Indian Journal of Innovations and Developments Vol 5 (4), April, 2016
ISSN (online) : 2277-5390
ISSN (Print) : 2277-5382

Efficient Big Data Classification through Distributed Kernel-Based Extreme Learning Machine Approach

Karthick N[*1] , Dr. X. Agnes Kalarani[2]

[1]Research Scholar, Dept of Computer Science, Karpagam Academy of Higher Education, Coimbatore-641021, Tamil Nadu, India
[2]Professor, Dept of Computer Application, Karpagam Academy of Higher Education, Coimbatore-641021, Tamil Nadu, India
*[1]karthickphd333@gmail.com , [2]agneskala72@gmail.com

Abstract

Objectives: The main objective of this work is to provide effective big data classification by analyzing the classification performance of the existing machine learning algorithms.
Methods/Statistical analysis: In MapReduce, after pre-processing and initial partitioning, the features are selected by reducing the dimensionality using the SPARK ITFS technique and then the classification is performed at each node. The Extreme Learning Machine (ELM) is utilized in this work to provide efficient big data classification with reduced computation and network traffic cost. The work is also aimed to reduce the overall memory consumption and reduce time complexity.
Findings: The various research works has been analyzed and evaluated. From the analysis, the classification is performed using machine learning techniques such as Artificial Neural Networks (ANN), Gradient Boosting, Support Vector machine (SVM), Random forests, Naïve Bayes, etc. Though the techniques have good classification performance, the classification performance can be further enhanced by using more advanced machine learning approaches. The proposed Distributed Kernel-ELM (DK-ELM) gives better performance in terms of accuracy, precision, recall and response time compared with existing machine learning algorithms such as SVM, ANN and Partial Least Square-Discriminant Analysis (PLS-DA) utilized in the partition based aggregation methods.
Application/Improvements: The findings of this work prove that the DK-ELM provides better result than other approaches.

Keywords: Big data classification, Artificial Neural Networks, Support Vector machine, Partial Least Square-Discriminant Analysis, Distributed Kernel-ELM

1. Introduction

Big data is a huge volume of datasets where data are heterogeneous and distributed either in structured or semi structured or unstructured format. We go with big data when the data cannot be handled by traditional storage and analysis technologies. In real world applications big data plays an important role because of its huge storage capacity. It is to gain knowledge from the data by applying different techniques. Data in big data can be defined by three different properties of data. Those are the extreme volume of data, the wide variety of types of data and the velocity at which the data must be must processed. By these properties of data, researchers found difficulties to classify the data. Classification of huge amount of data is a tedious process.

Data mining is the process of finding hidden patterns in data. From this useful information can be identified from huge volume of data. This technique categorizes bulk of data based on content-aware partition or network aware partition. Then partitioned the bulk of data into sub partition and feed into different mappers. With the help of SPARK ITFS the features of the data are selected. Finally aggregate the results by reducers from the machine learning algorithms.

In [1] proposed a method to improve Hierarchical Cluster Analysis with the help of PLS. It overcomes the problem of Euclidean distance in Hierarchical Cluster Analysis. PLS extracts only the important information of samples thus it can reduce the dimension of classification variables. The clustering method deals with double data tables.

In [2] proposed Partial Least Square Discriminant Analysis for bankruptcy prediction. PLS-DA is derived from PLS which combines features from Principal Component Analysis (PCA) and Multiple Linear Regression. In this PLS-DA is compared with eight algorithms LDA, LR, MLP, KNN, NB, SVM, C4.5, and BRT and it proves PLS-DA is the best for classification.

In [3] proposed a novel hybrid classifier which constructs SVM on kernel partial least square discriminant space (KPLSDS) for high dimensional data mining. The hybrid classifier reduce computational loading and it also enhance the performance. First KPLSDS maximize the covariance between input and output by optimal projection of original data space to a low dimensional subspace. Further classification is carried out by constructing S4VM on low dimensional KPLSDS.

In [4] presented a hybrid fault diagnosis model composed of Black Box Neural Network (BBNN) and SVM component. It utilizes the advantage of black box modeling to make robustness to SVM approach. It can detect any type of faults in HVAC system by integrating online SVM Classifier with ANN black box model. Additionally it can detect the faults within few seconds or time and with minimum errors.

In [5] proposed a new feature selection method which is called support vector machine (SVM)-based recursive feature elimination (SVM-RFE) with a normalized mutual information feature selection (NMIFS). It combines the RFE with an SVM classifier. It improves some features of SVM by reducing redundancy in feature selection and cost. A fuzzy c-means (FCM) is integrated with level-set-based segmentation method to reduce labor cost and feature selection methods are investigated for the mass classification problem.

In [6] discussed Arabic text categorization model based on artificial neural network. Here the feature reduction methods are combined with PCA to reduce high dimension feature space. Additionally it improves the categorization performance and it reduces the computational time in the back propagation neural networks by reducing size of vectors.

In [7] proposed SVM for Khmer character classification system. Khmer is one of the most complex languages. In his method printed Khmer character feed as an input image and it converted into machine codes document format. It utilize three different SVM kernels are Gaussian, Polynomial and Linear Kernel on training data set and the result shows the Gaussian is the best SVM kernel. If we choose the best training data sets that leads to better results in SVM classification.

In [8] proposed to discover salient regions of pix at some point of training and fold the data to reduce the effect of inappropriate regions. First salient detector is used to attain high classification score then fold the images that contain only salient areas which reduce classification errors and classify the images by SVM based on LBP features. Finally the results are computed with reported Image CLEF for error score evaluation.

In [9] proposed a new method for large number of short text classification and mining useful information from short text. Classification methods like Naive Bayes (NB), Support Vector Machines (SVM), Neural Network (NN), Decision Tree (DT), k-Nearest Neighbor (KNN) show better performance for large text. But the above classification methods show less performance for short text. To improve the performance of classification semi-supervised learning and SVM is used to learn and label the unlabeled samples in the short text.

In [10] used a Map Reduce-based framework for incremental big data analyzing that combined a fine-grain incremental engine and a set of effective techniques for incremental iterative computation. Re-computation of the data can be reduced by i2MapReduce which supports key value pair and it modeled as MRBG-Store to retrieve fine-grain states for incremental processing. Project API function is proposed for General-purpose iterative computation.

In [11] presented an Integrated ANN approach to predict monthly mean clearness index based on environmental and meteorological factors. This approach is used where the measurement equipments are not available. This approach shows high accuracy in terms of mean absolute percentage error (MAPE) from those results geographic information system (GIS) is plotted it illustrates the distribution of solar energy.

In [12] introduced Content based Spark-ITFS feature selection and Network based Spark-ITFS [13] feature selection to remove irrelevant information in big data. This filter selects relevant information from the huge amount of data in order to improve performance of big data and reduce network traffic cost.

In our previous research work, the data mining is done effectively by various machine learning algorithms like SVM, ANN and PLS-DA. But there is some drawback in classification of those machine learning algorithms. In SVM technique the drawbacks are it is not suited for all datasets, computational cost is high and storage problems. The downsides of ANN technique are time complexity is high, error rate may increase and the PLS-DA technique takes more in computation. These issues need to be resolved with an effective classification method for better classification of data.

In this research work a new machine learning technique called DK-ELM is introduced to improve the efficiency and reduce computation and network traffic cost of big data classification and it also compared with existing machine learning techniques to prove its efficiency and cost consideration. The efficiency of DK-ELM can be estimated in terms of accuracy, precision and recall. In DK-ELM classify the data based on feature selection of SPARK ITFS and merge the data with unique key values by using reducer.

2. Data classification

Data Classification is an important process in data mining. The data are aggregated from different sources in a database it may contain irrelevant data or noisy data. In order to remove the irrelevant data or mine the important information from the database various classification techniques are used. Such techniques namely ANN, SVM and PLS-DA are discussed here.

In the SVM classification work is of two phases. The training phase includes creation of a training model on which the nodes are compared in testing phase. The Support Vector machine trains and tests the nodes with the data assigned into different classes. The nodes are trained by creating a model for marginal hyper-plane of the data and the binary classification of the nodes performed. The separating hyper-plane is the hyper-plane that maximizes the distance between the two parallel hyper-planes. Then in the testing phase the nodes are compared with the trained classes and the testing nodes are classified. The downside of this technique is storage cost, computation cost is high and it is suited for all datasets.

In ANN classification technique each input sample, it can be positive, negative or neutral in the data sample. For the number of input training sample we have to perform the analysis of reviews. For each input sample the neuron network perform the output sample based on the prior knowledge. It is used to search the more accurate results along with hidden layer and this layer is used to map the semantic results for corresponding input sample. Though it overcomes the drawbacks in SVM still it has some drawbacks in ANN are time complexity is high and errors may occur.

In PLS-DA model is a statistical method that bears some relation to principal components regression; instead of finding hyper-planes of minimum variance between the response and independent variables, it finds a linear regression model by projecting the predicted variables and the observable variables to a new space. Because both the X and Y data are projected to new spaces, the PLS-DA method is known as bilinear factor models. Thus the classification based on the partial least square can improve the classification accuracy.

Thus the various classification technique principles and disadvantages of classification techniques are discussed. To enhance the performance of classification technique a new technique called DK-ELM is proposed and explained in proceeding section.

3. Distributed Kernel-Extreme Learning Machine

In previous research, the data can be classified by using machine learning techniques namely SVM, ANN, PLS-DA. But still there are some more problems in those techniques such as matrix inversion and matrix multiplication. So here we proposed a machine learning technique called DK-ELM based on MapReduce. It provides an alternative realization of kernel matrix inversion by using matrix decomposition and multiplication of matrix with vector.

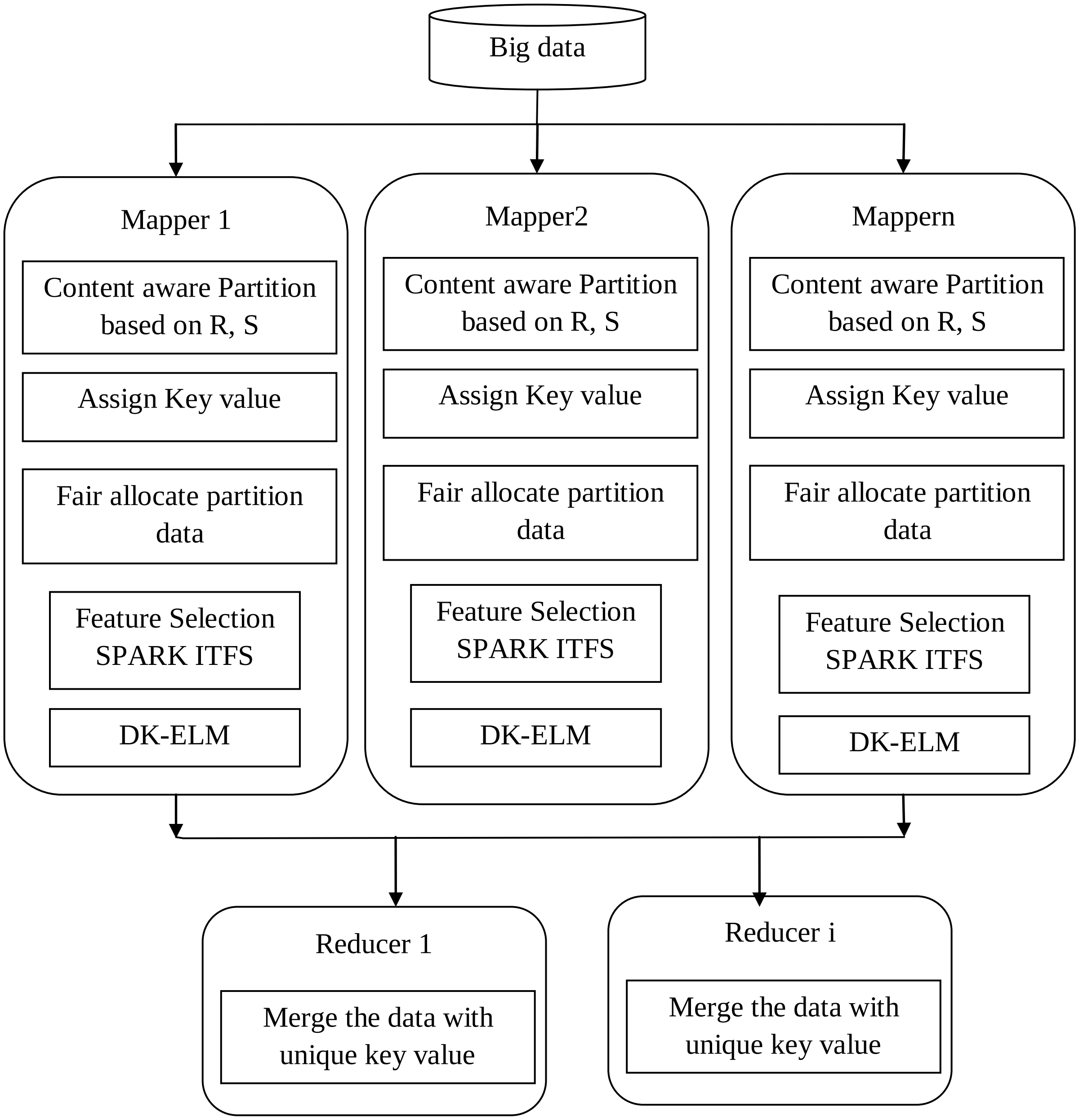
In MapReduce technique the data from big data are partitioned into n number of partitions and it will allocate to n number of Mappers which is across distributed system. The data in the database may be either content based or network based. It can be automatically partitioned and computed in Map. For each partition assign key values and allocate the appropriate data to the nodes from the allocated data the features are extracted using filters and classify the data based on features by using Distributed Kernel-Extreme Learning Machine. Finally aggregate the intermediate results in Reduce to gain useful information from big data. These all are carried out as parallel calculation.

In this research work, the Distributed Kernel-Extreme Learning Machine technique is used to classify the data and enhances the performance of classification. The data may be partitioned based on either content-aware or network aware. Thus the content-aware partition with DK-ELM and network aware partition with DK-ELM are explained in the following section.

3.1. Content-Aware Partition With DK-ELM

Content aware Partition with DK-ELM approach is of three features: the data partition, Computation model and the communication and data exchange. The ELM tree with the distributed kernel is modeled into a kernel matrix Ω and the weight of the matrix is estimated after normalizing and decomposing the matrix into two orthogonal matrices. Then the Distributed RBF accepts the training samples x in columnar format as input. Each input key-value pair contains a column ID as key and the corresponding elements to this column as value. The Map function of D-RBF first transforms the value content into a vector K. The output key of Map is the index of kernel matrix Ω, then the Reduce function is activated to reduce the computation cost. Then the distributed matrix-vector multiplication is carried out in the MapReduce to reduce the memory consumption. The work flow of content-aware partition with DK-ELM is shown in Figure 1.

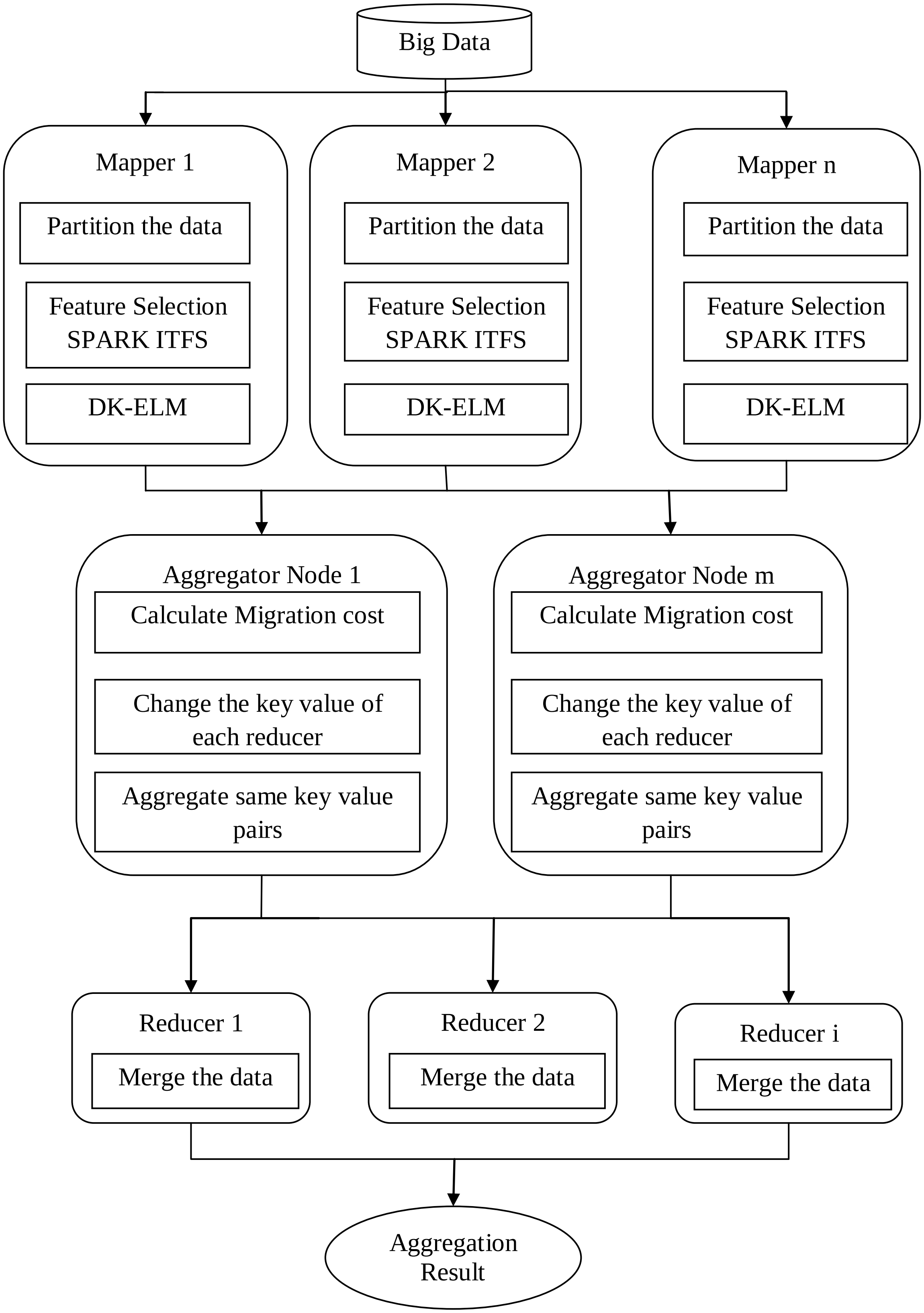
Figure 1. Work flow of content-aware partition with DK-ELM



3.2. Network aware partition with DK-ELM

The network aware partition with DK-ELM is similar to the content aware DK-ELM but the network aware partition with DK-ELM focuses more on reducing the network cost than the content aware method. In this method, in order to avoid the great memory consumption of large matrix operations of ELM with kernels in single-machine environment, DK-ELM is proposed and implemented on MapReduce. The DK-ELM is modeled into a kernel matrix Ω which is decomposed into two orthogonal matrices. Then the distributed RBF is implemented on MapReduce and the matrix-vector multiplications are performed with reduced memory constraints. DK-ELM includes three major parts: distributed RBF kernel matrix calculation by D-RBF on MapReduce; decomposition of the kernel matrix and performing its inverse matrix calculation in parallel; Distributed matrix-vector multiplication is proposed to implement the multiplications of the intermediate matrices with vectors. Thus the network aware DK-ELM can reduce the memory consumption along with reduction in network traffic cost. The work flow of network-aware partition with DK-ELM is shown in Figure 2.

Figure 2. Work flow of network-aware partition with DKELM



All matrices can be decomposed as either by rows or by column. The matrix can be decomposed as

$$\Omega' = P\Sigma Q \qquad (1)$$

In equation 1 Σis a diagonal matrix with nonnegative real numbers on the diagonal. P and Q are orthogonal matrices.

The inverse matrix is calculate as

$$(\Omega^{'})^{-1} = (\mathrm{P}\Sigma\mathrm{Q}^{\mathrm{T}})^{-1} = \left(\mathrm{Q}^{\mathrm{T}}\right)^{-1}\Sigma^{-1}\mathrm{P}^{-1} = \mathrm{Q}\Sigma^{-1}\mathrm{P}^{\mathrm{T}} \qquad (2)$$

Considering $(\Omega^{'})^{-1}$M as an *N*×1 output weight vector **V**, $\mathrm{Q}\Sigma^{-1}$ as an *N*×*N* matrixR and $\mathrm{P}^{\mathrm{T}}T$as an *N* × 1 vector **S**, we have the output weight as

$$\mathrm{V} = \mathrm{RS} = Q\Sigma^{-1}\mathrm{P}^{\mathrm{T}}\mathrm{M} \qquad (3)$$

DK-ELM is described as DK-ELM algorithm

DK-ELM Algorithm

Input: training data
Output: output weight V
1: Call distributed kernel functions to calculate matrix Ω and then $\Omega^{'}$;
2: Decompose $\Omega^{'}$into P, Σ and Q on MapReduce;

3: Calculate output weight $Q\Sigma^{-1}\mathrm{P}^{\mathrm{T}}$M**;**

In the above algorithm the distributed kernel matrix **Ω** and is calculated (Line 1) in the proceeding section 3.3. The Stochastic Singular Value Decomposition (SSVD) method was used in $\Omega^{'}$to avoid N rounds of iteration (Line 2). The output weight is calculated (Line 3). In this we calculate $Q\Sigma^{-1}$and P^{T}Mrespectively and simultaneously**.** This detailed description is explained in section 3.4. Thus the calculation problem in matrix operation is resolved and the performance of learning phase is improved.

3.3. Spherical kernel Function

In this section, spherical kernel function is described. It has a linear behavior at the origin. This has huge advantage in certain applications with massive data sets where Gama matrix G will be sparse. Sparse linear algebra techniques are used to resolve the Gama matrix linear systems.
The spherical kernel is defined as:

$$W(x,y) = 1 - \frac{3}{2}\left(\frac{||x-y||}{\sigma}\right) + \frac{1}{2}\left(\frac{||x-y||}{\sigma}\right)^{3} \qquad (4)$$

wherex,y are coordinates and sigma is adjustable parameter. It plays a major role in the performance of the kernel, and should be carefully tuned to the problem at hand.

Spherical Kernel Algorithm

Input: training data **y**in columnar format, kernel parameters *σ*
Output: kernel matrix Ω
1: **function** Map (*CID, data*)
2: Parse *data* into Vector *W*
3: **for** *i*=1 to *W* size () **do**
4: **for** *j*=1 to *i* **do**
5: *W*(*i, j*) = (*W*[*i*] −*W*[*j*]) ∗(*W*[*i*] −*W*[*j*]);
6: *context*.write (<*i,j*>,*W*(*i, j*));
7: **end for**
8: **end for**
9: **end function**
10: **function** Reduce (<*i, j* >,list[*W*(*i, j*)],*σ*)
11: Initiate spherical = 0;

12: **for all** *W* ∈lis [(*W*(*i, j*)] **do**
13: spherical= *spherical*+ *W*;
14: **end for**
15: **if** $||x - y|| < \sigma$ then
16: $spherical = 1 - \frac{3}{2}\left(\frac{||x-y||}{\sigma}\right) + \frac{1}{2}\left(\frac{||x-y||}{\sigma}\right)^3$
17: **else**
18: $\mathrm{spherical} = 0$
19: **end if**
20: *context*.write (<*i,j*>,*spherical*);
21: *context*.write (< *j, i*>,*spherical*);
22: **end function**

In the above Spherical Kernel Algorithm accepts input as training data y in columnar format. Each data contain key as CID and corresponding elements are as values. In the Map function the value data is converted into vector W. That is, for the *i*-th column, we have W[*j*] = **x***ij*. Elements of each two samples in vector Ware subtracted and squared to get the partial summation of the 2-norm of each two samples difference (Lines 3-8). The output key of Map is the index <*i,j*>of kernel matrix **Ω**, while value is the partial summation *K*(*i, j*). Then the Reduce function adds all the partial summations corresponding to a same <*i,j*> index (Lines 12-14). Finally we have each final element value Ω*i,j*(Lines 15-17). Note that kernel matrix **Ω** is a symmetric matrix. For optimization, only an upper triangular matrix or a lower triangular matrix needs to be calculated (Lines 3,4,15,16), which reduces the computation cost by half.

3.4. Distributed Matrix-Vector Multiplication

In this section, the parallel multiplication of matrix with vectors is done by proposed Distributed Matrix-Vector Multiplication to calculate $Q\Sigma^{-1}$ and$\mathrm{P^T}$M. Distributed Matrix-Vector Multiplication transforms square matrix Σinto Σ' N×1 vector the reciprocal of ΣisΣ'. It combines multiple MapReduce job into single MapReduce job by multiplying N×N matrix by N×1 matrix. Finally we get R N×N matrix and N×1 vector S.

Distributed Matrix-Vector Multiplication Algorithm
Input: matrix **P** and **Q**
Output: matrix **R** and vector **S**
function Map (<*matID, rID*>*, data*)
Vector *U*= DistributedCache.get (”vectorSigma”);
Vector *M*= DistributedCache.get (”vectorM”);
Parse *data*into Vector *Z*;
for *i*= 1 to *Z*.size () **do**
if *matID*== ”Q” **then**
CID = *i*;
result= *Z*[*i*]*U*[*i*];
context.write (<*matID, rID, CID* >*, result*);
else if *matID*== ”P” **then**
CID = *null*;
result= *X*[*i*]*T*[*i*];
context.write (<*matID, rID, CID* >*, result*);
end if
end for
end function
function Reduce (<*matID, rID, CID* >,list[*result*])
Initiate *content*= 0;
for all *result*∈ *list* [*result*] **do**
content=*content*+ *result*;
end for
context.write (<*matID, rID, CID* >*, content*);
end function

In the above algorithm, the inverse matrix of Σis calculated. Thus $Z[i]U[i]$ is a final element in R which is indexed by <rID, CID > where rID is the row number and CID is the column number of R. matID check the key-value pair belongs to which matrix (Line 9). This is for the case $Q\Sigma^{-1}$(Lines 7-9). For $\mathrm{P}^{\mathrm{T}}\mathrm{M}$ (Lines 11-13)**,** each$Z[i]U[i]$ (Line 12) is the a partial summation of vector S element. All the partial summation results to the same element in S will be marked by a same key *<matID, rID, CID >* (Line 13), where *matID* identifies the result matrix, *rID* indicates the element index of **S**. Since the result of $\mathrm{P}^{\mathrm{T}}\mathrm{M}$ is an $N \times 1$ vector, *CID* is no use and set to *null* (Line 11). In Reduce function, all the partial summation values of the same final element are added up (Lines 18-21). The final output of Reduce function is in the form of *<<matrixID, rowID, columnID>, element >*(Line 22).

4. Experimental results

In this section the performance of each machine learning technique is calculated and compared with proposed technique based on content based partition and network based partition. To extract useful information and to compare the effectiveness of our technique KDD 99 data set is used here. The comparison is made between proposed techniques DK-ELM and existing techniques namely ANN, SVM, PLS-DA in terms of accuracy, recall, precision and time measure. This can be explained briefly in the following section.

Accuracy

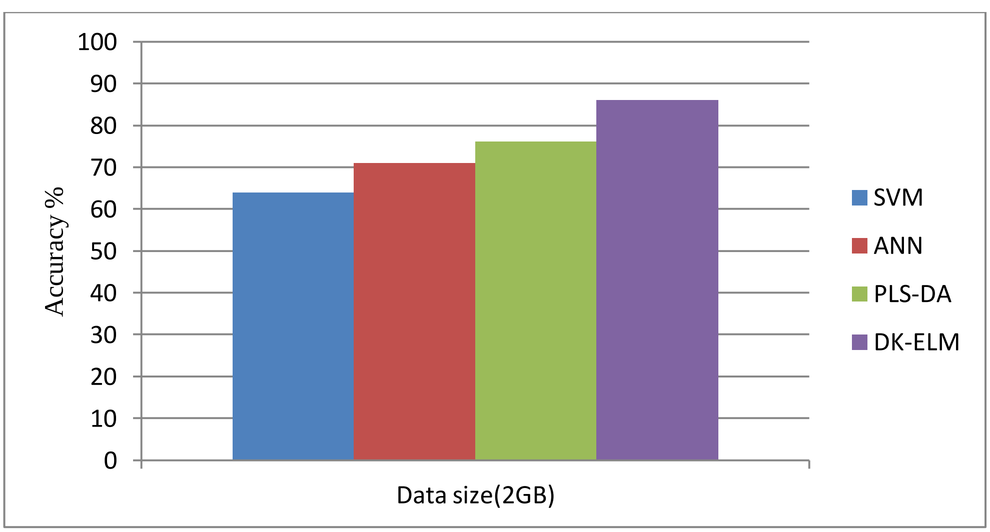
Social Accuracy is defined as the proportion of true positives and true negatives among the total number of results obtained. Accuracy is evaluated as,

$$Accuracy = \frac{(Truepositive + Truenegative)}{(Truepositive + Truenegative + Falsepositive + Falsenegative)}$$

The accuracy of proposed technique can be evaluated based on content aware partition and network aware partition.

From the figure 3, 4 it is proved that the proposed classification technique classify the data effectively. In content aware partition DK-ELM provides 86% accuracy rate whereas SVM provides 64%, ANN provides 71%, PLS-DA provides 76%. In network aware partition DK-ELM provides 89% accuracy rate whereas SVM provides 68%, ANN provides 77%, PLS-DA provides 82%. From this it is proved the proposed classification technique provides better accuracy than other classification techniques.

Figure 3. Accuracy comparison based on content aware partition



Precision

Precision value is evaluated according to the relevant information at true positive prediction, false positive.

$$Precision = \frac{Truepositive}{(Truepositive + Falsepositive)}$$

The Precision value of proposed classification technique can be evaluated based on content aware partition and network aware partition.

From the figure 5, 6 it is proved that the proposed classification technique classify the data effectively. In content aware partition DK-ELM provides 0.89 precision values whereas SVM provides 0.62, ANN provides 0.69, PLS-DA provides 0.76. In network aware partition DK-ELM provides 0.91precision value whereas SVM provides 0.65ANN provides 0.71, PLS-DA provides 0.81. From this it is proved the proposed classification technique provides better precision than other classification techniques.

Figure 4. Accuracy comparison based on network aware partition

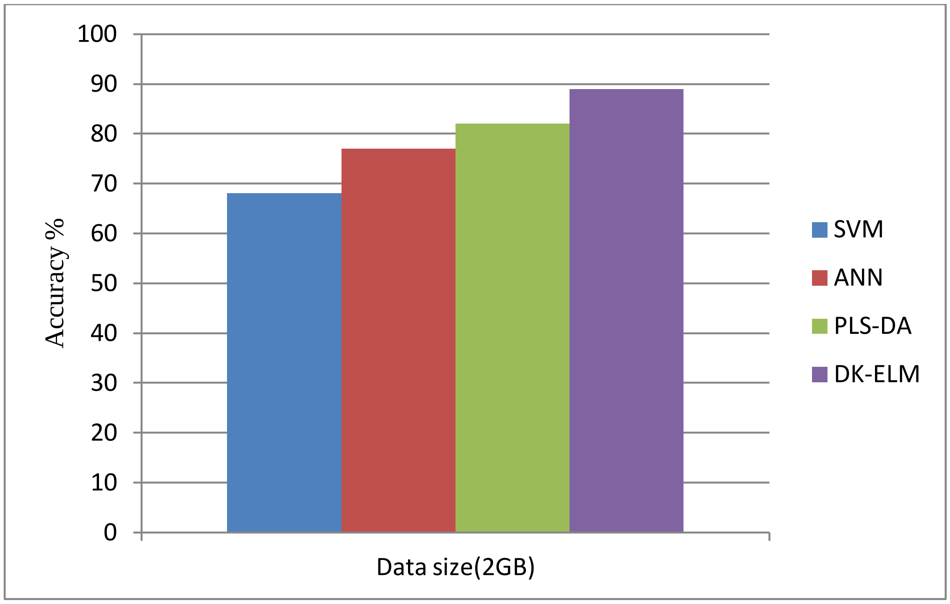


Figure 5.Precision comparison based on content aware partition

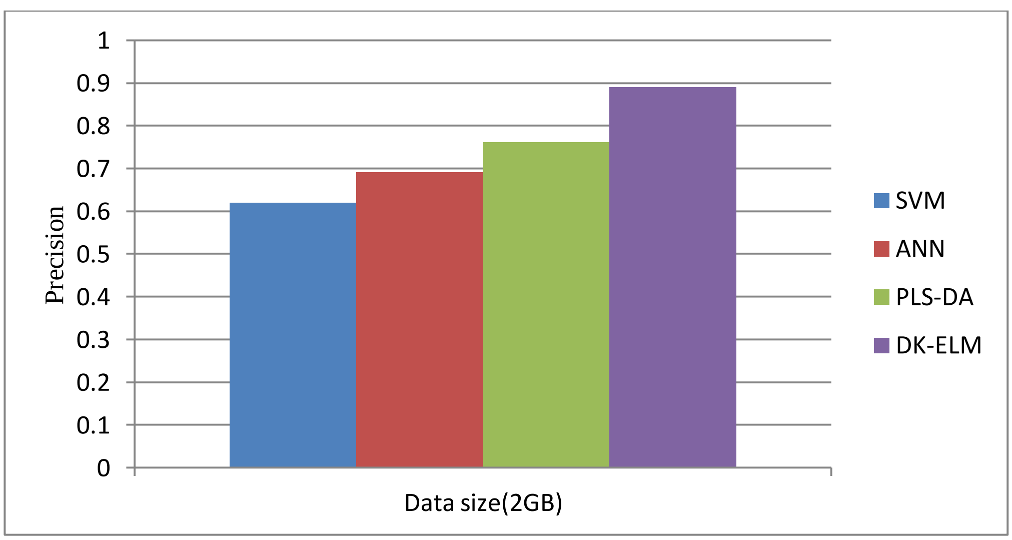


Figure 6. Precision comparison based on network aware partition

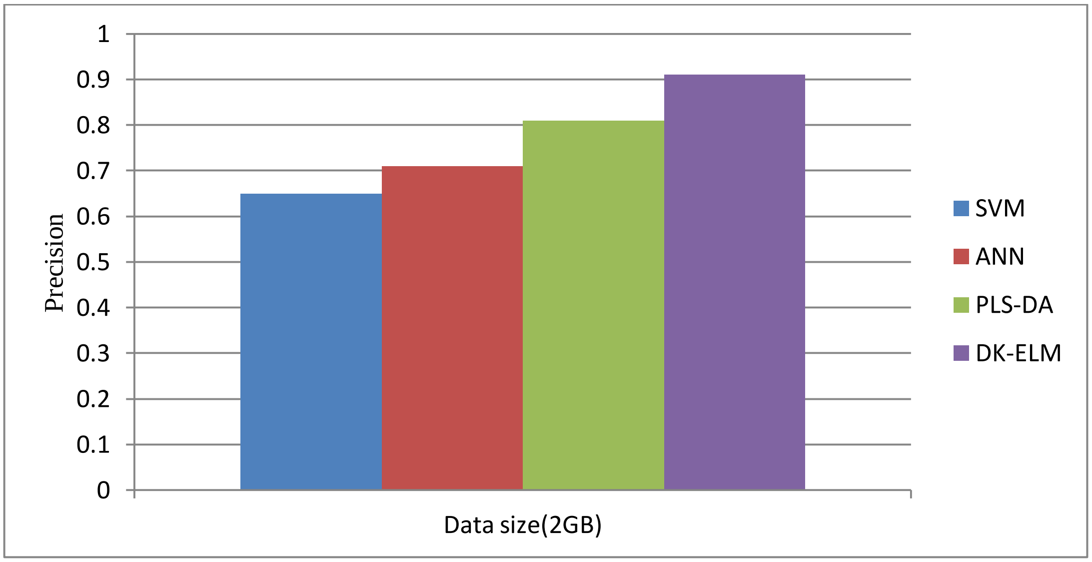


Figure 7. Recall comparison based on content aware partition

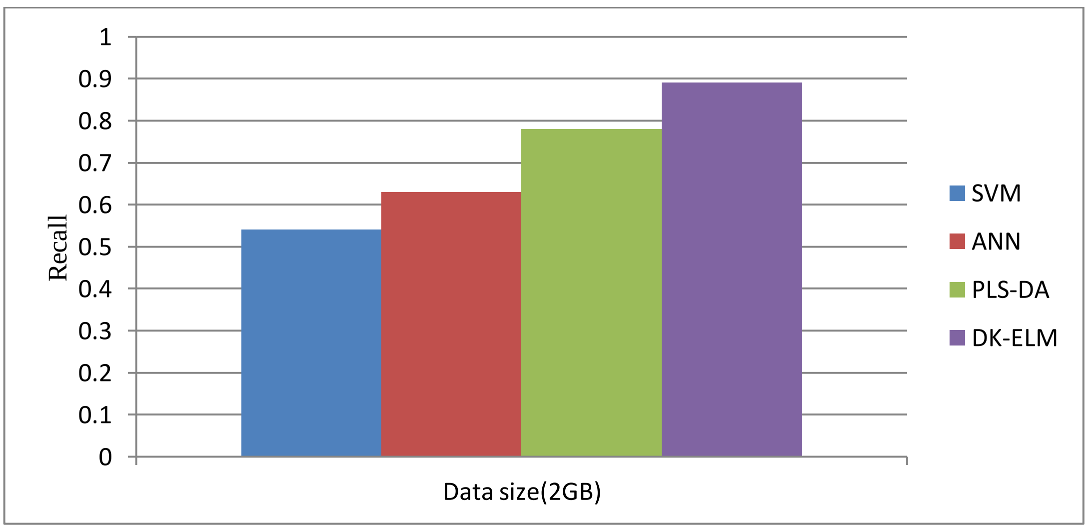
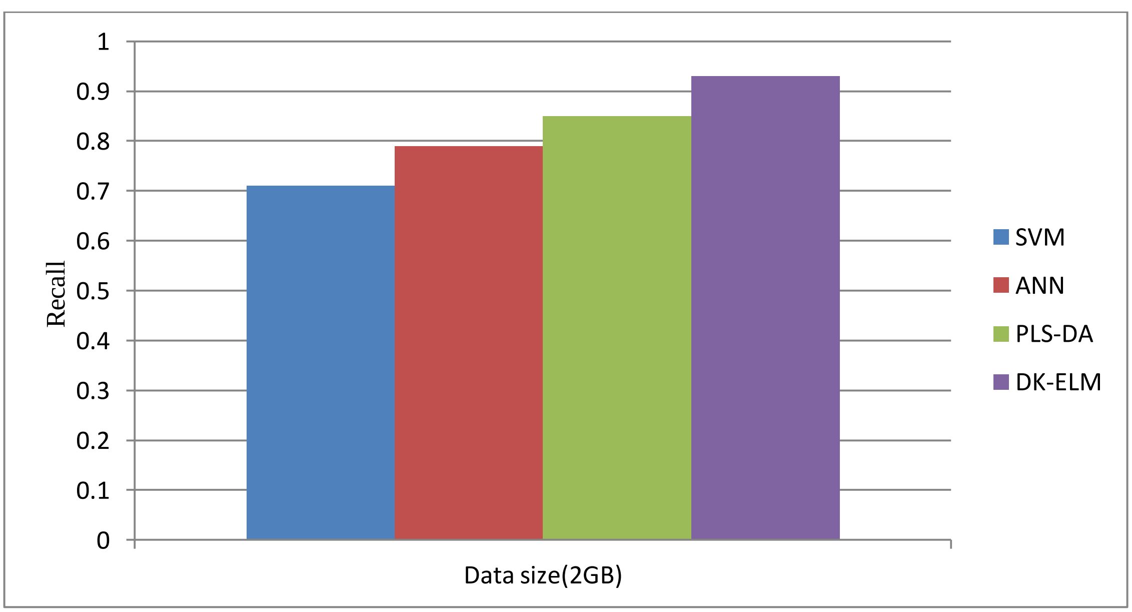


Figure 8. Recall comparison based on network aware partition



Recall

The Recall value is evaluated according to the retrieval of information at true positive prediction, false negative.

$$Recall = \frac{Truepositive}{(Truepositive + Falsenegative)}$$

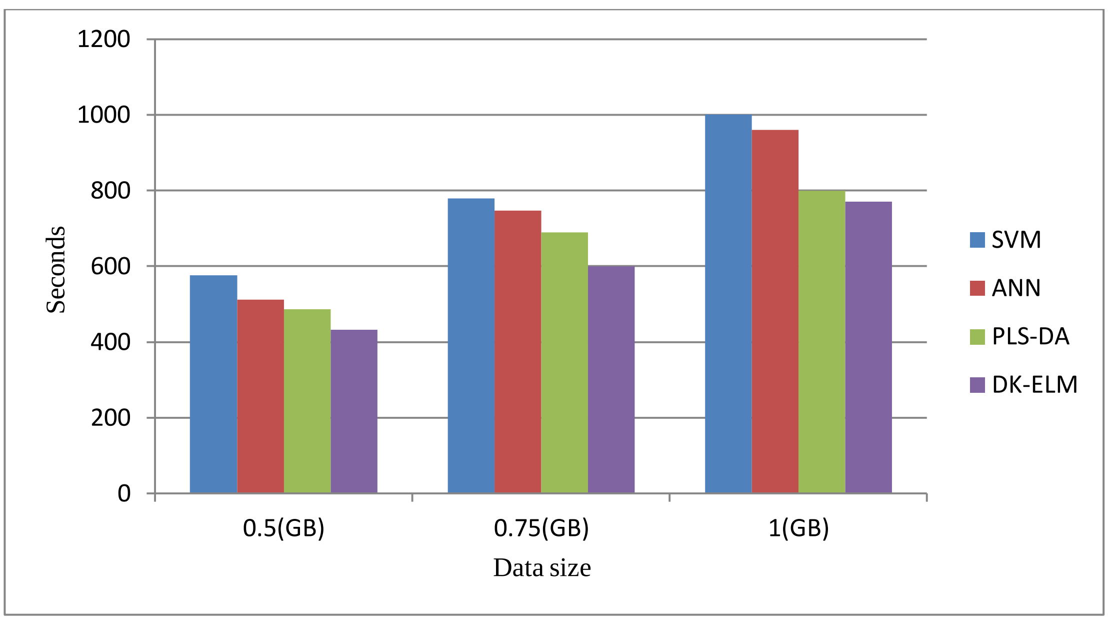
The Recall value of proposed classification technique can be evaluated based on content aware partition and network aware partition.

From the figure 7, 8 it is proved that the proposed classification technique classify the data effectively. In content aware partition DK-ELM provides 0.89 recall value whereas SVM provides 0.54, ANN provides 0.63, PLS-DA provides 0.78. In network aware partition DK-ELM provides 0.93 recall value whereas SVM provides 0.71 ANN provides 0.79, PLS-DA provides 0.85. From this it is proved the proposed classification technique provides better recall than other classification techniques.

Response Time

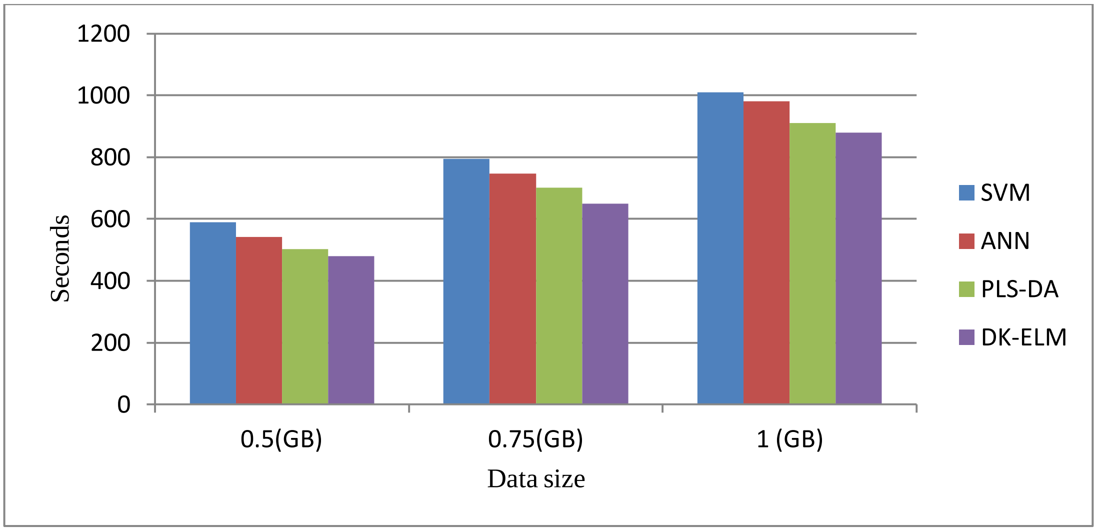
Response Time Measure is evaluated based on the run time of classification techniques. The proposed DK-ELM requires less time to classify the data.

Figure 9. Response Time comparison based on content aware partition



From the figure 9, 10 it is proved that the proposed classification technique classify the data effectively. In content aware partition DK-ELM provides response time as 770s whereas SVM provides 1000s, ANN provides 800s, PLS-DA provides960s. In network aware partition DK-ELM provides response time as 880 whereas SVM provides 1010 ANN provides 980, PLS-DA provides 910. From this it is proved the proposed classification technique provides better response time than other classification techniques.

Figure 10. Response Time comparison based on network aware partition



5. Conclusion

In this paper we proposed a new classification technique called DK-ELM which uses KDD 99 as data set. The three important features of DK-ELM are: 1. Spherical kernel is used for matrix calculation on MapReduce 2. To decompose a matrix and realize its inverse matrix Stochastic Singular Value Decomposition is used. 3. To implement the multiplications of the intermediate matrices with vectors DMXV is proposed. From the experimental results it proved that the proposed DK-ELM is superior to other machine learning classification techniques.

6. References

1. J. L., Liu, Y. Bai, J. Kang, N. An. A new approach to hierarchical clustering using partial least squares. In 2006 International Conference on Machine Learning and Cybernetics. IEEE .2006 Aug, pp. 1125-1131.
2. C. Serrano-Cinca, B. Gutiérrez-Nieto. Partial Least Square Discriminant Analysis (PLS-DA) for bankruptcy prediction. Working Papers CEB, 2011, 11.
3. S. C. Huang, T. K. Wu. Robust semi-supervised SVM on kernel partial least discriminant space for high dimensional data mining. In 2012 International Conference on Information Science and Applications. IEEE, 2012, May, pp. 1-6.
4. D. Dehestani, S. Su, H. Nguyen, Y. Guo. Robust fault tolerant application for HVAC system based on combination of online SVM and ANN black box model. In Proceedings of European Control Conference (ECC), 2013, July, pp. 2976-2981.
5. X. Liu, J.Tang. Mass classification in mammograms using selected geometry and texture features, and a new SVM-based feature selection method. IEEE Systems Journal. 2014, 8(3), 910-920.
6. F. A. Zaghoul, S. Al-Dhaheri. Arabic Text Classification Based on Features Reduction Using Artificial Neural Networks. In Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on. IEEE. 2013, April, pp. 485-490
7. P. Sok, N. Taing. Support Vector Machine (SVM) Based Classifier For Khmer Printed Character-set Recognition. In Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, IEEE , 2014 Dec, pp. 1-9.
8. Z. Camlica, H. R. Tizhoosh, F. Khalvati. Medical image classification via svm using lbp features from saliency-based folded data.arXiv preprint arXiv:1509.04619. 2015.

9. C. Yin, J. Xiang, H. Zhang, J. Wang, Z.Yin, J. U. Kim. A New SVM Method for Short Text Classification Based on Semi-Supervised Learning. In 2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS) 2015, IEEE, Aug, pp. 100-103.
10. Y. Zhang, S. Chen, Q. Wang, G.Yu. iMapReduce: Incremental MapReduce for Mining Evolving Big Data. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(7), 1906-1919.
11. S. Kheradmanda, O. Nematollahi, A. R. Ayoobia. Clearness index predicting using an integrated artificial neural network (ANN) approach. Renewable and Sustainable Energy Reviews, 2016, 58, 1357-1365.
12. N. Karthick, X. A. Kalarani. An Effective Filtering on Big Data by Finding Relevant Features to Extract Useful Information. International Journal of Applied Engineering Research, 2016, 11(6), 3805-3810.
13. N. Karthick, X. A. Kalarani. An improved method for handling and extracting useful information from big data. Indian Journal of Science and Technology, 2015, 8(33), 1-7.

The Publication fee is defrayed by Indian Society for Education and Environment (iSee). www.iseeadyar.org

Citation:

Karthick N , Dr. X. Agnes Kalarani. Efficient Big Data Classification through Distributed Kernel-Based Extreme Learning Machine Approach. *Indian Journal of Innovations and Developments.* 2016; 5 (4), April.