Fast and Robust Fuzzy-based Hybrid Data-level Method to Handle Class Imbalance

Kamlesh Upadhyay[1], Prabhjot Kaur[2], Ritu Sachdeva[3]
Research Scholar, Lingayas Vidyapeeth, Faridabad, New Delhi, India[1]
Department of Information Technology, Maharaja Surajmal Institute of Technology[2]
GGSIP University, New Delhi, India[2]
Professor, Lingayas Vidyapeeth, Faridabad, UP, India[3]

***Abstract*—Conventional classification algorithms do not provide accurate results when the data distribution (class sizes) is unequal or data is corrupted with noise because the results are biased towards the bigger class. In many real life cases, there is a requirement to uncover unusual/smaller classes. There are a bundle of examples where importance of smaller/rare class is much-much higher than the bigger class for example- brain tumor detection, credit card fraud or anomaly detection and many more. This is usually called as problem of imbalance classes. The situation becomes worst when the data is corrupted with extra impurities like noise in data or overlapping of class or any other glitch in data because in this scenario traditional methods produce more poor results. This paper proposed a fast, simple and effective data level hybrid technique based on fuzzy concept to overcome the class imbalance problem in noisy condition. To appraise the classification performance of the offered technique it is tested with 40 UCI real imbalanced data sets having imbalance ratio ranges from 1.82 to 129.44 and compared with 12 other approaches. The outcome specifies that the presented hybrid data level technique performed better and in a fast manner when compared to other approaches.**

***Keywords*—Data level approaches; undersampling; oversampling; fuzzy concept; imbalanced data-sets; classification**

I. Introduction

Classification methods are very useful in solving many real life problems. In the research literature, so many classification techniques are proposed like Decision Tree, SVM, Neural networks etc. These classification techniques work efficiently in classifying the balanced data-sets wherein the number of instances in the classes are approximately equal. Their internal design favors the balanced data-sets. These techniques fail to detect classes when used with imbalanced data-set, because as per their internal design the results in case of unequal size of classes deviate towards the bigger class. These algorithms ignore the smaller class as noise.

In real life situations, sometimes there is a need to detect exceptional cases e.g. credit card frauds, tumor detection, fraudulent telephone calls, shuttle system failure, text classification, oil spill detection, web spam detection, risk management, information retrieval, intrusion detection, earthquake and nuclear explosion, helicopter gear-box fault monitoring [1-4], etc. In such cases, Traditional Classification Algorithms do not work well. This problem is identified as Class Imbalance Problem (CIP). Class Imbalance problem is the classification problem wherein we are using traditional classification algorithms to classify data with unequal size classes and our objective is to identify smaller class from the data. Researchers have addressed this problem in various diversified ways and a new field of research has emerged under the name Class Imbalance Learning (CIL) and it is evolving day by day. In many papers it is referred to as dealing with IDS (Imbalanced data sets) or with rare cases or dealing with skewed data sets (SDS) or skewed distributions. The smaller class in CIL is known as minority class and bigger class is known as majority class.

Class Imbalance Problem does not exist if the purpose is to identify majority class, it actually exists because the purpose is to identify minority class. The ratio of the number of majority to minority class data instances is called imbalance ratio. The problem becomes more risky as this ratio increases i.e. when data-set is highly imbalanced. The techniques proposed by researchers to solve the Class Imbalance Problem are majorly classified into data-level approaches (Pre-processing techniques), algorithm level approaches and their hybrid forms [5-6]. In data-level approach, the researchers have tried to balance the data-sets before applying traditional classification algorithms so that results may not be overwhelmed by the majority class [7-15]. In algorithm level approaches, the researchers have worked upon the internal algorithm structure and tried to work upon the sensitivity of algorithm towards the majority class. These algorithms come under the category of cost sensitive algorithms [16-35]. Third approach is the hybrid form, which is the combination of data-level and algorithm level approaches. The advantage of data level approaches is that, the researcher will work at the data level and balance the data before classification and hence same classification algorithms can be used. This paper proposed a fast and robust hybrid data level approach based upon fuzzy logic. The proposed method can work with any level of imbalance data. It is tested with 40 UCI real world imbalanced data sets and its performance is compared with 12 other methods. It is observed that performance of proposed method is best compared to other methods. Rest of the paper is organized as follows. Section II explains the background information required to develop the method. Section III describes the proposed approach followed by conclusion in Section IV.

II. Background Information

This section explains various techniques and terms, which are required to develop the proposed approach.

A. Density Oriented Fuzzy C means (DOFCM)

Density oriented FCM is a robust clustering approach, which identifies and removes noise from the data based upon the density of the data [36, 37]. It uses density factor (neighborhood membership) to remove the outliers from the data. Density factor of DOFCM is defined as:

$$DensityFactor(D) = \frac{\eta^{i}_{neighborhood}}{\eta_{max}} \; \forall \; i \; in \; D \tag{1}$$

Where $\eta^{i}_{neighborhood}$ is the total number of points around i

η_{max} is the maximum number of points around any point in the whole D. D is the complete data-set.

DOFCM clusters the data into clusters using the following Objective function:

$$DOFCM_{Obj\_fun} = \sum_{l=1}^{c+1} \sum_{m=1}^{n} u_{lm}^{\zeta} d_{lm}^{2} \tag{2}$$

Where d_{lm} is the distance between center of a cluster 'l' and a point 'm' in the data-set. u_{lm} is the fuzzy membership between 'l' and 'm'. ζ is the fuzziness index. The membership equation for DOFCM is as below:

$$u_{lm} = \begin{cases} \dfrac{1}{\sum_{j=1}^{c}\left(\frac{d_{lm}}{d_{jm}}\right)^{\frac{2}{\zeta-1}}} \; \forall \; l,m \; if \; densityfactor \geq Threshold \\ 0 \quad if \; densityfactor < Threshold \end{cases} \tag{3}$$

B. Modified SMOTE

Chawla et al. in 2002 proposed Synthetic Minority Oversampling approach (SMOTE) [7]. This approach randomly selects candidate points and uses interpolation method to generate synthetic points in between the selected candidate points. Although the method is very simple but the limitation of existing SMOTE is that, it is not effective in case the data-set is corrupted with noise. In that situation, SMOTE method may select noise points as the candidate points (Fig. 1) and generate synthetic data within the candidate points. This situation may end up by generating more noise points within the data-set.

In the proposed approach, authors have used the variation of existing SMOTE method in order to avoid the limitation. The proposed method doesn't use random approach to select candidate points. It uses those points as candidate points which have large fuzzy membership values, which means the selected points will be close to the center of the minority class. It then uses interpolation method to generate the synthetic data between selected candidate points and the center of the minority cluster. Fig. 2 shows the process of synthetic data generation in case of modified SMOTE. In the figure, 'c' is the center of cluster, 'r' is the selected candidate point and 'n' is the synthetic point generated through interpolation. This approach intelligently generates the synthetic points by selecting only those points as candidate points, which are close to the center point; hence works on the limitation of existing SMOTE.

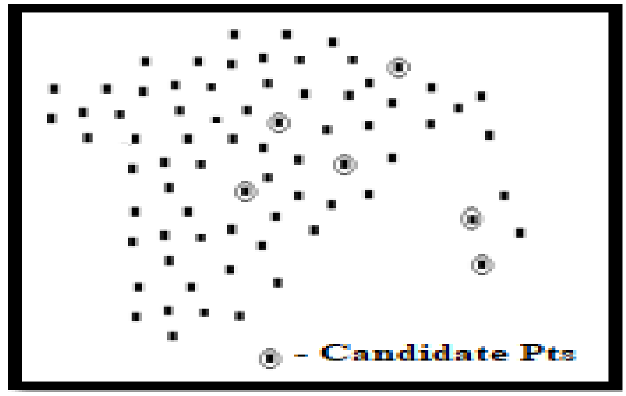

Fig. 1. Limitation of SMOTE

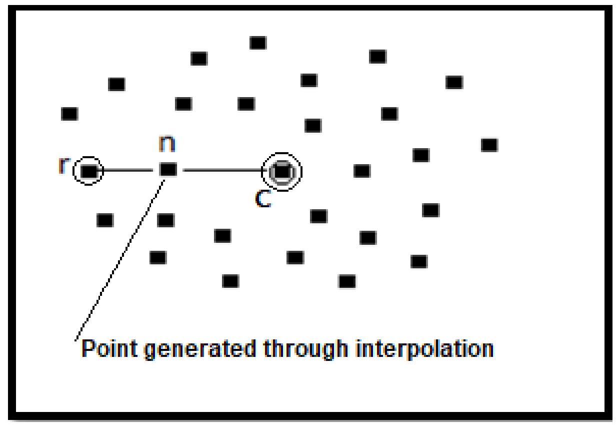

Fig. 2. Modified SMOTE.

C. Performance Criteria

Proposed approach used AUC (Area under the curve), F-measure and G-mean (Geometric mean) performance criteria's, which are majorly used by researchers in case of imbalanced data-sets, to compare the performance of proposed approach with their counterparts. As the focus of imbalance data sets is majorly to identify minority class so author considered minority class as the positive class in the confusion matrix (Table I) as mentioned.

AUC is a plot of false-positive rate on x-axis and true positive rate on y-axis. It is the best method to compare the performance of multiple classifiers. It is represented quantitatively by ROC and is calculated as the arithmetic mean of True Positive rate and True Negative rate.

$$AUC = \frac{TruePos_{Rate} + TrueNeg_{Rate}}{2} \tag{4}$$

Where $TruePos_{Rate}$ represents the amount of positive data categorized as positive and $TrueNeg_{Rate}$ represents negative data, which is correctly identified as negative.

TABLE I. CONFUSION MATRIX

	Minority (Positive)	**Majority (Negative)**
True	TP (True Positive)	TN (True Negative)
False	FP (False Positive)	FN (False Negative)

F-measure is the harmonic mean of precision and recall. Recall is the rate of total positive data, which is correctly identified as positive and Precision is the rate of correctly identified positive data out of total identified positive data. Recall is also known with the name as Sensitivity or True positive rate.

$$F - measure = \frac{(1+\gamma^2).Recall.Precision}{\gamma^2.Recall+Precision} \quad (5)$$

Where

$$Recall = \frac{TruePos}{TruePos+FalseNeg} \quad (6)$$

$$Precision = \frac{TruePos}{TruePos+FalsePos} \quad (7)$$

$'\gamma'$ parameter is used to set the importance of recall or precision.

Geometric mean represents the accuracy of every class. It is the geometric mean of True positive rate and True negative rate. It considers the performance of both the classes.

$$G - Mean = \sqrt{TruePos_{rate}.TrueNeg_{rate}} \quad (8)$$

III. Proposed Method

A. Description of the Proposed Method

This paper proposed fast, robust and effective hybrid data level approach based upon fuzzy concept to handle the imbalanced data. It is called fast and robust because it can handle any amount of noise in the data-set and has least time complexity compared to other methods. It is the most effective approach in case of real time datasets because of its noise resistant nature. It can work with any level of imbalance situation. (Refer to Section III-B). Fig. 3 and Fig. 4 show the algorithm and model of the proposed approach.

Input: *Imbalance Data corrupted with noise (I-DS)*
Output*: Balanced Dataset (B-DS)*

Step 1*: Cluster the dataset (I-DS) into minority and majority class using DOFCM clustering approach.*
Step 2*: Record the Fuzzy membership values of minority and majority data points.*
Step 3: *Reduce the size of majority class by removing those points whose fuzzy membership value is low.*
Step 4: *Enhance the size of minority class by using modified SMOTE concept (Refer to Section II-B)*
Step 5: *Combine the modified minority and majority class to generate balance dataset (B-DS).*

Fig. 3. Algorithm of Proposed Method.

B. Results and Simulations

To assess the performance of proposed approach, it is tested with 40 UCI real time imbalanced datasets [38] having an imbalance ratio ranging from 1.82 to 129.42. The properties of 40 UCI data sets are listed in Table VI (Appendix A). MATLAB R2018A [39] and Python framework are used to do the simulations. Its performance is compared with 12 other approaches namely RUSBoost [40], SMT-ENN [43], BalanceRandomForest (BRForest)[42], One Sided Selection (OSS) [43], ADASYN [44], SVMSMOTE [45], SMOTETomek (SMT-TL)[46][41], BorderLineSMOTE (B-SMT) [45], Edited Nearest Neighbor (ENN) [47], Condensed Nearest Neighbor (CNN) [48], Neighborhood Cleaning Rule (NCR) [49] and GradiantBoosting (GBoosting)[50]. In these simulations, authors have used Decision Tree method (C4.5) as the base classifier because in most of the research papers, C4.5 is widely used by the researchers to compare the methods in imbalance domains [51, 52]. Table II, Table III and Table IV list the AUC, G-mean and F-measure values of all the methods corresponding to 40 UCI real time imbalanced data-sets. Table V lists the average execution time against every method. As it is not possible to plot all the values hence authors plotted the average values of AUC, G-mean and F-measure in Fig. 5, Fig. 6 and Fig. 7. Average execution time in seconds is shown in Fig. 8.

C. Visual Interpretations and Discussions

It is observed from the Table II, Table III, Table IV and Fig. 5, Fig. 6, Fig. 7 that the performance of proposed data level hybrid method is best and consistent compared to all other methods irrespective of any imbalance ratio. It is seen that CNN performed worst in every case. The performance of RUSboost, GBoosting, ENN, OSS, NCR varies with the variation in imbalance ratio. Their performance degrades with the highly imbalance data-sets (abalone-19). Performance of SMT-TL, SMYSVM, ADASYN, B_SMT and SMT-ENN is almost similar in case of every data-set.

In case of execution time (Table V, Fig. 8), it is reported that the execution time in case of proposed method is least compared to other methods. Other methods are also taking less than one second in execution except CNN, which took the maximum time (up to three seconds).

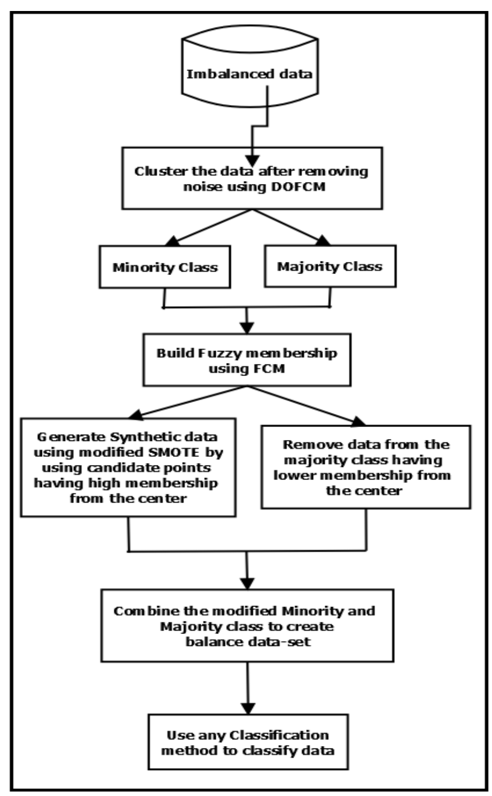

Fig. 4. Model of Proposed Method.

TABLE II. AUC VALUES OF 13 APPROACHES

Data Set	**Proposed method**	**RUSBoost**	**BRForest**	**SMT-ENN**	**SMT-TL**	**GBoosting**	**SMTSVM**	**ADASYN**	**B-SMT**	**ENN**	**CNN**	**OSS**	**NCR**
abalone19_b	**1**	0.57	0.72	0.98	0.96	0.5	0.98	0.97	0.99	0.49	0.4	0.5	0.49
abalone9_18	**1**	0.67	0.73	0.92	0.92	0.64	0.93	0.91	0.93	0.61	0.6	0.59	0.71
ecoli0137_vs_26	**1**	0.98	0.98	0.99	0.99	0.98	0.96	0.95	0.97	0.95	0.88	0.96	0.98
ecoli0_vs_1	**1**	0.98	0.98	1	1	0.98	0.96	0.95	0.97	0.95	0.88	0.96	0.98
ecoli1	**1**	0.85	0.9	0.97	0.92	0.86	0.92	0.87	0.9	0.63	0.9	0.92	0.97
ecoli2	**1**	0.85	0.9	0.97	0.92	0.86	0.92	0.87	0.9	0.63	0.9	0.92	0.97
ecoli3	**1**	0.85	0.9	0.97	0.92	0.86	0.92	0.87	0.9	0.63	0.9	0.92	0.97
ecoli4	**1**	0.85	0.9	0.97	0.92	0.86	0.92	0.87	0.9	0.63	0.9	0.92	0.97
glass_016_vs_2	**1**	0.67	0.83	0.92	0.95	0.58	0.92	0.92	0.92	0.42	0.42	0.54	0.44
glass_0123_vs_456	**1**	0.93	0.98	1	0.98	0.89	0.92	0.95	0.95	0.85	0.66	0.77	0.97
glass0	**1**	0.93	0.98	0.97	0.98	0.89	0.92	0.95	0.95	0.85	0.66	0.77	0.97
glass1	**1**	0.93	0.98	1	0.98	0.89	0.92	0.95	0.95	0.85	0.66	0.77	0.97
glass2	**1**	0.93	0.98	0.97	0.98	0.89	0.92	0.95	0.95	0.85	0.66	0.77	0.97
glass4	**1**	0.93	0.98	0.98	0.98	0.89	0.92	0.95	0.95	0.85	0.66	0.77	0.97
glass6	**1**	0.93	0.98	1	0.98	0.89	0.92	0.95	0.95	0.85	0.66	0.77	0.97
haberman	**1**	0.55	0.79	0.85	0.73	0.61	0.76	0.73	0.78	0.66	0.48	0.67	0.71
new_thyroid1	**0.99**	0.98	0.97	**0.99**	0.96	0.91	0.94	0.97	0.98	0.93	0.75	0.92	0.96
new_thyroi2	**0.99**	0.98	0.97	**0.99**	0.96	0.91	0.94	0.97	0.98	0.93	0.75	0.92	0.96
pima	**1**	0.69	0.75	0.88	0.76	0.73	0.72	0.7	0.7	0.84	0.59	0.67	0.8
segment0	**1**	**1**	**1**	**1**	0.99	0.99	0.99	1	0.99	0.99	0.91	0.99	0.98
shuttle_c2_vs_c4	**1**	**1**	**1**	**1**	0.99	0.99	0.99	1	0.99	0.99	0.91	0.99	0.98
shuttlec0_vs_c4	**1**	**1**	**1**	**1**	1	1	1	1	1	1	0.49	1	1
vehicle1	**1**	0.61	0.8	0.89	0.84	0.65	0.8	0.84	0.83	0.78	0.57	0.71	0.74
vehicle2	**1**	0.61	0.8	0.89	0.84	0.65	0.8	0.84	0.83	0.78	0.57	0.71	0.74
vehicle3	**1**	0.61	0.8	0.89	0.84	0.65	0.8	0.84	0.83	0.78	0.57	0.71	0.74
vowel0	**0.99**	0.98	0.97	0.98	0.99	0.88	0.99	0.99	0.99	0.96	0.64	0.93	0.98
wisconsin	**1**	0.92	0.99	0.97	0.97	0.96	0.97	0.98	0.93	0.94	0.6	0.74	0.93
yeast_05679_vs_4	**0.99**	0.66	0.82	0.94	0.89	0.72	0.94	0.91	0.9	0.73	0.6	0.8	0.77
yeast1	**0.99**	0.67	0.74	0.91	0.79	0.69	0.78	0.76	0.78	0.65	0.56	0.68	0.79
yeast1v6	**0.99**	0.67	0.74	0.91	0.79	0.69	0.78	0.76	0.78	0.65	0.56	0.68	0.79
yeast1v7	**0.99**	0.65	0.76	0.93	0.91	0.67	0.93	0.91	0.93	0.76	0.48	0.8	0.65
yeast2_vs_4	**0.99**	0.92	0.92	0.98	0.98	0.9	0.95	0.94	0.95	0.79	0.59	0.86	0.94
yeast2_vs_8	**0.99**	0.56	0.7	0.98	0.95	0.67	0.95	0.94	0.93	0.75	0.89	0.75	0.68
yeast3	**0.99**	0.87	0.93	0.94	0.94	0.86	0.96	0.94	0.94	0.87	0.78	0.87	0.93
yeast4	**0.99**	0.87	0.93	0.94	0.94	0.86	0.96	0.94	0.94	0.87	0.78	0.87	0.93
yeast4_u	**0.99**	0.87	0.93	0.94	0.94	0.86	0.96	0.94	0.94	0.87	0.78	0.87	0.93
yeast5	**0.99**	0.87	0.93	0.94	0.94	0.86	0.96	0.94	0.94	0.87	0.78	0.87	0.93
yeast6	**0.99**	0.87	0.93	0.94	0.94	0.86	0.96	0.94	0.94	0.87	0.78	0.87	0.93
yeast1289_vs_7	**0.99**	0.67	0.78	0.94	0.93	0.55	0.93	0.96	0.95	0.69	0.5	0.63	0.62
yeast1458_vs_7	**0.99**	0.64	0.69	0.94	0.92	0.55	0.9	0.9	0.93	0.58	0.47	0.53	0.52
Average	**0.996**	0.81425	0.884	0.952	0.92775	0.80325	0.91525	0.913	0.919	0.78925	0.678	0.79725	0.85575

TABLE III. G-MEAN VALUES OF 13 APPROACHES

Data Set	Proposed Method	RUSBoost	BRForest	SMT-ENN	SMT-TL	GBoosting	SMTSVM	ADASYN	B-SMT	ENN	CNN	OSS	NCR
abalone19_b	**0.996**	0.437	0.755	0.975	0.959	0	0.984	0.967	0.991	0	0	0	0
abalone9_18	**0.995**	0.585	0.757	0.916	0.92	0.541	0.93	0.908	0.93	0.519	0.55	0.457	0.657
ecoli0137_vs_26	**0.995**	0.985	0.757	0.981	0.973	0.541	0.977	0.968	0.968	0.994	0	0	0
ecoli0_vs_1	**1**	0.982	0.969	1	1	0.982	0.963	0.953	0.973	0.951	0.866	0.96	0.981
ecoli1	**0.987**	0.875	0.914	0.97	0.882	0.87	0.916	0.867	0.896	0.903	0.606	0.92	0.971
ecoli2	**0.987**	0.875	0.914	0.97	0.882	0.87	0.916	0.867	0.896	0.903	0.606	0.92	0.971
ecoli3	**0.987**	0.875	0.914	0.97	0.882	0.87	0.916	0.867	0.896	0.903	0.606	0.92	0.971
ecoli4	**1**	0.875	0.914	0.97	0.882	0.87	0.916	0.867	0.896	0.903	0.606	0.92	0.971
glass_016_vs_2	**1**	0.665	0.784	0.921	0.952	0.408	0.917	0.961	0.924	0.429	0.382	0.496	0
glass_0123_vs_456	**1**	0.944	0.981	1	0.977	0.887	0.92	0.95	0.947	0.838	0.658	0.754	0.966
glass0	**1**	0.944	0.981	1	0.977	0.887	0.92	0.95	0.947	0.838	0.658	0.754	0.966
glass1	**1**	0.944	0.981	1	0.977	0.887	0.92	0.95	0.947	0.838	0.658	0.754	0.966
glass2	**1**	0.944	0.981	1	0.977	0.887	0.92	0.95	0.947	0.838	0.658	0.754	0.966
glass4	**1**	0.944	0.981	0.98	0.977	0.887	0.92	0.95	0.947	0.838	0.658	0.754	0.966
glass6	**1**	0.944	0.981	0.98	0.977	0.887	0.92	0.95	0.947	0.838	0.658	0.754	0.966
haberman	**1**	0.646	0.746	0.842	0.718	0.535	0.756	0.725	0.777	0.651	0.482	0.667	0.711
new_thyroid1	**0.995**	0.96	0.98	0.98	0.956	0.907	0.942	0.974	0.983	0.936	0.707	0.92	0.961
new_thyroi2	**0.995**	0.96	0.98	0.98	0.956	0.907	0.942	0.974	0.983	0.936	0.707	0.92	0.961
pima	**1**	0.709	0.728	0.884	0.758	0.718	0.712	0.702	0.704	0.833	0.592	0.67	0.802
segment0	**1**	1	0.998	0.997	0.995	0.995	0.994	0.997	0.994	0.99	0.908	0.987	0.983
shuttle_c2_vs_c4	**1**	1	0.998	0.997	0.995	0.995	0.994	0.997	0.994	0.99	0.908	0.987	0.983
shuttlec0_vs_c4	**1**	1	1	1	1	1	1	1	1	1	0	1	1
vehicle1	**1**	0.689	0.799	0.885	0.84	0.587	0.804	0.836	0.83	0.784	0.571	0.697	0.733
vehicle2	**1**	0.689	0.799	0.885	0.84	0.587	0.804	0.836	0.83	0.784	0.571	0.697	0.733
vehicle3	**1**	0.689	0.799	0.885	0.84	0.587	0.804	0.836	0.83	0.784	0.571	0.697	0.733
vowel0	**0.989**	0.936	0.968	0.981	0.993	0.874	0.991	0.991	0.992	0.962	0.593	0.926	0.978
wisconsin	**1**	0.95	0.99	0.968	0.966	0.964	0.966	0.985	0.928	0.935	0.468	0.715	0.927
yeast_05679_vs_4	**0.989**	0.32	0.795	0.972	0.891	0.637	0.943	0.907	0.896	0.706	0.544	0.787	0.742
yeast1	**0.963**	0.663	0.729	0.913	0.785	0.649	0.779	0.756	0.783	0.636	0.557	0.675	0.789
yeast1v6	**0.963**	0.663	0.729	0.913	0.785	0.649	0.779	0.756	0.783	0.636	0.557	0.675	0.789
yeast1v7	**0.933**	0.573	0.752	0.925	0.909	0.603	0.933	0.914	0.934	0.734	0.377	0.783	0.589
yeast2_vs_4	**0.984**	0.919	0.926	0.981	0.979	0.897	0.95	0.936	0.95	0.769	0.566	0.849	0.963
yeast2_vs_8	**0.993**	0.562	0.709	0.984	0.948	0.577	0.946	0.936	0.928	0.707	0.889	0.707	0.621
yeast3	**0.993**	0.847	0.941	0.985	0.944	0.853	0.956	0.944	0.943	0.865	0.777	0.86	0.933
yeast4	**0.993**	0.847	0.941	0.985	0.944	0.853	0.956	0.944	0.943	0.865	0.777	0.86	0.933
yeast4_u	**0.993**	0.847	0.941	0.985	0.944	0.853	0.956	0.944	0.943	0.865	0.777	0.86	0.933
yeast5	**0.993**	0.847	0.941	0.985	0.944	0.853	0.956	0.944	0.943	0.865	0.777	0.86	0.933
yeast6	**0.993**	0.847	0.941	0.985	0.944	0.853	0.956	0.944	0.943	0.865	0.777	0.86	0.933
yeast1289_vs_7	**0.993**	0.659	0.782	0.948	0.929	0.332	0.929	0.958	0.953	0.647	0.413	0.558	0.495
yeast1458_vs_7	**0.993**	0.574	0.566	0.925	0.922	0.946	0.905	0.904	0.929	0.439	0.387	0.309	0.293
Average	**0.992**	0.80535	0.87605	0.960	0.922975	0.762125	0.9152	0.914125	0.9192	0.792925	0.585575	0.741075	0.794225

TABLE IV. F-MEASURE VALUES OF 13 APPROACHES

Data Set	**Proposed Method**	**RUSBoost**	**BRForest**	**SMT-ENN**	**SMT-TL**	**GBoosting**	**SMTSVM**	**ADASYN**	**B-SMT**	**ENN**	**CNN**	**OSS**	**NCR**
abalone19_b	**0.996**	0.06	0.043	0.977	0.96	0	0.979	0.968	0.991	0	0	0	0
abalone9_18	**0.996**	0.41	0.358	0.918	0.919	0.435	0.908	0.909	0.93	0.267	0.4	0.273	0.421
ecoli0137_vs_26	**0.996**	0.41	0.358	0.981	0.978	0.435	0.968	0.972	0.972	0.667	0	0	0
ecoli0_vs_1	**1**	0.98	0.964	1	1	0.982	0.969	0.959	0.98	0.923	0.979	0.957	0.98
ecoli1	**0.987**	0.76	0.772	0.97	0.883	0.783	0.918	0.868	0.9	0.852	0.792	0.857	0.913
ecoli2	**0.987**	0.76	0.772	0.97	0.883	0.783	0.918	0.868	0.9	0.852	0.792	0.857	0.913
ecoli3	**0.987**	0.76	0.772	0.97	0.883	0.783	0.918	0.868	0.9	0.852	0.792	0.857	0.913
ecoli4	**0.987**	0.76	0.772	0.97	0.883	0.783	0.918	0.868	0.9	0.852	0.792	0.857	0.913
glass_016_vs_2	**1**	0.4	0.375	0.921	0.948	0.286	0.909	0.959	0.923	0.2	0.2	0.125	0
glass_0123_vs_456	**1**	0.91	0.917	1	0.977	0.818	0.911	0.941	0.943	0.786	0.72	0.692	0.966
glass0	**1**	0.91	0.917	1	0.977	0.818	0.911	0.941	0.943	0.786	0.72	0.692	0.966
glass1	**1**	0.91	0.917	1	0.977	0.818	0.911	0.941	0.943	0.786	0.72	0.692	0.966
glass2	**1**	0.91	0.917	1	0.977	0.818	0.911	0.941	0.943	0.786	0.72	0.692	0.966
glass4	**1**	0.91	0.917	1	0.977	0.818	0.911	0.941	0.943	0.786	0.72	0.692	0.966
glass6	**1**	0.91	0.917	1	0.977	0.818	0.911	0.941	0.943	0.786	0.72	0.692	0.966
haberman	**1**	0.5	0.613	0.873	0.769	0.4	0.752	0.732	0.766	0.49	0.48	0.5	0.612
new_thyroid1	**0.991**	0.88	0.933	0.984	0.95	0.857	0.938	0.97	0.98	0.833	0.96	0.88	0.96
new_thyroi2	**0.991**	0.88	0.933	0.984	0.95	0.857	0.938	0.97	0.98	0.833	0.96	0.88	0.96
pima	**1**	0.64	0.667	0.89	0.764	0.654	0.707	0.684	0.716	0.8	0.646	0.6	0.798
segment0	**1**	1	0.989	0.997	0.995	0.995	0.994	0.997	0.994	0.99	0.965	0.973	0.951
shuttle_c2_vs_c4	**1**	1	0.989	0.997	0.995	0.995	0.994	0.997	0.994	0.99	0.965	0.973	0.951
shuttlec0_vs_c4	**1**	1	1	0.91	1	1	1	1	1	1	0.974	1	1
vehicle1	**1**	0.57	0.693	0.915	0.856	0.475	0.811	0.834	0.838	0.662	0.557	0.574	0.646
vehicle2	**1**	0.57	0.693	0.915	0.856	0.475	0.811	0.834	0.838	0.662	0.557	0.574	0.646
vehicle3	**1**	0.57	0.693	0.915	0.856	0.475	0.811	0.834	0.838	0.662	0.557	0.574	0.646
vowel0	**1**	0.84	0.779	0.982	0.992	0.852	0.99	0.99	0.992	0.962	0.83	0.897	0.947
wisconsin	**1**	0.95	0.987	0.968	0.966	0.961	0.966	0.984	0.925	0.929	0.946	0.932	0.919
yeast_05679_vs_4	**0.972**	0.16	0.528	0.955	0.895	0.5001	0.945	0.914	0.905	0.5	0.444	0.667	0.593
yeast1	**0.966**	0.54	0.609	0.922	0.787	0.564	0.78	0.769	0.793	0.5	0.635	0.558	0.731
yeast1v6	**0.966**	0.54	0.609	0.922	0.787	0.564	0.78	0.769	0.793	0.5	0.635	0.558	0.731
yeast1v7	**0.942**	0.22	0.261	0.936	0.906	0.4	0.929	0.912	0.931	0.552	0.222	0.643	0.3

yeast2_vs_4	**0.983**	0.65	0.686	0.962	0.978	0.71	0.949	0.934	0.949	0.692	0.629	0.774	0.759
yeast2_vs_8	**0.983**	0.32	0.37	0.945	0.947	0.5	0.945	0.932	0.925	0.667	0.5	0.667	0.421
yeast3	**0.988**	0.68	0.748	0.936	0.946	0.776	0.957	0.945	0.944	0.771	0.804	0.731	0.848
yeast4	**0.988**	0.68	0.748	0.936	0.946	0.776	0.957	0.945	0.944	0.771	0.804	0.731	0.848
yeast4_u	**0.988**	0.68	0.748	0.936	0.946	0.776	0.957	0.945	0.944	0.771	0.804	0.731	0.848
yeast5	**0.988**	0.68	0.748	0.936	0.946	0.776	0.957	0.945	0.944	0.771	0.804	0.731	0.848
yeast6	**0.988**	0.68	0.748	0.936	0.946	0.776	0.957	0.945	0.944	0.771	0.804	0.731	0.848
yeast1289_vs_7	**0.96**	0.14	0.155	0.953	0.93	0.167	0.908	0.958	0.953	0.276	0.2	0.214	0.3
yeast1458_vs_7	**0.96**	0.16	0.116	0.933	0.922	0.182	0.871	0.905	0.932	0.211	0.19	0.1	0.087
Average	**0.989**	0.66	0.693275	0.955	0.92575	0.666028	0.911875	0.913225	0.9204	0.693675	0.648475	0.6532	0.726175

TABLE V. AVERAGE EXECUTION TIME (SECONDS)

Algorithms	Proposed	RUSBoost	BRForest	SMT-ENN	SMT-TL	GBoosting	SMT-SVM	ADASYN	B-SMT	ENN	CNN	OSS	NCR
Average Time (Sec.)	0.004	0.513	0.330	0.079	0.054	0.156	0.045	0.019	0.021	0.046	3.506	0.048	0.050

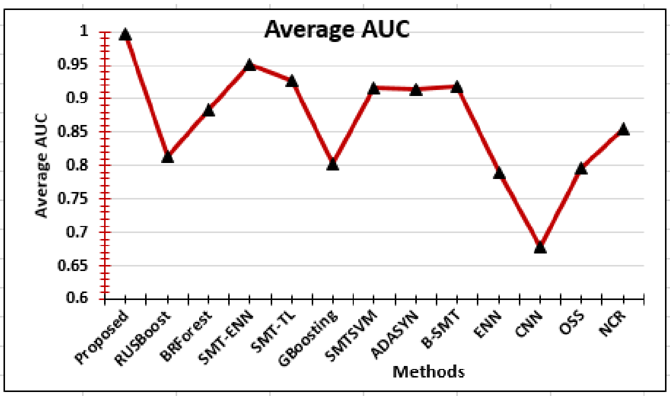

Fig. 5. Average AUC Results of 13 Methods.

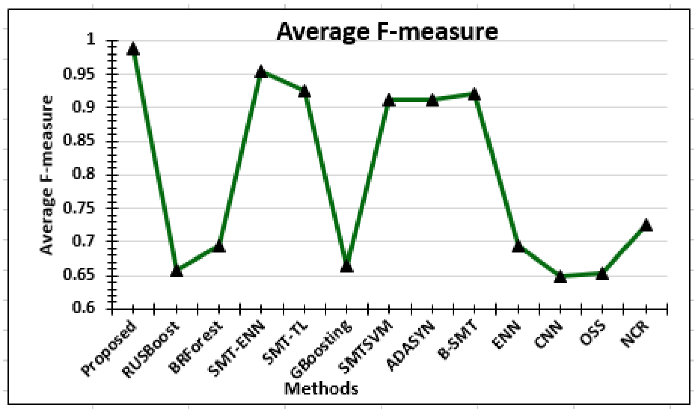

Fig. 7. Average F-measure Results of 13 Methods.

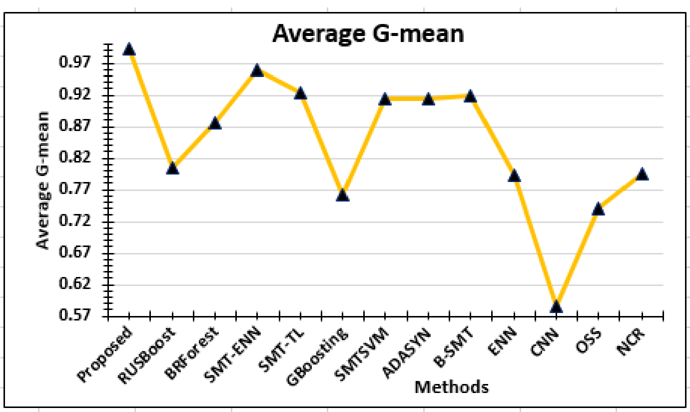

Fig. 6. Average G-mean Results of 13 Methods.

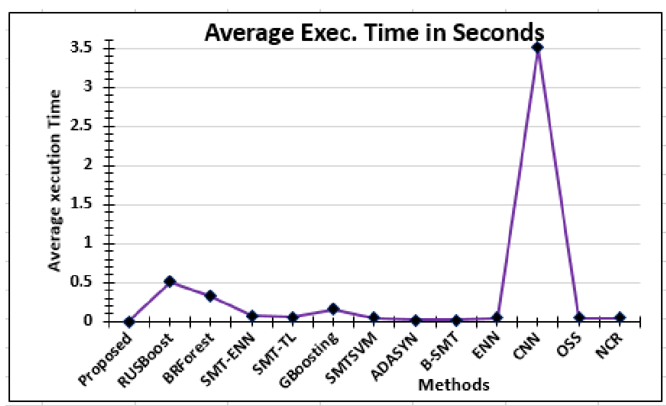

Fig. 8. Average Execution Time (Seconds) of 13 Methods.

From the simulations and observations, it is concluded that proposed method is a robust and fast approach to balance the data because it works consistently for any kind of data set within least time.

IV. Conclusion

In this paper, authors proposed fuzzy based fast and robust hybrid data level approach to balance the data. Its performance is tested with 40 UCI real time data-sets (Imbalance ratio- 1.82 to 129.44) and is compared with 12 other methods. After conducting the simulations, it is observed that proposed method can perform consistently with any level of imbalanced data compared to others and converge with the least execution time.

References

[1] Yang Yong, “The Research of Imbalanced data-set of sample sampling method based on K-means cluster and Genetic algorithm”, Energy Procedia, Vol. 17, pp 164-170, 2012 Sciverse ScienceDirect.

[2] V. Garcia et al., “The class imbalance problem in pattern classification and learning”, Pattern analysis and learning group, Conreso Espanol de Informatica; pp 283-291,2007.

[3] Sofia Visa, and Anca Ralescu, “Issues in Mining Imbalance data-sets – A Review paper”, Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference, pp. 67-73, 2005.

[4] Guo Hongyu and Herna L. Viktor, “Learning from imbalanced data sets with boosting and data generation: the databoost-im approach.”, ACM Sigkdd Explorations Newsletter, Vol. 6, No.1, pp. 30-39, 2004.

[5] Napierała Krystyna, Jerzy Stefanowski and Szymon Wilk, “Learning from imbalanced data in presence of noisy and borderline examples.” In Proceedings of International Conference on Rough Sets and Current Trends in Computing, Springer, Berlin, Heidelberg, 2010, pp. 158-167.

[6] K. P. N. V. Satyashree and J. V. R. Murthy, “An Exhaustive Literature Review on Class Imbalance Problem”, Int. Journal of Emerging Trends and Technology in Computer Science, Vol. 2, No.3, pp. 109-118, May 2013.

[7] N. V. Chawla et al., “SMOTE: Synthetic Minority Over-sampling Technique”, Journal of artificial Intelligence Research, Vol. 16, pp. 321-357, 2002.

[8] Bunkhumpornpat Chumphol, Krung Sinapiromsaran and Chidchanok Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem", In Proceedings of Pacific-Asia conference on knowledge discovery and data mining, Springer, Berlin, Heidelberg, 2009.

[9] Han Hui, Wen-Yuan Wang and Bing-Huan Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning", In Proceedings of International Conference on Intelligent Computing, Springer, Berlin, Heidelberg, 2005.

[10] S. Hu et al., “MSMOTE: Improving classification performance when training data is imbalanced”, In proceedings of 2nd Int. Workshop Computer Sci. Eng., Vol. 2, pp. 13-17, 2009.

[11] Nakamura Munehiro et al., "Lvq-smote–learning vector quantization based synthetic minority over–sampling technique for biomedical data", BioData Mining, Vol. 6, No. 1, pp. 16, 2013.

[12] García Salvador and Francisco Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy", Evolutionary computation, Vol. 17, No.3, pp. 275-306, 2009.

[13] Barua Sukarna et al., "MWMOTE--majority weighted minority oversampling technique for imbalanced data set learning," IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No.2, pp. 405-425, 2014.

[14] Rahman, M. Mostafizur and D. Davis, "Cluster based under-sampling for unbalanced cardiovascular data", In Proceedings of the World Congress on Engineering, 2013, Vol. 3.

[15] Ying Mi, “Imbalanced classification based on Active Learning SMOTE”, Research Journal of Applied Sciences, Engg. And Tech., Vol. 5, issue 3, pp. 944-949, 2013.

[16] R. Akbani, S. Kwek and N. Japkowicz, “Applying Support Vector Machines to Imbalanced Datasets”, In Proceedings of ECML 2004, LNAI 3201, pp. 39-50, 2004. Springer-Verlag Berlin Heidelberg.

[17] Fernández Alberto et al., "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets", Fuzzy Sets and Systems, Vol. 159, No. 18, pp. 2378-2398, 2008.

[18] R. Batuwita and V. Palade, “FSVM-CIL: Fuzzy Support vector machine for class imbalanced learning”, IEEE Transactions on Fuzzy Systems, Vol. 18, No. 3, pp. 558-571, 2010.

[19] H-L. Dai, “Class Imbalance Learning via a Fuzzy Total Margin based Support Vector Machine”, Applied SoftComputing, Vol. 31, pp.172-184, 2015.

[20] A. Fernandez et al., “A study of the behaviour of linguistic fuzzy rule base classification systems in the framework of imbalanced data-sets”, Fuzzy Sets and Systems, Vol. 159, issue 18, pp. 2378-2398, 2008.

[21] Galar, M., et al. (2013) ‘Dynamic classifier selection for One-vs-One strategy: Avoiding non-competent classifiers’, Pattern Recognition, Vol. 46, pp. 3412-3424.

[22] Gu, X., et al. (2014) ‘New Fuzzy Support Vector machine for the Class Imbalance Problem in Medical data-sets Classification’, The Scientific World Journal. Vol. 2014, pp. 1-12, Hindawi Publishing Corporation.

[23] He Haibo et al., "ADASYN: Adaptive synthetic sampling approach for imbalanced learning", In Proceedings of IEEE International Joint Conference on Neural Network, 2008, IEEE.

[24] T. Iman, K. Ting and J. Kamruzzaman, “z-SVM: An SVM for improved classification of imbalanced data”, In proceedings of the 19th Australian joint conference on Artificial Intelligence, springer-verlag, 2006, pp. 264-273.

[25] Y. Tang, B. Jin and Y. Q. Zhang, “Granular support vector machines with association rules mining for protein homology prediction”, Artificial Intelligence in Medicine, Vol. 35, No.1-2, pp. 121-134, 2005.

[26] Y. Tang, B. Jin, Y. Q. Zhang, H. Fang, B. Wang, “Granular support vector machines using linear decision hyperplanes for fast medical binary classification”, In Proceedings of FUZZ'05, The 14th IEEE International Conference on Fuzzy Systems, 2005, May 25, pp. 138-142.

[27] Y. C. Tang, Y.Q. Zhang, Z. Huang, Hu XT and Y. Zhao, “Granular SVM-RFE feature selection algorithm for reliable cancer-related gene subsets extraction on microarray gene expression data”, In Proceedings of IEEE Symp. Bioinformatics and Bioeng, 2005, pp. 290-293.

[28] Fan Wei et al., "AdaCost: misclassification cost-sensitive boosting", In Proceedings of Icml, Vol. 99, 1999.

[29] S. Wu and S. Amari, “Conformal Transformation of kernel functions: A data-dependent way to improve the performance of support vector machine classifier”, Neural Networks Letter, Vol. 15, 2002.

[30] G. Wu and E. Chang, “Kba: Kernel Boundary alignment considering imbalanced dataset distribution”, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, pp. 786-795, 2005.

[31] Fernández Alberto, María José del Jesus and Francisco Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets", International Journal of Approximate Reasoning, Vol. 50, No. 3, pp. 561-577, 2009.

[32] Z. Chi, H. Yan and T. Pam, “Fuzzy algorithms with application to image processing and pattern recognition”, Vol. 10, World Scientific, Singapore, 1996.

[33] H. Ishibuchi and T. Yamamoto, “Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in Data mining”, Fuzzy Sets and Syste ms, Vol. 141, No.1, pp. 59-88, 2004.

[34] H. Ishibuchi and T. Yamamoto, “Comparison of Heuristic criteria for fuzzy rule selection in classification problems”, Fuzzy Optim. Decision making, Vol. 3, No. 2, pp. 119-139, 2004.

[35] H. Ishibuchi, and T. Yamamoto, “Rule weight specification in fuzzy rule based classification systems”, IEEE Trans. Fuzzy Systems, Vol. 13, pp. 428-435, 2005.

[36] Prabhjot, Kaur, I. M. S. Lamba, and Gosain Anjana. "DOFCM: a robust clustering technique based upon density." International Journal of Engineering and Technology 3.3 (2011): 297.

[37] Kaur, Prabhjot, and Anjana Gosain, "Density-oriented approach to identify outliers and get noiseless clusters in Fuzzy C—Means." International Conference on Fuzzy Systems. IEEE, 2010.

[38] Alcalá-Fdez Jesús et al., "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework", Journal of Multiple-Valued Logic & Soft Computing, Vol. 17, 2011.

[39] Natick, Massachusetts MATLAB version 8.1 (2013): The MathWorks Inc., 2013.

[40] C. Seiffert et al., "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance", IEEE Trans. on Sys. Man and Cyber.-Part A, Vol. 40, No.1, pp. 185-197, 2010.

[41] Gustavo EAPA Batista, Ana LC Bazzan, and Maria Carolina Monard, "Balancing training data for automated annotation of keywords: a case study", In WOB, 10–18. 2003.

[42] Chen, Chao, Andy Liaw, and Leo Breiman. "Using random forest to learn imbalanced data." University of California, Berkeley 110 (2004): 1-12.

[43] M. Kubat, S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," In ICML, vol. 97, pp. 179-186, 1997.

[44] He, Haibo, Yang Bai, Edwardo A. Garcia, and Shutao Li. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," In IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322-1328, 2008.

[45] H. M. Nguyen, E. W. Cooper, K. Kamei, "Borderline over-sampling for imbalanced data classification," International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1), pp.4-21, 2009.

[46] G. Batista, B. Bazzan, M. Monard, "Balancing Training Data for Automated Annotation of Keywords: a Case Study," In WOB, 10-18, 2003.

[47] D. Wilson, Asymptotic, "Properties of Nearest Neighbor Rules Using Edited Data," In IEEE Transactions on Systems, Man, and Cybernetrics, vol. 2 (3), pp. 408-421, 1972.

[48] P. Hart, "The condensed nearest neighbor rule," In Information Theory, IEEE Transactions on, vol. 14(3), pp. 515-516, 1968.

[49] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," Springer Berlin Heidelberg, 2001.

[50] Basha, S.; Vellore Institute of Technology University; Rajput, D.; Vandhan, V, "Impact of Gradient Ascent and Boosting Algorithm in Classification" Int. J. Intell. Eng. Syst. 2018, 11, 41–49.

[51] J. Demšar, "Statistical comparisons of classifiers over multiple datasets", Journal of Machine Learning Research, Vol. 7, pp. 1–30, 2006.

[52] S. García and F. Herrera, "An extension on statistical comparisons of classifiers over multiple datasets for all pairwise comparisons", Journal of Machine Learning Research, Vol. 9, pp. 2677–2694, 2008.

APPENDIX A

TABLE VI. PROPERTIES OF DATA SETS

Sr. No	Data Sets (Imbalance Ratio)	Dimensions	Total Size
1	glass1(1.82)	9	214
2	ecoli-0_vs_1(1.86)	7	220
3	wisconsin(1.86)	9	683
4	pima(1.87)	8	768
5	iris0(2.00)	4	150
6	glass0(2.06)	9	214
7	yeast1(2.46)	8	1484
8	haberman(2.78)	3	306
9	vehicle2(2.88)	18	846
10	vehicle1(2.90)	18	846
11	vehicle3(2.99)	18	846
12	glass-0-1-2-3_vs_4-5-6(3.20)	9	214
13	ecoli1(3.36)	7	336
14	new-thyroid2(5.14)	5	215
15	new-thyroid1(5.14)	5	215
16	ecoli2(5.46)	7	336
17	segment0(6.02)	19	2308
18	glass6(6.38)	9	214
19	yeast3(8.10)	8	1484
20	ecoli3(8.60)	7	336
21	yeast-2_vs_4 (9.08)	8	514
22	yeast-0-5-6-7-9_vs_4 (9.35)	8	528
23	vowel0 (9.98)	13	988
24	glass-0-1-6_vs_2 (10.29)	9	192
25	glass2 (11.59)	9	214

26	shuttle-c0-vs-c4 (13.87)	9	1829
27	yeast-1_vs_7 (14.30)	8	459
28	glass4 (15.46)	9	214
29	ecoli4 (15.80)	7	336
30	abalone9-18 (16.40)	8	731
31	glass-0-1-6_vs_5 (19.44)	9	184
32	shuttle-c2-vs-c4 (20.50)	9	129
33	yeast-1-4-5-8_vs_7 (22.10)	8	693
34	yeast-2_vs_8 (23.10)	8	482
35	yeast4 (25.08)	8	1484
36	yeast-1-2-8-9_vs_7 (30.57)	8	947
37	yeast5 (32.78)	8	1484
38	ecoli-0-1-3-7_vs_2-6 (39.14)	7	281
39	yeast6 (41.40)	8	1484
40	abalone19 (129.44)	8	4174