Handling Class Imbalance in Google Cluster Dataset Using a New Hybrid Sampling Approach

Jyoti Shetty * and G. Shobha

Department of Computer Science, RV College of Engineering, Bangalore, India; Email: shobhag@rvce.edu.in (G.S.)
*Correspondence: jyothis@rvce.edu.in (J.S.)

***Abstract*—Class imbalance is a classical problem in data mining, where the classes in a dataset have a disproportionate number of instances. Most machine learning tasks fail to work properly with an imbalanced dataset. There exist various approaches to balance a dataset, but suffer from issues such as overfitting and information loss. This manuscript proposes a novel and improved cluster-based undersampling method for handling two and multi-class imbalanced dataset. Ensemble learning algorithm integrated with the pre-processing technique is used to address the class imbalance problem. The proposed approach is tested using a publicly available imbalanced Google cluster dataset, in case of imbalanced dataset the F1-score value for each class has to be checked, it is observed that the existing approaches F1-score for class 0 was not good, whereas the proposed algorithm had a balanced F1-score of 0.97 for class 0 and 0.96 for class 1. There is an improvement in F1-score of about 2% compared to the existing technique. Similarly for multi-class problem the proposed novel algorithm gave balanced AUC values of 0.87, 0.83 and 0.97 for class 0, class 1 and class 2 respectively.**

***Keywords*—imbalanced dataset, hybrid sampling, google cluster**

I. Introduction

Class imbalance is a challenge in machine learning, where the number of instances among the classes of a dataset varies significantly [1]. When the number of instances of one class far exceeds others, any classification algorithm tends to treat the features of the minority class as noise and disregard them [2]. Consequently, the classifier predicts the majority class instances correctly while misclassifying the minority class instances. A minority class instance has a higher probability of being incorrectly classified than a majority class instance.

In cases of an imbalanced dataset, the accuracy metric to measure the performance of a classifier is misleading, a high accuracy does not imply that the model is performing well [3]. The accuracy metric gives equal weight to false positives and false negatives, for example, suppose there are 100 instances with 10 belonging to the positive class and 90 belonging to negative class. The classifier predicts 0 true positives, 0 false positives, 90 true negatives, and 10 false negatives. The accuracy of the classifiers is (0+90)/(0+90+0+10)=0.90 or 90%, which seems pretty good performance, but the fact is that the classifier completely failed to predict the positive instances. This is called an accuracy paradox, where the accuracy metric reports a good performance while the model completely failed to predict the minority class instances. Such a metric can be deceptive, especially in applications where positive class prediction is very crucial such as cancer prediction, fraud prediction and so on. Accuracy can be used when both positive class and negative class predictions are of equal importance or where the dataset is balanced. Various resampling techniques such as oversampling, undersampling, bagging and boosting are used to address this issue [4]. But these techniques have other issues such as loss of information, overfitting, more processing time and memory constraints. Google Cloud dataset [5] used in this study is a use case of class imbalance in large datasets [6]. Google Cluster dataset describes various traces from parts of the Google cluster management software and systems. The dataset consists of details of the jobs and tasks running on the cloud. It is 40 GB data collected from 12,500 nodes for 29 days, with six data tables of machine events, machine attributes, job events, task events, task usage, and task constraint [5]. Each job/task status is given in the form of events: Submit(0), schedule(1), evict(2), fail(3), finish(4), kill(5) and lost(6). This study focuses on failure information of the task, hence only fail(3), finish(4) and kill(5) events are interest. The dataset is highly imbalanced as there are about 370,000 instances of the finished class (class 0) and about 9000 instances of the failed class (class 1).

The use of traditional classifiers with this dataset will predict the majority class instances successfully however misclassifying the minority class i.e., finished class instances as discussed earlier. A solution to this problem is to resample the instances to balance the number of instances among the classes in the dataset. So, the proposed system examines different resampling techniques that includes: random oversampling, random undersampling, Synthetic Minority Over Sampling and cluster-based undersampling approaches to balance the dataset.

While random oversampling increases the number of instances in the minority class by randomly reproducing them, random undersampling randomly removes examples from the majority class in an effort to maintain a balanced

Manuscript received September 17, 2022; revised November 12, 2022; accepted December 22, 2022; published September 18, 2023.

class distribution in the dataset [7]. The problem with Random oversampling is it may overfitting the model, while Random undersampling results in information loss. Synthetic minority oversampling uses a heuristic approach to increase minority instances to avoid overfitting with increased processing costs. This work proposes a modified cluster-based undersampling to address the issue of an imbalanced dataset. The contributions of the proposed work are as follows.

1. Empirical evaluation and analysis of various traditional sampling techniques with google cluster dataset
2. Propose a novel Informed Cluster-based Undersampling algorithm for two-class imbalanced dataset
3. Propose a hybrid approach that extends the two-class imbalanced classification approach to multiclass imbalanced classification

The further sections of the paper are organized as follows: first, the related work is discussed, and then the proposed approach, followed by results and finally the conclusion.

II. Related Work

Class imbalance problem is inherent to most of the real world dataset and has gained wide attention recently. The Google cloud dataset used in this research work suffers from class imbalance problem. Lot of research work addressing class imbalance problem has been proposed so far to handle class imbalance issue in general, this section reviews the most important ones related to the study in this research work.

Random Undersampling (RUS) approach randomly eliminates instances of majority class until the instances in majority and minority classes are balanced. RUSBoost [7] is a hybrid approach applies RUS to boosting to improve the classification performance of weak learners. The problem with the approach is that random undersampling could eliminate potentially useful information important for building classifiers i.e., it could result in information loss. An improvement over this approach is Cluster-based Undersampling (CUS) [8].

The cluster-based undersampling approach first divides the data into training and testing sets, then the training set is stratified into majority and minority class strata, following this random undersampling of majority class starta is carried out to remove the instances of majority class, finally the reduced majority class starta and minority class starta are combined to form balanced dataset. By eliminating instances cluster wise CUS approach tries to reduce the important information loss. CUSBoost [9] approach applies CUS to every iteration of Adaboost to improve the performance of weak learners.

Near-Miss [10] is another undersampling approach, which eliminates majority class samples closer to minority class samples, thereby increasing the separation among the majority class and minority class data samples, helping the classifiers to build a better decision boundary. There are variations of near-miss algorithm which use different criteria for selection of instances of majority class for undersampling. Tomek-link [11] undersampling approach selects pair of nearest borderline instances of the majority and minority classes, and eliminates the majority class instance with the intuition that they are noise or borderline instances and eliminating them increases the boundary space between the two-classes.

Synthetic Minority Oversampling Approach (SMOTE) [12, 13] is an oversampling approach that increases the minority class instance count using artificial or man-made hypothesized samples based on a minority sample and its near neighbors. However, with a large dataset, the problem using this technique is the additional memory used for the increased samples of minority class and it may also result in overfitting. Another drawback of SMOTE is it is applicable to binary class problem and requires adequate number of minority class samples for estimating accurate probability distribution of actual data. To enhance the accuracy further Modified Synthetic Minority Oversampling Technique (MSMOTE) is proposed [14]. In this approach, the minority class instances are separated as safe, border and latent noise instances based on their distance from the all other samples in the dataset. With MSMOTE algorithm a new instance is generated based on the strategy - randomly select a data point from the set of safe instances; from border instance set select the nearest neighbor; for latent noise instance set do nothing.

Another Oversampling technique developed is the Adaptive Synthetic Sampling approach [15]. This technique aims to lessen minority class samples' learning bias. The synthetic data is generated using minority class samples rather than random samples or easier to learn samples. Sampling techniques take a lot of memory for execution and also may lead to overfitting. Hence ensemble techniques are used to reduce overfitting and also decrease the use of memory.

Adaptive Boosting (AdaBoost) is an ensemble technique known to reduce the impact of imbalanced data, it calls a weak classifier iteratively and combines their output to form a stronger classifier [16]. AdaBoost maintains a set of weights over the training set and updates the weights adaptively according to classification errors in each iteration. Bagging or bootstrap aggregation is another technique for reducing the impact of an imbalance dataset on classification [17]. Bagging is an ensemble technique that involves training many base classifiers using random subsets of the training data. Then for testing individual predictions from the base classifiers are aggregated using either the majority vote or weighted vote technique to infer the final class to be assigned to an observation. However, ensemble techniques such as bagging or boosting themselves are not sufficient and can work better if a balanced dataset is provided as input. These are the important approaches for handling the data imbalance, and form the base of other approaches discusses next.

A modification of undersampling approach for multi-label classification is Inverse Random Under Sampling (IRUS) [18]. It uses the ratio of unbalanced cardinality to determine number of instances to be eliminated from the majority class. It is called inverse as it uses the unbalanced information for the purpose. Following this the decision

boundary between the majority and minority class is learned. However, the approach just determines number of samples to be eliminated and not which data to be eliminated. Hence the approach could lead to loss of potential information.

An Ant Colony Optimization (ACO) based under sampling is proposed in [19]. It uses modified ACO to separate out the important majority class samples while eliminating the less important ones to balance the dataset. Thus, the approach is better compared to RUS as it is paying attention to the data. The approach is evaluated using DNA microarray data. Another approach uses Noise Filter to balance the dataset [20]. It uses the noise filter to remove the noise in Minority class, while the majority class is randomly under sampled. The usual approach is to remove the noise from majority class, while this approach in other direction is focusing on minority class. The results showed improved performance on most of the dataset.

A fast cluster based under sampling is proposed in [21]. It clusters the minority instances while selecting equal number majority class instances from each cluster. This is followed by building a classifier for each cluster. The cluster specific classifiers return the classification then the result is weighted by inverse distance from the cluster. If the instance does not belong to any cluster, then it is classified as majority class. The approach is evaluated for accuracy and speed against other approaches.

A cluster based over sampling with boosting is proposed in [22]. It applies oversampling to minority class followed by AdaBoost for classification. The technique seems to be simple hybrid oversampling approach and does not pay attention to the data, also it increases the dataset size which is an issue especially in large dataset used in proposed research work. A density and clustering based approach for class imbalance is proposed in [23]. Here K-means clustering is adopted to cluster the majority and minority classes, then densities are calculated, then oversampling of minority class is done by selecting the denser samples using roulette wheel concept. Similarly, the majority class is undersampled by selecting the denser samples using roulette wheel concept. This way the approach gives more importance to the denser samples. Then the majority and minority classes are combined to get a balanced dataset and SVM model is used for classification.

In Ref. [24], the effectiveness of two data resampling techniques, SMOTE and Deep Belief Network (DBN), is compared to the effectiveness of two cost-sensitive learning techniques, focal loss and weighted loss in the churn prediction problem. The empirical findings demonstrate that for the churn prediction problem, the focal loss and weighted loss approaches have higher overall predictive performance than SMOTE and DBN. With Classification Based on Associations (CBA) using Class Association Rules (CARs), the conventional technique defines the two key measures of task interest for pruning: minimum Support and minimum Confidence [25]. However, although uninteresting rules are being pruned at the same time, some interesting rules that have low Support or Confidence are also removed. Positive-Class CARs often have a significantly lesser number than negative-Class CARs, and this problem typically arises in datasets with an imbalanced Class ratio. Most Positive-Class CARs are discovered to have low Support or Confidence and require trust in order to be used without uninteresting rules. The study outlines the issue in respect to a dataset on breast cancer and employ a pruning task to identify interesting positive CARs even when the Support or Confidence is low.

Thus, this section provides a detailed discussion of various algorithms and their advantages and disadvantages in addressing the class imbalance issue. However, no technique is best for all the scenarios. Depending on the data a particular technique may outperform other. Also, it is observed that none of the algorithms addressed the class imbalance issue of Google cloud data.

III. Proposed Approach

For a two-class imbalanced dataset first preprocessing is done using Informed cluster-based undersampling given in Algorithm 1 followed by classification using Xgboost classifier.

$$Proportional\ size\ = \frac{number\ of\ instances_i}{number\ of\ instances_{maj}} \quad (1)$$

$$Dispersion = \frac{stddev_i}{mean_i} \quad (2)$$

$$w_i = \left((Propotional\ size_i \times Dispersion_i)\right) \quad (3)$$

Algorithm 1: Informed CUS

Input: Majority class instances, $majority_1$, to be undersampled

Output: Undersampled majority instances $majority_1'$

1. Determine the number of clusters N using elbow or silhouette coefficient measure
2. Form *N* clusters using MiniBatchKmeans clustering
3. $majority_1' = \{\emptyset\}$

For *each cluster i in N* **do**

$size_i =$ number of instances in $cluster_i$

$point_i =$ instances in $cluster_i$

$centroid_i$ = center point of $cluster_i$

$DistanceMatrix =$ Euclidian distance of each point from $centroid_i$

$SortedPoints$= Sort points based on Euclidean distance

W_c =Calculate weight for $cluster_i$ as per Eq. (3)

$BoundaryPoints =$ Points after removing $W_c \times size_i$ from $SortedPoints$

$majority_1' = majority_1'\ U\ BoundaryPoints$

end

Return $majorit\ y_1'$

For the majority class instances, clusters are created using the MinibatchKmeans clustering algorithm, which is an alternative to the Kmeans clustering algorithm for large datasets. The number of clusters to be formed is decided using the elbow method. Following clustering, samples are picked up from clusters as the cluster represents a different population distribution of underlying data. Rather than performing random undersampling from the clusters an informed approach is used, where the instances that form a

boundary for classification are retained while eliminating other instances. To retain the boundary points Euclidean distances of the instances from the center instance is computed, then the instances are sorted to eliminate the instances nearer to the center. Further rather than uniform elimination of instances from each cluster the proposed undersampling approach uses a measure to determine the number of instances to be eliminated from each cluster. Each cluster is assigned a weight that determines the number of instances to be eliminated given by Eq. (3). This weight is based on the Proportional size of the cluster concerning the majority class and Dispersion of the instances within the cluster. The smaller the cluster and the dispersion, the smaller is the weight assigned and hence less number of instances are eliminated from it vice versa. The Proportional size is the ratio of the number of instances of the cluster to the total number of majority class instances given by Eq. (1). The Dispersion of cluster is defined as the coefficient of variation, which is the ratio of the standard deviation to the mean of the distances of the instances of the cluster as given by Eq. (2). Thus the number of samples eliminated is proportional to size proportion and sparsity of the cluster.

Most of the existing studies focused on imbalance learning of two-class, however, those techniques perform poor and are not effective in handling multiclass imbalance problems. There are challenges in handling multiclass imbalanced data over and above two-class imbalanced data such as improving the performance of one class may decrease the performance of other classes. There is a lack of systematic research on this topic and existing solutions are limited. There two types of multi-class imbalanced data: multi-majority and multi-minority. Multi-majority imbalanced dataset where there are more majority classes and a few minority class, for example, a dataset with 7 overall classes has 5 classes with more instances and 2 classes with few instances. Similarly multi-minority class is where there are multiple minority classes and few majority classes. The Google cloud data is an example of a multi-class dataset, however as per the need for the proposed research only failed, killed and finished classes are considered. It is a multi-minority dataset with failed and killed and minority class and finished as majority class. The proposed algorithm combined with oversampling is used to address multi-class imbalanced learning of the Google Cloud dataset. At first, the instances of interest i.e killed, failed and finished instances are filtered from the data. It is observed that failed class and killed are minority class, while finished class forms the majority class. The proposed approach is hybrid where it uses oversampling and informed undersampling to balance the data, followed by the ensemble of the one-vs-rest classification approach. The failed class instances are oversampled using SMOTE and appended to the dataset, followed by the finished class instances are undersampled using informed undersampling and added to the dataset, the killed class instances remain unchanged. The balanced data is then trained using the One-Vs-Rest Xgboost classifier and tested using test dataset. The classification algorithm is as shown in Algorithm 2. The performance is measured using Precision, Recall, Average Precision-recall and F1-Score.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1\ Score = 2 \times \frac{Precesion\ X\ Recall}{Precesion+Recall} \quad (6)$$

$$APR = \sum_n (Recall_n - Recall_{n-1}) Precision_n \quad (7)$$

Algorithm 2: Imbalanced Multiclass Classification

Input: Multiclass Imbalanced Dataset D_s
Output: Multiclass Imbalanced Dataset

1. Let $minority_1, minority_2, majority_1$ be the instances belonging to the dataset such that $size(minority_1) \leq size(minority_2) < size(majority_1)$
2. Oversample $minority_1$using SMOTE $minority_1' = SMOTE()$
3. Oversample $majority_1$using InformedCUS given in Algorithm1
4. $majority_1' = InformedCUS()$
5. Construct the balanced dataset $D_s' = minority_1' \cup minority_2 \cup majority_1'$
6. Split the dataset D_s' using 10-fold cross-validation into test and train sets
7. Train Xgboost classifier using the training dataset
8. Predict the classes for the test dataset
9. Evaluate the model performance using various measures as in Eqs. (4), (5), (6) and (7)

End

IV. Results and Discussion

The objective of this section is to illustrate the performance impact of the proposed approach compared with existing approaches. To achieve this, first, the experimental work involves modeling various existing sampling approaches and the proposed approach.

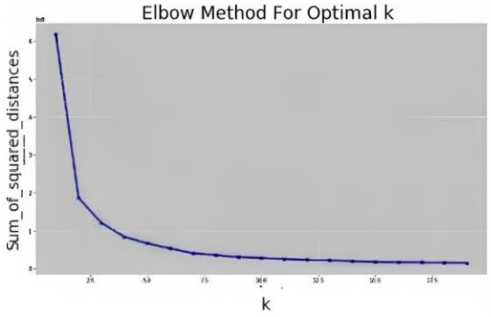


Figure 1. Finding *k* value for minibatch k-means using elbow method.

For the two-class imbalanced dataset, the sampling approaches used are RUS, ROS, CUS, and Informed CUS. MinibatchKmeans clustering algorithm is used for CUS, in which the *k* value, i.e., the number of clusters to be formed is determined as 7 using elbow method as shown in Fig. 1. The number of minority and majority class instances for the

imbalanced dataset and balanced dataset after application of various techniques is as in Table I.

Once the dataset is balanced the Xgboost classifier is trained over the balanced dataset for prediction. The performance of the classifier over each of the balanced datasets is measured and tabulated as shown in Table II. All the models are implemented using Scikit Learn library.

TABLE I. NUMBER OF INSTANCES

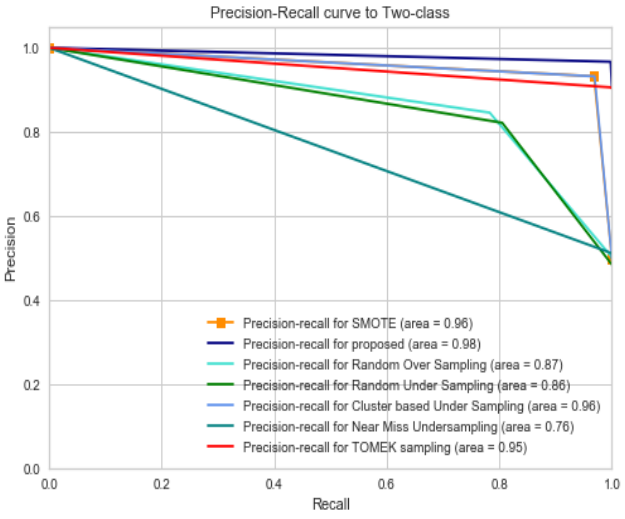
Techniques	Minority Class/ Failed class	Majority Class/ Finished class
Imbalanced	10124	385581
Under sampling	10124	10124
Over sampling	385581	385581
CUS	80240	81000
Informed CUS	80240	81000

TABLE II. PERFORMANCE COMPARISON OF VARIOUS APPROACHES FOR TWO-CLASS IMBALANCED GOOGLE DATASET

Technique	Class	Precision	Recall	F1-score	AUC	APR	Accuracy
Imbalanced dataset+Xgboost	Class 0	0.94	0.53	0.68	0.58	0.02	98.7
	Class 1	0.91	1.00	0.99			
CUS+Xgboost	Class 0	0.50	0.93	0.65	0.96	0.96	96.92
	Class 1	1.00	0.98	0.98			
RUS+Xgboost	Class 0	0.80	0.86	0.83	0.86	0.45	81.61
	Class 1	0.85	0.78	0.82			
ROS+Xgboost	Class 0	0.80	0.86	0.83	0.87	0.77	81.61
	Class 1	0.85	0.79	0.82			
SMOTE+Xgboost	Class 0	0.97	0.90	0.95	0.96	0.92	94.98
	Class 1	0.93	0.97	0.95			
NearMiss+Xgboost	Class 0	0.04	0.61	0.07	0.76	0.98	54.05
	Class 1	0.98	0.54	0.69			
TomekLinks+Xgboost	Class 0	0.04	0.64	0.07	0.95	0.98	56.75
	Class 1	0.98	0.57	0.72			
Informed CUS+Xgboost	Class 0	**0.98**	**0.90**	**0.94**	**0.98**	**0.92**	94.88
	Class 1	**0.93**	**0.99**	**0.96**			

TABLE III. PERFORMANCE COMPARISON OF VARIOUS APPROACHES FOR MULTI-CLASS IMBALANCED GOOGLE DATASET

Technique	Class	AUC	Accuracy	Precision	Recall	F1-score	Micro Average precision-recall
InformedCUS+SMOTE+Xgboost	Class 0	**0.87**	88.42	**0.932**	**0.901**	**0.917**	**0.87**
	Class 1	**0.83**					
	Class 2	**0.91**					
CUS+ROS+Xgboost	Class 0	0.86	87.28	0.929	0.895	0.911	0.86
	Class 1	0.87					
	Class 2	0.90					
RUS + ROS+Xgboost	Class 0	0.85	87.64	0.930	0.900	0.914	0.87
	Class 1	0.85					
	Class 2	0.90					
Imbalanced Xgboost	Class 0	0.32	94.82	0.938	0.755	0.804	0.93
	Class 1	0.95					
	Class 2	0.93					



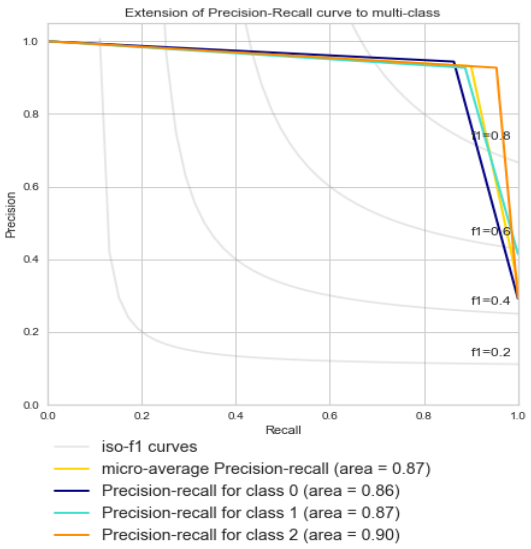
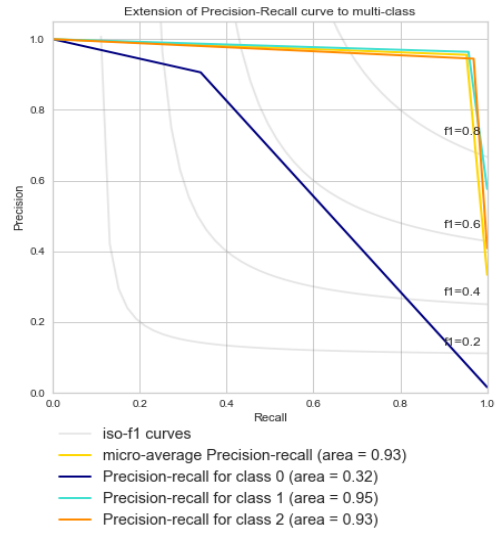
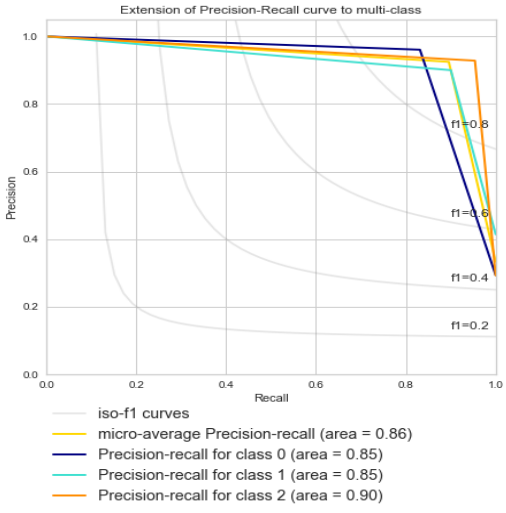


Figure 2. Performance comparison graph of various approaches for two-class imbalanced Google dataset.

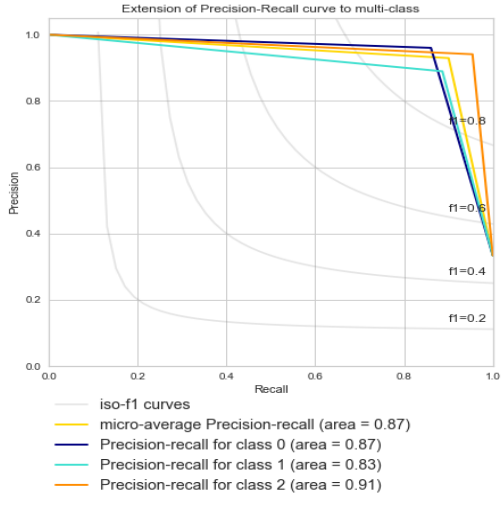
(a) Proposed approach



(b) Imbalanced Xgboost



(c) CUS+ROS+Xgboost



(d) RUS+ROS+Xgboost

Figure 3. Performance comparison graph of various approaches for multi-class imbalanced Google dataset.

An insight into the performance metrics tabulated as in Table II it can be observed that after balancing the dataset the classifier is able to perform well with respect to minority class (Class 0) as the precision and recall values are improved, among all the approaches, it can be seen that the proposed approach outperforms other approaches. The graph in Fig. 2 shows the precision-recall curve, it can be seen the proposed approach has got the best AUC of 0.98, followed by CUS and other approaches. This improvement in performance justifies that the proposed algorithm is able to retain the information of the majority class while undersampling. To handle multi-class imbalance problems, a hybrid approach is proposed, where SMOTE oversampling is applied to the minority class and Informed Cluster-based undersampling is applied to majority samples.

The Table III depicts the performance comparison of various approaches and Fig. 3 shows performance graphs of same, from the graph it is seen that the performance of the classifier on imbalanced dataset is very poor with respect to minority class (class 0) with AUC of 0.32 as compared to those classifiers with balanced dataset, also it is observed that the proposed approach outperforms all other approaches with AUC of 0.87, 083, 0.91 for class 0, 1, and 2 respectively.

V. Conclusion

One of the major problems with building a prediction model is imbalanced class dataset, the Google cloud data used for failure prediction in this research work has disproportionate instances of classes. The research work proposed and implemented novel algorithms for handling two-class and multi-class imbalance problems. In case of imbalanced dataset, the F1-score value for each class has to be checked, it is observed that the existing approaches F1-score for class 0 was not good, whereas the proposed algorithm had a balanced F1-score of 0.97 for class 0 and 0.96 for class 1. There is an improvement in F1-score of about 2% compared to the existing technique. Similarly for multi-class problem the proposed novel algorithm gave balanced AUC values of 0.87, 0.83 and 0.97 for class 0, class 1 and class 2 respectively.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Jyoti Shetty proposed the problem, carried out the study, designed the methodology and carried out implementation with guidance from G. Shobha. Both Jyoti Shetty and G. Shobha drafted the paper and the results. All authors had approved the final version of the paper.

Acknowledgment

The authors wish to thank RV College of Engineering for its support and encouragement during the research.

References

[1] Q. Yang and X. Wu, "10 challenging problems in data mining research," *Int. J. Inf. Tech. Decis.*, vol. 5, no. 4, pp. 597–604, 2006.
[2] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder *et al.*, "A survey on addressing high-class imbalance in big data," *J. Big Data*, vol. 5, 42, 2018.

[3] V. López, A. Fernandez, S. Garcia, 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.
[4] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, July 2012.
[5] Cluster-data. [Online]. Available: https://github.com/google/cluster-data
[6] J. Shetty, R. Sajjan, and G. Shobha, "Task resource usage analysis and failure prediction in cloud," in *Proc. 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2019, pp. 342–348. doi: 10.1109/CONFLUENCE.2019.8776612
[7] C. Seiffert, T. M. Khoshgoftaar, J. 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, Jan. 2010. doi: 10.1109/TSMCA.2009.2029559
[8] S. J. Yen and Y. S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," in *Proc. the 8th International Conference*, 2006, pp. 427–436.
[9] 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 *Proc. 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, Bengaluru, India, 2017, pp. 1–5, doi: 10.1109/CSITSS.2017.8447534
[10] I. Mani and J. Zhang, "KNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. the Workshop on Learning from Imbalanced Data Sets*, 2003, pp. 1–7.
[11] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, no. 6, pp. 448–452, 1976.
[12] 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, June 2002.
[13] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *Proc. European Conference on Principles of Data Mining and Knowledge Discovery*, Springer, 2003, pp. 107–119.
[14] S. Hu, Y. Liang, L. Ma, and Y. He, "MSMOTE: Improving classification performance when training data is imbalanced," in *Proc. 2009 Second International Workshop on Computer Science and Engineering*, Qingdao, 2009, pp. 13–17.
[15] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Hong Kong, 2008, pp. 1322–1328.
[16] Y. Sun, M. S. Kamel, and Y. Wang, "Boosting for learning multiple classes with imbalanced class distribution," in *Proc. the Sixth International Conference on Data Mining (ICDM'06)*, Hong Kong, 2006, pp. 592–602.
[17] J. Błaszczynski, J. Stefanowski, and Ł. Idkowiak, "Extending bagging for imbalanced data," in *Proc. the 8th International Conference on Computer Recognition Systems (CORES 2013)*, Springer, 2013, pp. 269–278.
[18] M. A. Tahir, J. Kittler, and F. Yan, "Inverse random under sampling for class imbalance problem and its application to multi-label classification," *Pattern Recognition*, vol. 45, no. 10, pp. 3738–3750, 2012. https://doi.org/10.1016/j.patcog.2012.03.014
[19] H. Yu, J. Ni, and J. Zhao, "ACO sampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data," *Neurocomputing*, vol. 101, pp. 309–318, 2013. https://doi.org/10.1016/j.neucom.2012.08.018
[20] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 47, no. 12, pp. 4263–4274, 2017. https://doi.org/10.1109/TCYB.2016.2606104
[21] N. Ofek, L. Rokach, R. Stern, and A. Shabtai, "Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem," *Neurocomputing*, vol. 243, pp. 88–102, 2017. https://doi.org/10.1016/j.neucom.2017.03.011
[22] G. Rekha, V. K. Reddy, and A. K. Tyagi, "A novel approach for solving skewed classification problem using cluster based ensemble method," *Mathematical Foundations of Computing*, vol 3, issue 1, 2020.
[23] B. Mirzaei, B. Nikpour, and H. Nezamabadi-pour, "CDBH: A clustering and density-based hybrid approach for imbalanced data classification," *Expert Systems with Applications*, vol. 164, 2021.
[24] N. N. Nguyen and A. T. Duong, "Comparison of two main approaches for handling imbalanced data in churn prediction problem," *Journal of Advances in Information Technology*, vol. 12, no. 1, pp. 29–35, February 2021. doi: 10.12720/jait.12.1.29-35
[25] P. Liewlom, "Class-association-rules pruning by the profitability-of-interestingness measure: Case study of an imbalanced class ratio in a breast cancer dataset," *Journal of Advances in Information Technology*, vol. 12, no. 3, pp. 246–252, August 2021. doi: 10.12720/jait.12.3.246-252