The Impact of Data Locality on the Performance of Cluster-Based Under-Sampling

Ahmed Shabab Noor, Muhib Al Hasan, Ahmed Rafi Hasan, Rezab Ud Dawla,
Afsana Airin, Akib Zaman, and Dewan Md. Farid

Department of Computer Science and Engineering, United International University,
United City, Madani Avenue, Vatara, Dhaka 1212, Bangladesh,
{anoor193024,mhasan191083,ahasan191131,rdawla191187,aairin191172}@bscse.uiu.ac.bd,
{akib,dewanfarid}@cse.uiu.ac.bd,
https://www.uiu.ac.bd

Abstract. Class-imbalanced classification is one of the most challenging issues in supervised learning. Traditional machine learning classifiers are generally biased toward to the majority class instances and ignore the minority class data. Although a good number of data-sampling techniques have been introduced in the last decade, learning from imbalanced data remains a problematic issue and a major research priority. In this paper, we have studied the impact of cluster locality on various types of imbalanced datasets employing cluster-based under-sampling. Existing under-sampling methods removes a large portion of majority data to make the dataset balanced, so there is high chance that we may lose the informative majority class instances. Cluster-based under-sampling technique addresses this issue and applies ensemble learning to boost-up the classification performance. This paper presents an extensive study on cluster-based under-sampling with ensemble learning using 31 imbalanced datasets to figure out which cluster locality is most important for data sampling.

Keywords: Class-imbalanced classification, Clustering, Data sampling

1 Imbalanced Data Classification

Classifying a class imbalance data in the machine learning field is a major issue [11]. Knowledge mining from imbalanced data has received substantial research attention in the last decade [9]. Many real-life applications face this class-imbalanced problem where the conventional algorithms of machine learning focus more on the majority class data and overlook the minority class data, e.g., cancer detection, credit card fraud detection, and network intrusion detection [5], [13]. For instance, data imbalance in fraud detection can go as low as 1 in 1000 and there have been reports of 1 in 100,000 instances in the real-world application. Conventional classification algorithms give accurate enough predictions for the majority class instances and perform poorly for the minority ones [3]. One of the ways to solve the class-imbalanced problem is dataset sampling technique. Data sampling adds or removes instances from the dataset to make the dataset balanced [12]. Among the sampling techniques, over-sampling and under-sampling are the two types of sampling methods that are used on the training dataset. Over-sampling methods generate artificial instances to make the dataset balanced, but oversampling may provoke the model to overfit as the data tends to become dense and redundant [8]. On the other hand, under-sampling removes large portion of data points from the majority class to make the balanced dataset [7]. As the under-sampling technique removes data, so it is very important to choose the right data. Exception of choosing the right data yields unsatisfactory results.

When it comes to under-sampling the majority class data, there remains a question of how the majority class instances should be removed or selected. Various cluster-based methods exist for under-sampling the imbalanced data, which takes data points from the different cluster locations, such as near the centroid of the cluster, near the border of the cluster, or even random instances selection throughout the cluster. However, all of these methods work differently for different types of datasets. This paper tries to understand which data point selection method from clusters is more informative for different types of datasets. On that note, we divide our datasets into five categories: (1) Continuous, (2) Discrete, (3) Nominal, (4) Continuous + Nominal, and (5) Continuous + Discrete. Then, we consider three types of data selection methods for cluster-based under-sampling: (1) random instances selection, (2) data points nearing the centre of the cluster, and (3) data points nearing the border of the cluster. After applying cluster-based under-sampling technique, we have built three ensemble models on the datasets: (1) Random Forest, (2) Bagging with decision trees, and (3) AdaBoosting algorithm. We have conducted an evaluation study with 31 imbalanced datasets (6 discrete, 9 continuous, 7 nominal, 6 continuous & nominal and 3 continuous & discrete) consisting of variable amount of imbalance ratio. The result demonstrates that random sampling from each cluster of majority samples yields better result than selective sampling from near the centroid or the border of the cluster.

2 Literature Review

Since the majority of machine learning models are biased toward the majority class assumption, it might be challenging to work with imbalanced data in supervised learning today. Consequently, researchers conducted several studies to address the prevailing data imbalance problem using data-sampling techniques, ensemble methods, and cost-sensitive learning methods. LIUBoost is an approach proposed by Sajid et al.[2] designed to handle datasets with class imbalance. LIUBoost applies under-sampling for balancing the dataset yet retaining important information of each instance using the weight update equation of AdaBoost in the form of cost terms. The novelty LIUBoost brings is, it creates an ensemble model which is cost-efficient and it does not suffer from information loss. Arafat et al. [6] introduced another under-sampling technique leveraging SVM (Support

Vector Machine) for making imbalanced datasets balanced by making decision boundaries using cases from the majority and minority classes. To create a balanced dataset, it only takes into account the support vectors for the majority class and those closest to them when combined with cases from the minority class. When there are more instances, this model greatly outperforms other approaches. For handling multi-class, extremely imbalanced data, Arafat et al. [4] have suggested a cluster-based under-sampling technique leveraging the Random Forest algorithm. Here, cluster-based under-sampling is used for selecting the informative majority class instances and considered informative instances close to the center of the cluster and the border of the cluster. In a similar note, Farshid et al. [15] developed Mixing Estimators with Boosting for Imbalanced Data Classification (MEBoost) which is a unique boosting algorithm that mixes two weak classifiers, Decision Tree (DT) and Extra Tree (ET), on the training instances. MEBoost takes advantage of both learners instead of single-base learners.

To categorize unbalanced data, Farshid et al. [14] suggested a novel method called CUSBoost ,which combines the under-sampling and AdaBoost approaches. The imbalanced dataset is split using this approach into two groups: instances belonging to the majority class and instances belonging to the minority class. Using the k-means clustering technique, the majority classes are divided into many groups, and then specific examples of the majority class are picked from each group to create a balanced dataset.CUSBoost is considered as an efficient alternative algorithm to RUSBoost [16] (random under-sampling with AdaBoost) and SMOTEBoost [8] (synthetic minority oversampling with AdaBoost). RUSBoost is a hybrid algorithm which combines boosting and data sampling and SMOTEBoost combines smote algorithms and boosting.

Farid et al. [10] proposed a clustering-based multi-class imbalance data handling method. This proposed method makes multiple balanced datasets from an extensive imbalance dataset by clustering the majority instances into several groups/ clusters and selecting the most informative instances in each cluster where the number of majority and minority class instances are equal. This approach is compared with random under-sampling, random over-sampling, bagging, and boosting. This method outperforms the others since it does not suffer from over-fitting or the loss of potentially helpful information. Ahmed et al. [1] proposed two novel approaches, namely ADASYNBagging and RSYNBagging, for dealing with class imbalance problems. The ADASYNBagging uses an ADASYN-based oversample technique with a bagging algorithm and applies it to minority class instances while keeping the majority class instances untouched. The RSYNBagging uses random under-sampling and ADASYN-based over-sample technique with a bagging algorithm. The proposed methods are compared with existing bagging hybrid approaches, and the performance of the proposed methods is strongly encouraged.

Analysing the prevailing studies, we have found that under-sampling technique is certainly recognised with several success in recent years. Several literatures clustered the majority dataset and selected few instances from each cluster to create an updated majority class. Along with choosing random instances, instances near centroid of the clusters and near border of the clusters are prominent choices to select instances from the clusters. These techniques were widely used on various types of dataset (discrete, continuous, nominal, etc.) and found to be effective. However, the comparison of the performance of these under-sampling techniques in mentioned variants of datasets are one of the under explored areas. Thus, we develop our research framework to explore the performances of there prominent under-sampling techniques (Selection of instances near centroid, random location or near borders) on various types of dataset.

3 Cluster-based Under-sampling

We have collected imbalanced datasets for binary-class classification with various degrees of features, instances and imbalance ratios. We have segmented the dataset based on the characteristics of the features and divide datasets into five specific segments: (1) Discrete, (2) Continuous, (3) Nominal, (4) Continuous and Nominal, and (5) Continuous and Discrete. For example, a dataset that falls into the discrete segment only has features with discrete values. Similarly, a dataset that falls into the continuous and nominal segment has some features with continuous data and some with nominal data. The proposed research methodology is illustrated in Figure 1.

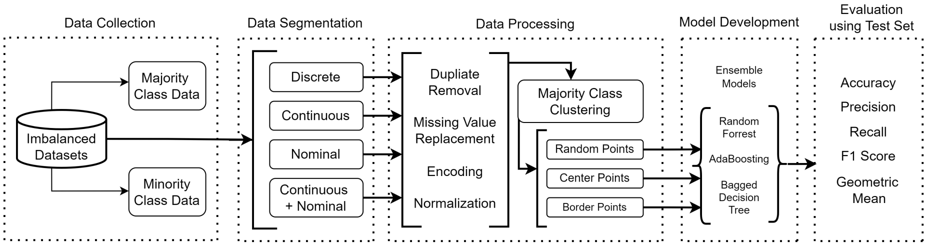


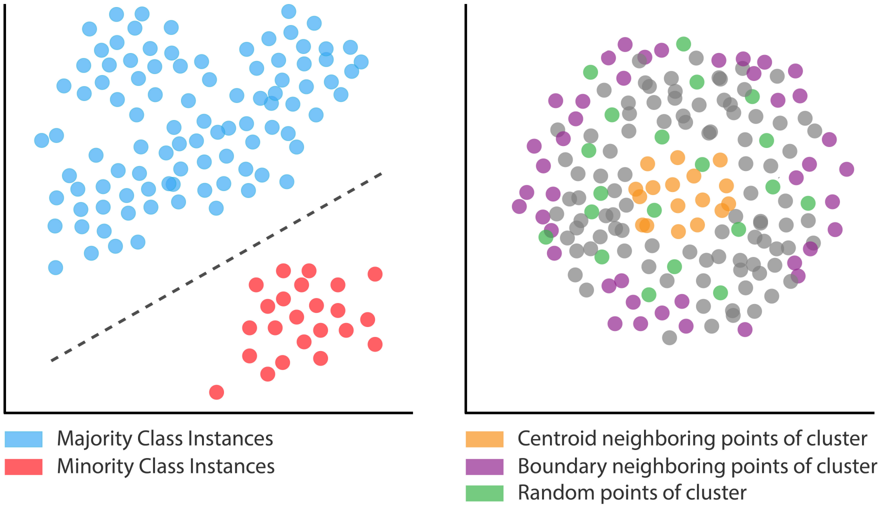
Fig. 1: Research Methodology

We have divided the instances of each dataset into majority class and minority class (Fig. 2a shows a visual representation of how the majority and minority class instances might reside in an imbalanced dataset). We kept the minority class and clustered the majority class using the k-means clustering algorithm. For the number of k, we used the silhouette score to find the optimal number of clusters. Here, $S(i)$ is data point i's silhouette coefficient. $a(i)$ is the average distance from every other point to data point i. And, $c_{min}(i)$ is the minimum average distance from data point i to every cluster except its own. After clustering the majority class instances with the optimal number of clusters, we sampled data points in three ways and discarded the rest. We took data points near the centroids of the clusters, along the boundary, and at random locations across the cluster (Fig. 2b shows a visual representation of these data localities). To take the centre and border data points, we sorted the examples based on their distance from the centroid and then took n examples from the head or tail respectively. Here n is selected in such a way, so that more examples are taken from larger clusters and the total number of examples that are taken are equal to the total number of minority examples.

$$S(i) = \frac{c_{min}(i) - a(i)}{max\{a(i), c_{min}(i)\}} \tag{1}$$

$$n = \frac{cluster\ size}{majority\ examples\ size} \times minority\ examples\ size \tag{2}$$

We took the under-sampled majority instances and all the minority instances as training data. Finally, Random Forest, AdaBoost and Bagged Decision Tree algorithms were used to develop the models and train using the training set.


Majority Class Instances
Minority Class Instances
Centroid neighboring points of cluster
Boundary neighboring points of cluster
Random points of cluster

(a) Majority and minority class

(b) Various types of cluster locality

Fig. 2: Clustering-based Under-sampling

Random Forest (RF) is an ensemble learning algorithm commonly used to solve both classification and regression problems. It creates random decision trees with attribute bagging technique and uses majority vote technique for final prediction. One of the essential characteristics of the Random Forest Algorithm is that it can handle datasets with both continuous and categorical variables, as in regression and classification. AdaBoost (Adaptive Boosting) is a popular boosting method that combines multiple weak classifiers to create a single strong classifier. AdaBoost can be used to improve any machine learning algorithm's performance. It updates the weights of instances based on how they were classified. Weights of misclassified instances are increased, and weights of correctly classified instances are decreased. AdaBoost technique uses majority weighted voting for making final predictions. Bagging, also known as bootstrap aggregation, is the ensemble learning method commonly used to reduce variance within a noisy dataset. Bagging is the process of selecting a random sample of data from a training set with replacement—that is, the individual data points might be picked many times. These weak models are trained individually after multiple data samples have been collected. Depending on the kind of job (regression or classification, for example), the average or majority of such forecasts yields a more accurate estimate.

4 Evaluation

4.1 Datasets

We have collected 31 imbalanced datasets from the KEEL (Knowledge Extraction based on Evolutionary Learning) repository (http://www.keel.es). Table 1 shows the datasets with their category.

4.2 Experimental Setup

The experiments were carried out in Google Colab, or 'Colaboratory', which allows us to write and execute Python 3.10 codes in web browser (https://colab.research.google.com). We have used Python library scikit-learn (https://scikit-learn.org/stable/) for model building. The performance of the ensemble models are tested applying 10-fold cross-validation with the following metrics:

- *F1 Score*: F1 Score unifies Precision and Recall by taking their harmonic mean. It is used to compare different classifiers more generally.

$$F1Score = \frac{precision \cdot recall}{precision + recall} \tag{3}$$

- *G-Mean*: Geometric Mean or G-Mean equals the N-th root of the multiplication of all the values, where N is the number of values.

$$G - Mean = \sqrt[N]{item_1 + item_2 + item_3 + ... + item_N} \tag{4}$$

We have used the Jaccard similarity index to evaluate the difference of the undersampling techniques being tested.

- *Jaccard Similarity Index*: Jaccard Similarity Index or Jaccard Score shows us the similarity between two sets. Jaccard score is equal to 1 if the two sets are identical and 0 if they are mutually exclusive. Equation 5 shows the Jaccard similarity index of two sets A and B.

$$J(A, B) = \frac{A \cap B}{A \cup B} \tag{5}$$

Table 1: Imbalanced datasets collected from KEEL data repository.

Discrete	dermatology-6
	poker 8 vs. 6
	poker 8-9 vs. 5
	shuttle 2 vs. 5
	shuttle c0 vs. c4
	vehicle0
Continuous	led7digit-0-2-4-5-6-7-8-9 vs. 1
	segment0
	winequality-red-4
	winequality-white-3-9 vs. 5
	yeast-0-2-5-6 vs. 3-7-8-9
	yeast-0-2-5-7-9 vs. 3-6-8
	yeast4
	yeast5
	yeast6
Nominal	car-good
	car-vgood
	flare-F
	kr-vs-k-three vs. eleven
	kr-vs-k-zero vs. eight
	kr-vs-k-zero vs. fifteen
	kr-vs-k-zero-one vs. draw
Continuous + Nominal	abalone 20 vs. 8-9-10
	abalone19
	kddcup-buffer_overflow vs. back
	kddcup-guess_passwd vs. satan
	kddcup-land vs. satan
	kddcup-rootkit-imap vs. back
Continuous + Discrete	page-blocks 1-3 vs. 4
	page-blocks0
	vowel0

4.3 Results

From the experiment, We found various results regarding the impact of clustering locality while performing cluster-based under-sampling on the ensemble models. The summary of the results is highlighted in table 3. In case of discrete data, random sampling techniques performs better than centre or border instances with an average f1-score of 0.82 and a g-mean of 0.882. Bagging of Decision tree contributed the most having a f1-score of 0.88. On a similar note, random sampling performs better on continuous, having an average f1-score of 0.756 and a g-mean of 0.805. AdaBoost performs well here for the f1-score and g-mean. Random sampling also performs well for the nominal data, with a f1-score of 0.87 and a g-mean of 0.919. In nominal data, there is no highest contributor. All models perform randomly here. If the dataset is combined with continuous and nominal values, random under-sampling gives a higher f1-score of having 0.905

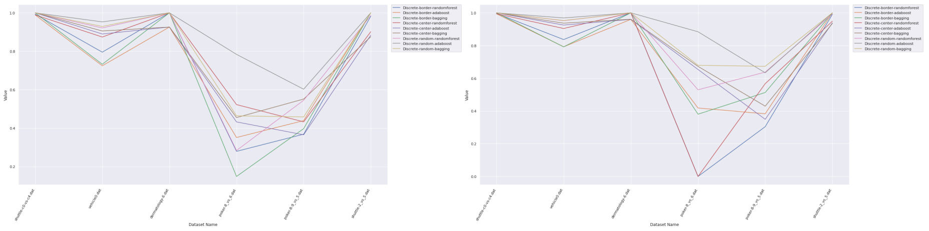
and centre under-sampling gives a higher g-mean of 0.886. Random sampling again performs well if the dataset is combined with discrete and continuous values. The f1-score is 0.953 and the g-mean is 0.969. In summary (see figure 8a and Fig. 8b), Random under-sampling performs well in most datasets as it takes data points randomly from the whole bunch of the cluster acquiring the necessary diverse characteristics of all instances. On Contrary, taking instances near to the centroid and border demonstrates a lower performance metrics scores as it preserves a specific zone characteristics rather than the full set of instances.

Fig. 3, 4, 5, 6, and 7 show the F1 score and G-Mean of the segments Discrete, Continuous, Nominal, Continuous+Discrete, and Continuous+Nominal respectively. Here the individual F1 score and G Mean score for all datasets can be seen in the aforementioned figures.

In table 2, we show the difference of the data samples between different clustering methods using Jaccard Similarity Index. This indicates how the methods we are comparing are actually different from each other and not the same. We see that all the scores are below 0.5 meaning that the techniques are less than 50% similar. We have calculated this score for each dataset and averaged them according to their segment. We see Jaccard similarity index as low as 0.16 and no greater than 0.46 for the chosen datasets and undersampling techniques.

Table 2: Jaccard Score for different types of clustering methods.

Segmentation	Center vs. Border	Border vs. Random	Random vs. Center
Discrete	0.35	0.38	0.39
Continuous	0.16	0.19	0.2
Nominal	0.43	0.45	0.46
Continuous + Nominal	0.46	0.46	0.46
Continuous + Discrete	0.4	0.44	0.42



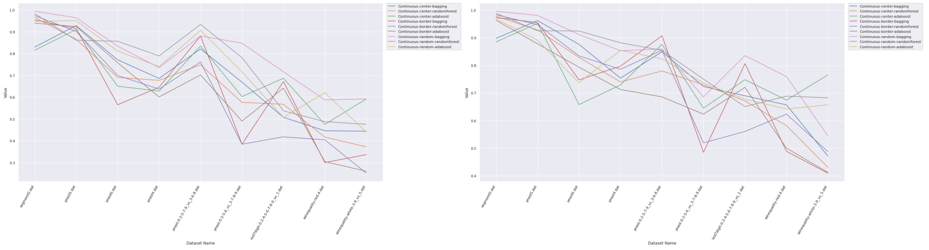
(a) F1 Score

(b) G Mean

Fig. 3: Discrete datasets

Table 3: Experiment results using cluster-based under-sampling techniques.

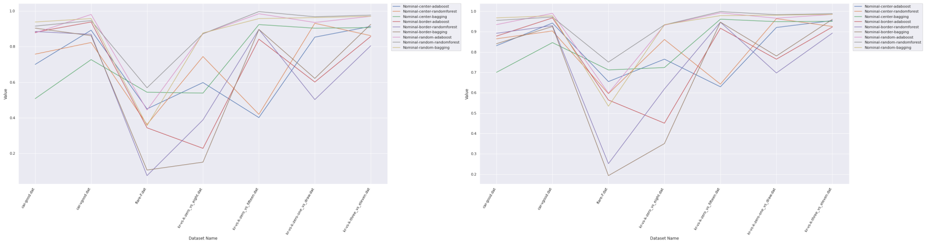
Type of Data	Cluster Based Under-sampling Technique	Models	F1 Score		Gmean	
			Unit	Average	Unit	Average
Discrete	Random	Random Forest	0.788	**0.82**	0.851	**.882**
		AdaBoost	**0.88**		**0.913**	
		Bagging	0.8		0.883	
	Center	Random Forest	0.785	0.774	**0.823**	0.817
		AdaBoost	0.75		0.805	
		Bagging	**0.787**		**0.823**	
	Border	Random Forest	0.736	0.749	0.688	0.713
		AdaBoost	**0.798**		**0.738**	
		Bagging	0.713		0.713	
Continuous	Random	Random Forest	0.74	**0.756**	0.815	**0.805**
		AdaBoost	0.738		0.777	
		Bagging	**0.79**		**0.825**	
	Center	Random Forest	0.655	0.671	0.741	0.759
		AdaBoost	**0.688**		**0.772**	
		Bagging	0.67		0.764	
	Border	Random Forest	0.602	0.615	**0.733**	0.72
		AdaBoost	**0.623**		0.697	
		Bagging	0.62		0.732	
Nominal	Random	Random Forest	**0.89**	**0.87**	**0.938**	**0.919**
		AdaBoost	0.86		0.912	
		Bagging	0.86		0.907	
	Center	Random Forest	0.69	0.698	0.821	0.822
		AdaBoost	0.685		0.811	
		Bagging	**0.72**		**0.834**	
	Border	Random Forest	0.63	0.643	0.747	0.747
		AdaBoost	**0.67**		**0.78**	
		Bagging	0.63		0.714	
Continuous + Nominal	Random	Random Forest	0.91	**0.905**	0.876	0.88
		AdaBoost	**0.93**		**0.898**	
		Bagging	0.875		0.868	
	Center	Random Forest	0.82	0.833	0.866	**0.886**
		AdaBoost	0.82		0.896	
		Bagging	**0.86**		**0.898**	
	Border	Random Forest	**0.67**	0.656	0.735	0.727
		AdaBoost	0.63		0.705	
		Bagging	**0.67**		**0.743**	
Continuous + Discrete	Random	Random Forest	0.933	**0.953**	0.956	**0.969**
		AdaBoost	**0.963**		0.973	
		Bagging	**0.963**		**0.98**	
	Center	Random Forest	0.74	0.741	0.836	0.843
		AdaBoost	0.72		0.83	
		Bagging	**0.763**		**0.863**	
	Border	Random Forest	0.713	0.759	0.806	0.838
		AdaBoost	**0.79**		0.853	
		Bagging	0.776		**0.856**	



(a) F1 Score

(b) G Mean

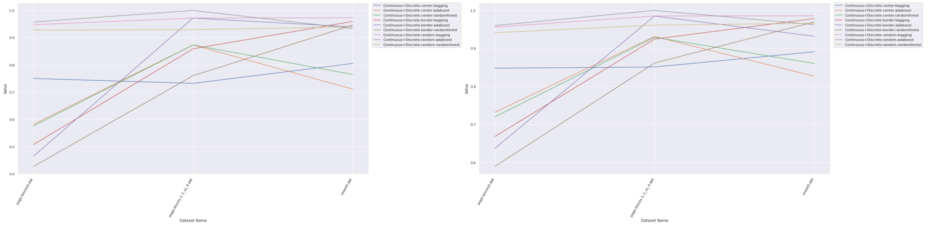
Fig. 4: Continuous datasets



(a) F1 Score

(b) G Mean

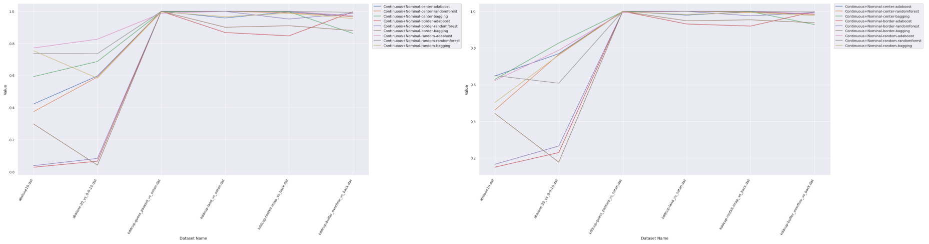
Fig. 5: Nominal datasets



(a) F1 Score

(b) G Mean

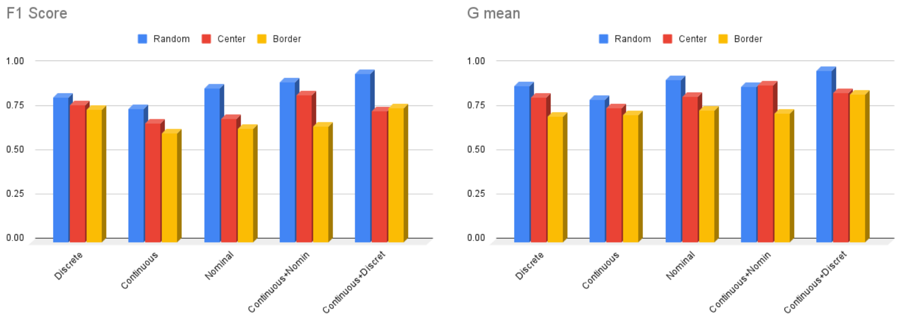
Fig. 6: Continuous and Discrete datasets



(a) F1 Score

(b) G Mean

Fig. 7: Continuous and Nominal datasets



(a) F1 Score

(b) G Mean

Fig. 8: Comparative performance analysis

5 Conclusion

Most of the machine learning algorithms do not give attention to the minority class instances whereas focus on majority class instances for classifying imbalanced data. It is a challenging task to create a balanced dataset and classify correctly both the majority and minority class instances. For dealing with class imbalance difficulties, combining sampling technique with ensemble classifiers became very popular nowadays. This paper presented a comparative study of cluster locality in imbalanced classification using cluster-based under-sampling and ensemble learning algorithms. We used the cluster-based under-sampling technique to select the majority class instances in three different ways. We applied three different ensemble learning techniques in our experiment to find the pattern and behaviours of different kinds of datasets. Experimental results show that the random selection of instances from clusters to make balanced datasets performs better in most datasets. In future, we will apply different clustering algorithms and assign weights to the instances to select majority class instances.

References

1. Ahmed, S., Mahbub, A., Rayhan, F., Jani, M.R., Shatabda, S., Farid, D.M.: Hybrid methods for class imbalance learning employing bagging with sampling techniques. In: 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS). pp. 126–131. Bengaluru, India (December 2017)
2. Ahmed, S., Rayhan, F., Mahbub, A., Jani, M.R., Shatabda, S., Farid, D.M.: LIUBoost: Locality informed under-boosting for imbalanced data classification. In: International Conference on Emerging Technology in Data Mining and Information Security (IEMIS). pp. 1–12. Kolkata, India (February 2018)
3. Ahmed, S., Rayhan, F., Mahbub, A., Jani, M.R., Shatabda, S., Farid, D.M.: Liuboost: Locality informed under-boosting for imbalanced data classification. pp. 133–144. Springer, Singapore, Singapore (2019)

4. Arafat, M.Y., Hoque, S., Farid, D.M.: Cluster-based under-sampling with random forest for multi-class imbalanced classification. In: 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), and IEEE Xplore Digital Archive. pp. 1–6. Colombo, Sri Lanka (December 2017)
5. Arafat, M.Y., Hoque, S., Xu, S., Farid, D.M.: Machine learning for mining imbalanced data. IAENG International Journal of Computer Science **46**(2), 332–348 (May 2019)
6. Arafat, M.Y., Hoque, S., Xu, S., Farid, D.M.: An under-sampling method with support vectors in multi-class imbalanced data classification. In: 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA). pp. 1–6. Island of UKULHAS, Maldives (August 2019)
7. Arafat, M.Y., Hoquef, S., Xuf, S., Farid, D.M.: Advanced data balancing method with svm decision boundary and bagging. In: IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). pp. 1–7. IEEE, Melbourne, Australia (December 2019)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
9. Farid, D.M., Nowé, A., Manderick, B.: Ensemble of trees for classifying high-dimensional imbalanced genomic data. In: SAI Intelligent Systems Conference (IntelliSys). pp. 115–122. London, UK (September 2016)
10. Farid, D.M., Nowé, A., Manderick, B.: A new data balancing method for classifying multi-class imbalanced genomic data. In: 25th Belgian-Dutch Conference on Machine Learning (Benelearn). pp. 1–2. Kortrijk, Belgium (September 2016)
11. Farid, D.M., Shatabda, S., Abedin, M.Z., Islam, M.T., Hossain, M.I.: Mining imbalanced big data with julia. In: JuliaCon. University of Maryland Baltimore (UMB), Baltimore, MD, USA (July 2019)
12. Hoque, S., Arafat, M.Y., Farid, D.M.: Machine learning for mining imbalanced data. In: International Conference on Emerging Technology in Data Mining and Information Security (IEMIS). pp. 1–10. Kolkata, India (February 2018)
13. Miah, M.O., Khan, S.S., Shatabda, S., Farid, D.M.: Improving detection accuracy for imbalanced network intrusion classification using cluster-based under-sampling with random forests. In: International Conference on Advances in Science, Engineering & Robotics Technology (ICASERT), and IEEE Xplore Digital Archive. pp. 1–5. Dhaka, Bangladesh (May 2019)
14. Rayhan, F., Ahmed, S., Mahbub, A., Jani, M.R., Shatabda, S., Farid, D.M.: CUSBoost: Cluster-based under-sampling with boosting for imbalanced classification. In: 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS). pp. 70–75. Bengaluru, India (December 2017)
15. Rayhan, F., Ahmed, S., Mahbub, A., Jani, M.R., Shatabda, S., Farid, D.M., Rahman, C.M.: MEBoost: Mixing estimators with boosting for imbalanced data classification. In: 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), and IEEE Xplore Digital Archive. pp. 1–6. Colombo, Sri Lanka (December 2017)
16. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: Rusboost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **40**(1), 185–197 (2009)