Measuring Accuracy between Ensemble Methods: AdaBoost.NC vs. SMOTE.ENN

S.Lavanya
Department of CSE
Anna University Regional Centre
Coimbatore, India
slavanyamtech@gmail.com

Dr.Palaniswami
Department of CSE
Government College of Engineering
Bodinayakanur, India
joegct81@yahoo.com

Abstract— The imbalanced class distribution is one of the main issue in data mining. This problem exists in multi class imbalance, when samples containing in one class are greater or lower than that of other classes. Most existing imbalance learning techniques are only designed and tested for two-class scenarios. The new negative correlation learning (NCL) algorithm for classification ensembles, called AdaBoost. NC, produce smaller error correlation along with training epochs. In existing system, an AdaBoost.NC is proposed to handle multiclass imbalance problems that fall into two major categories: data sampling and algorithmic modification. Cost-sensitive learning solutions combining both the data and algorithm level approaches assume higher misclassification costs with samples in the minority class and seek to minimize high cost errors. In this system, the main objective is to analyze the performance of data level proposals against algorithm level proposals, focusing on cost-sensitive models and thereby proposing a hybrid procedure that combines these two approaches. In order to analyze the over-sampling and under-sampling methodologies against Cost-sensitive learning approaches the “synthetic minority oversampling technique” (SMOTE) is used.

Keywords— Ensemble learning, SMOTE-ENN, Multi class imbalance problem, NCL.

I. INTRODUCTION

Imbalance dataset classification problem occurs when one class usually contains more samples than the other classes and the classes are not equally distributed. Majority and minority classes are formed in that class distribution so the prediction task will pay more attention to majority class and the minority class is ignored. It impacts on several issues such as data loss, dense data, misclassification problem, over-fitting, degradation, accuracy and performance. The imbalanced data distribution is a common type of real-world problem. Several type of learning algorithms are proposed to solve imbalanced dataset classification but with meager accuracy. Most existing imbalance learning techniques are only designed for and tested in two-class scenarios. They have been shown to be less effective or even cause a negative effect in dealing with multiclass tasks. In the existing system, multiclass makes an imbalanced problem harder and new approaches to tackle the difficulties. The impact of multiclass on the performance of random over-sampling and under-sampling techniques are studied by discussing “multi-minority” and “multi-majority” cases in depth. Both “multi-minority” and “multi-majority” negatively affect the overall and minority-class performance. In particular, the “multi-majority” case tends to be more harmful. Random oversampling does not help the classification and suffers from over fitting. Increment of minority class results in weak under-sampling technique. When multiple majority classes exist, random under sampling can cause a great performance reduction to these majority classes.

Over-sampling and under-sampling in data analysis are techniques used to adjust the class distribution of a data set. Over-sampling and under-sampling are opposite and roughly equivalent techniques. Both techniques involve selecting samples more for one class; this is used for balancing the class. In this situation the over sampling will take place. Training this data will lead the classifier to wrongly represent the data from the source. Many real-world applications contain multi class pattern. To manage good customer relationship the multi class imbalance problem is solved in balanced pattern. In supervised learning method, learning can be initiated by the guidelines. An input, a set of output model and a target is given in supervised learning, so that, predicting the samples into particular class is easier.

AdaBoost, means Adaptive Boosting, which is a machine learning algorithm, discovered by Yoav Freund and Robert Schapiro. It is a meta-algorithm and can be used in conjunction with many other learning algorithms to improve their performance. AdaBoost is sensitive to noisy data. However some problems, such as less susceptibility to the over fitting problem is prevailed when compared with most learning algorithms. The classifiers it uses can be weak, but as long as their performance is not random, they will lead to improved final model. Classifiers of higher error rate would be useful, since they will have negative co-efficient in the final linear combination of it. One hypothesis is generated from a set of machine learning hypotheses is known as ensemble method, i.e., a set of base learners are collected and they are combined. Some techniques lead out of boundary data (over-fit) while other ensemble like bagging tend to reduce over-fitting of training data.

In classifier algorithms majority class is given higher priority during the learning process in favor of the accuracy metric and the minority class is not considered. The motivation of the project is to identify the minority class, with intrinsic characteristics of the imbalanced classification problem. The scope of the system is to address the imbalance dataset classification problem and classification learning performance. Finally the proposed classifier algorithm tries to improve the performance accuracy.

II. LITERATURE SURVEY

The Non-Modular classifiers tend to introduce high internal interferences because a strong coupling between their have hidden-layer weights can appear [7]. The Mod-NN architecture was used to face the multi-class imbalance problem. In that Mod-NN architecture, each module was a single-output NN. Imbalances are commonly referred to as intrinsic, i.e., the imbalance was a direct result of the nature of the data space [8]. Variable factors such as time and storage also give rise to data sets that are
imbalance. Imbalances of that type are considered extrinsic, i.e., the imbalance was not directly related to the nature of the data space. When standard learning algorithms are applied to imbalanced data, the induction rules are often fewer and weaker, since the majority class was often outnumbered and underrepresented.

Classification and prediction of biological entities have long been a central research theme in bio-informatics [1]. The major problem of applying discriminating classical machine learning techniques in that situation was: generated of a trivial Rejector classifier or positive samples over-fit by generating large decision trees are highly complex neural networks. Discriminative learning approaches mostly apply recursive partitioning of the instance space into regions labeled with the majority class in that region.

Imbalanced data distribution was common in real-world problems and has resulted in serious deterioration of the performance of most well-known classification techniques, as a result of being biased towards the majority class and hence misclassifying most of the minority samples to be of the majority class [9]. That imbalanced situation was even more complicated in multi-class classification. Most methods designed to solve that problem are based on splitting the classification problem into a k-class of smaller two-class sub problem. For each sub-problem, an independent binary classifier was built. Then, the results of the binary classifiers are combined to get the classification result. Several techniques were proposed for decomposing the multi-class problem, including the two popular approaches; One-Against-All (OAA) and One-Against-One (OAO). OAA and OAO approaches have been employed successfully in different domains.

A CEL (Co-operation Ensemble Learning System) have adequate theoretical support in the regression context and has been applied to solve classification problems, a theoretical gap in the classification context still needs to be filled [10]. CELS suffer some other drawbacks. So far, it was only applicable to neural networks. Other base learners are not suitable for its training procedure. The pattern-by-pattern weight updating strategy results in very high computational cost. The training can take a long time before it converges to the expected error threshold for a very large data set. To speed up the learning process, another NCL algorithm, negative correlation learning via correlation-corrected data (NCCD), was proposed by Chan and Kasabov. NCL algorithms for classification ensembles, AdaBoost. NC. AdaBoost’s flexibility is used to overcome the existing disadvantages of NCL algorithm, the over-fitting problem of AdaBoost are solved by introducing the concept of diversity. Low diversity will be penalized by using the ambiguity term derived in that paper. Different from AdaBoost, CELS and NCCD. AdaBoost. NC introduces the error correlation information into the weights of the training data, when the updating happens, or in other words, after each classifier was constructed. It was applicable to both binary-class and multi-class problems.

Solutions for the bi-class applications can be categorized as data level and algorithm level approaches [3]. Solutions for bi-class problems are not applicable directly to multiclass cases. One possible solution was to convert a multi-class problem into a number of bi-class problems, i.e., classifying each individual class versus all the other classes. Without the cost matrix, experiments for bi-class problems were conducted by setting up a range of cost factors manually. Such a strategy was not applicable to multiple class cases since to figure out satisfactory cost values manually for multiple classes was a non-trivial job. Hence, searching an efficient cost setup becomes a critical issue in applying the cost-sensitive boosting approach to multiple class applications.

Cost-sensitive boosting algorithms are therefore developed such that the boosting process may cater to the costly class. [2] However, most reported cost-sensitive boosting algorithm neglect the effects of cost items when choosing the weight update parameter, which was crucial in converting a “weaker” learning algorithm into a strong one, three versions of cost-sensitive AdaBoost algorithms by inducing the misclassification costs into the weight update formula of AdaBoost. During each version, weight update parameter was calculated continuously by considering the misclassification costs. These adaptations retain the best feature of AdaBoost while becomes sensitive to different levels of learning the importance of different classes.

The method was first used by Cohen et al. (2006). It involves three major steps: (1) using an agglomerative hierarchical clustering algorithm such as single linkage to form a dendogram, (2) gathering clusters from all levels of the dendogram and computing the cluster centroids as synthetic examples, and (3) concatenating centroids with the original minority class examples [4]. Though not clear whether it is in the original procedure, we remove the redundancies of centroids that might be found in more than one layer in the implementation. Difference between the mean of a data preprocessing method and the mean of no data preprocessing method, and conclude that both methods are statistically indifferent if zero is below the upper confidence bound. The second analysis intends to determine the statistically indifferent best data preprocessing methods. Nearest neighbor classifiers outperform the minimum distance classifier.

Regularized linear regression (RLR) represents the simplest example of a Decision function [6]. Combined with quadratic loss function it has an essential advantage: using gradient-based search procedure we can optimize the value of the step size. Consequently, we will observe a rapid decline in the target function. By definition, regression coefficients may be regarded as natural measurements of influence of the corresponding features. Ensemble classifiers are learning algorithms that construct a set of many individual classifiers (called base-learners) and combine them to classify test data points by sample average. It is well-known that ensembles are more accurate than the base-learners that make them. In the case of SVM that are interested to deal with limited sample size this is equal to the dimension of the corresponding kernel matrix. The well known bagging technique is relevant. According to this technique each base-learner used in the ensemble is trained with data that are randomly selected from the training sample.

Several boosting algorithms have been proposed for semi-supervised learning [5]. They essentially operate like self-training where the class labels of unlabeled examples are updated iteratively: a classifier trained by a small number of labeled examples is initially used to predict the pseudo-labels for unlabeled examples; a new classifier is then trained by both labeled and pseudo-labeled examples; the processes of training classifiers and predicting pseudo-labels are altered iteratively till stopping criterion is reached. Multi-Class Semi-Supervised Boosting (MCSSB), a semi-supervised boosting framework that is designed for multi-class semi-supervised learning problems. The problems of converting a multi-class classification problem into a number of binary ones are avoided by directly solving a multi class problem. Moreover, unlike the existing semi-supervised boosting methods that only assign pseudo-labels to the unlabeled examples with high classification confidence; the proposed framework decides the pseudo labels for unlabeled examples based on both the classification confidence and the similarities among examples. It therefore effectively explores both the manifold assumption and the clustering assumption for semi-supervised learning.

III. EXISTING SYSTEM

Most existing imbalance learning techniques are only designed and tested for two-class scenarios. They have been shown to be less effective or even cause a negative effect in dealing with multi-class tasks. Earlier work shows that it has the better generalization ability under two-class imbalance scenarios by exploiting ensemble diversity. The existing system aims to provide a better understanding of multi-class makes an imbalanced problem harder and new approaches to tackle the difficulties. So they proposed AdaBoost. NC combined with oversampling can better recognize minority class examples and better balance the performance across multiple classes with high G-mean without decomposing any cases. AdaBoost. NC means “negative correlation” version of AdaBoost, it should be capable of reducing the classifier correlation within the ensemble, or in other words, introducing diversity.

The AdaBoost. NC can produce lower error correlation and better classification boundaries than AdaBoost was shown. The strength of negative correlation learning and boosting are increased by using an ensemble learning algorithm AdaBoost. NC. The performance of algorithm will be greater.

It emphasizes ensemble diversity explicitly during training and shows very encouraging empirical results in both effectiveness and efficiency in...
comparison with the conventional AdaBoost and other NCL methods in
general cases. Some disadvantages of the existing system are more
investigation is need to be done for multi class imbalance problem to solve
all and search procedure is not finely tuned. Cost-sensitive approach is not
considered in the existing system.

IV. PROPOSED SYSTEM

Class imbalance is the most existing problem in the machine learning
real world applications. Most of the solutions for class imbalanced is
proposed for binary classes. If the classes are greater than two i.e., for multi
class their class Imbalance problem tends to be greater than the binary case.
Most of the solutions for multi class imbalance problem don’t solve the
whole class imbalance problem. AdaBoost algorithm along with
oversampling will solve the multi class imbalance problem in some cases
only not the entire problem is solved. And it is not focused on the cost
sensitive model so the time required to solve the problem will be greater.

For solving the class imbalance problem two main steps are followed in
cases. The first step is to modify the data samples in the training itself
and a standard classification is done and it will be followed by the
classifiers. The second step is to modify the algorithms so that the learning
methods are made to be flexible for class imbalance issues. The
 oversampling and under sampling methodologies are analyzed against cost
sensitive learning approaches, the “Synthetic Minority Over-sampling
Technique” (SMOTE) is used and its variant with the Wilson’s Edited
Nearest Neighbor (ENN) rule as they have shown to obtain a very robust
behavior among many different situations, a wrapper approach to combine
both. Applying a preprocessing step in order to balance the class
distribution is an effective solution to the imbalanced dataset problem.
Specifically, in this work an oversampling method is chosen which a well
known reference in the area is: the SMOTE algorithm and a variant called
SMOTE+ENN. For performing this algorithm 3 datasets are taken from the
UCI machine repository. The following table represents the datasets used in
the experiment.

<table>
<thead>
<tr>
<th>Data</th>
<th>Class</th>
<th>Size</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>6</td>
<td>214</td>
<td>70/76/17/13/9/29</td>
</tr>
<tr>
<td>Thyroid</td>
<td>3</td>
<td>215</td>
<td>150/35/30</td>
</tr>
<tr>
<td>Solarflare</td>
<td>6</td>
<td>1066</td>
<td>147/211/239/95/43/331</td>
</tr>
</tbody>
</table>

In that SMOTE algorithm, the positive class is over-sampled by taking each
minority class sample and introducing synthetic examples along the line
segments joining any/all of the k minority class nearest neighbors.
According to the required oversampling the neighbors are randomly chosen.
Inducing a classifier in such a situation can lead to over fitting. For that
reason a hybrid approach in that work is also considered, “SMOTE+ENN”,
where the Wilson’sENN rule is used after the SMOTE application to
remove from the training set an example misclassified by its three nearest
neighbors.

Cost-sensitive learning algorithms associate high misclassification
costs to positive instances which bias the search towards the positive class.
The problems of the both method propose a solution with support vector
machines where they integrate a cost-sensitive support vector machine with
the SMOTE technique of oversampling the minority instances. With that
behavior they manage to push the boundary away from the positive
instances and make the boundary better defined. The estimation is done
over a training and a test set. The training data is split into five partitions for
an internal five-fold cross validation. The wrapper applies this independent

V. SYSTEM ARCHITECTURE

Explanation of each block:
- Database- the UCI machine repository datasets are present
- Input data- From the available datasets a particular dataset can be selected.
- Multi class data-It has to be checked whether it is two class or multi class
imbalance problems because the experiment is focused on multi class
imbalance problem.
- Data Preprocessing- Summary about the datasets are collected and the
majority, minority classes are identified. AdaBoostNC- Existing system
algorithm.
- Train & Test data- Training and testing are applied.
- Estimate oversampling and under sampling- According to classes present
the oversampling and under sampling techniques are applied to minority
and majority classes respectively.
- SMOTE, SMOTE+ENN- Proposed system technique Synthetic Minority
Oversampling Technique with Wilson’s Edited Nearest Neighbor.
- Cost-sensitive, Wrapper approach- Proposed system is considered with the
cost sensitive method.
- Classify results- Existing and proposed system results are calculated and
compared in terms of accuracy.

VI. RESULTS

The performance results of the proposed system and existing system are
measured in terms of the classification accuracy. Classification accuracy is
measured in terms of precision, recall, F-Measure, MAUC and G-mean.
Fig1, Fig2 and Fig3 denote the accuracy between the proposed and existing
system. SMOTE-ENN with cost-sensitive algorithm shows better performance than the AdaBoost algorithm is proven in that graphs.

Majority and Minority classes are observed and instances are added to the minority classes to improve the performance because the cost of misclassifying a positive instance is higher than the cost of misclassifying the negative instances. Recall, Precision and F-measure are calculated for each class to measure completeness, exactness and to express tradeoff respectively. Extended G-mean and MAUC are measured to check the overall performance. They are calculated using the formula as follows. In a statistical hypothesis test, there are two types of incorrect conclusions that can be drawn. The hypothesis can be inappropriately. If the outcome from a prediction is p and the actual value is also p, then it is called a true positive (TP).

True positive rate (TPR) = TP/P
P = (TP+FN)
Where, P is the positive. TP is the True Positive.

A result that appears negative when it should not. For measuring the incorrect conclusion in both training and testing true and testing true negative rate is used. A true negative (TN) has occurred when both the prediction outcome and the actual value are n is the number of input data.

True negative rate (TNR) = TN/N
N = (TN+FN)
Where, N is the Negative value. TN is the True Negative.

A result that indicates that a given condition is present but it is not present. An example of a false positive would be if a particular test designed to detect cancer returns a positive result but the person does not have cancer. However if the actual value is n then it is said to be a false positive (FP).

False positive rate (α) = FP / (FP + TN)

A result that appears negative but that is not a negative. An example of a false negative would be if a particular test designed to detect cancer returns a negative result but the person actually does have cancer. False negative (FN) is when the prediction outcome is n while the actual value is p.

False negative rate (β) = FN / (TP + FN)

Precision value is calculated based on the retrieval of information at true positive prediction, false positive. In healthcare data precision is calculated the percentage of positive results returned that are relevant.

\[ \text{precision} = \frac{\text{truepositive}}{\text{truepositive} + \text{falsenegative}} \]

Recall value is calculated based on the retrieval of information at true positive prediction, false negative. In healthcare data precision is calculated the percentage of positive results returned that is recall in this context is also referred to as the True Positive Rate. Recall is the fraction of relevant instances that are retrieved.

\[ \text{recall} = \frac{\text{truepositive}}{\text{truepositive} + \text{falsepositive}} \]

F-measure is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct results divided by the number of all returned results and r is the number of correct results divided by the number of results that should have been returned. The F Measure score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0.

\[ \text{fmeasure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]

MAUC and extended G-mean are used to evaluate the overall performance as before. Recall and precision are recorded as the single-class performance measures to explain how an overall performance improvement or degradation happens. MAUC of the classifier can be calculated

\[ \text{MAUC} = \frac{2}{c \times (c-1)} \sum_{i<j}^{A_{ij} + A_{ji}} \]

where \( A_{ij} (A_{ji}) \) is the AUC value calculated from the \( i^{th} (j^{th}) \) column of the output matrix considering instances from class \( i \) and \( j \), notice that they are not necessary equal to each other.

Table1: Precision vs. dataset

<table>
<thead>
<tr>
<th>Data set</th>
<th>Ada Boost under sample</th>
<th>Ada Boost over sample</th>
<th>Smote under sample</th>
<th>Smote Over sample</th>
<th>Auto Under sampling</th>
<th>Auto Ove sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>70</td>
<td>73</td>
<td>78</td>
<td>81</td>
<td>84</td>
<td>91</td>
</tr>
<tr>
<td>1.25</td>
<td>69.5</td>
<td>72</td>
<td>78</td>
<td>80</td>
<td>83</td>
<td>90</td>
</tr>
<tr>
<td>1.5</td>
<td>67.5</td>
<td>72</td>
<td>77.5</td>
<td>79.5</td>
<td>81.5</td>
<td>89.5</td>
</tr>
<tr>
<td>2</td>
<td>65</td>
<td>71</td>
<td>77</td>
<td>78</td>
<td>80</td>
<td>88</td>
</tr>
<tr>
<td>2.25</td>
<td>63</td>
<td>70.5</td>
<td>76</td>
<td>77</td>
<td>79</td>
<td>86</td>
</tr>
<tr>
<td>2.5</td>
<td>61.5</td>
<td>69</td>
<td>75.5</td>
<td>76.5</td>
<td>79.5</td>
<td>85.5</td>
</tr>
<tr>
<td>2.75</td>
<td>60</td>
<td>68.5</td>
<td>75.3</td>
<td>76</td>
<td>78</td>
<td>85</td>
</tr>
<tr>
<td>3</td>
<td>57</td>
<td>68</td>
<td>75</td>
<td>76</td>
<td>76</td>
<td>84</td>
</tr>
</tbody>
</table>

Figure 1: Precision vs. dataset

In this graph the three dataset values are defined at the x axis Glass dataset values 1.00, thyroid dataset values 2.00 and finally solar flare dataset values are 3.00. Based on the dataset precision value comparison results changes .AdaBoost,NC algorithm for oversampling and under sampling ,Smote algorithm for under sampling and oversampling , Automatically algorithm for oversampling and under sampling. Finally automatic algorithm for oversampling and under sampling and found that high precision value that performs better results than the existing system.
Table 2: Recall vs. dataset

<table>
<thead>
<tr>
<th>Dataset values</th>
<th>AdaBoost under sample</th>
<th>AdaBoost over sample</th>
<th>SMOTE under sample</th>
<th>SMOTE over sample</th>
<th>Auto Under sample</th>
<th>Auto Over sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>62</td>
<td>63</td>
<td>70</td>
<td>73</td>
<td>85</td>
<td>90</td>
</tr>
<tr>
<td>1.25</td>
<td>61.5</td>
<td>62.5</td>
<td>68.5</td>
<td>72.5</td>
<td>83</td>
<td>88.5</td>
</tr>
<tr>
<td>1.5</td>
<td>61</td>
<td>61.5</td>
<td>68</td>
<td>72</td>
<td>81</td>
<td>86</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>61</td>
<td>67</td>
<td>71.7</td>
<td>79</td>
<td>84</td>
</tr>
<tr>
<td>2.25</td>
<td>59</td>
<td>60</td>
<td>67.5</td>
<td>70</td>
<td>77</td>
<td>81</td>
</tr>
<tr>
<td>2.5</td>
<td>58</td>
<td>59</td>
<td>66</td>
<td>69.5</td>
<td>75.5</td>
<td>79.5</td>
</tr>
<tr>
<td>2.75</td>
<td>55</td>
<td>57</td>
<td>65.5</td>
<td>69</td>
<td>73</td>
<td>79</td>
</tr>
<tr>
<td>3</td>
<td>53</td>
<td>56</td>
<td>65</td>
<td>67</td>
<td>71</td>
<td>77</td>
</tr>
</tbody>
</table>

Figure 2: Recall vs. dataset

In this graph the three dataset values are defined at the x axis. Glass dataset values are 1.00, thyroid dataset values are 2.00 and finally solarflare dataset values are 3.00. Based on the dataset precision value comparison results changes, AdaBoost.NC algorithm for oversampling and under sampling, Smote algorithm for under sampling and oversampling, Automatically algorithm for oversampling and under sampling. Finally automatic algorithm for oversampling and under sampling has high recall rate than the existing system.

Table 3: F-measure vs. dataset

<table>
<thead>
<tr>
<th>Dataset values</th>
<th>AdaBoost under sample</th>
<th>AdaBoost over sample</th>
<th>SMOTE under sample</th>
<th>SMOTE over sample</th>
<th>Auto Under sample</th>
<th>Auto Over sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>65</td>
<td>69</td>
<td>74</td>
<td>77</td>
<td>80</td>
<td>90</td>
</tr>
<tr>
<td>1.25</td>
<td>63</td>
<td>67</td>
<td>72</td>
<td>75.5</td>
<td>79.5</td>
<td>87</td>
</tr>
<tr>
<td>1.5</td>
<td>61</td>
<td>66</td>
<td>71</td>
<td>74</td>
<td>79</td>
<td>85</td>
</tr>
<tr>
<td>2</td>
<td>59</td>
<td>64</td>
<td>70</td>
<td>73</td>
<td>78</td>
<td>82</td>
</tr>
<tr>
<td>2.25</td>
<td>57</td>
<td>62</td>
<td>69</td>
<td>72</td>
<td>77</td>
<td>81</td>
</tr>
<tr>
<td>2.5</td>
<td>55</td>
<td>60</td>
<td>67.5</td>
<td>70.5</td>
<td>76.5</td>
<td>80</td>
</tr>
<tr>
<td>2.75</td>
<td>53</td>
<td>59</td>
<td>65</td>
<td>69</td>
<td>70</td>
<td>76</td>
</tr>
<tr>
<td>3</td>
<td>51</td>
<td>57</td>
<td>63</td>
<td>69</td>
<td>75</td>
<td>78</td>
</tr>
</tbody>
</table>

Figure 3: F-measure Vs Dataset

In this graph the three dataset values are defined at the x axis. Glass dataset values are 1.00, thyroid dataset values are 2.00 and finally solarflare dataset values are 3.00. Based on the dataset precision value comparison results changes, AdaBoost.NC algorithm for oversampling and under sampling, Smote algorithm for under sampling and oversampling, Automatically algorithm for oversampling and under sampling. Finally automatic algorithm for oversampling and under sampling has high F-measure value than the existing system.

The correlation analysis and Performance pattern analysis results show the behavior of cost-sensitive approach of SMOTE-ENN to be better than AdaBoost algorithm. MAUC and G-mean are used to evaluate the overall performance. Table 1. Show the results in terms of MAUC and G-mean for AdaBoost and SMOTE-ENN algorithms. SMOTE-ENN tends to be relatively stable with slightly better M-AUC and G-mean than the AdaBoost. Therefore the result clearly gives a path to solve Multi-class
imbalance problem in a cost-sensitive manner. Transformation of vast amount of raw data efficiently into information and knowledge representation is done in an easy and efficient manner. The datasets are defined within the boundary and the prediction of a data is obtained in a correct manner.

VII. CONCLUSION

AdaBoost. NC, on a set of benchmark data sets with multiple minority and/or majority classes with the aim of tackling multiclass imbalance problems effectively and efficiently. Analyzing multiclass imbalance problems since is unclear how an imbalance rate could be more appropriately defined. Multiclass Imbalanced datasets are collected and they are balanced by applying AdaBoost. NC algorithm.

The performance of the algorithm is carried out and their results are shown better by producing balanced data sets. The preprocessing performance in the framework of imbalanced datasets against other approaches is analyzed and the algorithm is performed without using any class decomposition. Two oversampling methods: SMOTE and SMOTE+ENN, a cost-sensitive version and a hybrid approach that tries to integrate both approaches together.

Thus the two approaches used to address the imbalanced problem and improve the overall performance in all the paradigms. SMOTE ENN used for solving multi class imbalance problem in cost sensitive approach results in a good enhancement approach than other multi class solution approaches.

REFERENCES