BIG DATA CLUSTERING USING HEURISTIC DATA INTENSIVE COMPUTING AND SELF-ORGANIZING MAP

[1]M.Divyasri, [2]K.Prasanna, [3]M.Edwin Jayasingh
[1]UG Student, [2]Associate Professor, [3]Associate Professor,
[1,2,3]Department of Computer Science and Engineering, AITS, Rajampet, India.

Abstract : Conventional information grouping calculations are having traps while finding effective bunches. As the information base size builds powerfully and the sensational changes in the utilization of information, will shows sufficient outcomes on grouping execution. A brought together structure is exhibited for enormous information bunching utilizing a Heuristic information serious registering (HDIC) and Self-Organizing Maps (SOM). It is actualized on a N-hub HDIC groups, driven by a wide scope of informational indexes made utilizing IBM engineered information generator and ongoing informational collections taken from UCI. A structure for the grouping of enormous information utilizing Heuristic Data Intensive figuring and Self-Organizing Maps calculation has been proposed.

IndexTerms – Self-Organizing Maps, Clustering, Big Data.

1. INTRODUCTION

1.1. BIG DATA

Enormous information is a term that portrays the huge volume of information both organized and unstructured that immerses a business on an everyday premise .But it's not the measure of information that is significant .It's organizations main thing with the information that issues .Big information can be investigated for experiences that lead to better choices and key business moves .While the expression "huge information" is generally new, the demonstration of social occasion and putting away a lot of data for possible examination is ages old. The idea picked up energy in the mid-2000s when industry expert Doug Laney explained the now-standard meaning of enormous information as the three V's:

- ➢ **Volume:** Organizations collect data from a variety of sources, including business transactions, social media and information from sensor or machine-to-machine data. In the past, storing it would've been a problem – but new technologies (such as Hadoop) have eased the burden.
- ➢ **Velocity:** Data streams in at an unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time.
- ➢ **Variety:** Data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents, email, video, audio, stock ticker data and financial transactions.

1.2 CLUSTERING

Grouping can be considered the most significant solo learning issue; along these lines, as each other issue of this sort, it manages finding a structure in an assortment of unlabeled information. A free meaning of bunching could be "the way toward arranging objects into bunches whose individuals are comparable here and there". A group is along these lines an assortment of articles which are "comparative" among them and are "different" to the items having a place with different bunches.

1.3 LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is taken into account for developing the proposed system.

1.4 Clustering Algorithms and their Challenges with Big Data

Big Data clustering techniques can be classified into two categories single machine clustering techniques and multiple machine clustering techniques.

2. Single Machine Clustering

2.1 Partitioning Based Clustering Algorithm

It divides a data set in a single partition using a distance to classify points based on their similarities. These clusters should fulfill the following requirements:
(1) Each group must contain at least one object
(2) Each object must belong to exactly one group.
Examples of this type of classification algorithms are K-means, k-mediods.

2.1.1 K-Means Clustering

1) Arbitrarily pick k objects from D dataset as the underlying centroids
2) Repeat

3) Reassign each article to the bunches to which the item is the most comparative, in view of the mean estimation of the items in groups
4) Update the bunch implies, that is, figure the mean estimation of the articles for each group.
5) Until no change

2.1.2 K-Medoids

It varies from the k-means algorithm mainly in its representation of the different groups or clusters.

Algorithm:
1) Starts from an underlying arrangement of medoids and bunches are produced by which are near separate medoids.
2) The calculation iteratively replaces one of the medoids by one of the non-medoids on the off chance that it improves the all-out separation of the subsequent grouping.

Challenges:
1) Poor at taking care of uproarious information and exceptions.
2) Works just on numeric information.
3) Empty group age issue.

2.2 Hierarchical Based Clustering

This technique manufactures groups in a progressive request, it structures settled bunches sorted out in a various levelled tree. It structures bunches by recursively or iteratively dividing the examples in either a top-down or base up style. Various levelled bunching technique is of two sorts:

1. **Agglomerative:** This is a base up approach. Right now, each item is considered as a different individual group. It at that point combines at least two appropriate bunches to frame new groups. This converging of groups is done recursively until an ideal bunch structure or a halting standard (wanted number of groups k) is come to.
2. **Divisive:** This is top down methodology. Right now, the whole dataset is considered as one group. The bunch is then separated in to sub-groups, which thus are progressively isolated into more sub-groups. This procedure is rehashed until the halting basis (wanted number of groups k) is met. Case of this sort of order calculation is BIRCH.

3. EXISTING SYSTEM

Information size has expanded progressively with the approach of the present innovation in numerous divisions, for example, Manufacturing, Business, Science and Web applications. The vast majority of the information are organized, and a few information are semi-organized while others are unstructured and blend in with various kinds of information, for example, reports, records, pictures and recordings. Assets of information are from Web applications, which produce an exceptionally enormous volume of information.

Support vector machines (SVM) are managed learning models with related learning calculations that break down information utilized for order and relapse examination. A SVM model is a portrayal of the models as focuses in space, mapped so the instances of the different classifications are isolated by a reasonable hole that is as wide as could be expected under the circumstances.

Grouping utilizing SVM adequately consolidate the distributional property of the preparation information into the preparation procedure. It is normal that the comparable thought can be utilized to improve other directed learning calculation like neural systems. The disservices are that the hypothesis just truly covers the assurance of the parameters for a given estimation of the regularization and bit parameters and decision of portion. In a manner the SVM moves the issue of over-fitting from improving the parameters to demonstrate choice.

3.1 K-Means:

Bunching is the task of a lot of perceptions into subsets (called groups) with the goal that perceptions in a similar group are comparative in some sense. Grouping is a strategy for solo learning, and a typical method for factual information investigation utilized in numerous fields.

K-means clustering is an unsupervised algorithm to classify or to group your objects based on attributes/features into K number of groups. K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Thus, the purpose of K-mean clustering is to classify the data.

The basic step of k-means clustering is simple. In the beginning, determine number of cluster K and assume the centroid or centre of these clusters. User can take any random objects as the initial centroids or the first K objects in sequence can also serve as the initial centroids. Hence the K means algorithm will do the three steps below until convergence:
Step 1. Begin with a decision on the value of k = number of clusters.
Step 2. Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following:
1) Take the first k training sample as single-element clusters.
2) Assign each of the remaining training sample to the cluster with the nearest centroid.
Step 3. Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

3.2 Limitations in Existing System

Unaided calculation absence of system that empowers programmed information dissemination, load adjusting and adaptation to non-critical failure on huge processing groups.

a. The fundamental impediment of this calculation is each time information is shown up, it reproduces the groups.
b. Takes additional time and memory space
c. In existing techniques, the bunches will contain a smaller number of highlights.
d. Difficulty to get exact bunches because of the information base versatility

4. PROPOSED SYSTEM

The proposed system for large information control is appeared in the Figure. It utilizes Heuristic Data Intensive Computing (HDIC) for information handling and Self-Organizing Maps (SOM) calculation is utilized for information bunching. The proposed structure begins with different information sources converged into an ace database. These databases formalize the information escalated registering which can deal with high volume information streams.

Artificial Neural Networks (ANN) based Self-Organized Maps (SOM) algorithm is applied on the processed data to find right number of clusters. SOM is an unsupervised approach and intended to solve NP-Hard problems.

5. SYSTEM METHODOLOGIES

IBM Database Generator: IBM Database Generator is utilized to produce test information without any preparation or from existing information. It will have the parameters like normal exchange size, length, relationship, number of traits and so on. Test information can be produced in an assortment of configurations, including SQL, or XML.

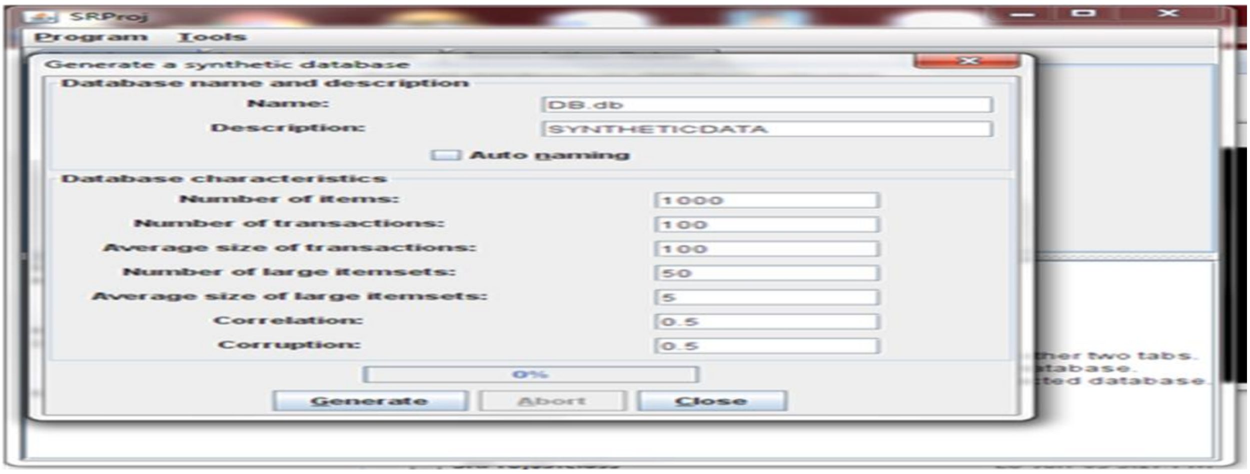
Heuristic Data Intensive Computing (HDIC): Heuristic methods set of decides that are utilized to expand the likelihood of taking care of an issue. Heuristic is utilized to give ideal arrangement. It can give an attainability to change over one type of information into another type of information. Information Intensive Computing is a class of equal registering applications which utilize an information equal way to deal with process huge volumes of information regularly terabytes or peta bytes in size and commonly alluded to as a Big Data. This Heuristic information concentrated processing is applied for uniform information. Here undesirable information or copy information is evacuated by utilizing this procedure.

Clustering using Self Organizing Maps (SOM)
The Self-Organizing Map (SOM) is one most popular neural network model. It is a unsupervised learning which means that no human interaction is needed during the learning. SOM is used for clustering data without knowing the class memberships of input data. The SOM can be used to detect features inherent to the problem and thus has also been called SOFM, the Self-Organizing Feature Map.

Cluster Visualization: Bunch Visualization renders your group information as an intelligent guide permitting you to see a snappy review of your bunch sets and rapidly drill into each group set to see sub bunches and reasonably related groups. In the event that there is any likelihood to do sub group, at that point sub bunching is finished.

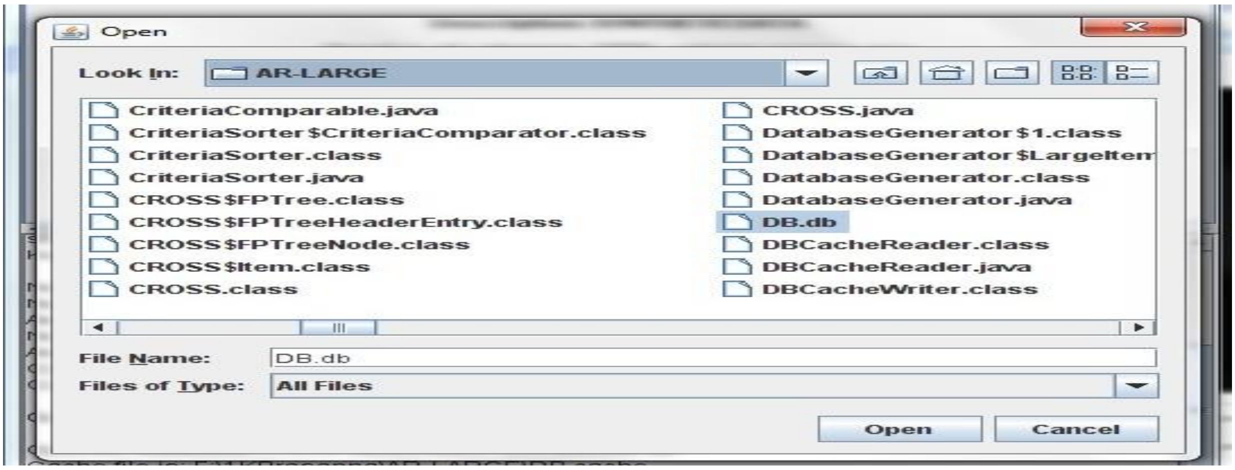
Building Knowledge Base: An information base in Data Quality Services (DQS) is a storehouse of information about your information that empowers you to comprehend your information and keep up its respectability. An information base comprises of areas, every one of which speaks to the information in an information field.

6. RESULTS



Database name and description
Name: DB.db
Description: SYNTHETICDATA
□ Auto naming
Database characteristics
Number of items: 1000
Number of transactions: 100
Average size of transactions: 100
Number of large itemsets: 50
Average size of large itemsets: 5
Correlation: 0.5
Corruption: 0.5
0%

Figure 1: Generating Synthetic Data Base



Open

Look In: AR-LARGE

```
CriteriaComparable.java                          CROSS.java
CriteriaSorter$CriteriaComparator.class          DatabaseGenerator$1.class
CriteriaSorter.class                             DatabaseGenerator$LargeItem
CriteriaSorter.java                              DatabaseGenerator.class
CROSS$FPTree.class                               DatabaseGenerator.java
CROSS$FPTreeHeaderEntry.class                    DB.db
CROSS$FPTreeNode.class                           DBCacheReader.class
CROSS$Item.class                                 DBCacheReader.java
CROSS.class                                      DBCacheWriter.class
```

File Name: DB.db

Files of Type: All Files

Figure 2: Opening a data file

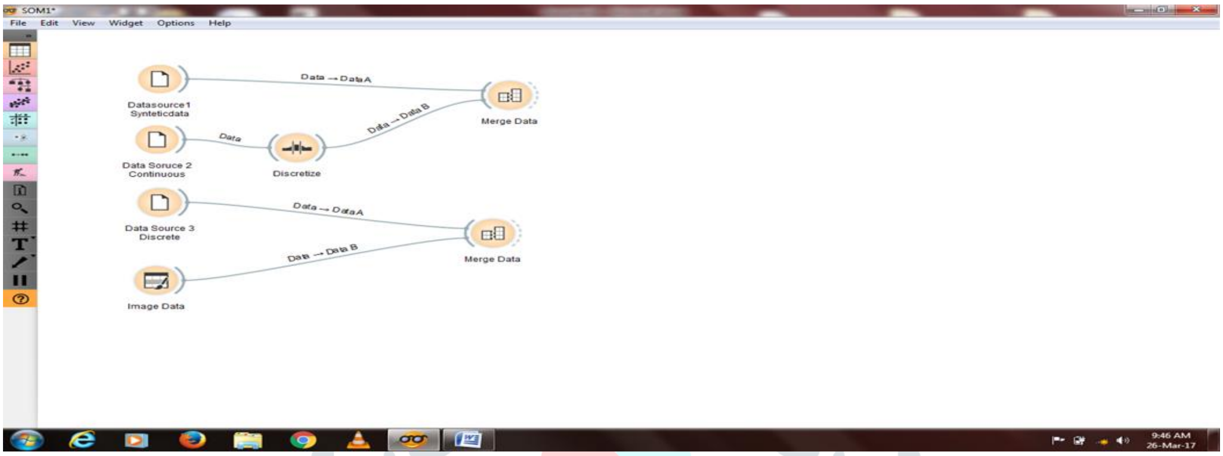


Figure 3: Showing different data sets are loaded in to the system for processing

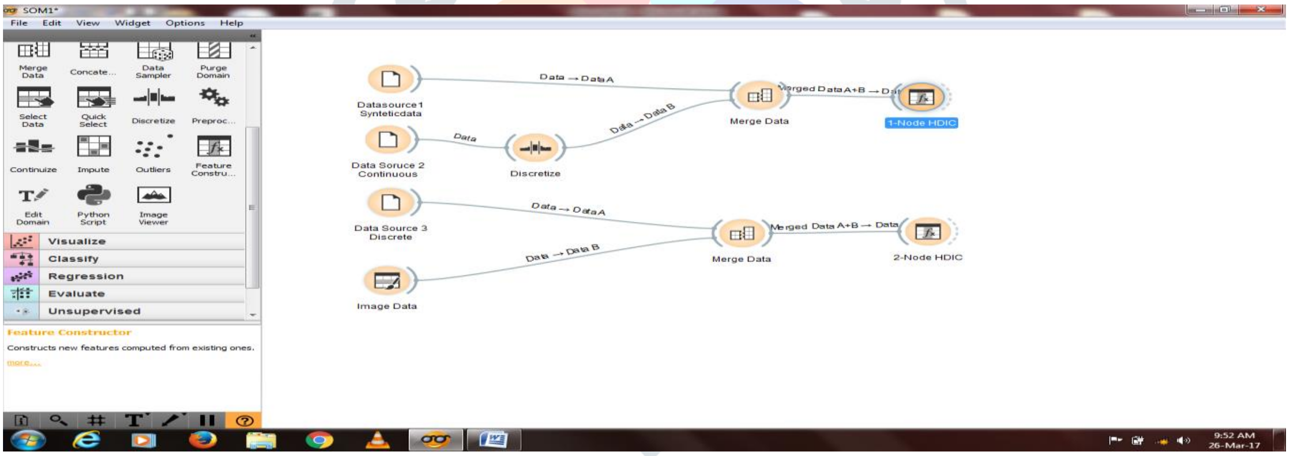


Figure 4: Two Data sets are formed into two DIC nodes

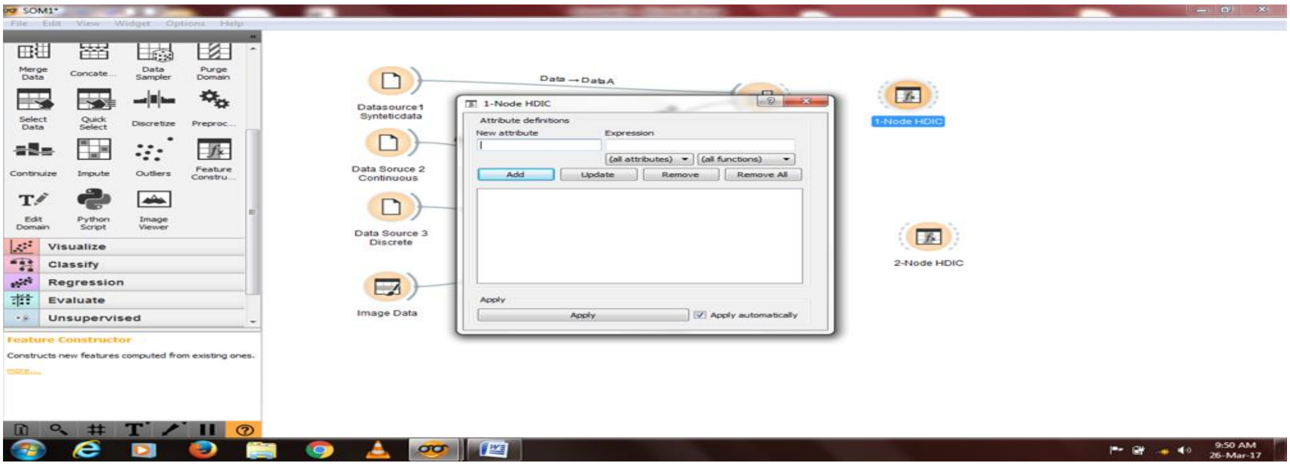
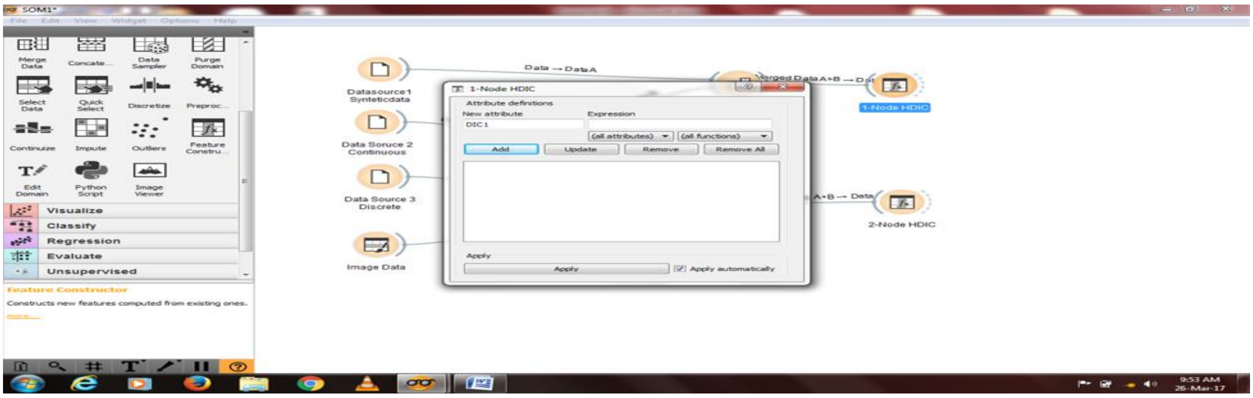


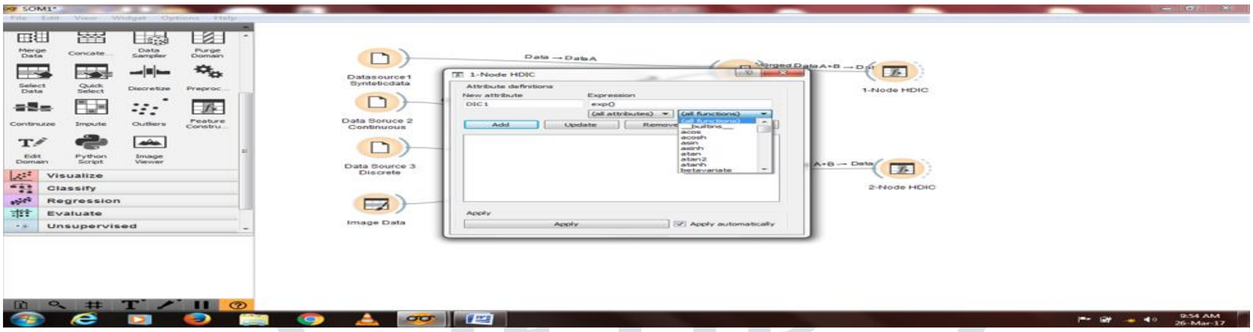
Figure 5: Defining new attribute for data intensive computing



New attribute: DIC1

Expression:

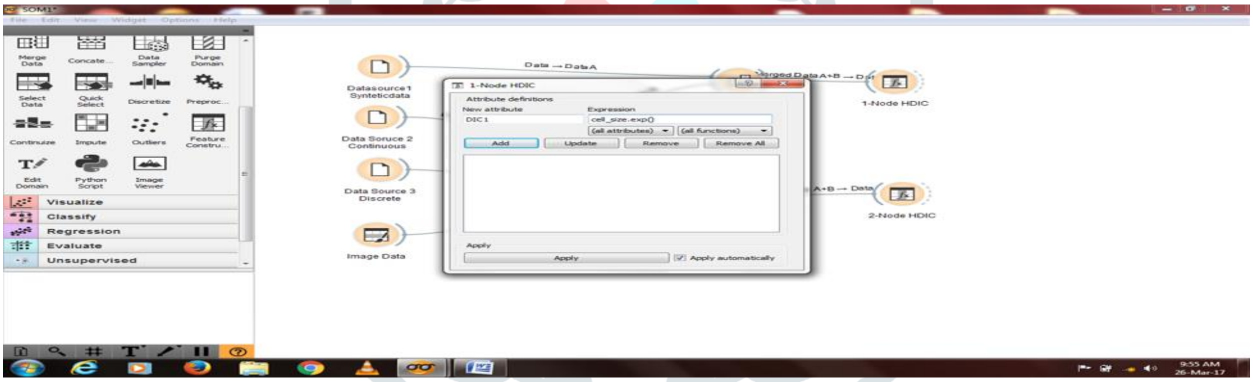
Figure 6: Giving name to new attribute



New attribute: DIC1

Expression: exp()

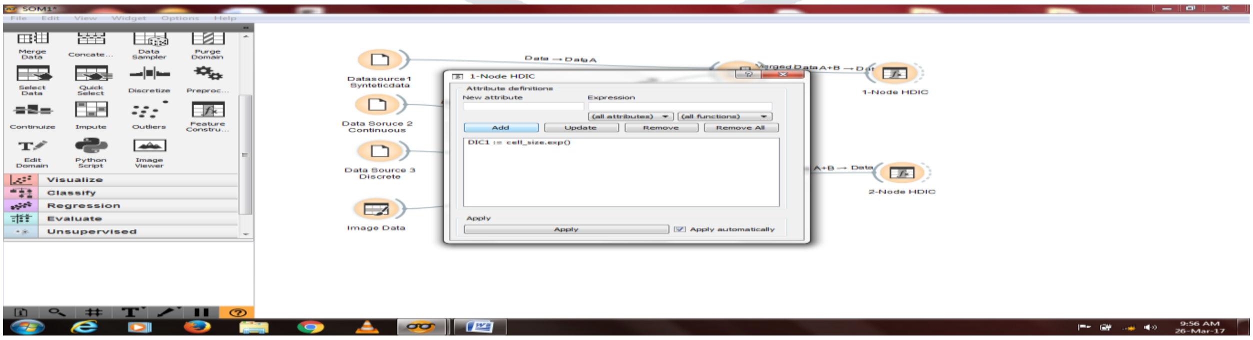
Figure 7: Selecting a function among list of functions



New attribute: DIC1

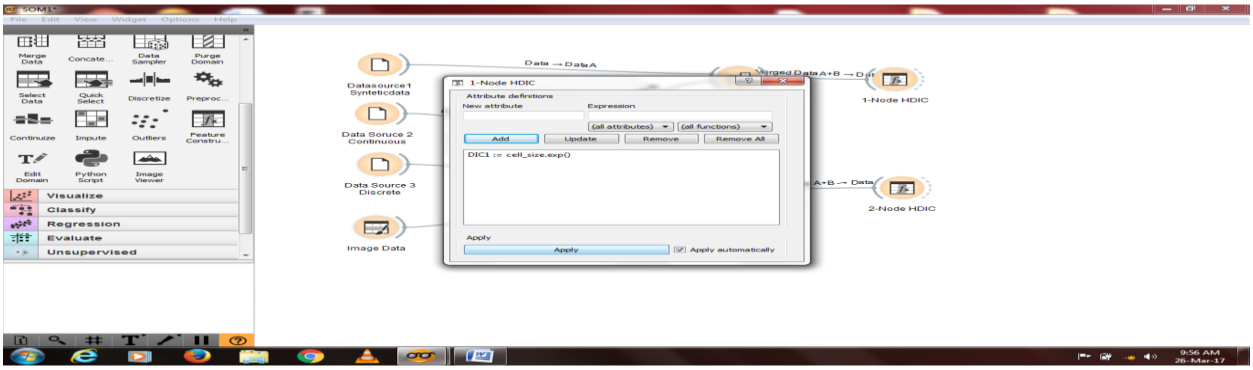
Expression: cell_size.exp()

Figure 8: Attribute is replaced with Expression



DIC1 := cell_size.exp()

Figure 9: Adding attribute to DIC node



1-Node HDIC

Attribute definitions

New attribute Expression

(all attributes) (all functions)

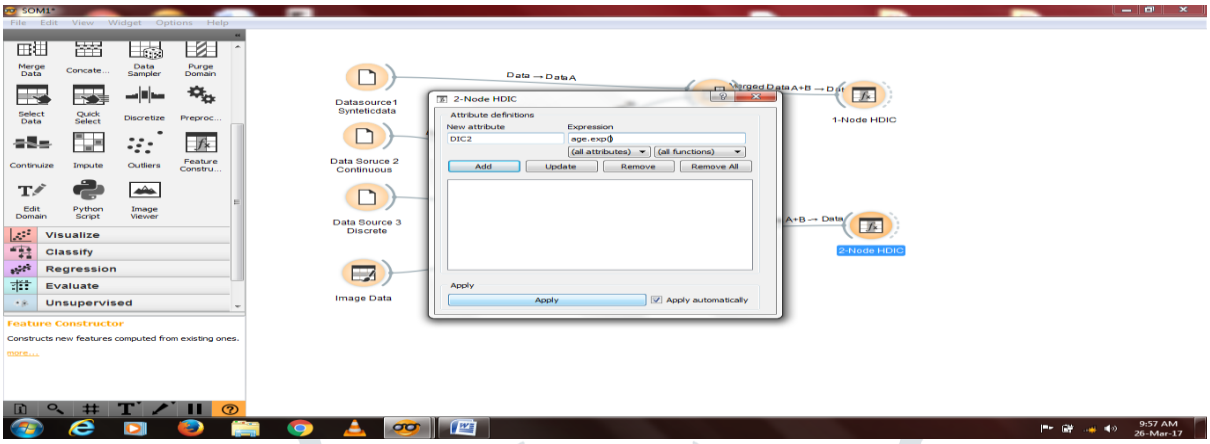
Add Update Remove Remove All

DIC1 := cell_size.exp()

Apply

Apply ☑ Apply automatically

Figure 10: Applying functions to DIC nodes



Feature Constructor

Constructs new features computed from existing ones.

more...

2-Node HDIC

Attribute definitions

New attribute: DIC2

Expression: age.exp()

(all attributes) (all functions)

Add Update Remove Remove All

Apply

Apply ☑ Apply automatically

Figure 11: Applying function to 2-Node HDIC

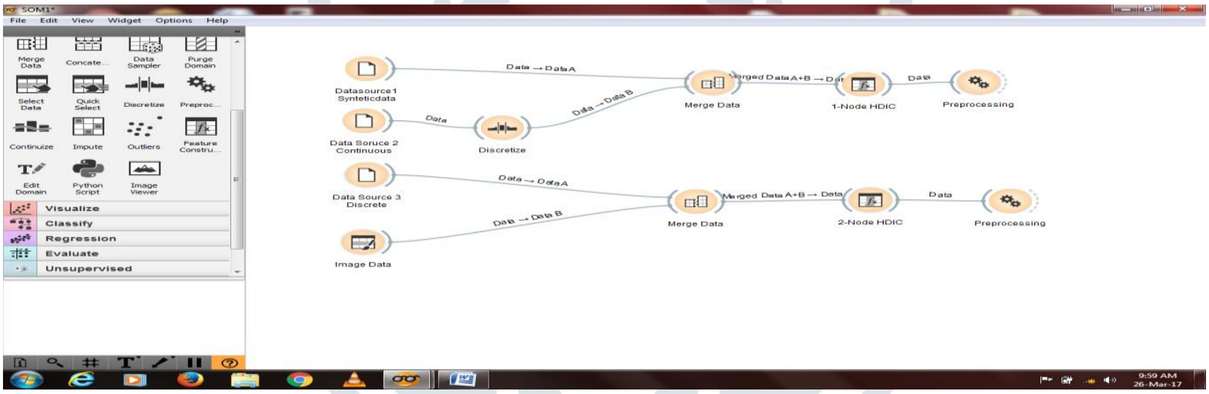
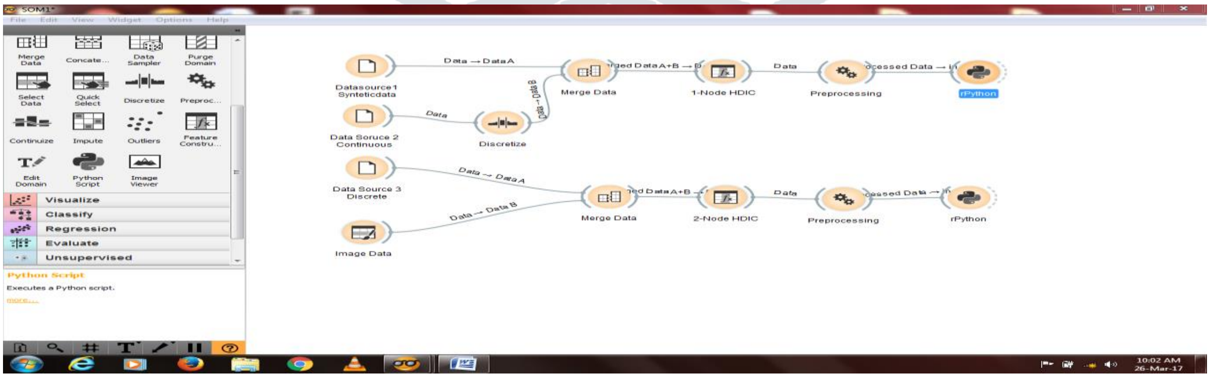


Figure 12: Pre-processing is applied on HDIC nodes

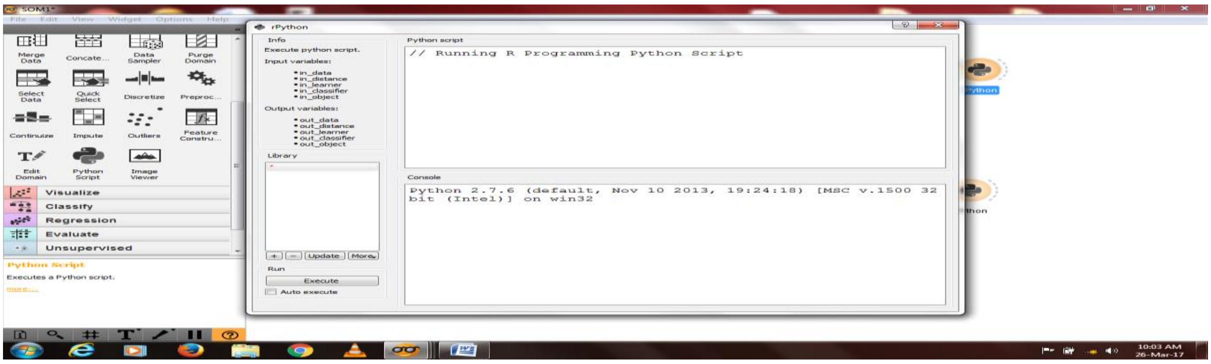


Python Script

Executes a Python script.

more...

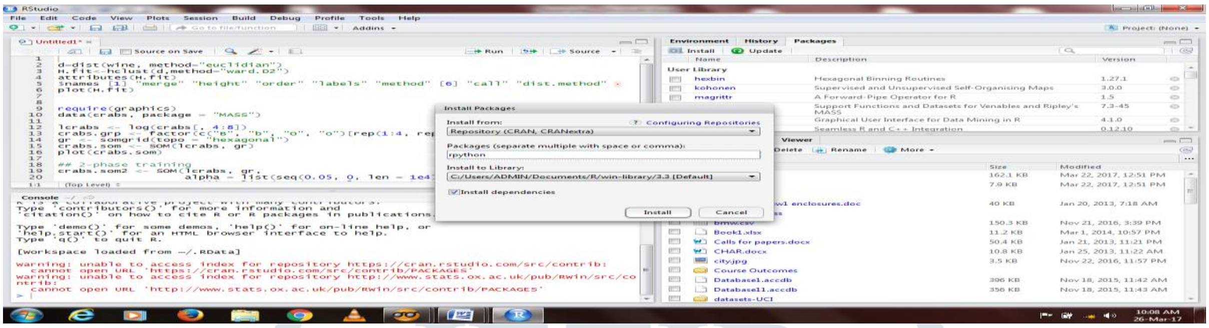
Figure 13: Adding Rpython to processed data



```
// Running R Programming Python Script
```

```
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32
bit (Intel)] on win32
```

Figure 14: Rpython window



```
d=dist(wine, method="euclidian")
H.fit<-hclust(d,method="ward.D2")
attributes(H.fit)
$names [1] "merge" "height" "order" "labels" "method" [6] "call" "dist.method"
plot(H.fit)


require(graphics)
data(crabs, package = "MASS")

lcrabs <- log(crabs[, 4:8])
crabs.grp <- factor(c("B", "b", "O", "o")[rep(1:4, rep
gr <- somgrid(topo = "hexagonal")
crabs.som <- SOM(lcrabs, gr)
plot(crabs.som)

## 2-phase training
crabs.som2 <- SOM(lcrabs, gr,
                  alpha = list(seq(0.05, 0, len = 1e4)
```

```
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[workspace loaded from ~/.RData]

Warning: unable to access index for repository https://cran.rstudio.com/src/contrib:
  cannot open URL 'https://cran.rstudio.com/src/contrib/PACKAGES'
Warning: unable to access index for repository http://www.stats.ox.ac.uk/pub/RWin/src/co
ntrib:
  cannot open URL 'http://www.stats.ox.ac.uk/pub/RWin/src/contrib/PACKAGES'
>
```

Install Packages

Install from: Repository (CRAN, CRANextra)

Packages (separate multiple with space or comma): rpython

Install to Library: C:/Users/ADMIN/Documents/R/win-library/3.3 [Default]

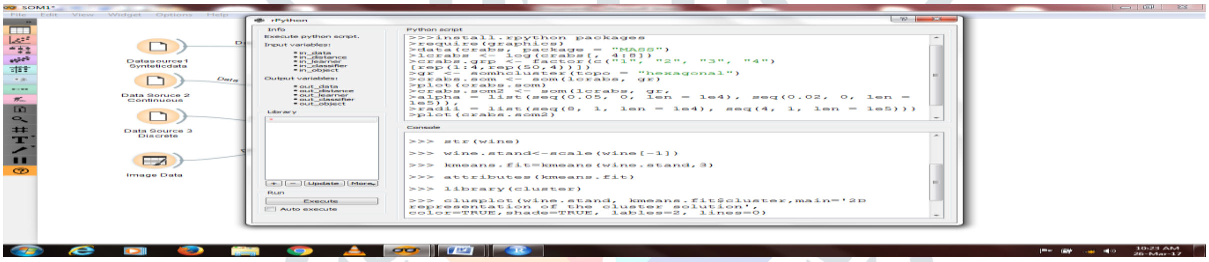
☑ Install dependencies

Install Cancel

Name	Description	Version
User Library		
hexbin	Hexagonal Binning Routines	1.27.1
kohonen	Supervised and Unsupervised Self-Organising Maps	3.0.0
magrittr	A Forward-Pipe Operator for R	1.5
	Support Functions and Datasets for Venables and Ripley's MASS	7.3-45
	Graphical User Interface for Data Mining in R	4.1.0
	Seamless R and C++ Integration	0.12.10

Name	Size	Modified
	162.1 KB	Mar 22, 2017, 12:51 PM
	7.9 KB	Mar 22, 2017, 12:51 PM
…w1 enclosures.doc	40 KB	Jan 20, 2013, 7:18 AM
bmw.csv	150.3 KB	Nov 21, 2016, 3:39 PM
Book1.xlsx	11.2 KB	Mar 1, 2014, 10:57 PM
Calls for papers.docx	50.4 KB	Jan 21, 2013, 11:21 PM
CHAR.docx	10.8 KB	Jan 25, 2013, 11:22 AM
city.jpg	3.5 KB	Nov 22, 2016, 11:57 PM
Course Outcomes		
Database1.accdb	396 KB	Nov 18, 2015, 11:42 AM
Database11.accdb	356 KB	Nov 18, 2015, 11:43 AM
datasets-UCI		

Figure 15: Rstudio



```
>>>install.rpython packages
>require(graphics)
>data(crabs, package = "MASS")
>lcrabs <- log(crabs[, 4:8])
>crabs.grp <- factor(c("1", "2", "3", "4")
[rep(1:4,rep(50,4))])
>gr <- somhcluster(topo = "hexagonal")
>crabs.som <- som(lcrabs, gr)
>plot(crabs.som)
>crabs.som2 <- som(lcrabs, gr,
>alpha = list(seq(0.05, 0, len = 1e4), seq(0.02, 0, len =
1e5)),
>radii = list(seq(8, 1, len = 1e4), seq(4, 1, len = 1e5)))
>plot(crabs.som2)
```

```
>>> str(wine)
>>> wine.stand<-scale(wine[-1])
>>> kmeans.fit=kmeans(wine.stand,3)
>>> attributes(kmeans.fit)
>>> library(cluster)
>>> clusplot(wine.stand, kmeans.fit$cluster,main='2D
representation of the cluster solution',
color=TRUE,shade=TRUE, lables=2, lines=0)
```

Figure 16: Executing R-programming code in orange

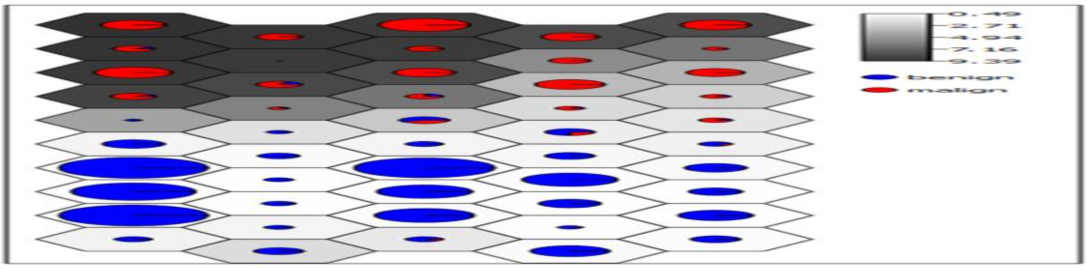


Figure 17: Hexagonal representation of cluster solution

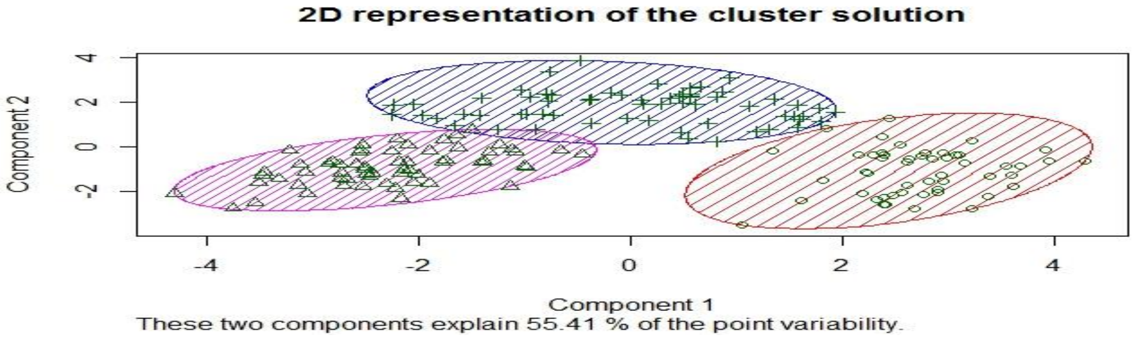


Figure 18: 2D representation of cluster

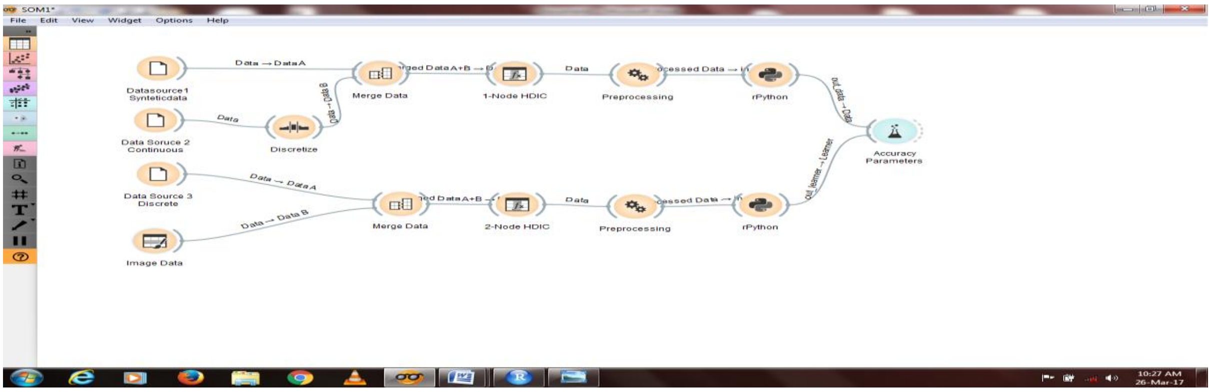
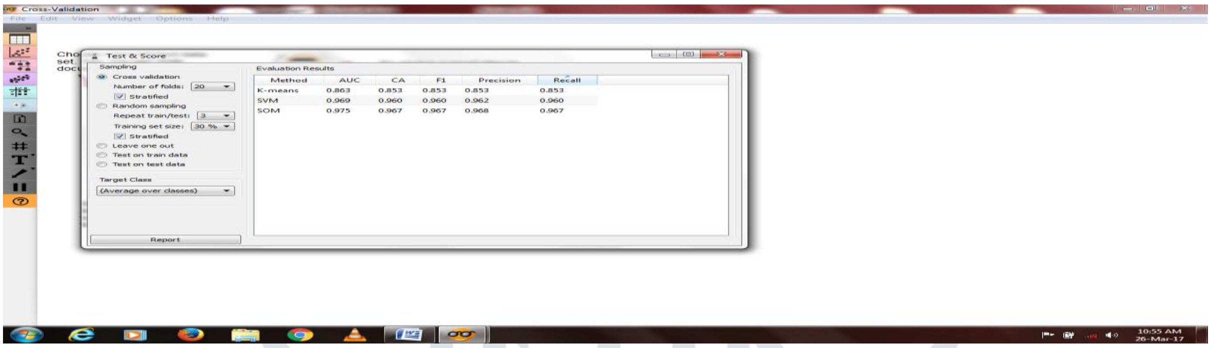


Figure 19: Cluster integration



Sampling

- ● Cross validation
  - Number of folds: 20
  - ☑ Stratified
- ○ Random sampling
  - Repeat train/test: 3
  - Training set size: 30 %
  - ☑ Stratified
- ○ Leave one out
- ○ Test on train data
- ○ Test on test data

Target Class: (Average over classes)

Evaluation Results

Method	AUC	CA	F1	Precision	Recall
K-means	0.863	0.853	0.853	0.853	0.853
SVM	0.969	0.960	0.960	0.962	0.960
SOM	0.975	0.967	0.967	0.968	0.967

Figure 20: Differences among clustering algorithms

7. CONCLUSION

A structure for the grouping of enormous information utilizing Heuristic Data Intensive figuring and Self-Organizing Maps calculation has been proposed. The Heuristic idea is to give ideal arrangement while SOM calculation is for the bunching of enormous information.

SOM calculation has numerous favorable circumstances to be utilized in huge information mining since it can scale with the size of the informational index, earlier information on the quantity of expected groups isn't required and simple to coordinate with bunches outfit model. Large information examination opens the entryway for some exploration regions and one of the most significant regions is the information security.

8. REFERENCES

[1] Agneeswaran, V. S. (2012). Big-data – theoretical, engineering and analytics perspective. In S.Srinivasa & V. Bhatnagar (Eds.), *Big Data Analytics SE – 2*Berlin, Germany: Springer-Verlag.

[2] Brzezniak, M., Meyer, N., Flouris, M., Lachaiz, R. & Bilas, A. (2008). Analysis of grid storage element architectures: high-end fiber-channel vs. emerging cluster-based networked storage. In M. Brzezniak, N. Meyer, M. Flouris, R. Lachaiz & A. Bilas (Eds.), *Grid middleware and services SE* , US: Springer.

[3] Chen, J., Chen, Y., Du, X., Li, C., Lu, J., Zhao, S. & Zhou, X. (2013). Big data challenge: a data management perspective. *Frontiers of Computer Science*.

[4] Das, S., Abraham, A. & Konar, A. (2009). Metaheuristic pattern clustering – an overview.*Metaheuristic Clustering*, Berlin, Germany: Springer-Verlag.

[5] Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. & Chretien, L. (1990). The dynamics of collective sorting robot like ants and ant like robots. *Proceedings of the 1stInternational Conference on Simulation of Adaptive Behaviour: From Animals to Animats*.

[6] Hall, L.O. (2013). Exploring big data with scalable soft clustering. In R. Kruse, M. R. Berthold, C.Moewes, M.Á. Gil, P. Grzegorzewski & O. Hryniewicz (Eds.), *Synergies of Soft Computing andStatistics for Intelligent Data Analysis*, Berlin, Germany: Springer-Verlag.

[7] Kim, B. (2012). A classifier for big data. In G. Lee, D. Howard, D. Ślęzak & Y. Hong (Eds.),*Convergence and Hybrid Information Technology*, Berlin: Germany: Springer-Verlag.

[8] Madheswari, A.N. & Banu, R.S.D.W. (2011). Communication aware co-scheduling for parallel jobscheduling in cluster computing. In A. Abraham, J. Lloret Mauri, J. Buford, J. Suzuki & S.Thampi (Eds.), *Advances in Computing and Communications*, Berlin, Germany:Springer.

[9] Qin, X. (2012). Making use of the big data: next generation of algorithm trading. In J. Lei, F. Wang, H.Deng & D. Miao (Eds.), *Artificial Intelligence and Computational Intelligence*, Berlin,Germany: Springer-Verlag.

[10] Strehl, A. & Ghosh, J. (2002). Cluster ensembles – a knowledge reuse framework for combiningmultiple partitions. *Journal of Machine Learning Research*.

[**11**] A. Fahad, N. Alshatri and Z. Tari, "A Survey of Clustering Algorithms for Big Data: Taxonomy", IEEE Transactions on Emerging Topics in Computing 2014.