding variables in principal component analysis I: arti¯cial data. Appl. Statist.,21:160{173, 1972
[20] I.T. Jolliffe. Discarding variables in principal component analysis II: real data. Appl. Statist., 22:21{31,1973.
[21] B. W. D. L. M. M. Daszykowski, "A Journey IntoLow-dimensional Spaces With Autoassociative Neural Networks," International Journal of Pure and Applied Analytical Chemistry Talanta, vol. 59, pp. 1095-1105, 2003.
[22] B. J. H. J. D. L. Mark J. Embrechts, "Augmented Efficient BackProp for Backpropagation Learning in Deep Autoassociative Neural Networks," in International Joint Conference on Neural Networks (IJCNN), Barcelona, 2010.
[23] B. J. H. J. D. L. Mark J. Embrechts, "AugmentedEfficient BackProp for Backpropagation Learning in Deep Autoassociative Neural Networks," in International Joint Conference on Neural Networks (IJCNN), Barcelona, 2010.

[24] Ibrahim M. El-Hasnony,Hazem M. El Bakry and Ahmed A. Saleh "Comparative Study among Data Reduction Techniques over Classification",International Journal of Computer Applications (0975 – 8887)Volume 122 – No.2, July 2015
[25] J.-C. G. Gaetan Kerschen, “Feature Extraction Using Auto-Associative Neural Networks,” Smart Materials and Structures, vol. 13, no. 1, 2004
[26] M.A. Kramer. Non-linear principal component analysis using autoassociative neural networks. AIChE Journal, 37:233{243, 1991.
[27] S. Kaski. Dimensionality reduction by random mapping: fast similarity computation for clustering.Proc. IEEE International Joint Conference on Neural Networks, 1:413{418, 1998.
[28] P. J. K. M. G. Ali Anaissi, “A Framework for High Dimensional Data Reduction in the MicroarrayDomain,” in IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010..
[29] J. Karhunen, P. Pajunen, and E. Oja. The nonlinear pca criterion in blind source separation: Relations with other approaches. Neurocomputing, 22:5{20, 1998.
[30] T. Kohonen et al. Self-organization of massive document collection. IEEE Transactions on Neural Networks, 11(3):574{585, May 2000.
[31] J.A. Spierenburg. Dimension reduction of images using neural networks. Master's thesis, Leiden University, 1997.
[32] M.L. Raymer et al. Dimensionality reduction using genetic algorithms. IEEE Transactions on Evolutionary Computation, 4(2):164{171, July 2000.

