Supervised Classification for Crystallographic Information Files

Adnan Ibne Khair
University of New Mexico
1 University Blvd
Albuquerque, NM 87131
adnanibnekhair@unm.edu

Towfiq Ahmed
Los Alamos National Laboratory
P.O. Box 1663
Los Alamos, NM 87545
atowfiq@lanl.gov

1. INTRODUCTION

Crystal structures play an important role in understanding physical properties and information of the materials. By evaluating crystal structures, researchers can extract the expert information which helps to contribute to their research substantially. About 100 years of this discovery of crystal structure using X-ray diffraction, researchers who learned about these structures archived their research result in scientific articles. Large amount of material crystal structures have been discovered and these structures are being archived in crystallographic databases like Inorganic Crystal Structure Database (ICSD), Crystallography Open Database (COD) etc. These crystal structures are archived as a Crystallographic Information File(CIF) in these databases. Researchers around the world use these databases to learn about material. The Crystallographic Information File contains the crystal information of the materials. Not all the CIF file archived in the databases are complete. Some of the files lack a good amount of information which describes the crystal structures. The reason these CIF files missing information could be because while generating the CIF files these information got missed by the generator or the creator didn't put it in their result. Our goal is to refine these information and generate these CIF information complete as much as possible. To do this, we have chosen a machine learning statistical approach to predict the missing value using the existing value.

2. BACKGROUND AND DEFINITION

In Crystallography, crystal structure describes the organization of the molecular structure. The crystal structure provides the gross structure of a system which describes the three-dimensional periodic arrangement of atoms, ions, or molecules in a crystal. The crystal structure also provides bond lengths, angles and electron density distribution. When crystal information in extracted from the material all these information are organized in scientific article. Later on, different organization extract the information from this published scientific article, put it to standard data exchange format and archive it. Inorganic Crystal Structure Database (ICSD), Cambridge Structural Database (CSD), Crystallography Open Database (COD) etc different organizations are responsible for archiving of these crystal structures information to the database.

3. WHY CIF CLASSIFICATION?

ICSD now contains 184,748 inorganic crystal structures and COD has around 362,577 crystal structures of organic, inorganic, metal-organic compounds and minerals, excluding bio polymers. These databases are built overtime with existing and newly discovered crystal structures. Not all the CIF files in these databases are complete. A CIF file for material could contain more than 100 fields value. Not all CIF for material is needed to have all of these fields. But some fields are important to describe the crystallographic structure of the material. There are 8 class of symmetry systems in the crystal system. These are:

cubic
hexagonal
monoclinic
orthorhombic
rhombohedral
tetragonal
triclinic
trigonal

From the Crystallography Open Database (COD), out of 362,577 crystal structures 71,964 CIF file doesn't have the _symmetry_cell_setting value. These CIF files are unable to describe the symmetry system of the material. So, we use these 362,577 crystal structures of Crystallography Open Database (COD) and use machine learning approach predictive model to predict the missing 71,964 CIF files _symmetry_cell_setting that is symmetry system.

4. DATABASE

There are 364,331 CIF files in COD database. CIF files are extensible, not all the CIF files have same set of keys or features. A single CIF file may contain 15 to more than 100 keys/features. We created a relational database 'CIF-Classification' and a table 'compound' to store these CIF files data. A tool was built 'CIFDataExtractor' which extracted the set of 56 keys and values from each CIF files and inserted into the database table 'compound' to see the data availability in these keys. Not all of these keys/features have strong information of the symmetry system and a big number of these keys in CIF files have missing values. For the classification purpose, we only picked up 10 keys from the CIF files which are most available and have most crystal structure information. Followings are the chosen keys from the CIF files:

_symmetry_cell_setting
_cell_angle_alpha
_cell_angle_beta
_cell_angle_gamma
_cell_formula_units_Z
_cell_length_a
_cell_length_b
_cell_length_c
_cell_volume
_space_group_IT_number

We put these information of each CIF files in the new database table 'classficationdata'. The keys/features we picked also have null values or incorrect information in the _symmetry_cell_setting column. So we filtered out these rows in the table by removing if any column in the row has null values or symmetry value is incorrect. After filtering we got 275926 data out of 364,331.

Label	Count
rhombohedral	821
hexagonal	3366
cubic	3935
trigonal	4846
tetragonal	7835
orthorhombic	45017
triclinic	70675
monoclinic	139431
Total	275926

We denote the CIF files with missing/incorrect symmetry value as a test class. There are 71,964 CIF files in with missing values/incorrect information. We will be predicting the missing/incorrect symmetry value of this 71,964 CIF files.

So, in the database 'classificaitondata' table '_symmetry_cell_setting ' is the target class and other 9 keys are features.

Though, these 9 features in the data suppose to have real values, some values in the features have a value in incorrect format like the following:

_cell_angle_alpha	95.560(5)
_cell_angle_beta	103.442(5)
_cell_angle_gamma	96.291(5)
_cell_length_a	9.899(5)
_cell_length_b	11.729(5)
_cell_length_c	12.
_cell_volume	1364.9(11)
_space_group_IT_number	?

so, a database function was written 'modify()' to change to all these incorrect values to correct format. After running this function to all the rows a real value like X.XX(X) changed to X.XX.

5. METHOD

Our objective here is to predict the missing/incorrect symmetry value of the 71,964 files in the Crystallography Open Database (COD). But right now we don't have the opportunity to validate our predicted class. For this reason, we use k-fold cross validation approach to verify the predication performance. We did 10 k-fold cross validation from the 275,926 label data and calculated the accuracy. In this process, we did 10 iterations. In each iteration, we picked 10% of the data as test from this label data and 90% a training data. That means in each iteration it picked 248,333 data a training data and 27592 data as test data. Then it trained the classification algorithm with 248,333 training data and tested with the 27592 test data. We accumulated all these 10 iterations result score and average the total score to get the accuracy. In this process, confusion matrix is also generated for all these 10 iterations. After 10 iteration, the average of the 10 confusion matrix was averaged. From this averaged confusion matrix we calculated the precision, recall and f-measure.

For a particular row in the data, the features value are like following:

_cell_angle_alpha	90
_cell_angle_beta	90
_cell_angle_gamma	120
_cell_formula_units_Z	6
_cell_length_a	14.2066
_cell_length_b	14.2066
_cell_length_c	33.41
_cell_volume	5839.7
_space_group_IT_number	167

I applied Z-score normalization method to scale the data.

6. RESULT

6.1 Classification

Different classification algorithms was run with different configurations on the k-fold cross validation process. These are 1) Logistic Regression 2) KNN 3) Support Vector Machine 4) Multilayer perceptron. The results are evaluated below

6.1.1 *Logistic Regression CIF classification*

Here is the result of logistic regression:

Data size	Accuracy	Precision	Recall	F-Measure
275926	90.4%	0.764900	0.655588	0.706

Confusion matrix figure-1:

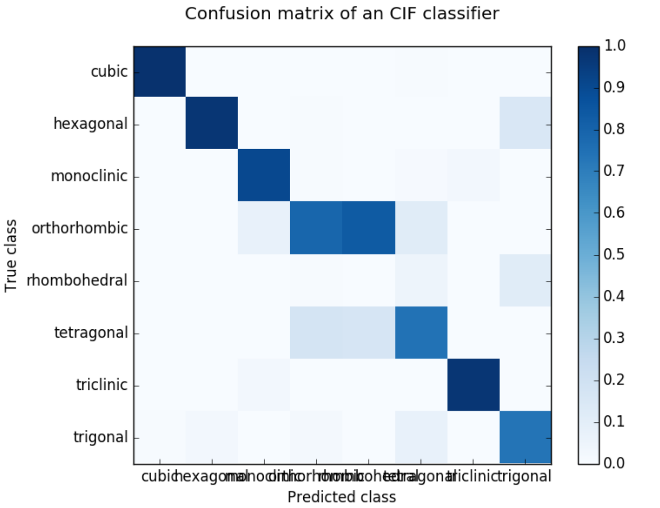

Figure 1: Confusion matrix of logistic regression CIF classification

6.1.2 *KNN CIF classification*

Here is the result of KNN CIF classification:

With Nearest Neighbor = 3, Metric = Euclidean

Data size	Accuracy	Precision	Recall	F-Measure
275926	98.1%	0.875740	0.861904	0.87

Confusion matrix figure-2:

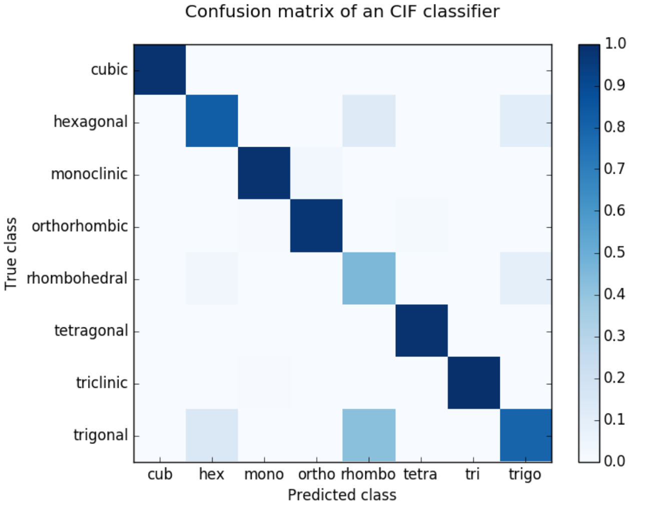

Figure 2: Confusion matrix of KNN (Nearest Neighbor = 3) CIF classification

With Nearest Neighbor = 5, Metric = Euclidean

Data size	Accuracy	Precision	Recall	F-Measure
275926	98%	0.873555	0.860282	0.882

Confusion matrix figure-3:

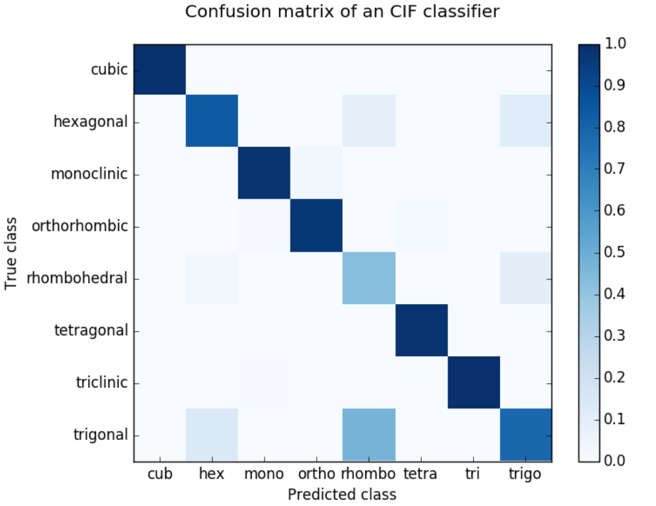

Figure 3: Confusion matrix of KNN (Nearest Neighbor = 5) CIF classification

With Nearest Neighbor = 7, Metric = Euclidean

Data size	Accuracy	Precision	Recall	F-Measure
275926	97.8%	0.873555	0.860282	0.882

Confusion matrix figure-4:

Here is the comparison chart figure-5:

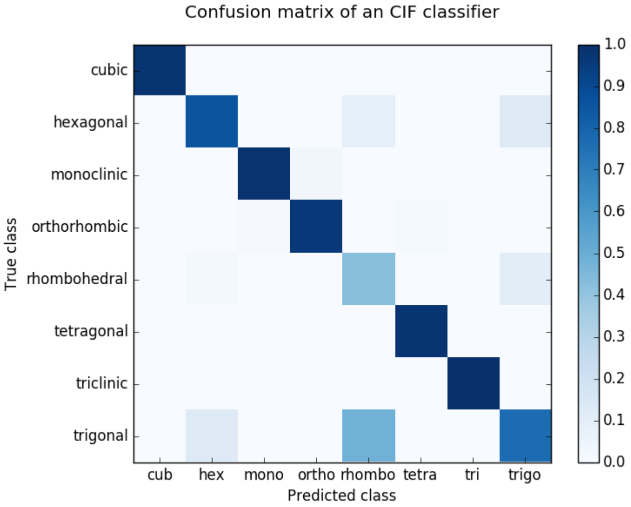

Figure 4: Confusion matrix of KNN(Nearest Neighbor = 7)CIF classification

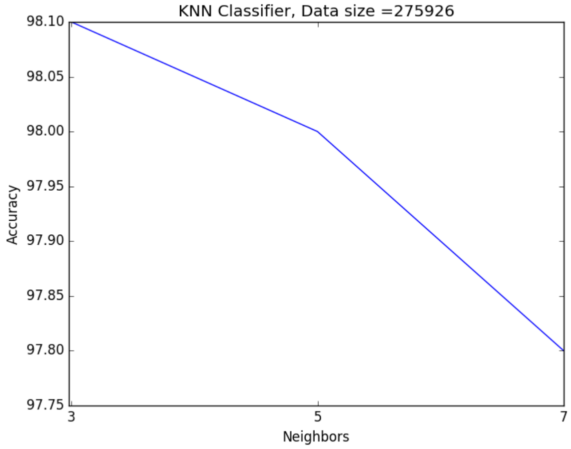

Figure 5: Comparison KNN classifications

6.1.3 *Multilayer perceptron CIF classification*

While running Multilayer perceptron algorithm with activity function 'tanh' over 275926 training data 99% accuracy was achieved. It means Multilayer perceptron was able to predict 99% data in k-fold cross validation with data size of 275926. From the confusion matrix we can see that Multilayer perceptron algorithm failed predict the class for rhombohedral. Out of this 275926 only 821(0.29%) data is rhombohedral. When I ran 10 k-fold cross validation, it picks up 90% out of 275926 data as a training set 10% data as test set. In this 10% test Multilayer perceptron was unable to find any rhombohedral in the test data. In that case, for label rhombohedral the precision can not be calculated. If we can't get the precision of rhombohedral then F-Measure also can not be calculated. For this reason, I have made a set of 6400 data taking 800 data from each group. Then I ran 10 k-fold cross validation . Here is the result :

Layer	Data size	Accuracy	Precision	Recall	F
1 (5,)	6400	77.5%	0.772	0.772	
1 (25,)	6400	89.1%	0.893,	0.891	
1 (50,)	6400	89.6%	0.900	0.897	
1 (100,)	6400	89.0 %	0.892	0.890	
2 (5,5)	6400	70.9 %	0.709	0.710	
2 (8,8)	6400	90.3 %	0.904	0.906	
2 (25, 25)	6400	90.5 %	0.908	0.906	
2 (50, 50)	6400	90.7 %	0.910	0.907	
3 (5, 5, 5)	6400	77.3 %	0.783	0.774	
3 (7, 7, 8)	6400	85.5 %	0.863	0.856	
3 (8,8,8)	6400	93.0 %	0.920	0.912	
3 (25, 25, 25)	6400	91.4 %	0.916	0.914	

And here is the comparison chart figure-6:

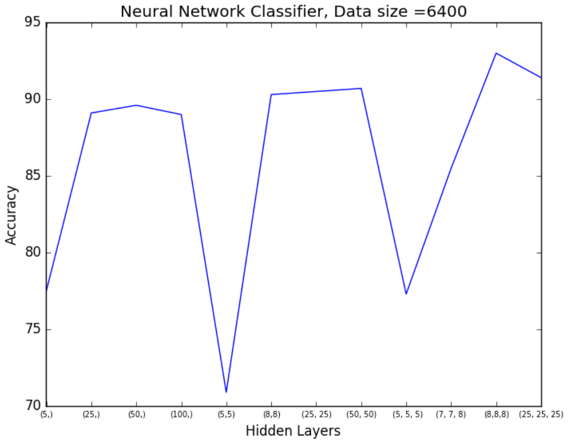

Figure 6: Comparison Neural Network classifications

6.1.4 *Support Vector Classification*

While running Support Vector Classifier over data size 275926 it fell into infinite loop. So, we picked data set of 6400 same as neural network and ran SVC with different kernel value. Here are the results:

With kernel = linear

Data size	Accuracy	Precision	Recall	F-Measure
6400	91.1%	0.915914	0.911864	0.91

Confusion Matrix figure-7:

With kernel = rbf

Data size	Accuracy	Precision	Recall	F-Measure
6400	90.6%	0.909812	0.905002	0.9

Confusion Matrix figure-8:

With kernel = sigmoid

Data size	Accuracy	Precision	Recall	F-Measure
6400	57.4 %	0.582944	0.575354	0.57

Confusion Matrix figure-9:

The chart comparison is given below figure-10:

6.2 Clustering

I ran some clustering algorithm on the data. The followings are the result:

6.2.1 *K-Means*

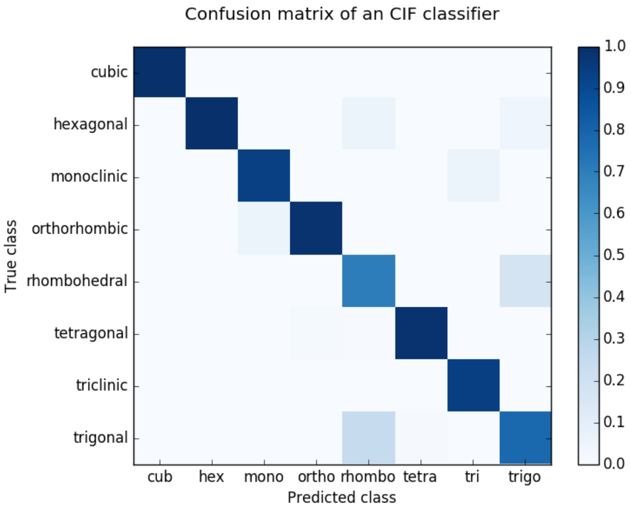

Figure 7: Confusion matrix of Support Vector (kernel=linear) classification

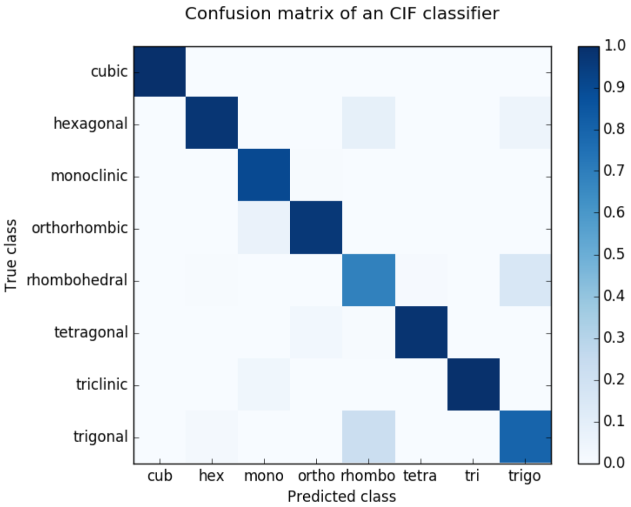

Figure 8: Confusion matrix of Support Vector (kernel=rbf) classification

K-Means algorithm was run over the data and k was set from 2 to 8.Total data size was 275926 and 100% instances were clustered. After clustering the data Accuracy, NMI, FMeasure, Homogeneity and Completeness were calculated to see the performance. Here is the performance table.

k	Accuracy	NMI	FMeasure	Homogeneity	Cmpltness
2	0.817	0.289	0.271	0.158	0.529
3	0.822	0.251	0.180	0.195	0.323
4	0.851	0.355	0.141	0.307	0.412
5	0.882	0.352	0.113	0.327	0.379
6	0.864	0.476	0.111	0.485	0.467
7	0.850	0.432	0.095	0.481	0.387
8	0.881	0.424	0.092	0.502	0.358

Here are the charts for the accuracy Figure 11, FMeasure Figure 12, NMI Figure 13, Completeness Figure 14 and Homogeneity Figure 15 :

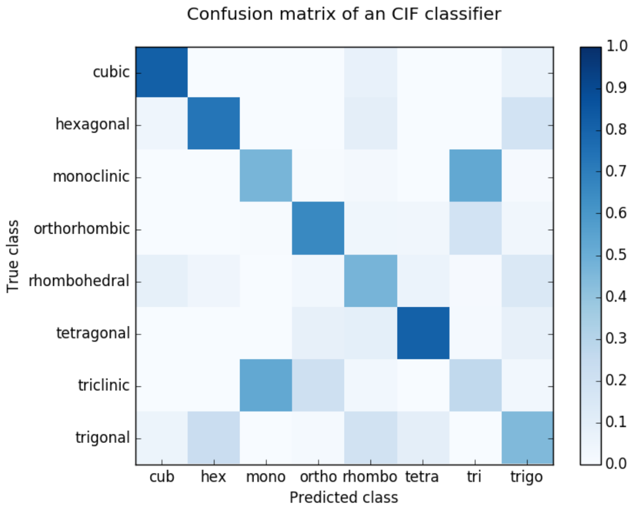

Figure 9: Confusion matrix of Support Vector (kernel=sigmoid) classification

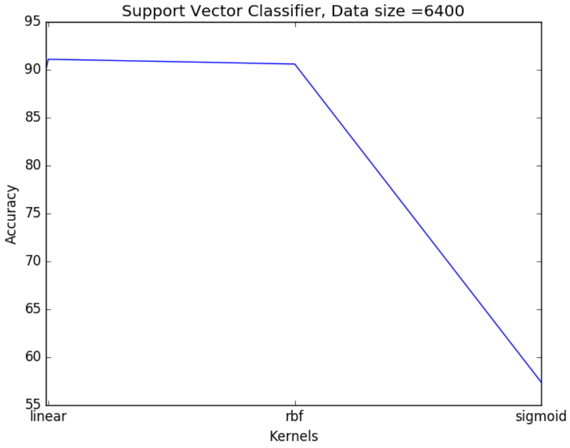

Figure 10: Comparison Support Vector classifications

Then the data is reduced to 2 dimension using PCA and ran the K-Means over reduced data setting the k=8. Here is the cluster in 2D space Figure 16

PCA(2)	KMeans	Performance
homo	compl	NMI
0.502	0.358	0.424

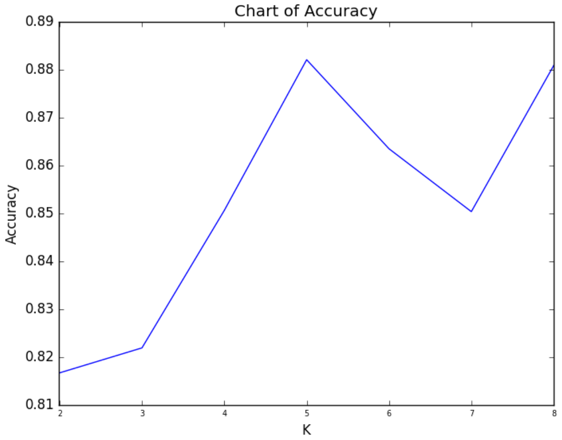

Figure 11: Accuracy of KMeans over CIF Clustering

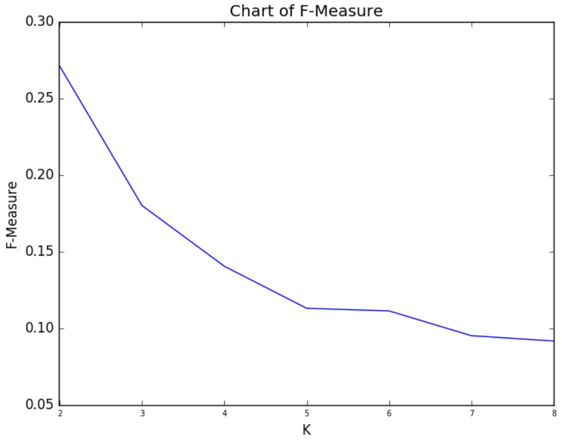

Figure 12: F-Measure of KMeans over CIF Clustering

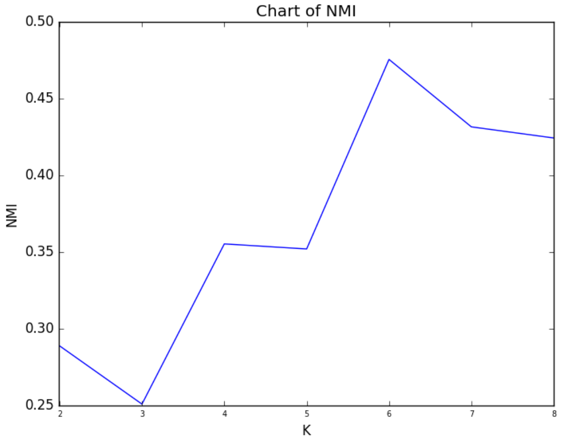

Figure 13: NMI of KMeans over CIF Clustering

6.2.2 DBScan

DBScan algorithm was run over the data where eps =

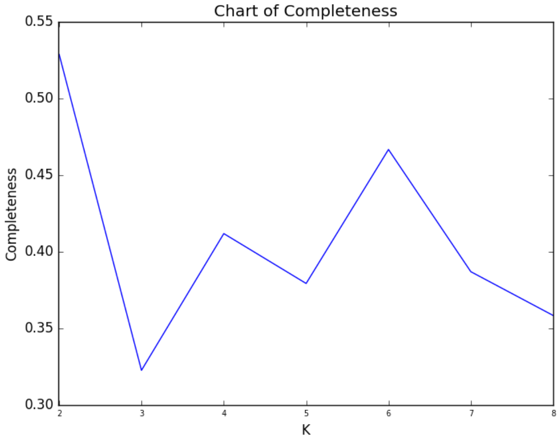

Figure 14: Completeness of KMeans over CIF Clustering

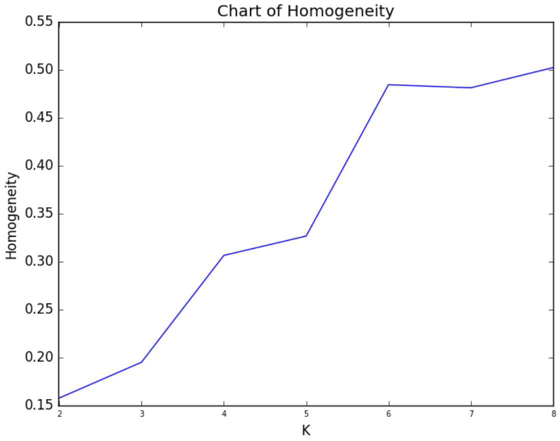

Figure 15: Homogeneity of KMeans over CIF Clustering

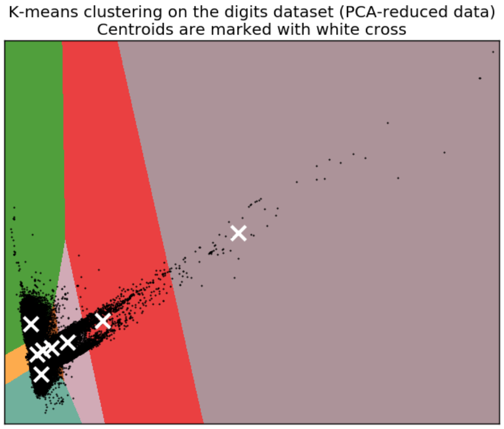

Figure 16: PCA-Based KMeans with k=8 over CIF Clustering

.5,.6,.8 and min-samples = 5,7,8,10,12 was set from 2 to 8. Total data size was 275926 and 95.51% instances were clustered. After clustering the data Accuracy, NMI, FMeasure, Homogeneity and Completeness were calculated to see the performance. In this case, 40 clusters were created for each settings. Here are the charts for the performance table. The performance table is shown below.

No-Cl	eps	minsamp	Acc.	NMI	FMsure	Homog	Cmpl
40	0.5	5	0.579	0.251	0.001	0.209	0.302
40	0.5	7	0.585	0.246	0.002	0.203	0.298
40	0.5	8	0.589	0.243	0.003	0.201	0.295
40	0.5	10	0.595	0.239	0.004	0.197	0.288
40	0.5	12	0.601	0.234	0.005	0.194	0.284
40	0.6	5	0.561	0.287	0.002	0.172	0.480
40	0.6	7	0.565	0.283	0.003	0.169	0.472
40	0.6	8	0.567	0.281	0.004	0.168	0.471
40	0.6	10	0.569	0.277	0.006	0.165	0.465
40	0.6	12	0.571	0.273	0.008	0.162	0.460
40	0.8	5	0.535	0.253	0.003	0.115	0.555
40	0.8	7	0.539	0.242	0.005	0.120	0.489
40	0.8	8	0.539	0.241	0.006	0.119	0.485
40	0.8	10	0.540	0.238	0.007	0.118	0.478
40	0.8	12	0.541	0.235	0.010	0.117	0.471

Here are the charts for the accuracy Figure 17, FMeasure Figure 18, NMI Figure 19, Completeness Figure 21 and Homogeneity Figure 20 :

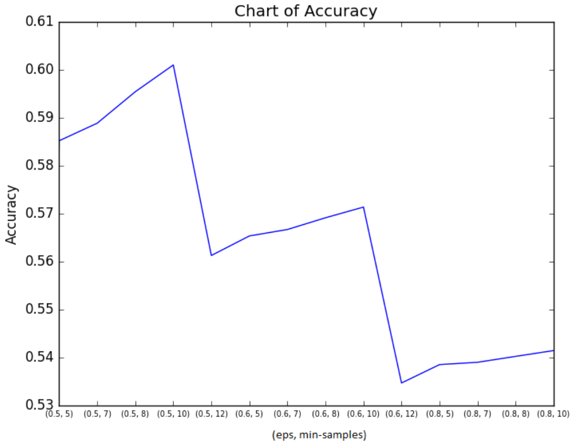

Figure 17: Accuracy of DBScan over CIF Clustering

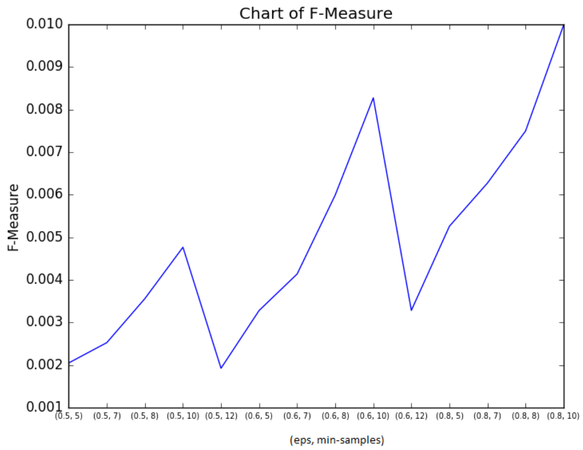

Figure 18: Fmeasure of DBScan over CIF Clustering

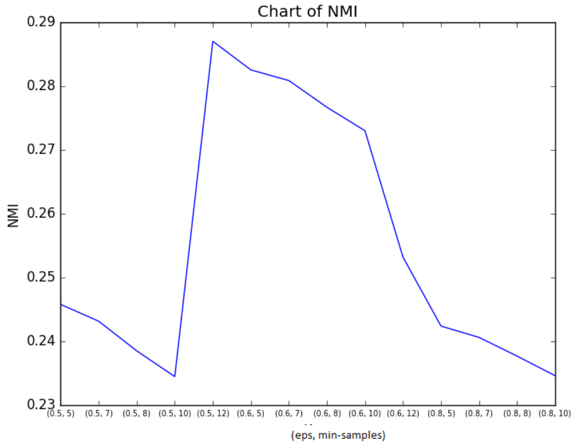

Figure 19: NMI of DBScan over CIF Clustering

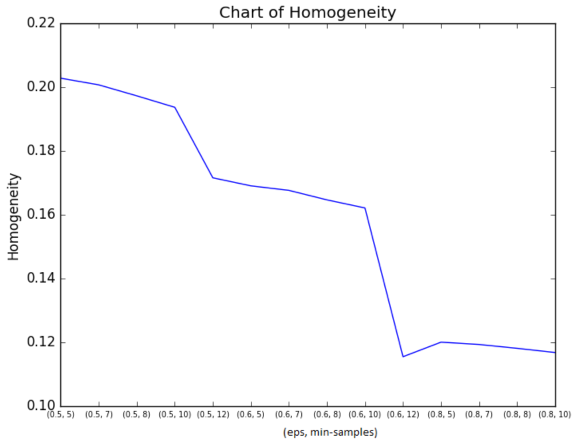

Figure 20: Homogeneity of DBScan over CIF Clustering

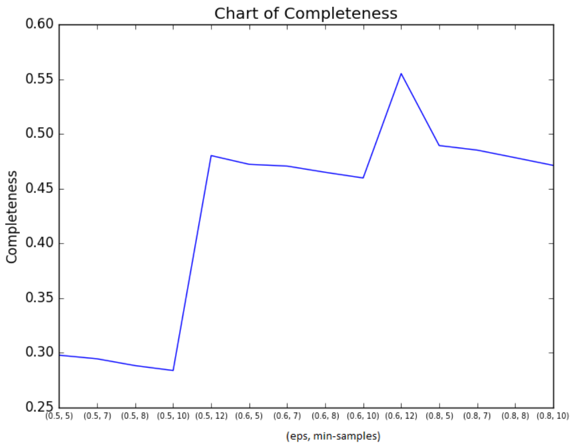

Figure 21: Completeness of DBScan over CIF Clustering

Then I take a sample of equally distributed data and ran DBScan over it. We took 6400 instances where size of each Here is the performance table.

No-Cl	eps	minsamp	Acc.	NMI	FMsure	Homog	Cmpl
10	0.5	5	0.691	0.538	0.014	0.507	0.570
10	0.5	7	0.705	0.536	0.022	0.486	0.590
10	0.5	8	0.719	0.535	0.028	0.478	0.600
10	0.5	10	0.751	0.534	0.052	0.460	0.619
10	0.5	12	0.738	0.527	0.049	0.452	0.614
10	0.6	5	0.525	0.450	0.011	0.400	0.507
10	0.6	7	0.546	0.445	0.019	0.380	0.523
10	0.6	8	0.550	0.442	0.018	0.376	0.519
10	0.6	10	0.575	0.441	0.026	0.367	0.530
10	0.6	12	0.586	0.440	0.048	0.351	0.552
10	0.8	5	0.385	0.419	0.014	0.316	0.556
10	0.8	7	0.404	0.413	0.022	0.309	0.551
10	0.8	8	0.414	0.409	0.028	0.305	0.549
10	0.8	10	0.430	0.407	0.035	0.304	0.546
10	0.8	12	0.441	0.402	0.037	0.301	0.536

6.2.3 tsne

I tried with the tsne method to cluster the CIF data. For the data size 275926 but it failed. Then it took sample data size 6400. The following is the result.

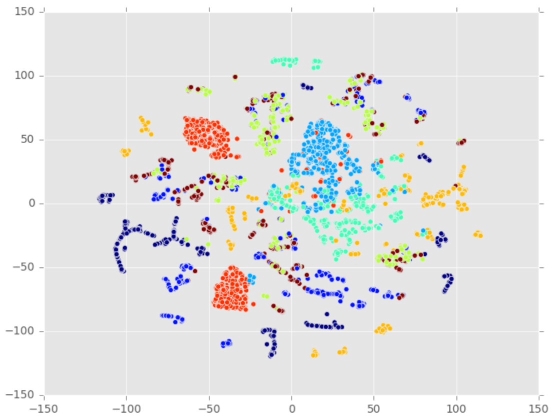

Figure 22: tsne CIF Clustering

6.3 Outlier

Outlier detection is determining the instances which are grossly different from or inconsistent with the remaining set of data. I used two outlier detection mechanism to detect outlier from the dataset.These are 1) local outlier factor 2) Isolation Forest.

6.3.1 Local outlier factor

To get the overview of lof, I first ran the lof over data size 6400. The the algorithm was run over data size 275926. Before running the outlier detection algorithm Multilayer Perceptron Classification algorithm was run. Then for each predicted class lof technique was applied. For preprocessing, the data was normalized using z-score and then ran PCA(n-components =2) was run over it. The following Figures Figure 23 Figure 24 shows the detected outlier for each classes after running lof. For data size 6400 the lof configuration was n-neighbors=50,contamination=0.1 and for data size 275926 the lof configuration was n-neighbors=500,contamination=0.1.

6.3.2 Isolation Forest

To get the overview of Isolation Forest , I first ran the Isolation Forest over data size 6400. The the algorithm was run over data size 275926. Before running the outlier detection algorithm Multilayer Perceptron Classification algorithm was run. Then for each predicted class Isolation Forest technique was applied. For preprocessing, the data was normalized using z-score and then ran PCA(n-components =2) was run over it. The following Figures Figure 25 Figure 26 shows the detected outlier for each classes after running Isolation Forest. The configuration for Isolation forest was (max-samples=50, contamination=.1,random-state=rng)

6.4 Ensemble

Previously I have used KNN and Multilayer perceptron for classification purpose. In ensemble technique, I have additionally used Random Forest classification technique to see if it increases the accuracy. First I ran these 3 classification technique individually and ran 10 k-fold cross validation over the data to calculate the accuracy score. Then ran combination of these classification ensemble technique with majority voting technique over the data. The following is the result.

Accuracy	Classification
0.98696	Multilayer Perceptron
0.98032	KNN
0.99366	Random Forest
0.99306	Ensemble

From the table we can see the Random Forest accuracy is the highest one. Ensemble techniques with majority voting accuracy is quite close to it. The accuracy of both of these techniques are already close to 100

Then I ran the ensemble technique excluding the knn technique. The result is following:

Accuracy	Classification
0.99101	Multilayer Perceptron
0.99366	Random Forest
0.99366	Ensemble

Excluding the knn from the ensemble increases the accuracy from 0.99306 to 0.99366.

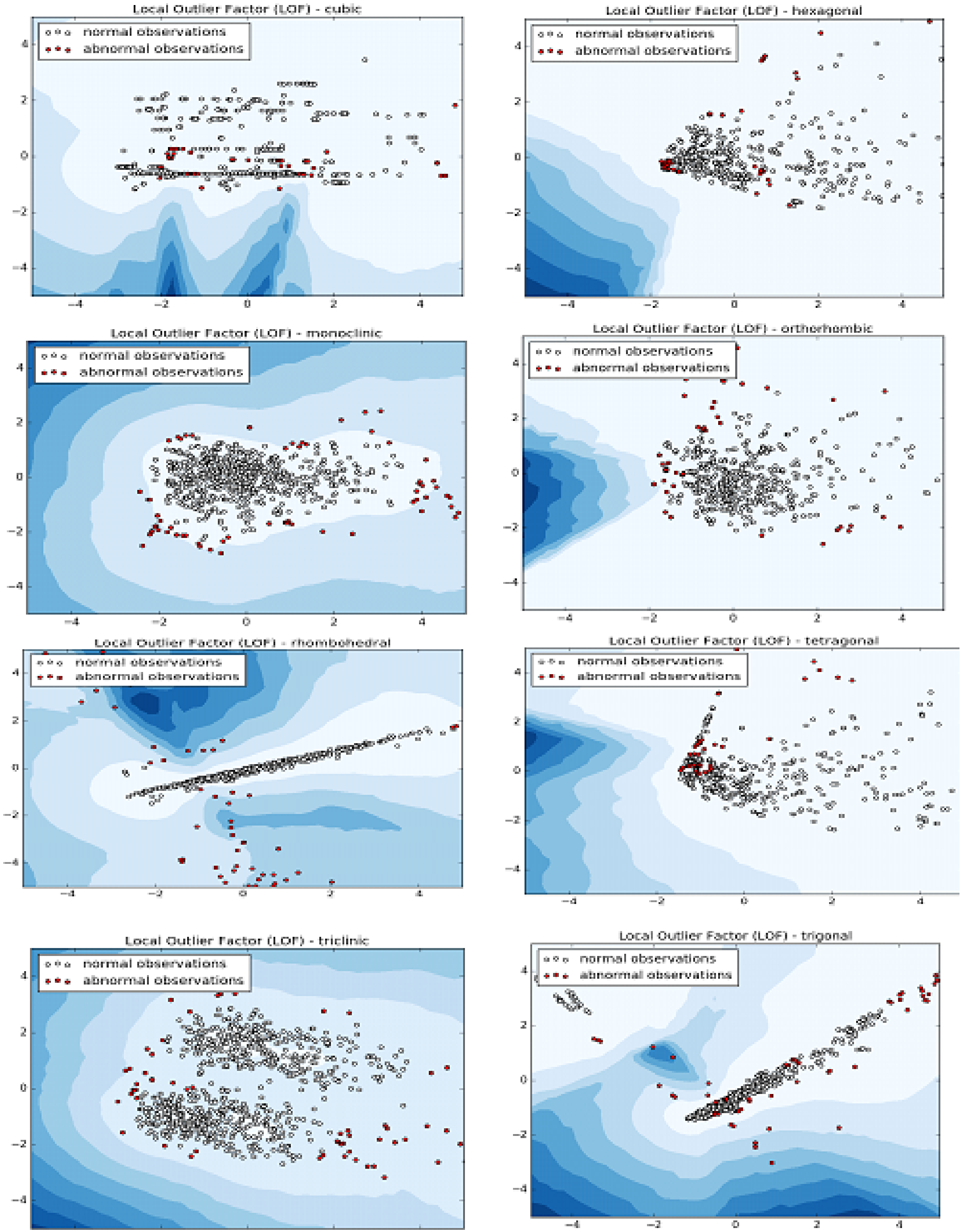

Figure 23: LOF over data size 6400

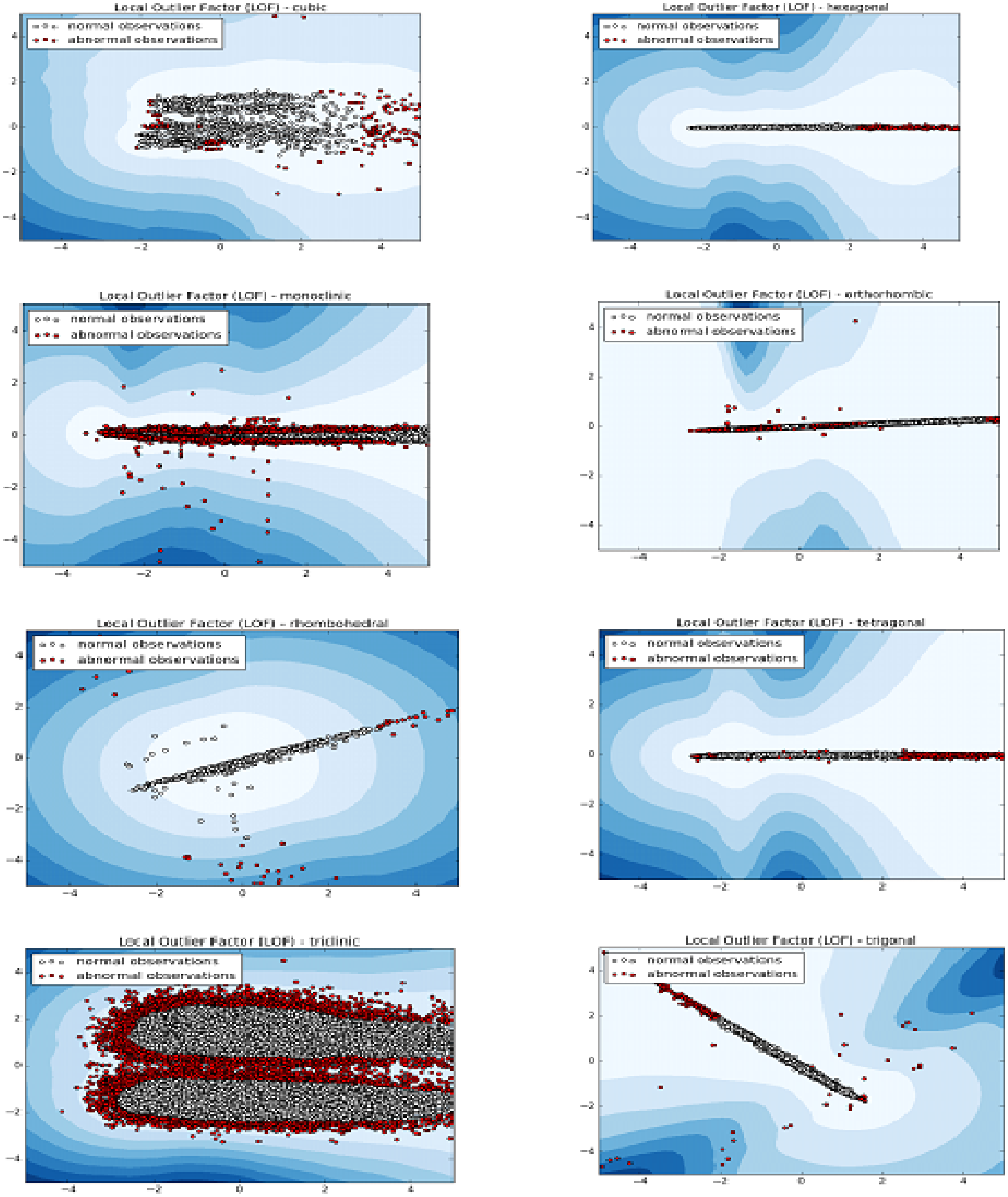

Figure 24: LOF over data size 275926

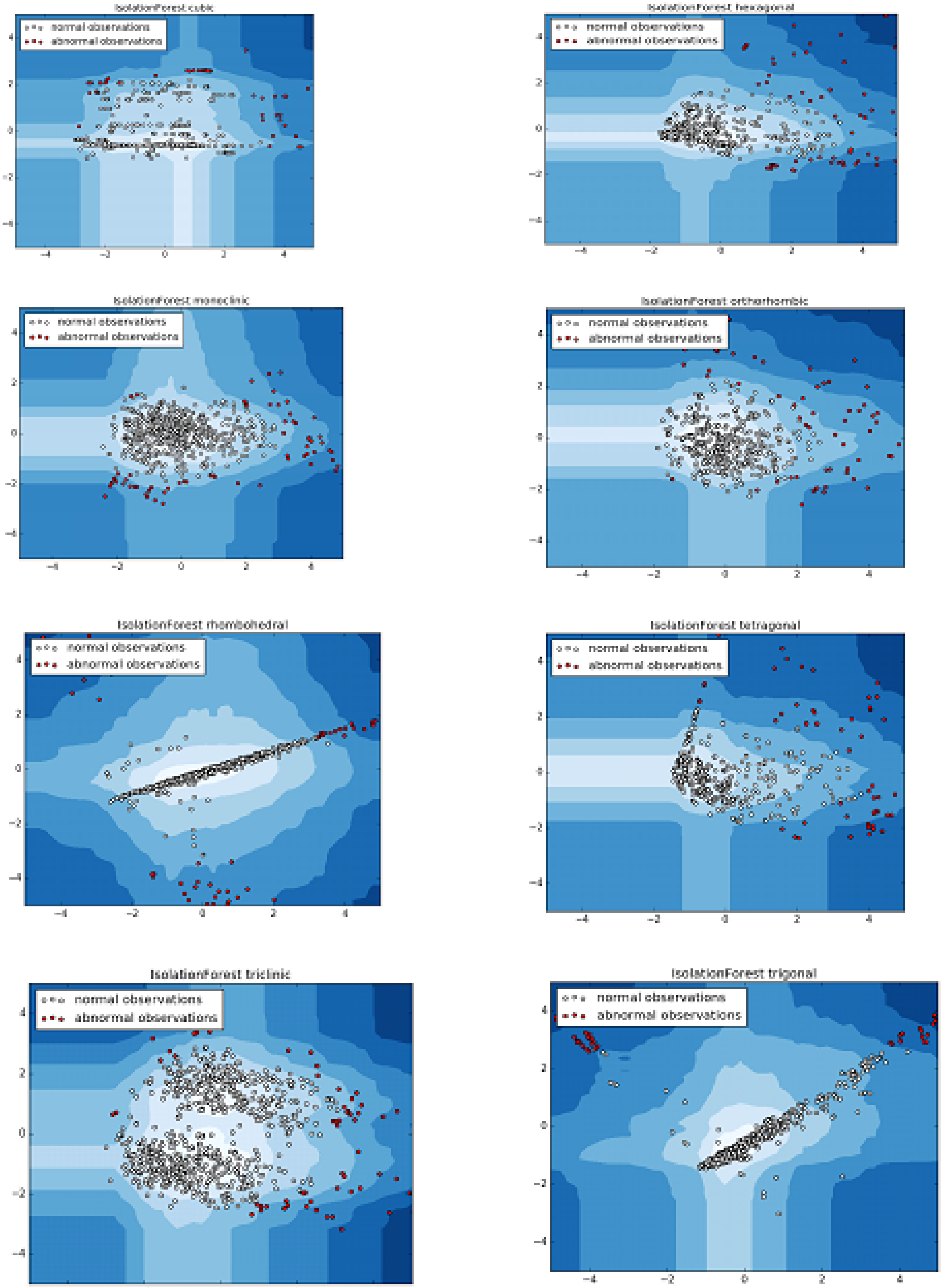

Figure 25: Isolation Forest over data size 6400

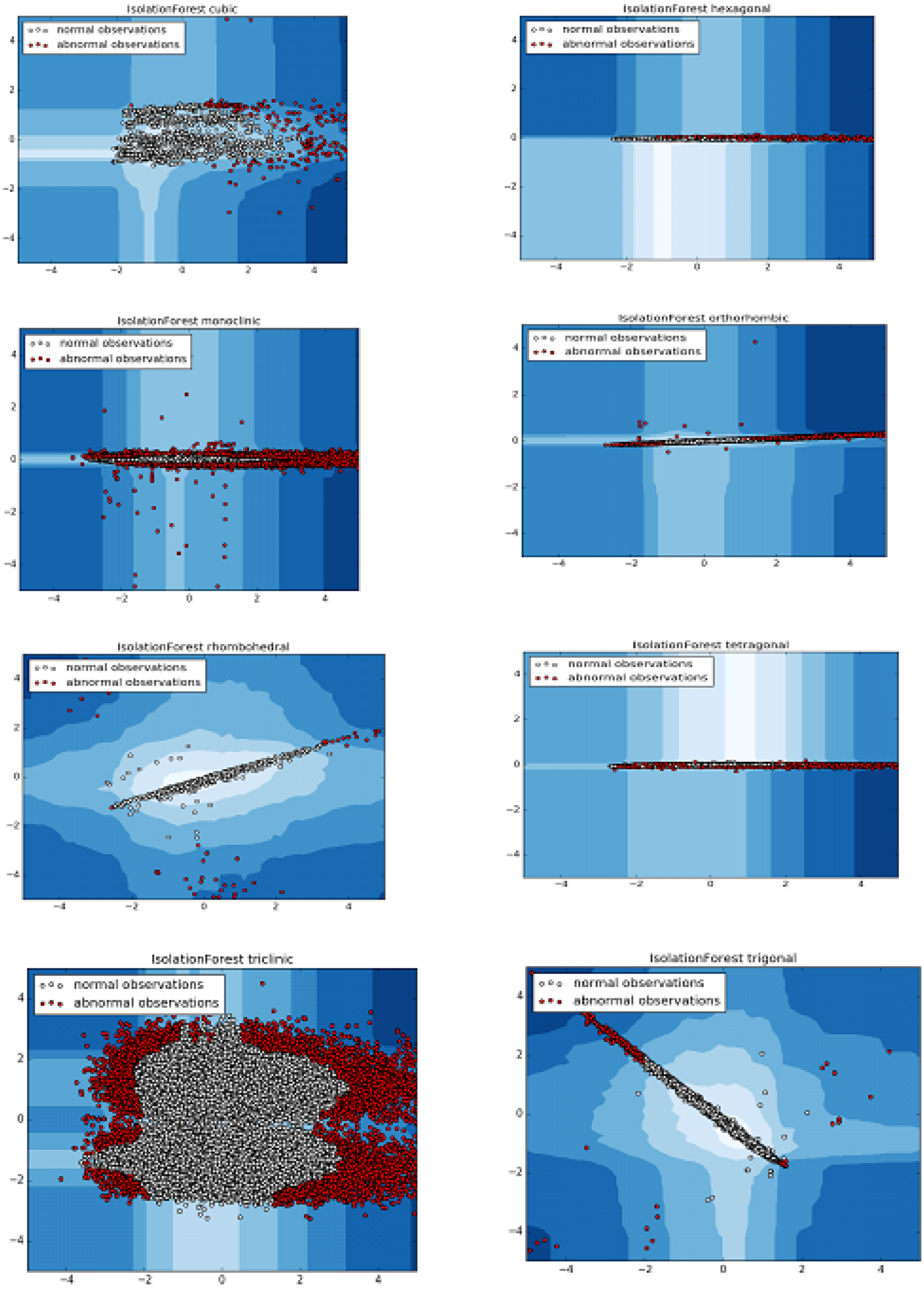

Figure 26: Isolation Forest over data size 275926

7. CONCLUSION

In this report, I tried to explored different characteristics of CIF data using different datamining technique. First, different classification techniques were used to classify and predict different classes of CIF. Next different clustering algorithm used to cluster the data and outlier algorithms were used to determine anamoly. Finally, ensemble technique was used to increase the accuracy of classification.

8. REFERENCES