A Clustering Based Priority Driven Sampling Technique for Imbalance Data Classification

Tasmia Ishrat Alam Chadni[1], Sajid Ahmed Khan[2], and Farzana Sharmin [3]

[1]Student Id: 012-193-001, 012-202-038, 012-202-008
[2]Department of Computer Science and Engineering, United International University
[3]Plot-2, United City, Madani Avenue, Badda, Dhaka-1212, Bangladesh.
[4]Email: tchadni193001@mscse.uiu.ac.bd,skhan202038@mscse.uiu.ac.bd,fsharmin202008@mscse.uiu.ac.bd

***Abstract*—Classification of Imbalance data is one of the most vital tasks in the field of machine learning because most of the real-life datasets available have an imbalanced distribution of class labels. The effect of imbalance data is severe where the predictive model trained on the imbalanced data faces some of the unprecedented problems like overfitting where the model gets biased towards the majority target class. Many techniques have been proposed over time to deal with the imbalanced distribution caused by problems like oversampling and undersampling where oversampling isn't able to match the performance acquired by the undersampling method. One such baseline method is clustering the majority data into multiple clusters and then randomly sampling some of the redundant data but we believe that randomly sampling the data sample might open the loophole to losing informative data samples. So, in this work, we would like to propose two clustering-based priority sampling methods which manage to boost the performance of the predictive model compared to the clustering-based random sampling techniques.**

***Index Terms*—Imbalanced Data; Clustering; Ensemble classifier; Priority Based Sampling; CUSBoost, RUSBoost**

I. Introduction

Big data refers to an incredibly vast amount of complex structured and unstructured data sets [1] that can't be stored, processed, and manage effectively with traditional techniques. Big data analytic can sort out the data by revealing patterns and trends [2]. Machine learning (ML) can speed up this interaction with the assistance of decision-making algorithms. It can classify the incoming data, pattern recognition, and interpret the data into insights helpful for business operations. In machine learning for data mining applications, supervised learning uses labeled data to help predict outcomes. Supervised learning [3] is divided into two classes one is classification and another one is regression. Classification is the method of identifying new/unknown examples or instances employing a classifiers algorithm based on a group of examples with known class membership (training data). Generally, real-world data sets are high-dimensional, multi-class, and class-imbalanced which decreases the classification accuracy of many machine learning algorithms. For that reason, in the last decade, several ensembles classifiers [4] with sampling methods have been proposed [5] for classifying binary-class low-dimensional imbalanced data. To better the performance of individual classifiers, which combine various hypotheses to generate an advanced hypothesis, ensemble classifiers use multiple machine learning algorithms. The sampling methods use under-sampling and oversampling techniques [6] to alter the original class distribution of imbalanced data. Under-sampling of the majority class instances. Over-sampling the minority class instances. The Random Under sampling method involves randomly selecting and discarding examples from the majority class. On the other hand, Random oversampling involves randomly selecting examples from the minority class, with replacement, and adding them to the training dataset. Most machine learning algorithms perform well when the number of examples of each class is roughly equal. When the number of examples of one class is too far from the others then problems arise which is known as a class imbalance [7]. In another word, class imbalanced data sets mean where the total number of a class of data (positive) is too less than the total number of another class of data (negative). However, the minority class examples are addressing the idea with more noteworthy interest than the majority class examples. The majority of traditional machine learning algorithms used in big data analytics such as k-nearest neighbors (KNN), Decision tree and Naive Bayes try to improve classification accuracy but fail to classify minority class examples accurately. The most permitted techniques for dealing with class imbalance issues are cost-sensitive learning methods, ensemble methods, and sampling techniques. The cost-sensitive [8] learning method assigns the different costs of mis-classification flaws for different classes. Generally, low cost for the majority class and high cost for the minority class. In this work we are proposing two clustering [9] based Centroid-Oriented Identity-Based priority sampling techniques to deal with the negative impact on performance caused by the imbalance distribution of the class label samples. For this we have picked two of the clustering sampling based baseline methods Cusboost [10] RusBoost [7] where we proposed some modification of the baseline methods to improve the performance and eradicate the impact of imbalance distribution [11].

II. Literature Review

In [12], they had proposed a new over-sampling technique, which is basically a hybrid approach that uses real-value negative selection (RNS) to generate artificial minority data with absence of actual minority data available. The generated minority data with the help of rare actual minority data (if available) are combined as input data. After testing it on different UCI repository and real-world imbalanced dataset they got better performance results in terms of both G-Mean F-Measure evaluation metrics.In [13] one way to deal with addressing imbalanced datasets is to oversample the minority class known as SMOTE (Synthetic Minority Over-sampling Technique). The simplest approach involves copying examples in the minority class, even though these examples don't add any new data to the model. This is a type of data increase for the minority class and is known as SMOTE. It generates synthetic data [14], in accordance with the feature space similarities among the minority class samples. The main idea is to create minority class samples according to the nearest neighbor. The new samples are generated with respect to the distinction between the feature vector of the sample under consideration and its nearest neighbors.

Discussing further, another study have shown that feature selection can be used as a method of handling imbalanced data. They proposed [15] an embedded feature selection method using a decision tree. Decision tree can easily detect features which can be used as a splitting node. After selecting features using Weighted Gini Index (WGI), it compares results with Chi2, F-statistic. If 20% or more features are chosen it performs excellent results. While facing real world problems [16] like fraud detection, cancer diagnosis these results are helpful to the practitioners.In [8] proposed a strategy particularly focusing on the noisy data. They name it [1] Radial-Based oversampling method. They discussed that instead of following the popular approaches like over-sampling of the minority objects, they had used radial basis function. It can find regions in which the synthetic objects from minority class can be generated. Their experimental results show that, guided synthetic over-sampling algorithm [5] offers an impressive alternative comparatively.

In [7] Seiffert and his team proposed a hybrid approach of RUS (random under-sampling) and standard boosting procedure AdaBoost which is known as RUSBoost, to better the model [9] performance of the minority class by removing majority class samples. The motivations for bringing RUS into the boosting process are performance, speed, and simplicity. RUSBoost is used to deal with class imbalance issues in data with discrete class labels. RUSBoost is based on the (synthetic minority over-sampling with AdaBoost) SMOTEBoost algorithm. The application of SMOTE has drawbacks for this purpose RUSBoost was developed to overcome.In [12] using the data balancing methods random splitting (SplitBal) or clustering (ClusterBal) Zhongbin Sun and his team proposed a method that converts the imbalanced data into multiple balanced data. After that, with a particular classification algorithm, multiple classifiers could build from this balanced data. Finally, to merge the classification results of these classifiers for the new data they use specific ensemble rules. When it's about solving the highly imbalanced problems [17] the experimental results on 46 imbalanced binary data sets from the Keel data set repository show that their proposed technique is far better than the traditional imbalanced data dealing methods [18].

III. Materials and Methods

In this section, we are going to elaborate on our proposed methods to handle the effect on the classification performance of an imbalanced dataset [19] which we can see in the figure-1. So In the initial stage, we have divided the primary dataset (D) into multiple subsets where each of the sets only contains unique label data samples where (D_M) is the particular set that contains only the majority class instances. Now on this dataset (D_M) we have applied the K-Means clustering algorithm to divide the majority instance dataset (D_M) into K number of clusters [20] and the value of K is parametrized according to the input dataset resembling the baseline method [10] where the authors have randomly eliminated 50% of the data samples in each of the clusters but in this work of ours we haven't sampled the data samples randomly from each of the clusters rather we are using two priority-based approaches to sample the extra data from the clusters to balance the distribution of the majority and minority instances. The methods we are using is centroid-oriented and uniqueness-driven priority sampling to flush out those data sample that has the least priority according to these two methods which are discussed in the below subsections.

A. Centroid-Oriented Priority Sampling

In this method, after dividing the majority data sample set (D_M) into K number of clusters we will iterate through all of the clusters, and in every cluster, the distance is calculated between all of the data samples in the cluster and the centroid of the cluster. After that, the cluster is sorted based on the distance to the centroid in decreasing order. From this centroid-based distance sorted cluster we will pick (n) number of data samples which is at the top of the sorted set which means those data samples that are nearest to the cluster centroid and discard the rest of the data samples. Here the value of (n) depends on the size of the cluster which is denoted if the size of the cluster is bigger then the value of n will also be bigger and the total number of priority given samples in all of the clusters is balanced with the size of the minority data instances. Below given is the algorithm-1 of the described method:

B. Identity-Based Priority Sampling

The Identity-Based Priority Sampling method focuses on the unique identity of the data samples in each of the clusters. So after dividing the Majority label dataset (D_M) into K clusters. In every respective cluster, each of the data samples will be iterated then the distance between the currently iterated sample and all of the other data samples is calculated after that each of

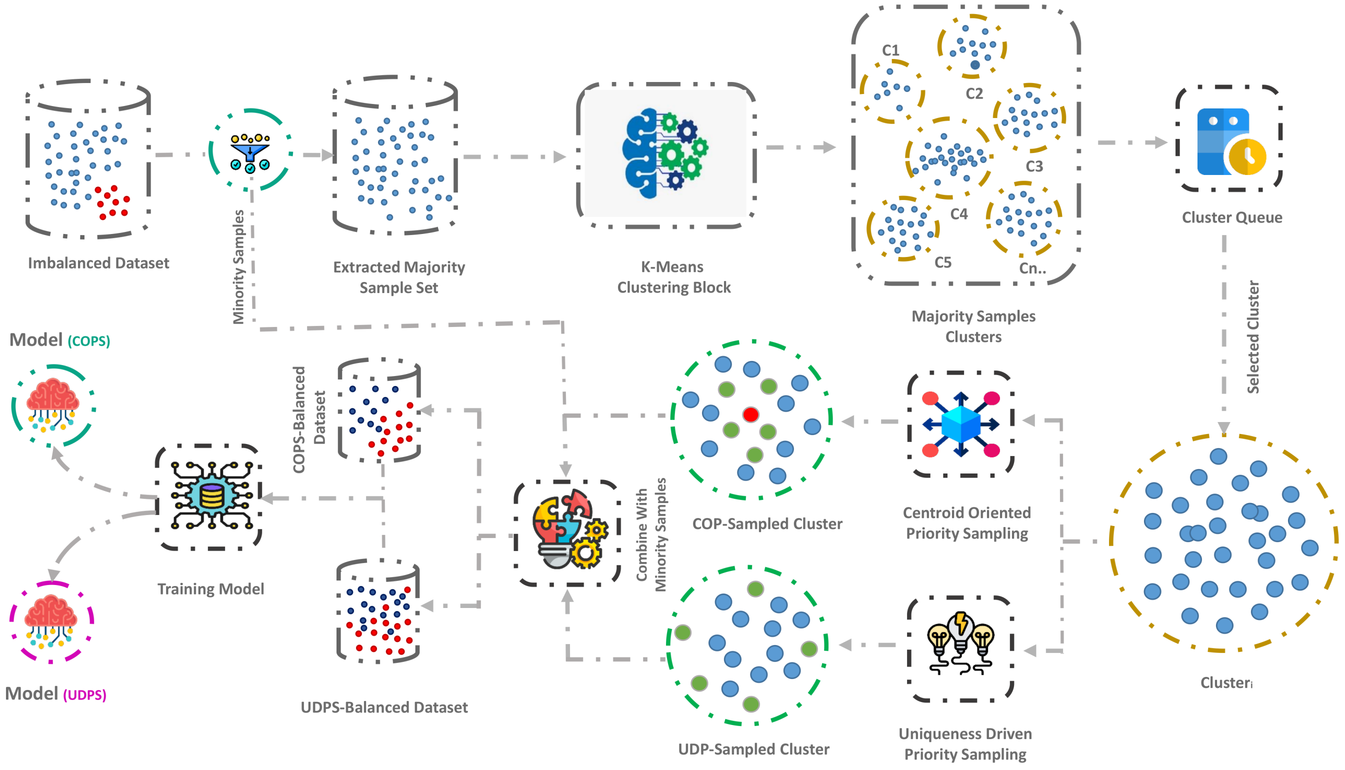

Fig. 1. Cluster-Priority Based Imbalanced Data Classification System Architecture

```
Algorithm 1: centroidPrioritySampling($ImbalanceDataset$,n)
$D_B \longleftarrow \phi$
$D \longleftarrow ImbalacedDataset$
$D_M \longleftarrow$D($label \in$MajorityGroup)
$Clusters \longleftarrow$K-Means($D_M$,K )
for each $cluster \in Clusters$ do
    $D_{ci} \longleftarrow \phi$
    $centroid \longleftarrow center(cluster)$
    for each $sample \in cluster$ do
        $dist\_sample \longleftarrow dist(sample, centroid)$
        $D_{ci}.append(sample, dist\_sample)$
    $D_{ci}.sort(reverse = True)$
    $i \longleftarrow 0$
    while $i \neq n$ do
    $D_B.append(D_{ci}[i])$
    $i \longleftarrow$i+1
  return $D_B$
```

the distances found for each of the other samples is summed up together and stored with the iterated data sample. After collecting all of the data sample distance pairs for the current cluster, the cluster will be sorted based on the summation of the distance between other data tuples in decreasing order. The key idea is those data samples that have the highest summed distance denoted that the particular data sample is the most unique among the other data samples and that is the identity of the selected data sample which will be given the highest selection priority and the rest of the data will be eradicated. Just like the previous method all of the selected data samples from each of the clusters will be balanced according to the size of the minority group. The algorithm-2 of the proposed method is given below:

```
Algorithm 2: identityPrioritySampling($ImbalanceDataset$,n)
$D_B \longleftarrow \phi$
$D \longleftarrow ImbalacedDataset$
$D_M \longleftarrow$D($label \in$MajorityGroup)
$Clusters \longleftarrow$K-Means($D_M$,K )
for each $cluster \in Clusters$ do
    for each $sample \in cluster$ do
        $D_{ci} \longleftarrow \phi$
        $dist\_sample \longleftarrow 0$
        for each $tuple \in cluster$ do
            $dist\_sample.add$(dist_sample,dist($sample$,tuple))
        $D_{ci}.append(sample, dist\_sample)$
    $D_{ci}.sort(reverse = False)$
    $i \longleftarrow 0$
    while $i \neq n$ do
    $D_B.append(D_{ci}[i])$
    $i \longleftarrow$i+1
  return $D_B$
```

IV. Experimental Analysis

In this section, we are going to highlight the performance result of the proposed methods Centroid-Oriented Priority Sampling Identity-Based Priority Sampling with the baseline methods CusBoost [10] and RusBoost [7]. We used the Adaboost algorithm for training the balanced dataset derived from the proposed sampling methods as Adaboost performed best compared to the other contemporary classifiers. The performance scale used for the comparison was ROC like the baseline method performance scale:

A. Baseline Vs Hypothesis

In this section we have compared the performances of all of the methods proposed and baselines that is comparing the performance of CusBoost [10], RusBoost [7] , centroid-oriented sampling priority sampling and identity-based priority sampling.

TABLE I
Comparison of our Proposed Methods with other two Baseline Methods

Dataset	Baseline Methods		Proposed Methdos	
	Cusboost	Rusboost	IDPS	COPS
abalone18	0.7243	0.5619	0.8729	**0.8879**
pima	0.6679	0.5898	**0.9031**	0.8903
dermatology	0.54	0.6025	**0.8945**	0.8876
segment0	**0.943**	0.5802	0.8901	0.8811
led7digit	**0.9412**	0.5516	0.8932	0.8814
yeast	0.7603	0.5113	0.8847	**0.8909**
poker 9vs7	**0.967**	0.8862	0.752	0.8837
passwd vs satan	0.8745	0.5291	**0.9005**	0.8744
yeast5	0.9	0.68	0.872	**0.8773**
ecoli	0.6589	0.6049	**0.8996**	0.899
abalone19	0.609	0.6049	**0.9011**	0.899
Page Blocks	**0.8981**	0.6037	0.8873	0.8964
Statlog (Shuttle)	0.8847	0.5438	**0.904**	0.8821

Below is the performance comparison graph of all the methods on the selected datasets. The x Co-ordinate refers to the MCC score acquired by the methods and the y Co-ordinate denotes the ROC score of the different method classifiers.

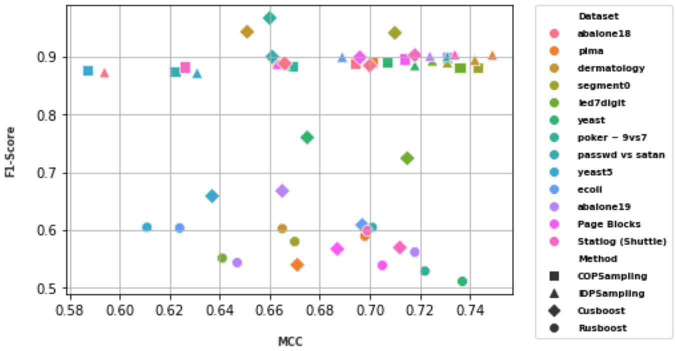

Fig. 2. Performance Comparison of the Baseline and Proposed methods.

From the performance graph we can see that our proposed method performs much better than the baseline methods Cusboost Rusboost in most of the dataset among the selected evaluation dataset.We have also conducted comparison experiment among the two of the proposed method centroid identity priority based sampling technique. Below is the comparison table where the MCC and ROC score achieved by the two respective methods are given:

B. Hypothesis-1 Vs Hypothesis-2

In this section we are focusing on the performances of the centroid-oriented priority sampling and identity-based priority sampling to see which method stands out from another.

TABLE II
Comparison Between Centroid and Identity Based Priority Sampling

Dataset	MCC		ROC	
	IDPS	COPS	IDPS	COPS
abalone18	0.594	**0.694**	0.8729	**0.8879**
pima	**0.749**	0.701	**0.9031**	0.8903
dermatology	**0.742**	0.663	**0.8945**	0.8876
segment0	**0.751**	0.743	**0.8901**	0.8811
led7digit	0.725	**0.736**	0.8732	**0.8814**
yeast	0.718	**0.737**	0.8847	**0.8909**
poker 9vs7	0.666	**0.669**	0.752	**0.8837**
passwd vs satan	**0.731**	0.622	**0.9005**	0.8744
yeast5	**0.631**	0.587	**0.872**	0.877
ecoli	0.689	**0.731**	0.8996	**0.899**
abalone19	**0.754**	0.731	**0.9011**	0.899
Page Blocks	0.663	**0.714**	0.8873	**0.8964**
Statlog (Shuttle)	**0.734**	0.626	**0.904**	0.8821

Based on the performance result table shown above we can see that Identity based priority sampling outperformed Centroid oriented sampling technique. Below is the comparison visualization graph of the two proposed hypothesis:

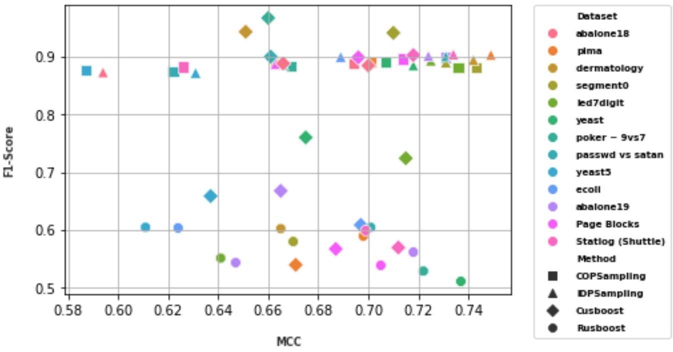

Fig. 3. Performance Comparison of the Proposed Hypothesis.

From the above result figure we can see that from the 13 evaluation dataset Identity Based priority method performed better in the 7 datasets and centroid-oriented priority sampling method managed to outperform Identity Based priority method in 6 datasets. So from this we can come to a conclusion that Identity Based priority method is the most dominant hypothesis over the other as the cases where Identity Based priority method had poor performance is pretty close to the performance scale of centroid-oriented priority sampling method.

V. Conclusion

In this work, we showed that the clustering-based random sampling method performs much better if we overlook the randomly sampling mechanism and use some priority-based data sample selection approach and also compared the performance result by testing the proposed methods on some of the open-source datasets along with the baseline methods. Some of the future experiments we would like to conduct are applying feature selection [21] techniques and some feature transformation techniques side by side with the priority-based sampling techniques to see if the performance can be pushed to a better scale.

References

[1] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International journal of pattern recognition and artificial intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

[2] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "Graphsaint: Graph sampling based inductive learning method," *arXiv preprint arXiv:1907.04931*, 2019.

[3] C. You, C. Li, D. P. Robinson, and R. Vidal, "Scalable exemplar-based subspace clustering on class-imbalanced data," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 67–83.

[4] Y. Wang, W. Gan, J. Yang, W. Wu, and J. Yan, "Dynamic curriculum learning for imbalanced data classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5017–5026.

[5] L. Yu, R. Zhou, L. Tang, and R. Chen, "A dbn-based resampling svm ensemble learning paradigm for credit classification with imbalanced data," *Applied Soft Computing*, vol. 69, pp. 192–202, 2018.

[6] G. Menardi and N. Torelli, "Training and assessing classification rules with imbalanced data," *Data mining and knowledge discovery*, vol. 28, no. 1, pp. 92–122, 2014.

[7] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2009.

[8] M. Koziarski, B. Krawczyk, and M. Woźniak, "Radial-based oversampling for noisy imbalanced data classification," *Neurocomputing*, vol. 343, pp. 19–33, 2019.

[16] V. Syrris, S. Ferri, D. Ehrlich, and M. Pesaresi, "Image enhancement and feature extraction based on low-resolution satellite data," *Ieee Journal Of Selected Topics In Applied Earth Observations And Remote Sensing*, vol. 8, no. 5, pp. 1986–1995, 2015.

[9] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409, pp. 17–26, 2017.

[10] F. Rayhan, S. Ahmed, A. Mahbub, R. Jani, S. Shatabda, and D. M. Farid, "Cusboost: Cluster-based under-sampling with boosting for imbalanced classification," in *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*. IEEE, 2017, pp. 1–5.

[11] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 4, pp. 42–47, 2012.

[12] X. Tao, Q. Li, C. Ren, W. Guo, C. Li, Q. He, R. Liu, and J. Zou, "Real-value negative selection over-sampling for imbalanced data set learning," *Expert Systems with Applications*, vol. 129, pp. 118–134, 2019.

[13] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information sciences*, vol. 250, pp. 113–141, 2013.

[14] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen *et al.*, "Dynamic sampling in convolutional neural networks for imbalanced data classification," in *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2018, pp. 112–117.

[15] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 703–715, 2019.

[17] S. Nejatian, H. Parvin, and E. Faraji, "Using sub-sampling and ensemble clustering techniques to improve performance of imbalanced classification," *Neurocomputing*, vol. 276, pp. 55–66, 2018.

[18] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[19] H. Patel, D. Singh Rajput, G. Thippa Reddy, C. Iwendi, A. Kashif Bashir, and O. Jo, "A review on classification of imbalanced data for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 16, no. 4, p. 1550147720916404, 2020.

[20] S. K. Popat and M. Emmanuel, "Review and comparative study of clustering techniques," *International journal of computer science and information technologies*, vol. 5, no. 1, pp. 805–812, 2014.

[21] L. I. Kuncheva, Á. Arnaiz-González, J.-F. Díez-Pastor, and I. A. Gunn, "Instance selection improves geometric mean accuracy: a study on imbalanced data classification," *Progress in Artificial Intelligence*, vol. 8, no. 2, pp. 215–228, 2019.