Participating Members	Name	ID
	Ala'a Mostafa	2013/2856
	Dalia Ashry	2013/1223
	Merna Osama	2013/0711
	Nora Eleish	2013/8636
Field of Research	Big Data/Data Mining	
Project Title	Capacity Monitoring Tool	
Academic Advisor	Dr Abd Al Rahman & Eng. Radwa Samy	
Gained Sponsorships/ Participated Competitions		

1. Abstract

Capacity monitoring tool consists a Java desktop application. The Java desktop version keeps monitoring the network resources as the capacity of the nodes. Moreover, the tool aims to serve telecommunication operators capacity team in monitoring the capacity of their network nodes in order to prevent and fix this critical capacity issues that might come up. Operations will be performed on the data coming from the nodes such as data analysis, classification and clustering. Data analysis that include the following operations: average, sum, maximum, minimum and extracting other columns from the datasets using Microsoft SQL Server. Classification is carried out to detect where overhead occurs. Clustering is for knowing when the peak occurs. In addition to providing a visual representation of detected nodes; detected nodes with overhead will be marked on the geomap. The files contain rows of data. The first row contains columns names, the rest are the data that corresponds to each column.

.

Problem Statement : Telecom operators have a crucial problem of monitoring the massive capacity of their network nodes. The late response for such an issue might result in complete or partial shutdown of a major server or an active node.

2. Project Objectives

Telecom operators have a crucial problem of monitoring the massive capacity of their

network nodes. The late response for such an issue might result in complete or partial shutdown of a major server or an active node.

3. Progress

Telecom Italia dataset is for measuring the interaction level between the users and the mobile phone network. The datasets provide data about the telecommunication activity in Milano city in Italy in december 2013.This dataset has 90 million records for 10,000 square id.This dataset consists of 14 columns which are square id of milano city that begins from 1 to 10,000 square id, 5 activities which are sms-in activity, sms-out activity, call-in activity, call-out activity and internet traffic activity, country code, time interval in milliseconds. Moreover, determining the start time of each activity by time interval and converting the start time from milliseconds to seconds and also converting it to date(Month/Day/Year), hours and minutes.Also, Determining the end time of each activity by time interval plus 60,0000 milliseconds and converting the end time from milliseconds to seconds and also converting it to date(Month/Day/Year), hours and minutes.Also, specifying holidays including weekends in italy and the public european holidays in december 2013.Also, specifying working hours portion in italy.The milano city is divided into 10,000 square id GRID.

A new table is designed having just 75 square id of milano city GRID, the sum of each activity(sms-in activity, sms-out activity, call-in activity, call-out activity and internet traffic activity), total summation of all activities, and activity level which determine low, moderate and high activity level for each 75 square id.As the 75 square id is divided into three sections: 25 square id for low activity level, 25 square id for moderate activity level and 25 square id for high activity level. This table is arranged ascendingly according to total sum of all activities. The objective of this table is to determine the activity level of each square id . Figure 1 shows the 75 square ids that are used in the new table.

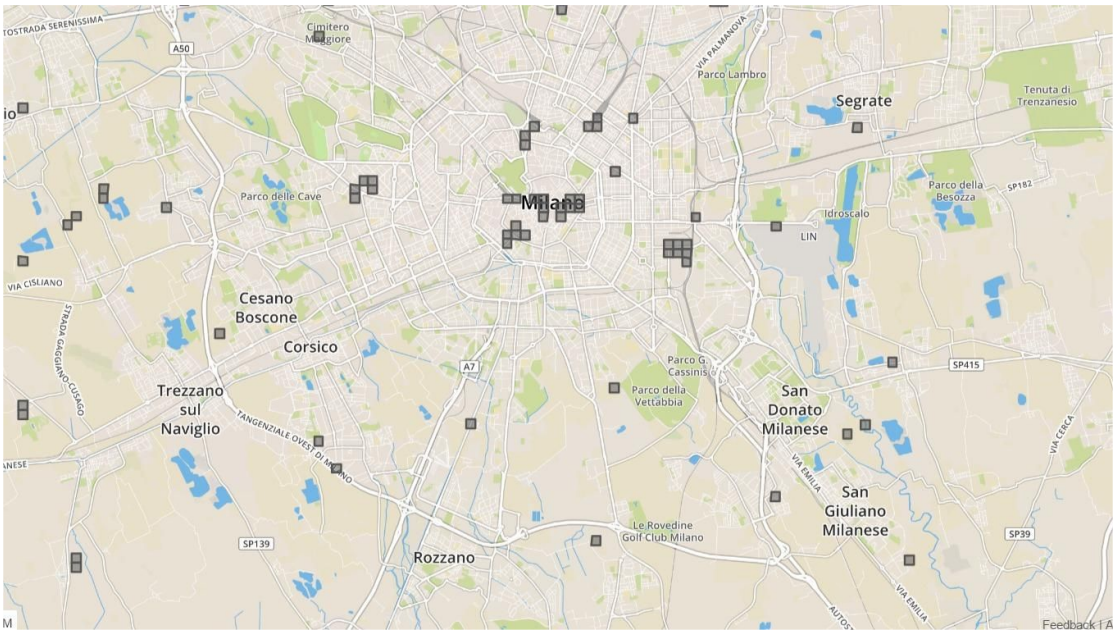

Figure [1]: Milano Grid with Selected 75 Square-IDs(25 with lowest activity, 25 with moderate activity,& 25with highest activity)

Operations done on Milano dataset as classification and clustering. The Milano dataset is divided into two parts training and testing, 90,000 record for training divided into three section 30,000 records for low activity level, 30,000 records for moderate activity level, 30,000 records for high activity level and 30,000 other records for testing divided into 10,000 for low activity level, 10,000 for moderate activity level, 10,000 for high activity level. The training and testing records are used for classification in support vector machine (SVM) and neural network (NN).

4. Classification

A new dataset has been created for classification having a new 120 thousand records and dividing them into 90 thousand records for training and 30 thousand records for testing.The 90 and 30 thousand records consists of 6 columns: smsInActivity, smsOutActivity,callInActivity, callOutActivity, InternetTracActivity and ActivityLevel. ActivityLevel column is a label, it consists of three classes: high, moderate and low as high equals to 3, moderate equals to 2 and low equals to 1. So we classify each activity acccording to the activityLevel label. And determining the square id having the highest activities according to label as the overhead node. The new 120 thousand records dataset is used in support vector machine (SVM)

smsInActivity	smsOutActivity	callInActivity	callOutActivity	internetTrafficActivity	activityLevel
0	0	0	1.026162125	0	1
0	0	0	2.05232425	0	1
0	0	0	1.026162125	0	1
0	0	0	3.078486375	0	1
0	0	0	5.130810625	0	1
0	0	0	1.026162125	0	1
0	0	0	2.05232425	0	1
0	0	0	1.026162125	0	1
0	0	0	2.05232425	0	1
0	0	0	5.130810625	0	1
0	0	0	6.15697275	0	1
0	0	0	3.078486375	0	1
0	0	0	4.1046485	0	1
0	0	0	3.078486375	0	1
0	0	0	5.130810625	0	1
0	0	0	8.209296999	0	1
0	0	0	1.026162125	0	1
0	0	0	5.130810625	0	1
0	0	0	8.209296999	0	1
0	0	0	4.1046485	0	1

Figure [2]: Columns of 120k records

4.1 Support Vector Machine

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification. Given binary classes or multi-classes, SVM trains a model forming an optimal hyperplane that separates the classes and then categorizes inputs into one of the classes through the test data.

LIBSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports multi-class classification. C and nu are parameters which help implement a penalty on the misclassifications that are performed while separating the classes. Thus helps in improving the accuracy of the output.

C ranges from 0 to infinity and can be a bit hard to estimate and use. A modification to this was the introduction of nu which operates between 0-1 and represents the lower and upper bound on the number of examples that are support vectors and that lie on the wrong side of the hyperplane.

Kernel Functions Equations According to LIBSVM Documentation:

1) Linear: u'*v
2) Polynomial: (gamma*u'*v + coef0)^degree
3) Radial basis function: exp(-gamma*|u-v|^2)

4) sigmoid: tanh(gamma*u'*v + coef0)

-d degree : set degree in kernel function (default 3)

-g gamma : set gamma in kernel function (default 1/num_features)

-r coef0 : set coef0 in kernel function (default 0)

-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1).

-n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)

-p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)

-m cachesize : set cache memory size in MB (default 100)

-e epsilon : set tolerance of termination criterion (default 0.001)

Figure 3 shows the dataset before classification.

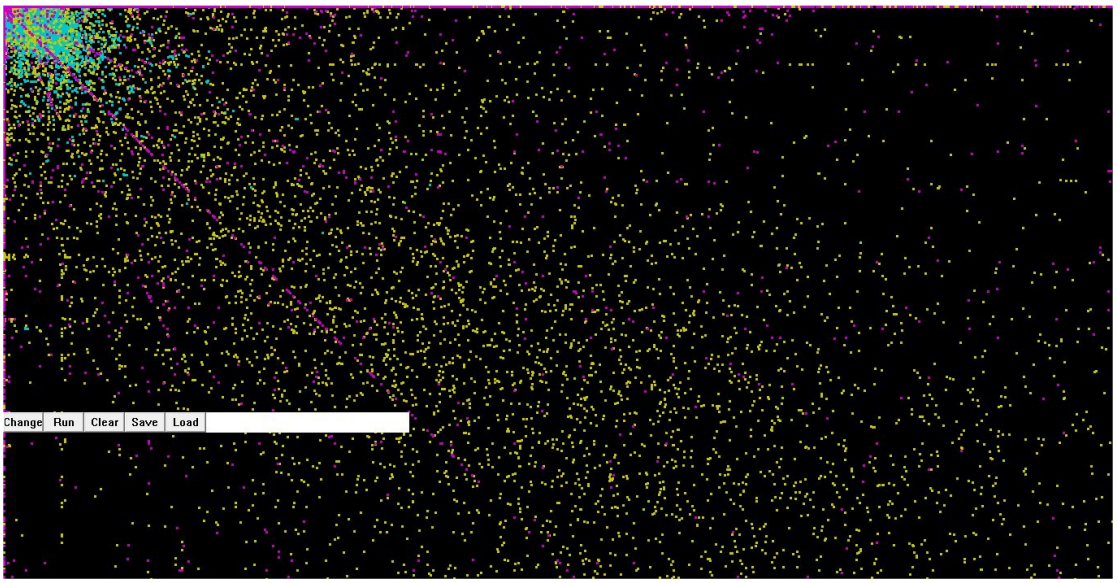

Figure [3]: Training data before classification.

Figure 4 shows dataset after classification(C-SVC type, and Linear Kernel function)

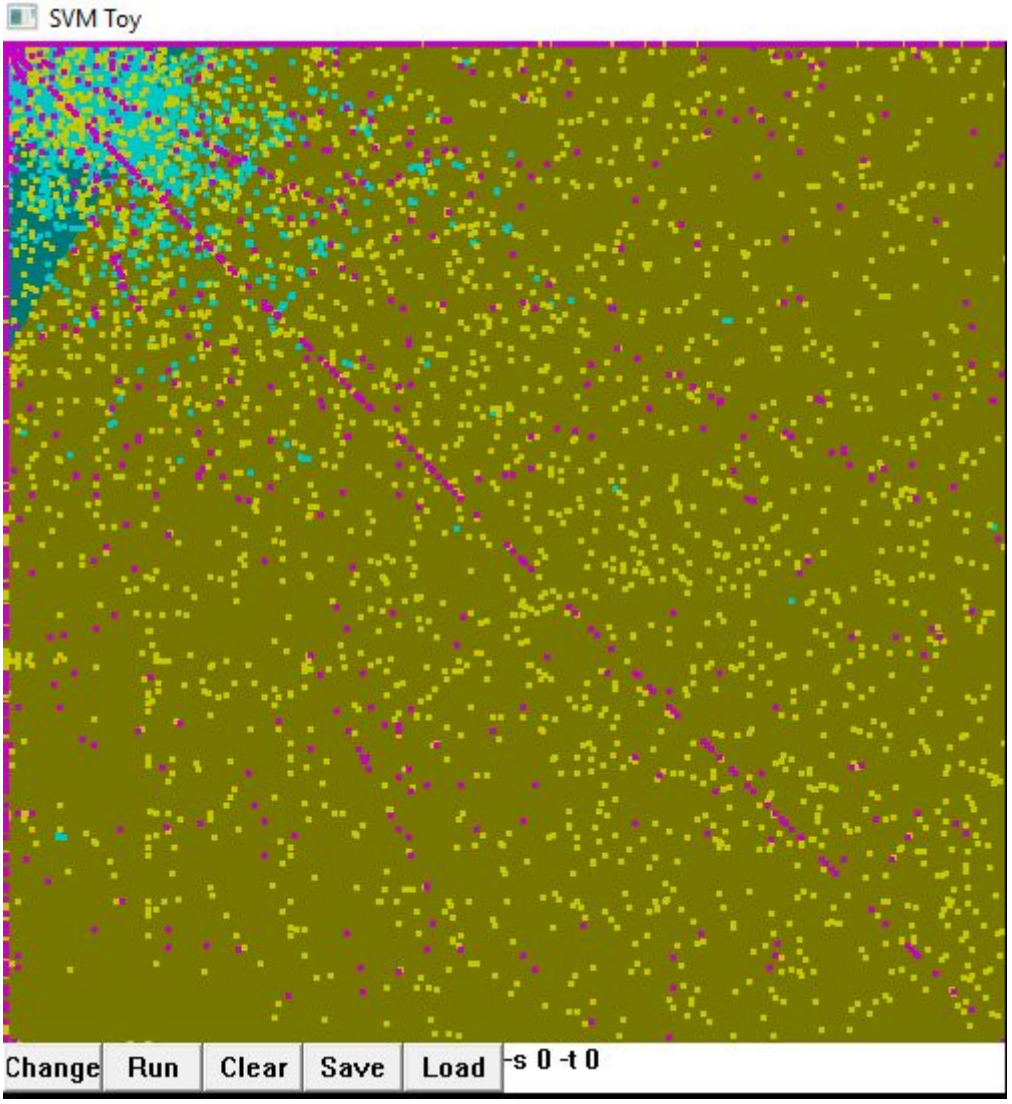

Figure [4]: Training data after classification with C-SVC type, and Linear Kernel function

Figure 5 shows the dataset after classification(C-SVC, and polynomial kernel function with default parameter values)

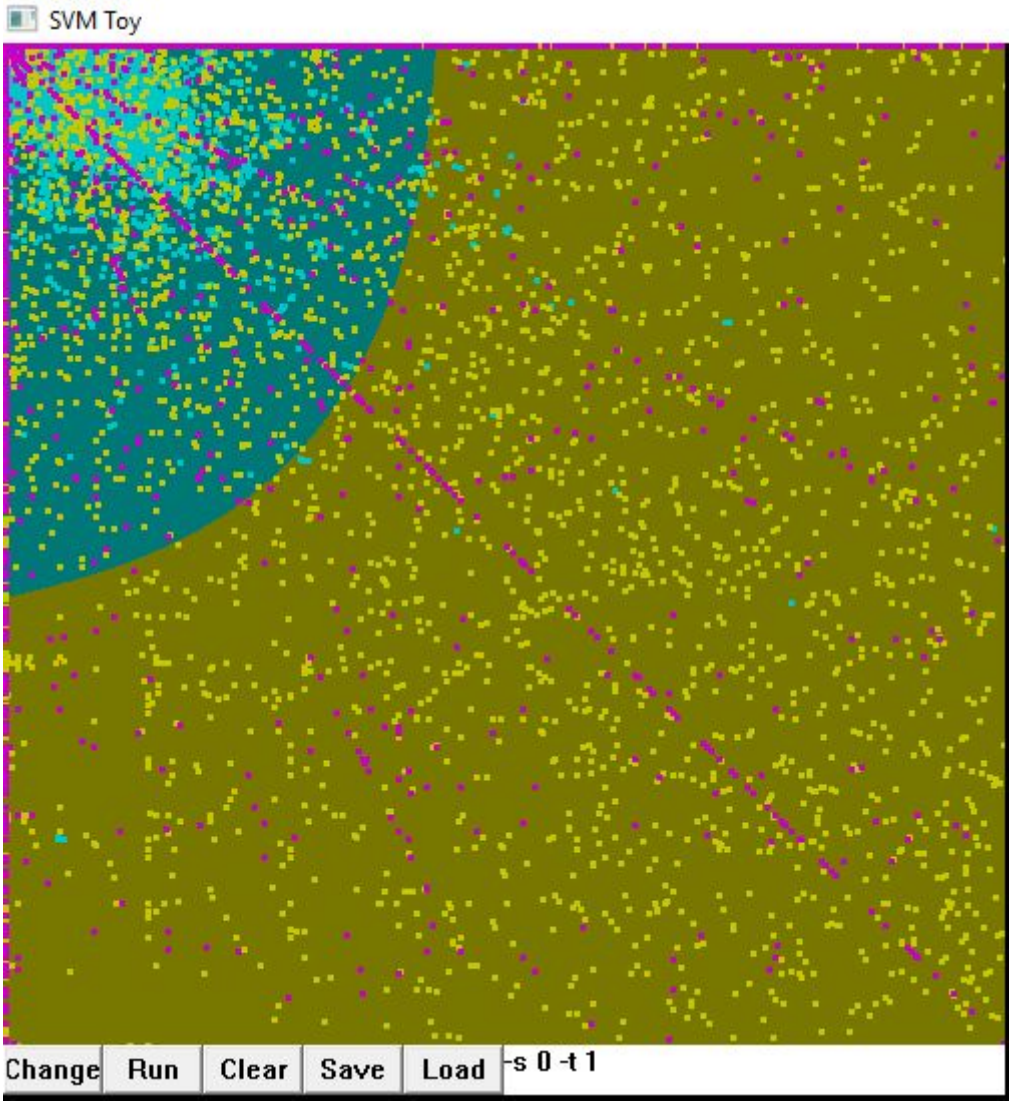

Figure [5]: Training data after classification with C-SVC type, and polynomial kernel function with default parameter values

Figure 6 shows the dataset after classification (C-SVC type, and Radial Basis Kernel function with default gamma value).

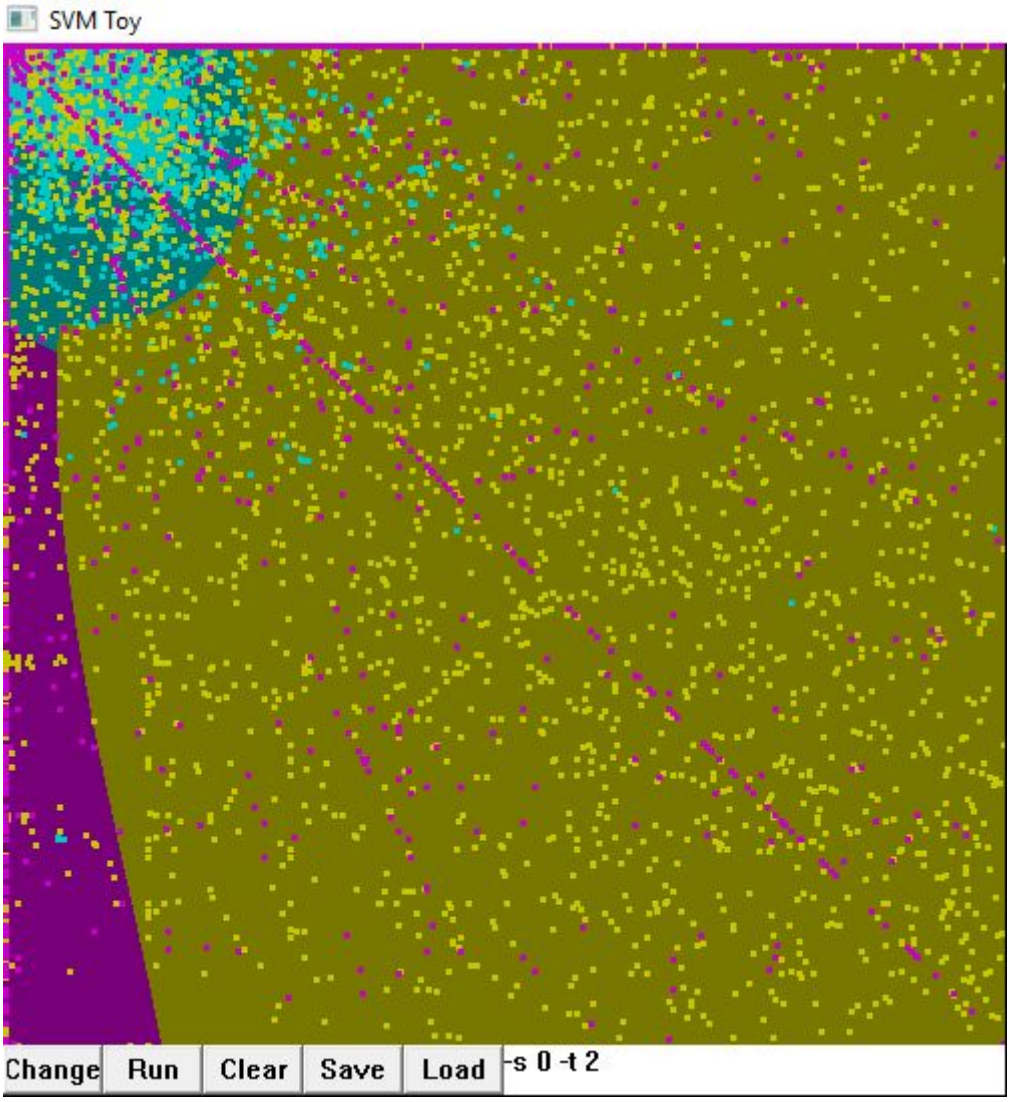

Figure [6]: Training data after classification with C-SVC type, and Radial Basis Kernel function with default gamma value.

Figure 7 shows the dataset after classification and after manipulating gamma parameter value(C-SVC type, and Radial Basis Kernel function with 1.8 gamma value).

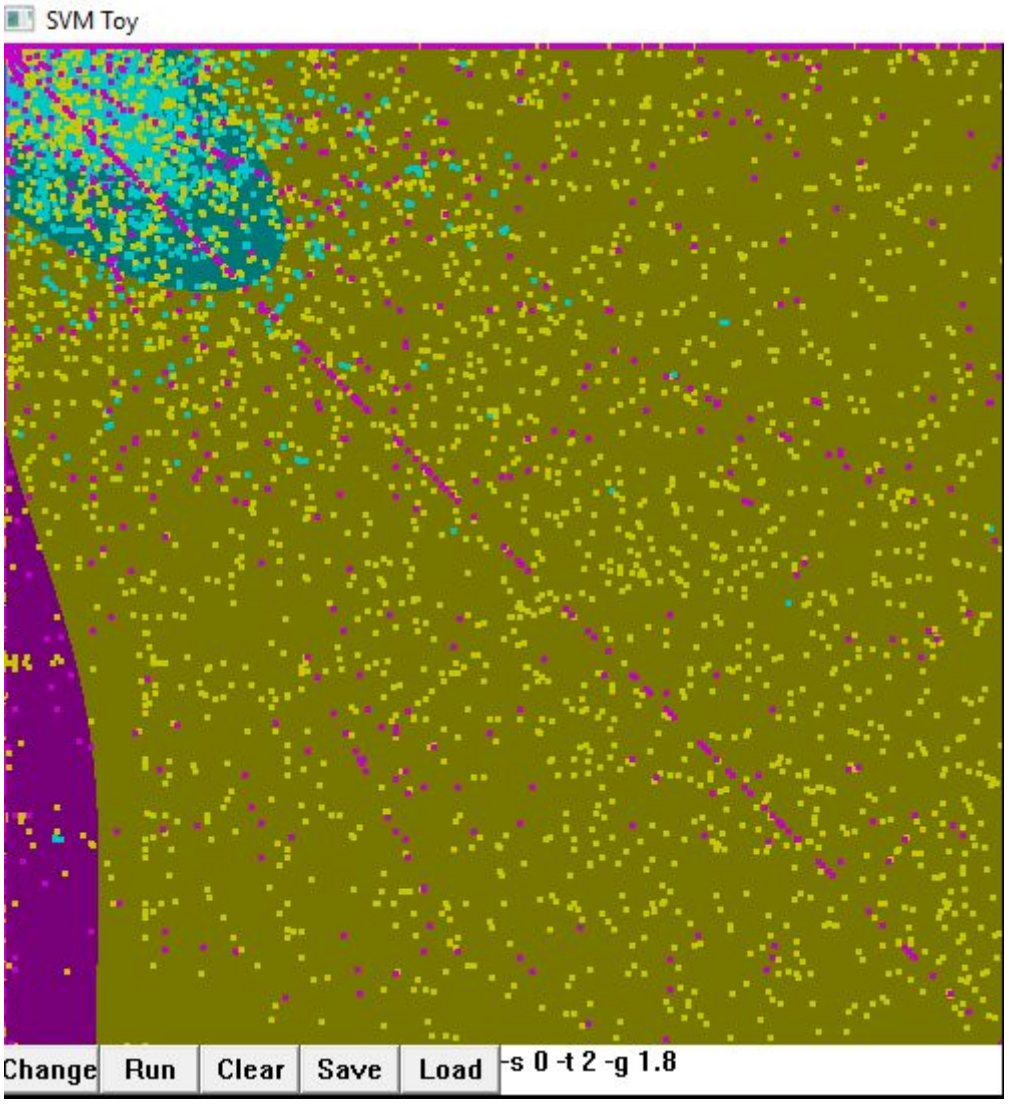

Figure [7]: Training data after classification with C-SVC type, and Radial Basis Kernel function with 1.8 gamma value

Figure 8 shows the dataset after classification and after manipulating gamma parameter value(C-SVC type, and sigmoid kernel function with default parameter values).

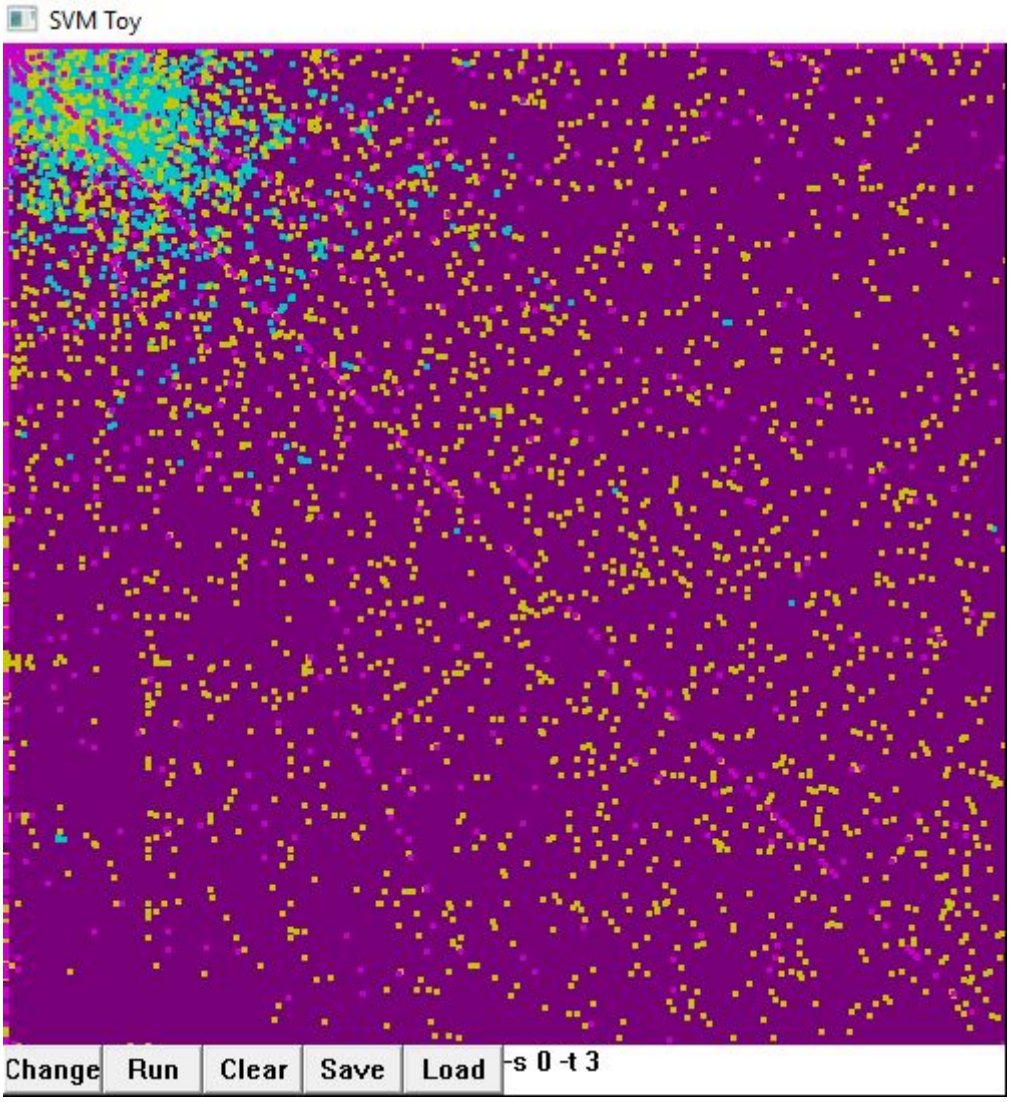

Figure [8]: Training data after classification with C-SVC type, and sigmoid kernel function with default parameter values

Figure 9 shows the dataset after classification(nu-SVC type, and Linear kernel function).

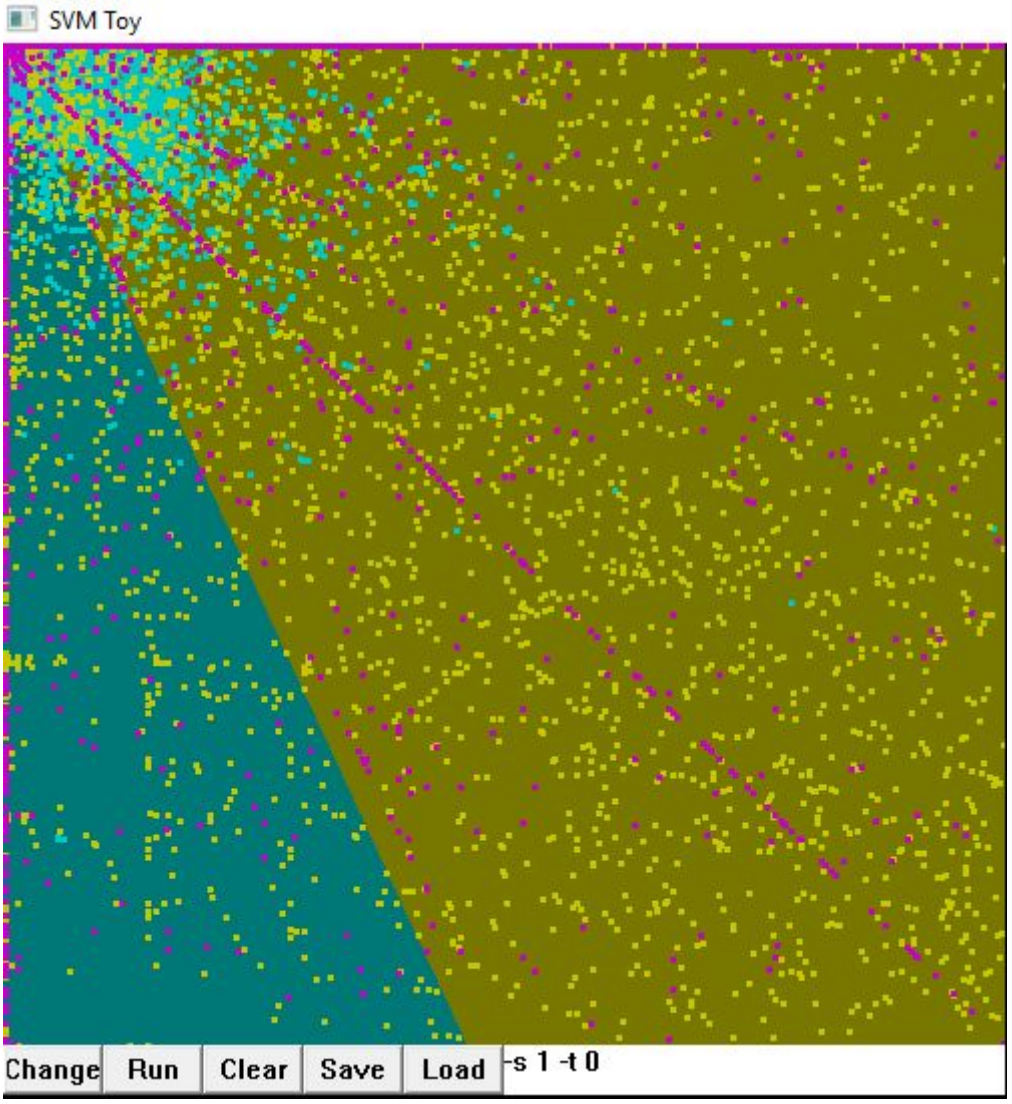

Figure [9]: Training data after classification(nu-SVC type, and Linear kernel function)

Figure 10 shows the dataset after classification(nu-SVC type, and polynomial kernel function with default parameter values).

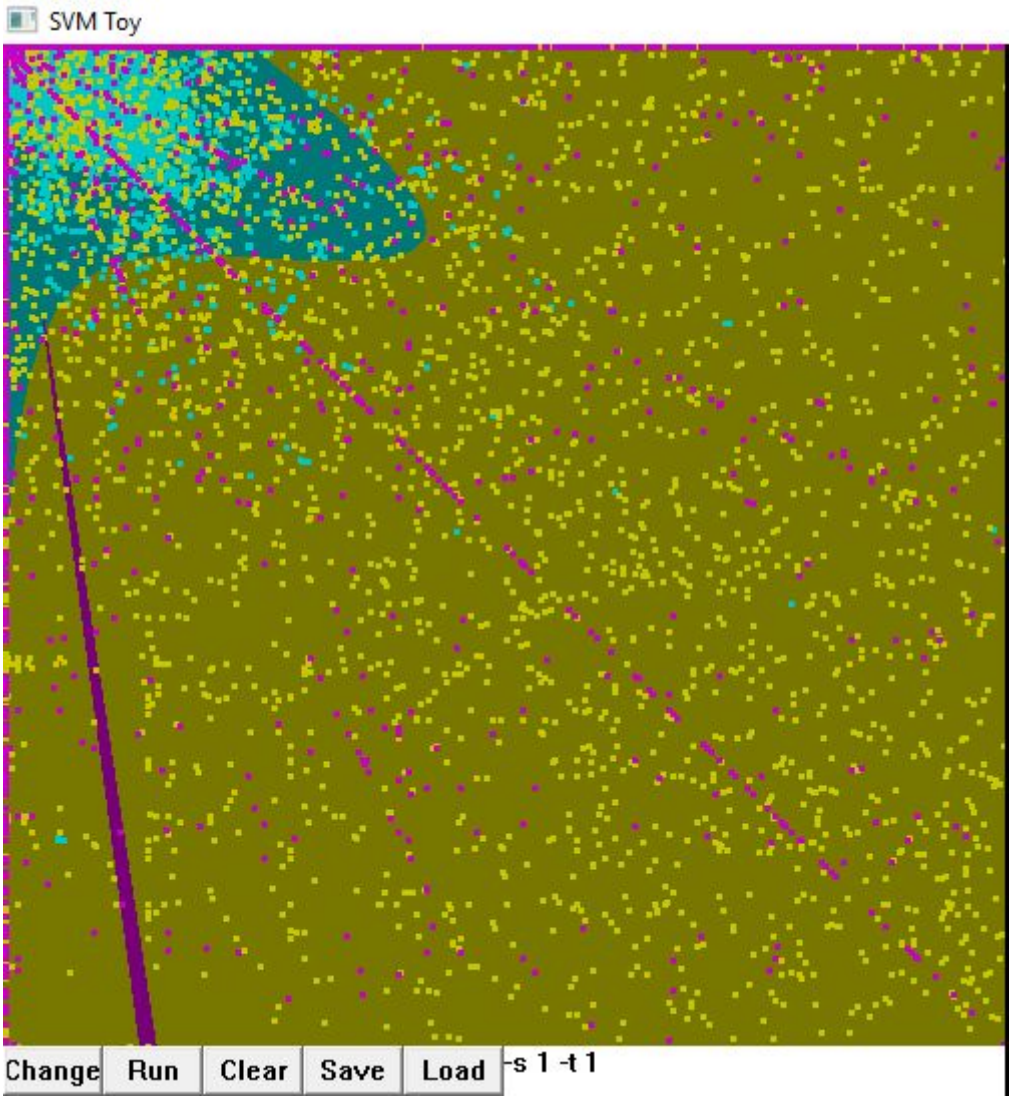

Figure [10]: Training data after classification(nu-SVC type, and polynomial kernel function with default parameter values)

Figure 11 shows the dataset after classification(nu-SVC type, and radial basis kernel function with default parameter values).

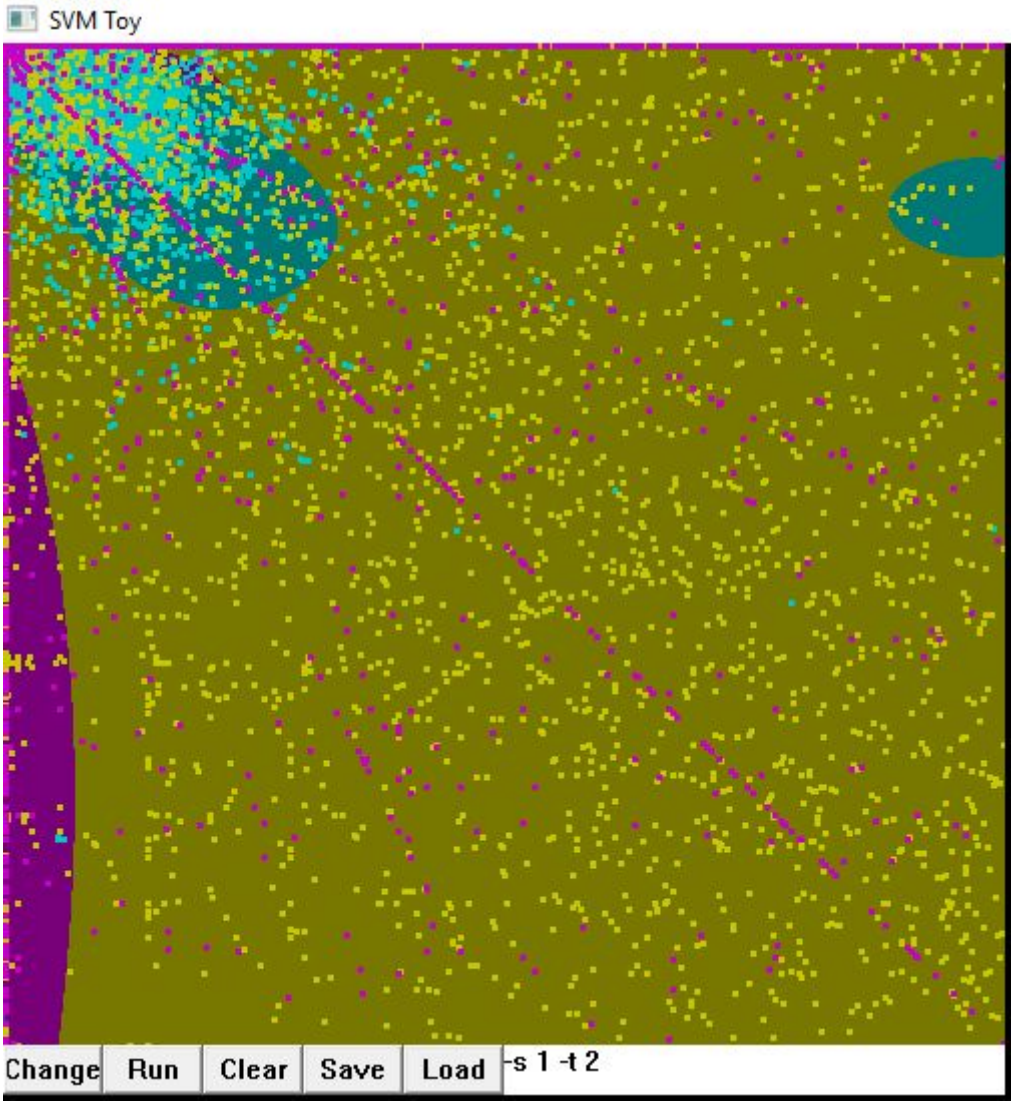

Figure [11]: Training data after classification(nu-SVC type, and radial basis kernel function with default parameter value)

Figure 12 shows the dataset after classification(nu-SVC type, and sigmoid kernel function with default parameter values).

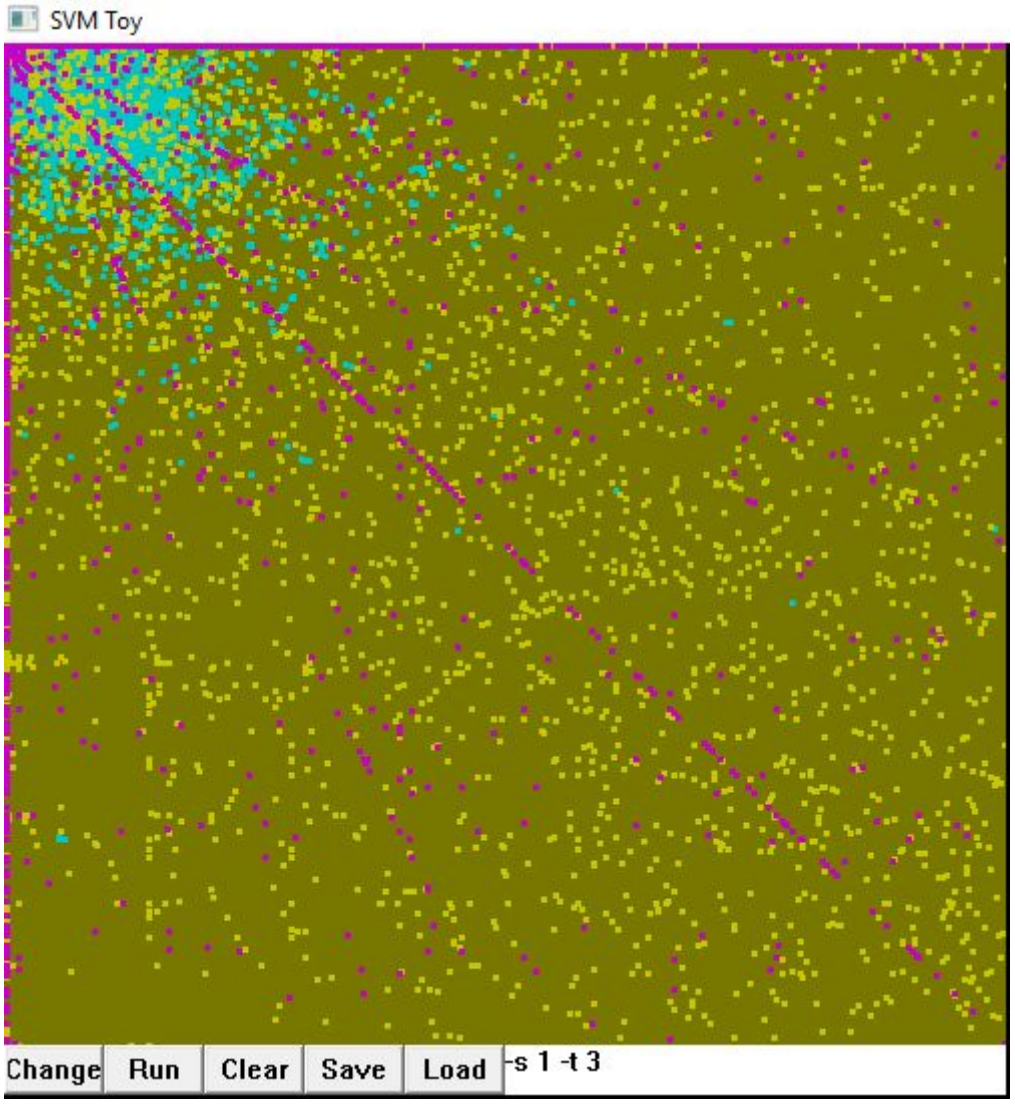

Figure [12]: Training data after classification(nu-SVC type, and sigmoid kernel function with default parameter value)

Dataset is composed of three classes, class(1) contains lowest activities(smsInActivity, smsOut,Activity, callInActivity, callOutActivity, and internetTrafficActivity) for 25 Square IDs, class(2) contains moderate activities for another 25 Square IDs, and class(3) contains highest activities for 25 Square IDs. Hence multi-class classification types are used (C-SVC and nu-SVC)with different kernel functions (Linear, polynomial, radial basis, and sigmoid). For the training dataset, 90 thousand records are used, and for testing 30 thousand records. Figure 12 shows the results for all of the SVM types used and kernel functions used by LIBSVM with their default parameter values and their accuracies for classification.

Kernel-Types	Linear	Polynomial	Radial Basis	Sigmoid
SVM-Types				
C-SVC (multi-class classification)	48.3035%	29.091%	65.8629%	35.957%
nu-SVC (multi-class classification)	46.1328%	40.7294%	65.3186%	43.2741%

According to the accuracy results, C-SVC type and Radial Basis kernel function achieved the highest accuracy. After further enhancement and manipulation for gamma parameter of Radial Basis kernel function, the following accuracies are achieved.

C-SVC, Kernel-Types	Radial Basis Kernel Function
gamma= 0.2(default)	65.8269%
gamma= 0.3	66.3372%
gamma= 0.6	69.0255%
gamma= 0.98	70.4114%
gamma= 0.99	70.4415%
gamma= 1.3	70.5884%
gamma= 1.8	71.1027%

The following figure shows the dataset before classification.

Figure [13]: Tables of Accuracies for SVM types & Kernel functions, with enhancements.

But we chose the median value according to accuracy percentage that corresponds to gamma value = 0.98, in order to avoid overfitting and underfitting of the model.

5. Clustering

A new dataset is created for clustering algorithms using 24 records which includes two columns: totalsum and timeinhours, as totalsum is total sum of all activities (smsInActivity, smsOutActivity, callInActivity, callOutActivity, internetTrafficActivity), this records are generated by getting the total sum of all activities with grouping by time in hours. The new 24 records dataset is used in K-medoids.

totalSumOfActivites	timeInHours
1159612	0
962107	1
860233	2
790473	3
773367	4
859879	5
1235026	6
2131567	7
3166303	8
3935817	9
4519424	10
4894686	11
5055317	12
4967417	13
4966804	14
5018127	15
4878320	16
4288993	17
3455439	18
2676909	19

Figure [14]: Dataset For Clustering.

5.1 K-Medoid

k-medoid clustering are used to know when the peak times happens through activities. For K-Medoids in producing the optimal clusters.

Row ID	totalSu...	tim...	Distance	partitio...
Row5	859879	5	5 [299738.0, 102232.0, 357.0, ...	5
Row6	1235026	6	6 [75420.0, 272924.0, 374797....	3
Row7	2131567	7	7 [971962.0, 1169466.0, 12713...	5
Row14	4966804	14	14 [3807206.0, 4004710.0, 410...	8
Row18	3455439	18	18 [2295845.0, 2493349.0, 259...	3

Figure [15]: Interactive Table results from K-medoids in new dataset

6. Related Work

System	Functionalities
Nokia	a) Problem: With mountains of data being generated every day, you need a solution to monitor your networks reliably, accurately and cost-efficiently. b) Tool: The new Capacity Advisor feature helps you predict the network capacity needs, and it recommends what actions to take and when. Thanks to this you can ensure that there is enough capacity in the network when you roll out your new service. c) They used the dashboards in KPI prediction and time slot classification. For showing as many dimensions as possible and showing relations between data. They used them in Predictive operations as the following: - Service Key Performance Indicator (KPI) monitor system health. - SLA agreement guarantees 99.x percent availability and steady values for normal operation.
NSN	a) Monitor network alarm and create a Trouble Ticket if needed after initial checks are done. b) Log the exact time of the alarm, as it appeared in OSS.

	c) Sent SMS to management team if the case is considered as critical. d) Notify the customer through emails for outage issues as per management directions. e) They used the dashboards in the User Mobility. -For showing Usage of traffic planning, ads planning and traffic jam prediction. -They also used them in cell classification and predicts preferred user movement.
The Proposed Tool	a) It will collect the files with different formats from the network nodes. b) Then, parsing them and save them in the database. c) There are some operations that can be made on the parsed files as the following: - classification, clustering, and data analysis. - A website for displaying the dashboards, the results of the operations that is done before, and the whole capacity monitoring process. - A mobile application will be available with a custom dashboard for viewing the data summary and receiving alarms.

7. Conclusion

Telecommunication is the most important service for human being to connect people with each other. Telecom operators are trying to enhance their services and promotions for customers using the huge data they get everyday from numbers of

calls, short messages service (SMS) and internet trac using data mining. Our proposed project detects and consults the capacity team for the over-head nodes that might result in partial or complete shutdown of a major server or an active node. We developed a tool consists a java desktop version. As the desktop application, its job is to detect the overhead nodes using classification algorithms as support vector machine (SVM), and to detect the peak time using clustering algorithms as K-Medoid. For classification, we achieved the accuracy 70.44% in support vector machine (SVM) algorithm and for clustering we reached the optimum clusters.