

# PSO Optimized Nearest Neighbor Algorithm

Manish Mahajan, Santosh Kumar, Bhasker Pant, Kireet Joshi, Vikas Tripathi



**Abstract:** Data mining can be considered to be an important aspects of information industry. Data mining has found a wide applicability in almost every field which deals with data. Out of the various techniques employed for data mining, Classification is a very commonly used tool for knowledge discovery. Various alternatives methods are available which can be used to create a classification model, out of which the most common and apprehensible one is KNN. In spite of KNN having a number of shortcomings and limitations in it, these can be overcome by with the help of alterations which can be made to the basic KNN algorithm. Due to its wide applicability, kNN has been the focus of extensive research and as a result, many alternatives have been performed with wide range of success in performance improvement. A major hardship being faced by the data mining applications is the large number of dimensions which render most of the data mining algorithms inefficient. The problem can be solved to some extent by using dimensionality reduction methods like PCA. Further improvements in the efficiency of the classification based mining algorithms can be achieved by using optimization methods. Meta-heuristic algorithms inspired by natural phenomenon like particle swarm optimization can be used very effectively for the purpose.

**Keywords :** Classification, Data mining, kNN, Particle Swarm Optimization, Principal Component Analysis.

## I. INTRODUCTION

Data mining can be thought of as a process that involves identifying, validating and finally analysing the data obtained from a variety of information sources in order to subsequent transformation of data into a useful form. The ever increasing volume of available data which needs to be searched for patterns has made the use of data mining tools an absolute necessity[1]. The data mining tools work toward analysing the data in such a manner so as to identify crucial information out of data. The results thus obtained by applying the data mining tools can then be used for making decisions. Data mining helps the decision makers of a wide range of application areas like marketing, credit card analysis,

financial analysis, medical diagnosis, network intrusion detection, manufacturing industry etc. There are a wide variety of data mining techniques being used for extracting knowledge in the form of patterns from the source data – some of them being outlier analysis, association rule mining, clustering, regression, classification, forecasting etc. These data mining methods make use of many algorithms based on machine learning in order to perform this task. The Classification technique is a very important data mining research area (especially where mining of big data is concerned)[2]. The classification technique is mainly used to predict labels attached to test data points. Classification involves building a classification model which is known as the classifier from the training data. The model thus obtained is capable of classifying test data tuple. The Classification technique uses class labeled data set to build this model or classifier. This is why among the data mining tools, the classification method is considered to exhibit supervised learning. There are a wide variety of classification algorithms in use today for example Rule-Based Classification, k-Nearest Neighbor (KNN), Bayesian Classifier, Decision Tree Induction, Support Vector Machine (SVM) etc.[3]. Out of these many alternatives available, the k-Nearest Neighbors classifier comes out as a simple yet powerful and effective method for data mining. The KNN method uses an approach that uses the nearest neighbor method for the purpose of classification of an unseen test tuple based on data tuples which are similar to it. The kNN algorithm uses the class labels of the k tuples from the data set which are nearest to the test tuple in order to classify it by assigning it an appropriate class label. The number of nearest neighbors to be considered, the parameter k, is a positive integer which can be passed as an input parameter to the Nearest Neighbor classifier integer which can be passed as an input parameter to the KNN classifier[4].

Revised Manuscript Received on December 30, 2019.

\* Correspondence Author

**Manish Mahajan**, Computer Science and Engg. Deptt., Graphic Era Deemed to be University, Dehradun, India, manishmahajan@geu.ac.in.

**Santosh Kumar**, Computer Science and Engg. Deptt., Graphic Era Deemed to be University, Dehradun, India drsantosh.cse@geu.ac.in

**Bhasker Pant**, Computer Science and Engg. Deptt., Graphic Era Deemed to be University, Dehradun, India, pantbhaskar2@gmail.com

**Kireet Joshi**, Computer Science and Engg. Deptt., Graphic Era Deemed to be University, Dehradun, India, joshikireet@gmail.com

**Vikas Tripathi**, Computer Science and Engg. Deptt., Graphic Era Deemed to be University, Dehradun, India, vikastripathi.be@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

### Standard kNN Algorithm

Input Parameters: Data set, k

Output: Classified test tuples

Step 1: Store all the training tuples.

Step 2: for each test tuple

A. Compute distance of it with all the training tuples using (1).

B. Find the k nearest training tuples to the test tuple.

C. The class which is most common in the k nearest training tuples to the test tuple is assigned to the test tuple.

End for

A number of modified versions of the basic kNN algorithm have been proposed by researchers. These variants of KNN have been critically discussed in [5]. The various kNN variants endeavor to remove the disadvantages of the basic kNN classifier. The basic kNN classification algorithm suffers from the drawback that its result can get easily influenced by a number of key factors including the parameter  $k$ , the distance measure being employed for calculating the similarity/ difference between two tuples, the assigned weights of attributes to name a few. A change or manipulation of these factors can result in the performance of KNN algorithm to be ameliorated.[6]

In order to perform efficient classification by using kNN or any other classifier, it often becomes important to decrease the number of dimensions to a manageable one. Feature extraction and feature selection are two common methodologies to achieve the reduction in the dimensionality of data sets. In the feature selection method, we choose the most relevant and significant dimensions are retained from the original data, where in the feature extraction method, we transform the original vector space into a novel one having particular special characteristics and it is the new vector space that is reduced in dimensions[7]. The Principle Component Analysis (PCA) method is one of the most widely used feature extraction methodology. The PCA is an unsupervised method for dimensionality reduction. PCA performs the reduction in the number of data dimension by identifying few linear combinations which tend to be orthogonal and have the largest variance, these are called the principal components PCs of the original dimensions [8].

Particle Swarm Optimization (PSO), a optimization algorithm based on the swarm intelligence concept, is a global optimization tool. In the case of standard PSO, each individual element of the population under study is updating by adjusting its trajectory iteratively toward its own hitherto previous best position, and also towards the previously attained best position of any member of the same generation[9]. PSO has been successfully used for the purpose of optimization of data mining tasks and has shown good results[10].

This study proposes a novel optimized kNN classification method. The  $k$  NN classifier helps to learn and model the local information about the data, and the PCA component is used to reduce the dimensionality of all data points while preserving the global nature of the data information. Finally the results of the PCA enhanced kNN are optimized by using a meta-heuristic algorithm, we have used PSO for this purpose. The selection of the fitness function plays a vital role in determining the performance of PSO. A well defined fitness function can lead to a prompt location of the optimized particle positions. We have selected the Schaffer's F6 function for the purpose of this study. The Schaffer's F6 function is essentially a function used for testing purposes. This function includes a number of peaks (local optima) which makes it difficult for the various hill-climbing techniques to converge. The F6 function used here is a standard one which has its peak situated at origin with a unit value [11]. For the purpose of evaluating the performance of the proposed methodology, three common datasets from the UCI repository have been used - the breast cancer dataset, the wine quality dataset and the Lower back pain dataset[12].

The remaining part of the paper is organized as; Section 2 lists the recent researches in the three areas that are of relevance viz., kNN classification, PCA and Particle Swarm Optimization. Section 3 elaborates the experimental setup and the results obtained by applying the proposed methodology on the three selected datasets. Section 4 covers the final conclusion drawn and the future scope for the work.

## II. RELATED WORK

### A. Knn Classifier

The kNN algorithm performs the task of classifying an unknown test data tuple with the help of the nearest neighbor technique to analyse the classes to which the nearest neighbors of this tuple belong to. A fixed subset of the nearest neighbours of the test tuple is allowed to participate in the classification process. The size of this fixed subset is identified by  $k$ , which obviously is an integer value greater than zero. When this parameter  $k=1$ , the classifier uses the class assigned to the training data tuple that is identified to be the most nearest one, i.e., has the least distance to the unknown test data tuple is used as the class of the test tuple. There are many distance measures that can be used for the purpose of finding the nearest neighbor. These include the Euclidian distance, the Manhattan distance etc. The most commonly used distance measure is the Euclidian distance and it the same that has been used in this study. KNN is identified as a nonparametric lazy learner. The available data set is first broken down into two components: one of them is used as the training set and the other is used as the test set. The training set is used as the input to the algorithm for developing the classifier model, while the test set is utilized to identify the accuracy and efficiency of the algorithm. The data set can be divided by using any one of the commonly used techniques like hold-out method, random sampling, cross validation etc. for this study, we have used random sampling. We have divided the data in the ratio of 80:20 for training to testing data component. The KNN algorithm classifies the new unknown test tuple by utilizing the training data tuples which are similar to it (have less distance). Since the kNN algorithm is quiet easy to implement and provides significant classification performance, it has become a very popularly used method in tasks involving data mining and has been also selected to be one of the top ten data mining algorithms[13]. In spite of all its features, the basic kNN algorithm suffers from two major weaknesses. First, the success rate of the basic kNN algorithm is highly dependent on the number of neighbors chosen, the  $k$  parameter. Second, the relationship between a data tuple and the class to which it is assigned is very rigid or crisp. There are no grey or fuzzy areas. Each data tuple has complete relationship with a single class exclusively and has absolutely no relationship with any other class. This crisp/strict relationship leads to inflexibility and inefficiency of the approach. The kNN classifier has been known to show efficient performance even on medium to large data sets and has shown significant performance levels even on very large data. Significant amount of effort has been made on designing the variants to the kNN methods.

These variants have efficiently proposed varying optimal-k parameter values to different sample data sets[14]. These variants have shown widely varying performance levels. The improvised variants of the kNN algorithm work in two stages, the first stage is used to learn and identify an optimal-k parameter value for each corresponding test data sample and subsequently the second stage uses this value on the traditional kNN classification method in order to predict the test tuple samples by using the optimal-k parameter value thus learned [15]. But this approach suffers from the disadvantage of being slow as the two stage process of first learning an optimal-k parameter value for each test sample and the subsequent process of scanning again all the training data samples for the purpose of finding the nearest data tuple neighbors of each given test tuple is time-consuming [16].

**B. Principal Component Analysis**

The data mining applications being used commercially often suffer from what is commonly called the “Curse of Dimensionality”. This refers to the problems faced by the data mining applications which are working with analyzing multivariate data. Commercial multivariate data usually comes with a large number of dimensions or attributes which need to be processed in order to extract useful information out of it. The goal of the methods being used for dimensionality reduction is to efficiently represent the input data using a lower dimensional space, which is achieved by preserving some of the initial original properties of the input data. The following equation illustrates how the high dimensional data space data [M] is reduced to lower -dimensional space [N].

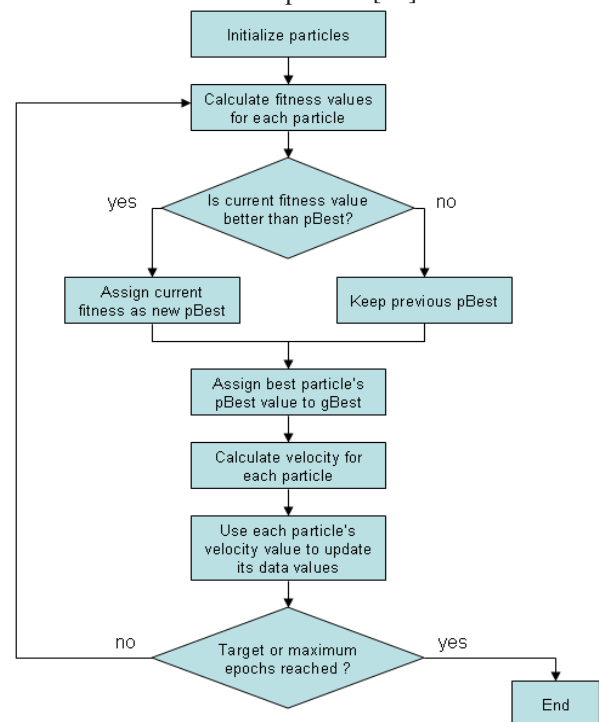
$$i/p = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \rightarrow \text{Dimensionality reduction} \rightarrow o/p = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad n \ll m$$

One widely used method employed for the purpose of reducing the dimensionality of the input feature space is the Principal Component Analysis (PCA) method. PCA helps to classify the similarities and differences inherent in the data so that they can be explored. The PCA approach works by analyzing the input data for the purpose of creating smaller variables which are known as the Principal Components (PCs) by using the theory of variance estimation [17]. By itself, the Principal component analysis (PCA) can be considered as a statistical procedure which employs an orthogonally implemented transformation in order to transform a set of input observations derived from a set of variables which might be possibly correlated and convert them into another set of values which are linearly uncorrelated. These transformed variables are called the principal[18]. The number of these principal components thus obtained is usually less than or in worst case equal to a value which represents the smaller quantity out of the number of originally available variables or the number of original observations. These transformations are carried out in such a manner that the first obtained principal component gets the largest possible variance, i.e., the first principal component is responsible for the maximum amount of variability in the data. The remaining components have the feature that each succeeding component has in turn the maximum value of variance which is possible considering that it must be is

orthogonal to its immediately preceding components. The Vectors thus obtained form a set which is inherently uncorrelated and forms a orthogonal basis set. [19].

**C. Particle Swarm Optimization**

A distributed solution for the purpose of solving complex problems is provided by the Swarm intelligence. PSO deploys the interactions occurring between the simple agents and the environment in which they are operating in order to solve complicated problems[20]. [20]. The technique involves several particles that keep a track of their own attributes, all of these particles are cumulatively contained by the Swarm. Every particle is defined by a set of attributes including, its current position which is provided by a n-dimensional vector and also its corresponding current velocity which is used to keep track of the direction and speed of the current motion of the particle [21].



**Fig 1 : Particle Swarm Optimization Flowchart**

A current fitness value which is calculated after evaluating the specified fitness function using the particle’s current position is also one of the particle’s attributes. A population of particles that are in actuality set of random potential solutions is given to the system at the time of initialization. A randomized velocity is assigned to each particle. These particles are then allowed to traverse within the (hyper) space that represents the set of potential solutions. The specifications for the Coordinates for the particle which has achieved the best fitness using the specified fitness function so far are made in the hyperspace and are kept track of by the particle and also its best fitness is also accounted for. A leader particle is appointed on the basis of better solutions among the best solutions given by the particles. The local best and the global best positions of the particle are taken and as their function, the velocity of the particle changes(accelerates) at each step. A random term is then used to weight this acceleration.

The new velocity value and all the relevant components of the position vector of the particle are summed up to calculate the new position of the particle in the solution space. [22].

Figure 1 illustrates the standard PSO algorithm with the help of a flowchart. In this, each individual component of the current population is termed as a particle. In case of standard PSO, once the population has been initialized, the position and velocity assigned to each particle is modified in each subsequent iteration. This update is based on the experience gained by the particle itself (pbest) and the experience considered as the best experience among all the particles (gbest) as shown in Fig 2. At the conclusion of each subsequent iteration, the updated performance from all the particles is then numerically evaluated by using the predefined cost or fitness functions. It is possible for a particle which is representing a candidate solution to move towards the optimal position by continually updating the specified position assigned to it and also its velocity. Figure 2 illustrates the typical mathematical equation and the various factors that contribute towards updating of the particle's velocity.

$$v_{k+1}^i = w v_k^i + c_1 \text{rand} \left( \frac{p^i - x_k^i}{\Delta t} \right) + c_2 \text{rand} \left( \frac{p_g^i - x_k^i}{\Delta t} \right)$$

velocity of particle *i* at time *k*+1

inertia factor range: 0.4 to 1.4

self confidence range: 1.5 to 2

swarm confidence range: 2 to 2.5

current motion

particle memory influence

swarm influence

Fig 2 : The Particle velocity update equation

III. EXPERIMENTAL SETUP AND RESULTS

The study has used comparative analysis of the application of reduced dimension simple kNN and reduced dimension optimized kNN to the three datasets. The experimental results have been shown in the form of data point plots. For the purpose of implementation of the PSO optimized kNN algorithm, we have used the following parameters:

Value of k	7
Testing to Training ratio	20:80
PSO Iterations	7
Fitness Function used	Schaffer's F6
Constant C1 for PSO	2.57
Constant C2 for PSO	2.57

The proposed methodology was applied on three datasets from the UCI repository.

- Breast Cancer Dataset
- Wine Quality Dataset
- Lower Back Pain Dataset

Table 1 : Experimental Results

Dataset	Accuracy Without PSO	Accuracy with PSO
Breast Cancer	91.61%	99.30%
Red Wine Quality	45.25%	94.01%
Spine Dataset	70.51%	94.87%
White Wine Quality	44.82%	98.45%

Table 1 shows the results obtained. It is clear that using PSO enhances the accuracy of the kNN operation significantly. This can be observed for all the datasets used. The subsequent figures visualize the results obtained graphically.

Figure 3 shows the result of using Simple kNN and PCA (for dimensionality reduction) on the wine quality data set.

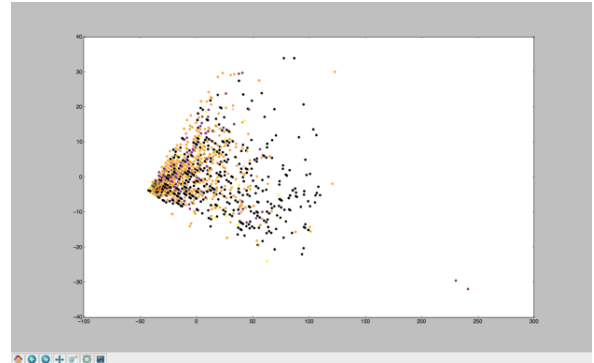


Fig 3 : Simple kNN and PCA on Red Wine Quality data

Figure 4 and 5 shows the improvement obtained by applying PSO Optimised kNN (5 and 7 iterations respectively) and PCA (for dimensionality reduction) on the red wine quality data set.

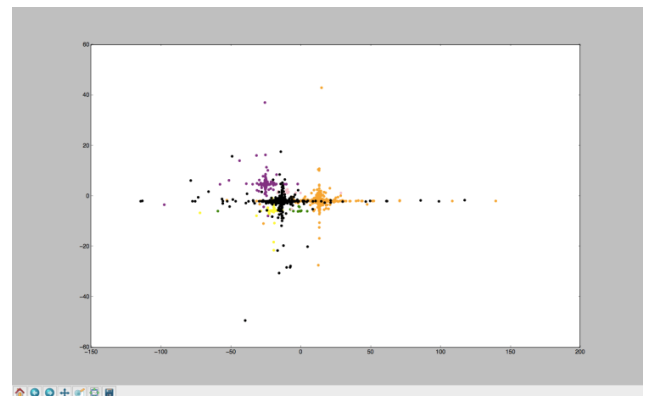


Fig 4 : PSO optimized kNN and PCA on Red Wine Quality data(5 Iterations)

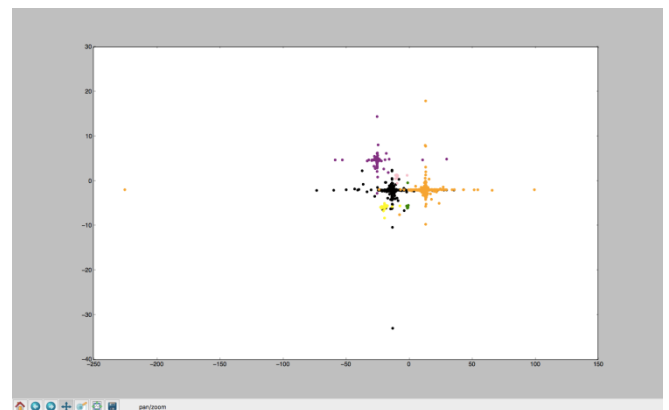


Fig 5 : PSO optimized kNN and PCA on Red Wine Quality data(7 Iterations)

Figures 6 and 7 show the results of applying simple kNN with PCA and PSO optimized kNN with PCA on the white wine quality data set respectively.

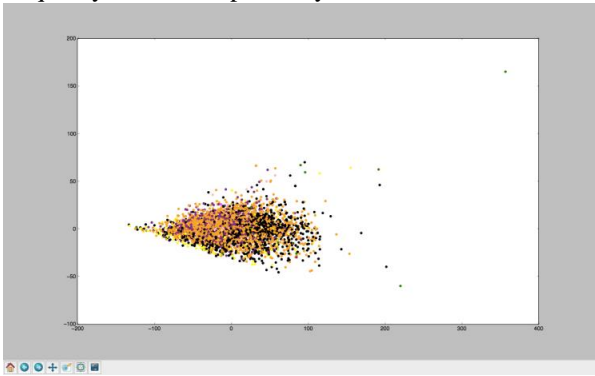


Fig 6 : Simple kNN and PCA on White Wine Quality data

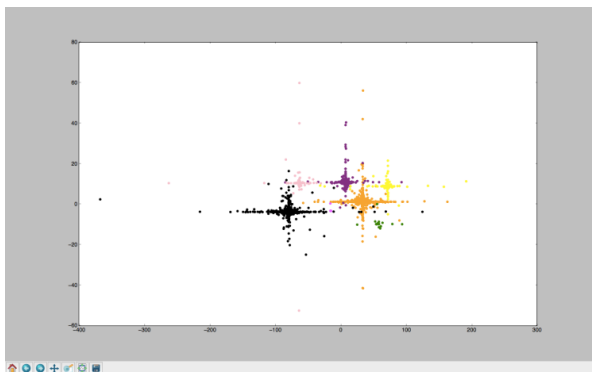


Fig 7 : PSO optimized kNN and PCA on White Wine Quality data

Figures 8 and 9 show the results of applying simple kNN with PCA and PSO optimized kNN with PCA on the Breast cancer data set respectively.

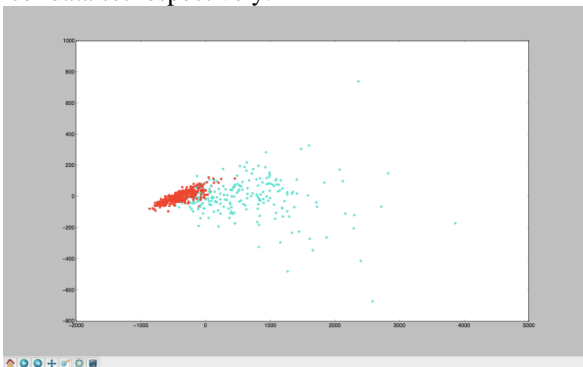


Fig 8 : Simple kNN and PCA on Breast Cancer data

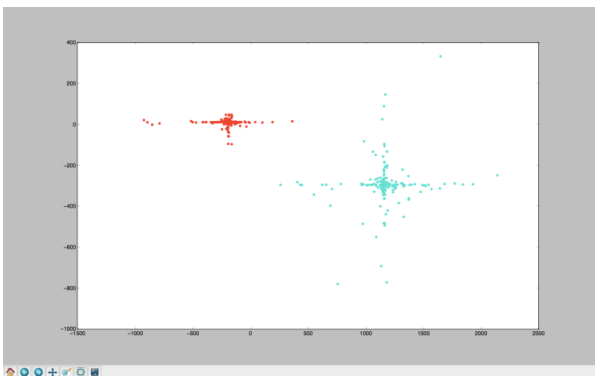


Fig 9 : PSO optimized kNN and PCA on Breast Cancer data

Figures 10 and 11 show the results of applying simple kNN with PCA and PSO optimized kNN with PCA on the Lower Back Pain data set respectively.

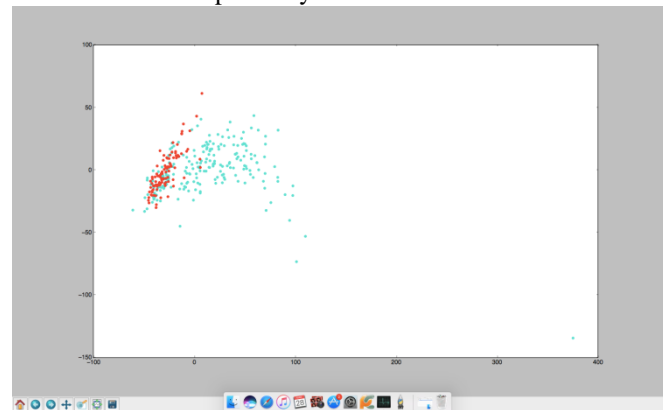


Fig 10 : Simple kNN and PCA on Lower Back Pain data

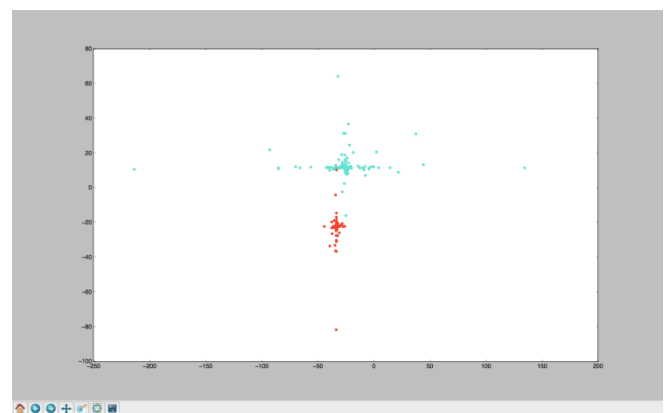


Fig 11 : PSO optimized kNN and PCA on Lower Back Pain data

#### IV. CONCLUSION AND FUTURE SCOPE

The results of applying PCA for dimensionality reduction on the three datasets used and subsequent optimization of the classification algorithm by using the Particle Swarm Optimization has clearly indicated that meta-heuristic optimizers definitely contribute towards improving the efficiency and performance of the kNN classifier. The comparison of results obtained before and after applying the optimization algorithm, clearly show the enhanced classification performance and the better suitability of the information thus gained for decision making purposes. The methodology thus proposed needs to be evaluated further with other, larger data sets for its validation. Moreover, other nature inspired optimizers like Ant Colony Optimization, Cuckoo Search, Firefly Algorithm, Bee Colony Optimization, to name a few must be tried for the optimization of the underlying classifier to evaluate their comparative performances.

REFERENCES

1. M. Mahajan, S. Kumar, and B. Pant, "A novel cluster based algorithm for outlier detection," in *Advances in Intelligent Systems and Computing*, vol. 810, 2018, pp. 449–456.
2. X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, "Robust Joint Graph Sparse Coding for Unsupervised Spectral Feature Selection," *IEEE Trans. Neural Networks Learn. Syst.*, 2017.
3. X. Zhu, X. Li, and S. Zhang, "Block-Row Sparse Multiview Multilabel Learning for Image Classification," *IEEE Trans. Cybern.*, 2016.
4. P. Strecht, L. Cruz, C. Soares, J. Mendes-Moreira, and R. Abreu, "A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance," *Proc. 8th Int. Conf. Educ. Data Min.*, 2015.
5. A. Lamba and D. Kumar, "Survey on KNN and Its Variants," *Int. J. Adv. Res. Comput. Commun. Eng.*, 2016.
6. R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Inf. Sci. (Ny)*, 2018.
7. M. Mahajan, S. Kumar, and B. Pant, "Outlier detection in climatology time series with sliding window prediction," *Int. J. Innov. Technol. Explor. Eng.*, 2019.
8. R. Bro and A. K. Smilde, "Principal component analysis," *Analytical Methods*. 2014.
9. A. Tharwat, H. Mahdi, M. Elhoseny, and A. E. Hassanien, "Recognizing human activity in mobile crowdsensing environment using optimized k-NN algorithm," *Expert Syst. Appl.*, 2018.
10. N. Kant and M. Mahajan, "Time-series outlier detection using enhanced k-means in combination with PSO algorithm," in *Lecture Notes in Electrical Engineering*, vol. 478, 2019, pp. 363–373.
11. X. Qiu and J. Liu, "A novel adaptive PSO algorithm on schaffer's F6 function," in *Proceedings - 2009 9th International Conference on Hybrid Intelligent Systems, HIS 2009*, 2009.
12. D. Dua and C. Graff, "{UCI} Machine Learning Repository." 2017.
13. N. Settouti, M. E. A. Bechar, and M. A. Chikh, "Statistical Comparisons of the Top 10 Algorithms in Data Mining for Classification Task," *Int. J. Interact. Multimed. Artif. Intell.*, 2016.
14. G. H. Chen and D. Shah, "Explaining the success of nearest neighbor methods in prediction," *Found. Trends Mach. Learn.*, 2018.
15. A. J. Gallego, J. Calvo-Zaragoza, J. J. Valero-Mas, and J. R. Rico-Juan, "Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation," *Pattern Recognit.*, 2018.
16. J. Derrac, F. Chiclana, S. Garcia, and F. Herrera, "Evolutionary fuzzy k-nearest neighbors algorithm using interval-valued fuzzy sets," *Inf. Sci. (Ny)*, 2016.
17. S. A. Shah and V. Koltun, "Robust continuous clustering," *Proc. Natl. Acad. Sci. U. S. A.*, 2017.
18. Q. Wang, Q. Gao, X. Gao, and F. Nie, "Angle principal component analysis," in *IJCAI International Joint Conference on Artificial Intelligence*, 2017.
19. J. Lever, M. Krzywinski, and N. Altman, "Points of Significance: Principal component analysis," *Nature Methods*. 2017.
20. M. A. Jabbar, "Prediction of heart disease using k-nearest neighbor and particle swarm optimization," *Biomed. Res.*, 2017.
21. M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.*, 2017.
22. D. Dou and S. Zhou, "Comparison of four direct classification methods for intelligent fault diagnosis of rotating machinery," *Appl. Soft Comput. J.*, 2016.

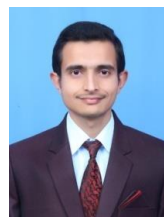
information technology. His research interests include data mining, outlier analysis, software testing, cyber and information security.



**Santosh kumar**, he received ph.d. From iit roorkee (india) in 2012, m. Tech. (cse) from aligarh muslim university, aligarh (india) in 2007 and b.e. (it) from c.c.s.university, meerut (india) in 2003. He has more than 13 years of experience in teaching/research of ug (b. Tech.) And pg (m.tech.) Level courses as a lecturer/assistant professor/ associate professor in various academic /research organizations. He has supervised 01 ph.d. Thesis and 22 m.tech thesis and presently mentoring 06 ph.d. Students (singly and jointly). He has also completed a consultancy project titled "manet architecture design for tactical radios" of drdo, dehradun in between 2009-2011. He is an active reviewer board member in various national/international journals and conferences. He has memberships of acm (senior member), ieee, iaeng, acee, isoc (usa) and contributed more than 52 research papers in national and international journals/conferences in the field of wireless communication networks, mobile computing and grid computing and software engineering. Currently holding position of associate professor in the graphic era deemed to be university, dehradun (india). His research interest includes wireless networks, manet, wsn, iot, and software engineering..



**Bhasker Pant**, Currently working as Dean Research & Development and Associate Professor in Department of Computer Science and Engineering. He is Ph.D. in Machine Learning and Bioinformatics from MANIT, Bhopal. Has more than 15 years of experience in Research and Academics. He has till now guided as Supervisor 3 Ph.D. candidates (Awarded).and 5 candidates are in advance state of work. He has also guided 28 MTech. Students for dissertation. He has also supervised 2 foreign students for internship. Dr.Bhasker Pant has more than 70 research publication in National and international Journals. He has also chaired a session in Robust Classification & Predictive Modelling for classification held at Huangshi, China.

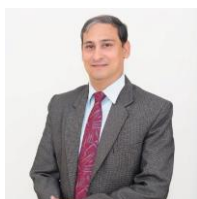


**Kireet joshi**, kireet joshi is an assistant professor at graphic era university,dehradun. He received his masters (computer science & engg.) From kumaoun engineering college,dwarahat. He has more than 7 years of professional experience.his research interests includes algorithm design, component based software engineering and data mining. He holds professional membership in acm etc.



**Vikas Tripathi**, He has done BE in information technology from Technocrats institute of technology, Bhopal , M. Tech in Software engineering from Indian institute of information technology Gwalior and PhD from Uttarakhand technical university, Dehradun. He is actively involved in research related to Software engineering, Computer Vision, Machine learning and Video Analytics. He has published many papers in reputed international conferences and journals. Currently he is working as an associate professor in Graphic era deemed to be university Dehradun, India.

AUTHORS PROFILE



**Manish mahajan**, he is pursuing ph.d. From graphic era university, dehradun. He received his b.e (electronics telecommn.) From marathwada university in 1992 and m.tech in 2005. He is a member of ieee, acm, scrs, iaeng, and has published 14 research papers in national and international journals/conferences in the fields of computer security, software testing and data mining. He has over 20 years of experience in teaching/research of ug( b.tech) and pg(m.tech) level courses as a lecturer/assistant professor/associate professor in organizations like abes it, ghaziabad, ipec, ghaziabad, abes ec ghaziabad and graphic era university, dehradun to name a few. He is currently associated with graphic era university as associate professor and is heading the of department of

