A Novel Method for Imbalanced Data Classification based on Label Reassignment

Ahmed Shabab Noor, Afsana Airin, Rezab Ud Dawla, Ahmed Rafi Hasan, Muhib Al Hasan,
Akib Zaman and Dewan Md. Farid*
Department of Computer Science & Engineering, United International University
United City, Madani Avenue, Badda, Dhaka 1212, Bangladesh
Email: {akib, dewanfarid}@cse.uiu.ac.bd

***Abstract*—Imbalanced data classification is one of the most challenging supervised learning problems in machine learning. Several data-sampling methods have been proposed in the last decade employing over-sampling and under-sampling techniques. Over-sampling technique engenders artificial minority-class instances that creates overfitting problems. However, under-sampling techniques suffer from informative information lost by removing major-part of majority-class instances. In this paper, we have proposed a novel approach for imbalanced data classification using label reassignment of misclassified majority-class instances. The idea is to consider the misclassified majority-class instances as minority-class instances to make the dataset balanced. We conceptualise the reassignment of majority-class instances in overlapping region to minority-class instances in Label Reassignment. We have conducted experiment on 11 imbalanced datasets and compared the performance of proposed method with well-known Synthetic Minority Oversampling Technique (SMOTE) method. The proposed method outperforms SMOTE in most of the experimented datasets with a significant improvement in terms of Recall and F1 Score.**

***Index Terms*—Imbalanced Classification; Imbalanced Data; Over-Sampling; Under-Sampling; SMOTE;**

I. Introduction

Class imbalance is one of the most challenging learning problems in the field of data classification in supervised machine learning. Class imbalance refers to the context where the number of data points/instances of one class is far less than that of another class/es [1]. Most of the real-world datasets are high dimensional, multi-class, and highly imbalanced and it is difficult to classify the real-world imbalanced data with traditional machine learning algorithms [2]. Many real-world problems include anomaly detection, face recognition, credit card fraud detection, and medical diagnosis where the ratio of data-classes are imbalanced. To address this class imbalanced classification a good number of data-balancing methods have been proposed over the past few decades [3]. The cost-sensitive methods and ensemble-learning are also widely applied in imbalanced data classification. The data sampling technique is basically added or removed data-sample to balance an imbalanced dataset. Removing majority-class instances randomly from the dataset is called the under-sampling technique while over-sampling method generates artificial data and adds it to the minority class instances to balance a dataset. Cost-sensitive learning for imbalanced classification is mainly assigning costs to the various kinds of classification errors that might occur and then employing specific techniques to take those costs into account. Ensemble learning initially ignores the minority-class instances in its initial iterations, but gradually gives more propriety to the minority-class instances as its weights became high e.g. AdaBoost (Adaptive Boosting) algorithm [4].

In imbalanced datasets, the majority-class contains less informative instances. On the other hand, the minority-class includes the most informative instances. But the conventional machine learning algorithms are biased toward the majority-class instances and ignore the minority-class instances. As a matter of fact, the prediction accuracy of classifiers gets influenced due to ignoring minority-class instances. Single model classifiers e.g. naïve Bayes classifier, Decision Tree induction, are created to improve the performance of classification appropriateness where the majority-class data samples are frequently correctly classified but the minority-class data samples are underestimated. Researchers came up with an ensemble method that performs better than these single model classifiers [5], [6], [7], [8]. Ensemble methods apply sampling techniques in its each iteration. In this study, we have proposed a novel algorithm named Label Reassignment (LR) of minority-class instances to balance imbalanced datasets and demonstrate the strength of the proposed algorithm compared with the recognized Synthetic Minority Over-sampling Technique (SMOTE). The proposed algorithm finds the over-lapping data points between the majority and minority class data and reassigns them to the minority class to increase the size of minority-class instances and thereby reduce the imbalance ratio. We have performed experiments on 11 imbalanced datasets using the proposed algorithm and SMOTE to balance the datasets and train on a base decision tree classifier. The result shows that the proposed method outperforms SMOTE in most cases on the selected datasets with an average increase of 4.98% in recall and 3.77% in F1 score.

II. Literature Review

Class imbalance problem is one of the oldest and most complex challenge to overcome for the supervised learning classifiers. Extensive studies were conducted on the class

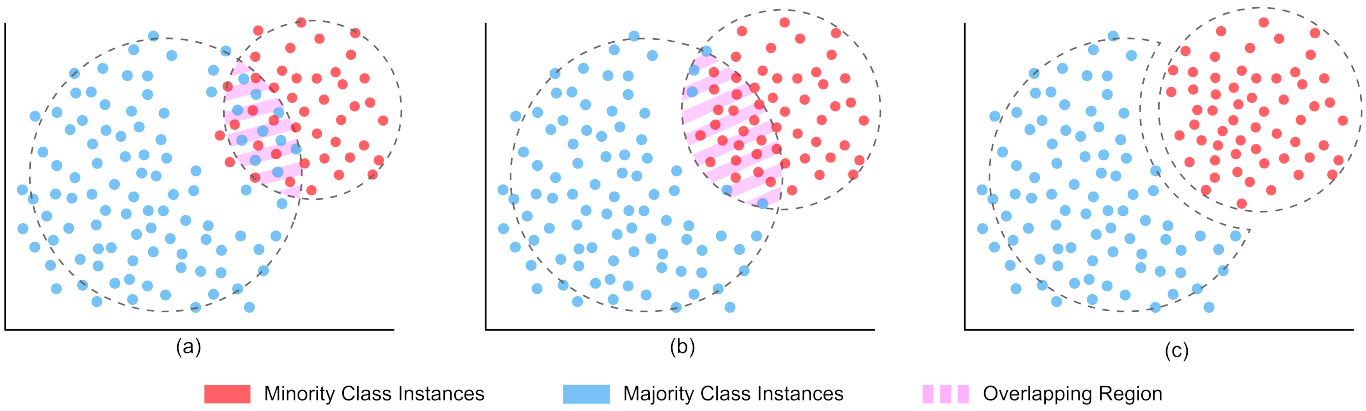


Fig. 1: Visualisation of proposed Label Reassignment (LR) method.

imbalance problem in the last decade. Farid et al. [9] presented a novel approach for handling multi class imbalance data. It's a clustering-based method where the majority-class instances are divided into several clusters. Then, the most informative majority-class instances are selected from each divided cluster. Using these clusters with the most informative instances, various balanced datasets are produced with an equal number of instances from minority and majority classes. In this method, the balanced data sets are trained by multiple classifiers to combine the new data instances for prediction. The worthiness of the proposed method is that it does not suffer from over-fitting and losing potential information. Arafat et al. [10] introduced an under-sampling method with support vector for multiclass imbalance data. This method has considered SVM decision boundary to create a decision line for separating majority and minority class instances and only selects majority class support vectors and its nearest neighbours. In the end, it combines the selected instances with the minority class instances to make a balanced dataset that produces high accuracy compared to the existing methods. Sajid et al. [11] proposed LIUBoost algorithm for dealing with class imbalance datasets. Using the weight update equation of AdaBoost, LIUBoost utilizes under-sampling to balance the dataset while maintaining the essential data for each instance in the form of cost terms.

MEBoost [12] uses two different estimators instead of a single estimator, unlike state-of-the-art models. The Decision Tree classifier and Extra Tree classifier are the estimators employed here. The base estimator changes after each iteration. It excludes estimators according to their performance, which means that estimators with bad performance are discarded. This algorithm does not offer any sampling technique from the training datasets. Instead, samplings are done by the estimators where the Decision tree uses information entropy, and the Extra Tree uses randomized tree classification. Thus, MEBoost gathers the benefits from these two estimators. Sajid et al. [13] proposed two methods called ADASYNBagging and RSYNBagging based on ADASYN and bagging algorithms. Inspired by SMOTEBagging algorithm, ADASYNBagging creates an ensemble model with C4.5 as base classifier. To train each base classifier, a bootstrap sample is created from the majority examples and the minority examples are oversampled using ADASYN algorithm. Then the bootstrapped majority data and the oversamples minority data are merged to create a training set. On the other hand, RSYNBagging uses a hybrid approach to fix class imbalance. Like ADASYNBagging, RSYNBagging also creates bootstrap samples from the majority class at first, but for half of the base classifiers, it randomly under-samples the bootstrap majority sample while for the other half it oversamples the minority sample using ADASYN.

Farshid et al. [14] proposed a novel cluster-based method named CUSBoost, which used cluster-based under-sampling. In CUSBoost, the majority instances are clustered using K-Means Clustering, and for each cluster, half of the instances are removed using random under-sampling. The under-sampled majority set is then merged with the minority instances to create the training set for each base classifier. Each classifier also assigns a weight to each training instance with a higher weight on the misclassified ones. To use this ensemble model, authors assign a weight to each classifier based on its error rate and add the weight to the class it predicts. The final prediction is the class with the most significant weight. Arafat et al. [15] also proposed a cluster-based under-sampling method with a Random Forest algorithm for dealing with multi-class, highly imbalanced data. Here, informative majority class examples are chosen using cluster-based under-sampling, and informative instances near the cluster's centre and border are also taken into consideration.

III. Methodology

In this section, we discuss current techniques for balancing imbalanced data and the proposed method for balancing binary class imbalanced dataset.

A. Sampling Technique

Sampling is a technique where we take specific amounts of samples from the dataset to create a balanced dataset. It is divided into two broad categories: under-sampling and over-sampling.

1) Under-sampling: In under-sampling, we eliminate samples from majority class and choose the rest to create a balanced dataset. The problem with this method is that we may lose valuable information as we discard some data and consider the rest.

2) Over-sampling: We artificially generate more data from the minority instances in over-sampling to create a balanced dataset. However, this method has the risk of over-fitting the data as this method tries to replicate data from the original dataset. SMOTE (Synthetic Minority Oversampling Technique) is a popular oversampling method that creates synthetic instances instead of picking random instances from the minority instances with replacement [8].

B. Cost-sensitive Learning

In cost-sensitive learning, we assign a weight to the classifications of the instances. Wrongly classified Minority class instances are given a more significant weight so that a cost-optimising algorithm can be used to reduce the cost and adequately classify the instances.

C. Ensemble Learning

In ensemble learning, multiple classifiers are combined into one single strong classifier that utilises the predictions of the to give a potentially better outcome. The ensemble model uses sampling techniques on each iteration to handle the imbalanced dataset.

D. Proposed Method

In the proposed method, Label Reassignment (LR) we reassign the labels of majority class instances in the overlapping region to minority-class to reduce the imbalance ratio. The algorithms targets the data points in the overlapping region and assigns them to the minority-class, but they originally belong to the majority class. When the learning model wrongly classifies any data points, it is likely that those points are very close to the overlapping region. That is why we change the labels of these specific data points in order to decrease the number of majority-class instances and increase the number of minority-class instances. Initially, we apply data pre-processing techniques such as duplicate removal, missing value replacement, encoding and normalization. The data samples are then divided into training (D_train), validation(D_val), and test(D_test) set all of which are imbalanced in nature. We have calculated the initial imbalanced ratio of the combination of training and validation dataset. We have used the Decision Tree (DT) algorithm (C4.5) on the training data to train a classifier and utilize the validation dataset to predict the accuracy of the classifier. While evaluating on the validation set, if any instance of the majority class is improperly classified, we have reassigned that instance of the majority-class to the minority-class. In summary, the *False Positives* (majority-class instance) are changed to *Positive* (minority-class instance). After reassigning the labels, we have combined the training set and the modified validation set to create a new training set (D'_{train}) with reduced imbalanced ratio I'_r. We have continued this iteration until the I'_r reache a saturation point i.e. the imbalanced ratio no more changes for previous 50 iterations. After reaching to the saturation point, we got the new training dataset with reduced imbalanced ratio I'_r. We have utilised the new training dataset to train another classifier based on decision tree algorithm. We have used the test dataset to evaluate the performance of the proposed method. Algorithm 1 illustrates the proposed method.

Algorithm 1 Proposed Method

Input: Training data D_{train}, Validation data D_{val}, Learning Model (C4.5 classifier)
Output: Balanced data D'_{train}, Final imbalanced ratio I'_r
Method:

$preds \leftarrow classifier(D_{val})$
for $i \leftarrow 1$ **to** $len(preds)$ **do**
  if $label(i) \neq preds(i)$ **then**
    if $label(i) = majority\ class$ **then**
      $label(i) \leftarrow minority\ class$
    end if
  end if
end for
$D'_{train} \leftarrow D_{train} + D_{val}$
Calculate I'_r
Continue updating the D'_{train} unless I'_r reaches $saturation$

IV. Evaluation

A. Dataset

We have chosen 11 datasets for this study, all of which are binary in nature. All of these datasets have been sourced from KEEL (Knowledge Extraction based on Evolutionary Learning) (http://www.keel.es). KEEL is an open source Java software tool that can be used for a large number of different knowledge data discovery tasks. Table I summarises the datasets chosen and their characteristics, such as number of attributes, instances and imbalance ratio. Among the evaluated datasets, we see three cases of how the majority and minority class instances are scattered. They are as follows:

- *Case 1.* minority and majority classes having an exclusive overlapping region.
- *Case 2.* the two classes having no overlapping region at all with clear boundary.
- *Case 3.* the minority class region being completely overlapped by the majority class region.

Name	Case	Attributes	Instances	IR
page-blocks0	1	10	5472	8.79
abalone19	1	8	4174	129.44
vowel0	1	13	988	9.98
car-vgood	1	6	1728	25.58
yeast6	1	8	1484	41.4
flare-F	1	11	1066	23.79
poker-8-9 vs. 5	1	10	2075	82
winequality white-3-9 vs. 5	1	11	1482	58.28
kr-vs-k-zero vs. fifteen	2	6	2193	80.22
shuttle-2 vs. 5	2	9	3316	66.67
segment0	3	19	2308	6.02

TABLE I: Selected datasets and their properties. (IR = Imbalance Ratio)

B. Experimental Setup

The experiments were conducted in Google Colab, also known as 'Colaboratory' (https://colab.research.google.com), which allows us to compose and execute Python code on the web. Python 3.10 was used to implement the proposed algorithm. For model construction, the widely used Python library SciKit Learn (https://scikit-learn.org) was used. We perform the imbalanced to balanced conversion of the dataset using the proposed LR method and SMOTE on each dataset and train them on same model based on decision tree algorithm. We have evaluated the performance of the methods using the following parameters:

- *Accuracy*: This metric shows the proportion between the total number of accurate predictions and the entire sample size. In other words, it reveals the accuracy of our forecasts. Equation 1 calculates the accuracy.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- *Precision*: By measuring precision, we may calculate the proportion of accurately anticipated positive samples to all positive samples. In essence, it informs us how many of all the samples that were expected to be positive are in fact positive. Equation 2 outlines the formula for calculating precision.

$$precision = \frac{TP}{TP + FP} \tag{2}$$

- *Recall*: The rate of accurate classification just within the positive samples is known as recall. False Negatives are more destructive than False Positives, and the Recall score tells us just that, thus it is crucial to understand how many of the positive samples are correctly identified.

$$recall = \frac{TP}{TP + FN} \tag{3}$$

- *F1 Score*: By calculating the harmonic mean of Precision and Recall, F1 Score combines the two. This metric is used to obtain a general comparison between different classifiers.

$$F1\ Score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{4}$$

- *G-Mean*: It is the geometric mean of Sensitivity and Specificity score. Sensitivity is also known as Recall. Specificity is complimentary to sensitivity as it tells us how well negative instances were predicted. G-mean combines these two metrics to give a balanced result.

$$specificity = \frac{TN}{TN + FP} \tag{5}$$

$$G - mean = \sqrt{recall \times specificity} \tag{6}$$

C. Results

Table II illustrates the comparative performance of the proposed method and conventional technique SMOTE on datasets using decision tree (C4.5) classifier. The proposed method has a better recall score in most cases and mixed performance when it comes to precision score. The proposed method outperforms SMOTE in 6 out of 11 datasets and matches with SMOTE in rest 5 datasets in recall score (Figure 3a). On the other hand, the proposed method outperforms SMOTE in 7 datasets, matches with 1 dataset and is outperformed in rest 3 datasets in precision score (Figure 3b). On average, LR demonstrates 4.983% better performance in case of recall and 3.285% in case of precision. The reduction of false negative instances during the process of balancing the dataset explains this phenomenon of better recall score. we have both minority and majority instances, i.e., positive and negative instances, and we turn all the instances in the overlapping region to be positive. However, in the test set, we have instances belonging to both the overlapping and non-overlapping regions for both positive and negative classes. Since the overlapping region's instances are now all reassigned as positive, the number of false negatives is reduced, and therefore the proposed method achieves a better recall score for most datasets. Conversely, the number of false positives increases because of the change in the overlapping region, resulting in a precision score to be less than SMOTE in some cases. Nevertheless, the scores are very close to SMOTE when the proposed method fails to outperform on the precision score with an average of 2.87%. Due to the better recall score and mixed precision score, the proposed method has a better F1 score and G-mean score than SMOTE in most cases (10 out of 11 datasets) (Figure 3c).

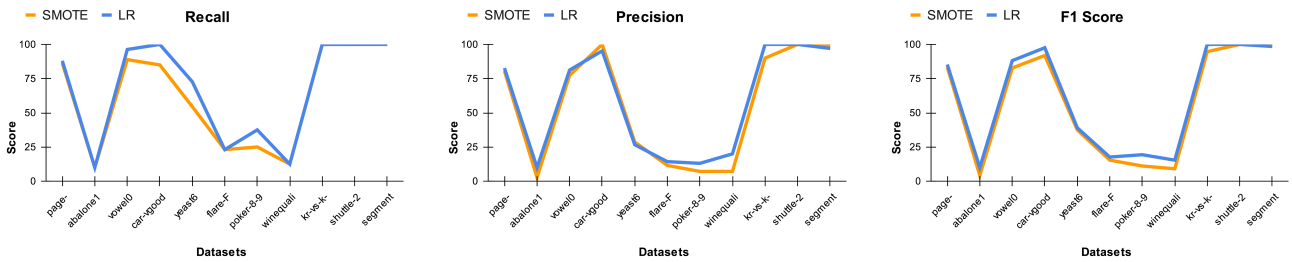
In Case 1, where majority and minority class instances have an exclusive overlapping region (see Figure 2a), the proposed method outperforms SMOTE in all datasets. The proposed method significantly affect the overlapping region to convert the majority-class instances into minority-class instances. This systematic conversion of borderline majority-class instances to

(a) Case 1 (b) Case 2 (c) Case 3

Fig. 2: Visualisation of various cases of the datasets.

Dataset Name	**Metrics**									
	Accuracy		Recall		Precision		F1 Score		G-mean	
	LR	SMOTE	LR	SMOTE	LR	SMOTE	LR	SMOTE	LR	SMOTE
page-blocks0	**96.89**	96.47	**88.01**	86.3	**82.68**	80.56	**85.3**	83.33	**9.29**	9.18
abalone19	**98.56**	96.4	**10**	**10**	**10**	2.7	**10**	4.2	**3.15**	3.11
vowel0	**97.64**	96.63	**96.3**	88.89	**81.25**	77.41	**88.14**	82.75	**9.7**	9.3
car-vgood	**99.81**	99.42	**100**	85	95.24	**100**	**97.56**	91.89	**9.99**	9.22
yeast6	94.39	**95.51**	**72.73**	54.54	26.67	**28.6**	**39.02**	37.5	**8.31**	7.26
flare-F	**91.25**	89.68	**23.08**	23.07	**14.29**	11.54	**17.65**	15.38	**4.66**	4.61
poker-8-9 vs. 5	**95.99**	94.86	**37.5**	25	**13.04**	7.14	**19.35**	11.11	**6.02**	4.89
winequality white-3-9 vs. 5	**97.53**	95.51	**12.5**	**12.5**	**20**	7.14	**15.38**	9.09	**3.52**	3.48
kr-vs-k-zero vs. fifteen	**100**	99.85	**100**	**100**	**100**	90	**100**	94.73	**10**	9.99
shuttle-2 vs. 5	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**10**	**10**
segment0	99.57	**99.85**	**100**	**100**	97.06	**99**	98.5	**99.49**	9.97	**9.99**

TABLE II: Proposed method performance compared to SMOTE on datasets and metrics on C4.5 classifier.



(a) Recall (b) Precision (c) F1 Score

Fig. 3: Performance of the proposed Label Reassignment (LR) method in comparison to SMOTE.

minority-class instance proves to be more effective than the generation of linearly interpolated minority instances generation process of SMOTE.

In Case 2, where there is no overlapping region between majority and minority class instances with a clear boundary (See Figure 2b), the proposed method does not have any positive or negative effect. For those datasets, we can see that the base classifier can achieve 100% accuracy without applying any sampling techniques. However, when it comes to SMOTE, it can sometimes have a negative effect on this type of datasets. In table II we see that for the dataset 'kr-vs-k-zero vs. fifteen', SMOTE fails to classify all instances correctly, whereas the proposed method does not. This is because this dataset has a clear boundary between the minority and majority class instances; thus, LR does not affect it. So, the base classifier can classify all the instances correctly. However, since SMOTE artificially increases the minority class instances, it creates some instances that either overlap with the majority class instances or are very close to them, which results in worse performance than the base classifier. Nevertheless, applying SMOTE for some datasets, we can also get 100% accurate prediction, which is also true for LR. This can be seen in table II for 'shuttle-2 vs. 5' dataset.

In Case 3, (See Figure 2c), where the majority class instances completely overlap the minority class instances, we see that LR fails to outperform SMOTE. This happens because there are too many majority-class instances in the overlapping region, which completely engulfs the minority class instances. Here the classifier does not wrongly classify the majority class instances a lot during the validation set phase, resulting in an insignificant reduction of the imbalance ratio. Thus, LR fails to outperform SMOTE but comes very close to it. This scenario can be seen in table II for the 'segment0' dataset, which is originally an image segmentation dataset. Here, SMOTE outperforms LR, but LR comes very close to SMOTE.

V. Conclusion

In this paper, we have proposed a new approach to deal with class imbalanced classification. To make the dataset balanced we have considered misclassified majority-class training examples as minority-class instance. In the experiments, we have applied C4.5 classifier as a baseline model. The C4.5 is an extension of Quinlan's earlier ID3 algorithm that can be used for classification task. We have compared the performance of proposed method with SMOTE technique. SMOTE is one of the well-known over-sampling methods. The proposed method outperforms the conventional SMOTE method in most of the cases and produce a better performance. Most importantly, the proposed method demonstrates a significant improvement in terms of predicting the minority-class instances since wrong minority-class prediction can lead to substantial damage in most of the real-life scenarios such as: Cancer detection, Fraud detection, etc. In future, we will apply AdaBoost technique with this proposed concept.

References

[1] M. Y. Arafat, S. Hoque, S. Xu, and D. M. Farid, "Machine learning for mining imbalanced data," *IAENG International Journal of Computer Science*, vol. 46, no. 2, pp. 332–348, May 2019.

[2] D. M. Farid, S. Shatabda, M. Z. Abedin, M. T. Islam, and M. I. Hossain, "Mining imbalanced big data with julia," in *JuliaCon*, University of Maryland Baltimore (UMB), Baltimore, MD, USA, July 2019.

[3] S. Hoque, M. Y. Arafat, and D. M. Farid, "Machine learning for mining imbalanced data," in *International Conference on Emerging Technology in Data Mining and Information Security (IEMIS)*, Kolkata, India, February 2018, pp. 1–10.

[4] D. M. Farid, A. Nowé, and B. Manderick, "Ensemble of trees for classifying high-dimensional imbalanced genomic data," in *SAI Intelligent Systems Conference (IntelliSys)*, London, UK, September 2016, pp. 115–122.

[5] M. O. Miah, S. S. Khan, S. Shatabda, and D. M. Farid, "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*, Dhaka, Bangladesh, May 2019, pp. 1–5.

[6] S. Ahmed, F. Rayhan, A. Mahbub, M. R. Jani, S. Shatabda, and D. M. Farid, "Liuboost: Locality informed under-boosting for imbalanced data classification," A. Abraham, P. Dutta, J. K. Mandal, A. Bhattacharya, and S. Dutta, Eds. Singapore: Springer, Singapore, 2019, pp. 133–144.

[7] M. Y. Arafat, S. Hoquef, S. Xuf, and D. M. Farid, "Advanced data balancing method with svm decision boundary and bagging," in *IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. Melbourne, Australia: IEEE, December 2019, pp. 1–7.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[9] D. M. Farid, A. Nowé, and B. Manderick, "A new data balancing method for classifying multi-class imbalanced genomic data," in *25th Belgian-Dutch Conference on Machine Learning (Benelearn)*, Kortrijk, Belgium, September 2016, pp. 1–2.

[10] M. Y. Arafat, S. Hoque, S. Xu, and D. M. Farid, "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)*, Island of UKULHAS, Maldives, August 2019, pp. 1–6.

[11] S. Ahmed, F. Rayhan, A. Mahbub, M. R. Jani, S. Shatabda, and D. M. Farid, "LIUBoost: Locality informed under-boosting for imbalanced data classification," in *International Conference on Emerging Technology in Data Mining and Information Security (IEMIS)*, Kolkata, India, February 2018, pp. 1–12.

[12] F. Rayhan, S. Ahmed, A. Mahbub, M. R. Jani, S. Shatabda, D. M. Farid, and C. M. Rahman, "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*, Colombo, Sri Lanka, December 2017, pp. 1–6.

[13] S. Ahmed, A. Mahbub, F. Rayhan, M. R. Jani, S. Shatabda, and D. M. Farid, "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)*, Bengaluru, India, December 2017, pp. 126–131.

[14] F. Rayhan, S. Ahmed, A. Mahbub, M. R. Jani, S. Shatabda, and D. M. Farid, "CUSBoost: Cluster-based under-sampling with boosting for imbalanced classification," in *2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, Bengaluru, India, December 2017, pp. 70–75.

[15] M. Y. Arafat, S. Hoque, and D. M. Farid, "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*, Colombo, Sri Lanka, December 2017, pp. 1–6.