

# Random Walk-steered Majority Undersampling<sup>\*</sup>

Payel Sadhukhan<sup>1</sup>[0000-0001-7795-3385], Arjun Pakrashi<sup>2</sup>[0000-0002-9605-6839],  
and Brian Mac Namee<sup>2</sup>[0000-0003-2518-0274]

<sup>1</sup> TCG Crest Kolkata, India  
payel0410@gmail.com

<sup>2</sup> School of Computer Science, University College Dublin, Ireland  
{arjun.pakrashi,brian.macnamee}@ucd.ie

**Abstract.** In this work, we propose *Random Walk-steered Majority Undersampling* (RWMaU), which undersamples the majority points of a class imbalanced dataset, in order to balance the classes. Rather than marking the majority points which belong to the neighborhood of a few minority points, we are interested to perceive the closeness of the majority points to the minority class. Random walk, a powerful tool for perceiving the proximities of connected points in a graph, is used to identify the majority points which lie close to the minority class of a class-imbalanced dataset. The visit frequencies and the order of visits of the majority points in the walks enable us to perceive an overall closeness of the majority points to the minority class. The ones lying close to the minority class are subsequently undersampled. Empirical evaluation on 21 datasets and 3 classifiers demonstrate substantial improvement in performance of RWMaU over the competing methods.

**Keywords:** class imbalance · undersampling · random walk · majority class

## 1 Introduction

Real-world data from the domain of medical [22], text [36], software defect prediction [2], and fraud detection [31] often have significant imbalance between target classes. In a binary classification dataset with class-imbalance, the class with more instances and the class with fewer instances are known as the majority and the minority class respectively. When a classifier is modelled on an imbalanced dataset, it often gets influenced to predict the majority class.

There are a number of different solutions to the class-imbalance problem in the literature. These can be categorized into: i) algorithmic methods [14], ii) data preprocessing [10] and iii) ensemble-based learning [8]. Data preprocessing is the most popular choice amongst these three as it is independent of model building.

---

<sup>\*</sup> This work has emanated from research supported in part by TCG Crest, Kolkata, India and a grant from Science Foundation Ireland under Grant number [16/RC/3835]. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

In data pre-processing, we *undersample* (remove points from) the majority class or *oversample* (add points to) the minority class in order to reduce the difference in representation between the two classes. Consequently, the bias towards the majority class is reduced.

The state-of-the-art undersampling schema primarily mark the majority points to be undersampled in one of two ways: i) finding a good representative set of the majority class by employing procedures like clustering, thereby discarding the majority points which lie near the clusters' periphery, or ii) marking the majority points which lie in the k-nearest neighborhood of the minority points. The k-nearest neighborhood based undersampling methods are generally focused to find the majority points which lie close to the minority points. While doing so, the methods somewhat become oblivious to the points' relative distances. Moreover, employing a crisp neighborhood-based protocol delivers a locally optimized nearness. In most cases, the methods are not motivated to quantify the overall closeness of the majority points to the minority class.

In this work, we present **Random Walk-steered Majority Undersampling (RWMaU)**, an undersampling technique which addresses these concerns. Instead of simply figuring out the k-nearest majority neighbors of the minority points, we are motivated to obtain the overall closeness of the majority points to the minority class. We employ random walk for this purpose. Random walk is a powerful tool for perceiving the mutual proximity of the nodes in a graph. It has been extensively used in the domain of social network analysis to find communities, compute feature representations, and find other relevant parameters of a graph [12,13]. RWMaU forms a directed graph from a class-imbalanced dataset, where each point is connected to its k-nearest neighbors. The edge-weights of the outward edges depend on the relative distances of the neighbors. We simulate a number of random walks from the minority points (as starting nodes) and study the visit frequencies of the majority nodes (along with the order of the visits) in these walks.

In particular, we use the visits and their orders to compute the proximity scores of the majority points with respect to the minority class, thereby finding the majority points which are close to the minority class as a whole. While undersampling the majority class, we remove the majority points in order of their decreasing proximity scores. A majority point which has the highest proximity to the minority class is removed first.

The key aspects of our work are summarized as follows.

- We quantify the majority nodes' visit frequencies and the order of the visits in random walks to compute the nearness of the majority points to the minority class.
- The majority points which lie close to the minority class are removed to alter the class distributions in favor of the under-represented minority class.
- An empirical study involving 21 datasets, 3 classifiers, 5 competing methods (4 undersampling methods and the original datasets) and 2 evaluating metrics indicates the effectiveness of the proposed method.

The remainder of the paper is organized as follows. In Section 2, we discuss the relevant existing work in the field of handling class-imbalance. We present the random walk preliminaries, motivation of our work, and the proposed algorithm in Section 3. The experimental design is described in Section 4 and the results of the experiments are discussed in Section 5. Finally, in Section 6 we conclude the paper.

## 2 Related Work

One of the early approaches in the field of class-imbalance learning is algorithm-based methods. Most of the schema from this domain are motivated to either shift the class boundary away from the minority class [21] or to add a cost-sensitive learning framework where the misclassification cost of the minority class is increased [19,15,32]. Other important classes of algorithm-based methods are active learning [33], multi-objective optimization based methods [27], kernel-based methods [30] and one class classifiers [16].

Data-preprocessing techniques form an important and popular choice to address class-imbalance of data. In undersampling, the points belonging to the majority class are selected and removed to reduce the difference in cardinalities of the two classes. Various techniques are proposed by the researchers in this domain to make a judicious choice of the majority points to be removed from the dataset. The two principal techniques to choose points to be undersampled are i) cluster based - clustering is done to recognize the key points to be kept for the classification phase [24,18,35] and ii) nearest neighbor based - the majority neighbors of the minority points are identified and are removed (with some additional condition checks) [4,17]. Oversampling of the minority class is another way of balancing the cardinalities of the two classes [3,5]. A number of diversified parameters like minority class density [11], oversampling near boundary [1], majority class non-encroachment [28] are considered by the researchers to effectively oversample synthetic minority points in the feature space. In recent years, random walk is learnt on graphs to choose the locations of minority oversampling [34]. In addition to these, hybrid data-preprocessing techniques also exist in literature which employs both minority oversampling and majority undersampling [26]. In some techniques, both data pre-processing and algorithm adaptation are considered to tackle the issue of class-imbalance.

The third category of class-imbalance learner deals with a set of classifiers (often at various hierarchies) along with boosting and bagging to obtain an improved learning over class-imbalanced datasets [8]. Minority oversampling is integrated with boosting to obtain an improved accuracy for both the minority and the majority class by [9]. In [25], the authors follow a hierarchical paradigm where a set of weak (preliminary) classifiers are trained on the imbalanced dataset followed by derivation of a strong classifier from these weak classifiers.

### 3 Random Walk-steered Majority Undersampling (RWMaU)

In this section, first we briefly explain related aspects of random walks followed by a brief discussion of the motivation and the core idea of our approach. Then we present the proposed algorithm, Random Walk-steered Majority Undersampling (RWMaU).

#### 3.1 Random Walk

A random walk is a sequence of discrete, finite length steps in random directions depending on probabilities. Random walks are often considered in context of a graph,  $G(V, E)$  where we have a set of nodes  $V = \{v_1, \dots, v_N\}$  and a set of edges,  $E = \{(v_i, v_j) | (v_i, v_j) \in V \times V \text{ and } i \neq j\}$  connecting the nodes. Each edge has a weight  $p_{ij}$  which connects  $v_i$  and  $v_j$ , which can be captured in an adjacency matrix of the graph. When we consider a random walk in a graph, we start from a node,  $v_i$ , and move to another node  $v_j$  with considering  $p_{ij}$  as the transition probability. This process of moving from one node to another node is repeated until we have performed a certain number of steps. The sequence of nodes which this process goes through is called a random walk. Details about random walk can be found in [20]. We use properties of random walk in our proposed method.

#### 3.2 Motivation and Overview

Our approach is to mark and remove the majority points which are close to the minority class overall. To do this, we compute a score for each majority point, which determines how close they are with respect to the minority points collectively. The majority points with high scores will indicate their closeness to the minority space. This score is ranked, and the higher scored majority points are removed. Random walk serves as the backbone of this entire procedure.

We construct a directed weighted graph from the given dataset on which the random walks will be performed. We assume that the majority points, which appear a) more frequently in the walk, and, b) earlier in the walk sequence, are closer to the minority class. Based on these two assumptions the scores for each majority points are assigned. We simulate a series of random walks from each minority point and record the visit frequencies of the majority points in the series of these walks to address the assumption (a). Also, the visit frequencies of a majority node is weighted based on how far in the walk the node was visited to incorporate the assumption (b). Therefore, a visit occurring earlier in the walk will be given more weight than a later one.

A proximity score of each majority point is computed from these two information, which will indicate a degree of closeness of the majority point relative to the minority class. These assigned scores are used to identify and remove the majority points.

**Algorithm 1** RWMAU

---

```

1: procedure RWMAU( $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}, \alpha, \gamma, k$ )
2:    $\mathcal{X}_{min} = \{\mathbf{x}_i | \forall_i y_i = 1\}$  and  $\mathcal{X}_{maj} = \{\mathbf{x}_i | \forall_i y_i = 0\}$ 
3:   Make graph  $G(\mathcal{X} = \{\mathbf{x}_i | i = 1, 2, \dots, n\}, E = \{p_{ij} | i, j = 1, 2, \dots, n\})$  use Eq. (2)
4:   for  $\mathbf{x}_i \in \mathcal{X}_{min}$  do
5:      $W_{\mathbf{x}_i} = \text{randomWalk}(G, \mathbf{x}_i, \gamma)$  (use Eq. (3))
6:   for  $\mathbf{x}_l \in \mathcal{X}_{maj}$  do
7:      $\nu_l = \sum_{\mathbf{x}_j \in \mathcal{X}_{min}} \sum_{\beta=1}^{\gamma} \frac{W_{\mathbf{x}_j}(\beta, \mathbf{x}_l)}{\beta}$ 
8:    $u = (|\mathcal{X}_{maj}| - |\mathcal{X}_{min}|) \times \alpha$ 
9:    $\tau = \text{sortDecreasing}(\nu)$  ▷ Get sorted order of  $\nu$ 
10:   $\mathcal{X}_{rem} = \{\mathbf{x}_{\tau_j} | \forall_j \nu_{\tau_j} \geq \nu_{\tau_u}\}$  ▷ Select top  $u$  points to remove
11:   $\mathcal{A} = \mathcal{X} - \mathcal{X}_{rem}$ 
12:  return  $\mathcal{A}$ 

```

---

**3.3 Algorithm**

In this section, we present Random Walk-steered Majority Undersampling (RWMAU). We will also describe the algorithm in details along with Algorithm 1, which summarises the scheme.

First, we represent the dataset is represented as a directed weighted graph  $G(\mathcal{X}, E)$ . In  $G$ , each vertex represents a point and the weight of the directed weighted edge from  $\mathbf{x}_i$  to  $\mathbf{x}_j$  is  $p_{ij}$  which is defined as

$$G(\mathcal{X} = \{\mathbf{x}_i | i = 1, 2, \dots, n\}, E = \{p_{ij} | i, j = 1, 2, \dots, n\}) \quad (1)$$

The  $p_{ij}$  indicates the reachability of  $\mathbf{x}_j$  from  $\mathbf{x}_i$  in the graph and will be used as the transition probabilities in the random walk. We define  $p_{ij}$  as follows

$$p_{ij} = \begin{cases} \frac{e^{-\frac{d_{ij}}{d_{iNN_k(i)}}}}{\sum_{m=1}^k e^{-\frac{d_{iNN_m(i)}}{d_{iNN_k(i)}}}} & , \text{if } \mathbf{x}_j \text{ is a } k\text{-nearest neighbor of } \mathbf{x}_i \\ 0 & , \text{otherwise} \end{cases} \quad (2)$$

here,  $d_{ij}$  is the Euclidean distance between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , and  $d_{iNN_m(i)}$  denotes the distance between  $\mathbf{x}_i$  and its  $m^{\text{th}}$  nearest neighbour.

We start a walk from each of the minority point  $\mathbf{x}_i \in \mathcal{X}_{min}$  and record the nodes visited during the walk along with the order of visit. The random walk starting at  $\mathbf{x}_i \in \mathcal{X}_{min}$  is represented as  $W_{\mathbf{x}_i}$  and is defined as follows

$$W_{\mathbf{x}_i}(\beta, \mathbf{x}_l) = \begin{cases} 1 & \text{if } \mathbf{x}_l \text{ is visited in } \beta^{\text{th}} \text{ step of a walk started at } \mathbf{x}_i \in \mathcal{X}_{min} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

here  $W_{\mathbf{x}_i}(\beta, l)$  indicates if node  $\mathbf{x}_i$  is visited in the  $\beta^{\text{th}}$  step of a random walk started at minority instance  $x_i \in \mathcal{X}_{min}$ . The datapoints in  $\mathcal{X}_{maj}$  through which

relatively more random walks go through, are the ones which are more likely to be removed during the undersampling process. Now we can calculate the minority proximity scores of the majority instances  $\mathbf{x}_l \in \mathcal{X}_{maj}$ . We denote the minority proximity score of  $\mathbf{x}_l$  by  $\nu_l$ , which integrates the information about visit frequency and order of  $\mathbf{x}_l$  in the different random walks.

$$\nu_l = \sum_{\mathbf{x}_j \in \mathcal{X}_{min}} \sum_{\beta=1}^{\gamma} \frac{W_{\mathbf{x}_j}(\beta, \mathbf{x}_l)}{\beta} \quad (4)$$

Once we have computed the values of  $\nu_l$  for all majority datapoints, we can then remove the ones which with a high value of  $\nu_l$ .

We will sort the elements of  $\mathcal{X}_{maj}$  in decreasing order of their  $\nu$  values. We will discard the first  $u$  points to get the set to remove  $\mathcal{X}_{rem}$

$$u = (|\mathcal{X}_{maj}| - |\mathcal{X}_{min}|) \times \alpha \quad (5)$$

In Eq. (5), let  $u$  be the number of points to be undersampled and  $\alpha$  be a constant such that  $0 < \alpha \leq 1$ .  $\alpha = 0$  signifies no undersampling, and if we set  $\alpha = 1$ , we will equate the cardinalities of the minority class and the majority class in the augmented set. We further denote the removed points from the majority class by  $\mathcal{X}_{rem}$ . Finally, the augmented training  $\mathcal{A}$  set is obtained by removing the set of points to be removed through  $A = \mathcal{X} - \mathcal{X}_{rem}$ .  $A$  is used to train the classifier.

## 4 Experimental Design

To evaluate the proposed method RWMaU, we have performed a detailed experiment involving 21 binary classification datasets with different degrees of class imbalance (Imbalance ratio ranging from 1.54 to 32.73). They are listed in Table 1 along with their basis statistics, where  $n$  is the number of datapoints,  $d$  is the number of dimensions and *Imb. Ratio* is the imbalance ratio of the dataset, which is the ratio of the number of majority class and minority class datapoints. The datasets are a part of [7,6] obtained from the KEEL project page <sup>3</sup>.

In the comparative study, we have included the original training dataset (without any oversampling or undersampling). Since majority class undersampling is the essence of the proposed work, we have included four undersampling schemes in this study namely – Random Undersampling (RUS) [23], Instance Hardness Threshold (IHT) [29], Undersampling with Cluster Centroids (CC) and Neighbourhood Cleaning Rule (NCR) [17]. K-Nearest Neighbour (with  $k=5$ ), C4.5 and C4.5 + Bagging classifiers were used to train the model at their default settings were used to train the model using various undersampling schemes. The original (unsampled) dataset’s performance on the above given classifier were also compared as a baseline. The value of  $k$  (in RWMaU) was selected from a range of 2, 3, . . . , 10 and  $\gamma$  (in RWMaU) was selected from a range of  $2k \pm 3$  in

<sup>3</sup> <https://sci2s.ugr.es/keel/imbalanced.php>

combination by optimizing over C4.5 Decision Tree classifier. The  $(k, \gamma)$  tuple which optimized the results of RWMaU on C4.5 Decision Tree were used in all the experiments. Also, the value of  $\alpha$  was set to 0.5 for all the undersampling methods in the comparative study. This was done to limit removal of too many majority points.

Table 1: Description of datasets

	n	d	Imb. Ratio
yeast5	1484	8	32.73
yeast1289v7	947	8	30.57
wine-red-4	1599	11	29.17
yeast4	1484	8	28.10
yeast1458v7	693	8	22.10
abalone9-18	731	8	16.40
ecoli4	336	7	15.80
led02456789v1	443	7	10.97
page-blocks0	5472	10	8.79
ecoli3	336	7	8.60
yeast3	1484	8	8.10
new-thyroid1	215	5	5.14
new-thyroid2	215	5	5.14
vehicle3	846	18	2.99
vehicle1	846	18	2.90
vehicle2	846	18	2.88
glass0	214	9	2.06
pima	768	8	1.87
glass1	214	9	1.82
wdbc	569	30	1.68
spam	4597	57	1.54

For each dataset, 80% of the points were selected randomly for training and the remaining 20% is used for testing. The training set was used with the sampling algorithms to get the undersampled dataset. This undersampled dataset was used to train the models using the previously mentioned algorithms. We have also used the original training dataset in the empirical study and reported its outcomes. The remaining 20% test datapoints were used to compute the model performance. The above process was repeated 10 times and the average AUC and F1-Scores were reported and compared. The same training-test partitions were used for all the competing methods and run on the same platform .

## 5 Results

The results of the experiments are shown in Table 2 and 3 respectively. The values in the table are the mean AUC (Table 2) and mean F1-Score (Table 3) of the ten runs of the experiment, as mentioned in Section 4. The values in the parentheses are the relative ranking for a sampling method and algorithm combination on the specific dataset. For example, for *ecoli4* dataset, when the proposed algorithm is used with kNN, attained an AUC value of 0.9963 and an rank of 1, when compared with kNN used with other sampling methods and original dataset. The last row of each table shows the average rank over all datasets for a specific sampling algorithm and classification algorithm pair.

The objective is to the relative efficacy of RWMaU in learning imbalanced datasets as compared to the competing paradigms. The comparison is done for each sampling method and classifier pair. With respect to kNN classifier, on both metrics, RWMaU did very well compared to other undersampling methods as well as the original dataset. RWMaU has also performed best on C4.5 Decision Tree with an average rank of 1.48. Particularly, the difference of average ranks of RWMaU and the next based ranked method is remarkable. For (Bagging + C4.5), RWMaU achieved the lowest average rank on both minority class F1 and AUC scores. However, it is worth noting that the difference with respect to next best average rank was not that significant as for the previous two cases. In case of minority F1 score, the thresholding was done using 0.5, we find RWMaU performing better overall with respect to the different classifiers.

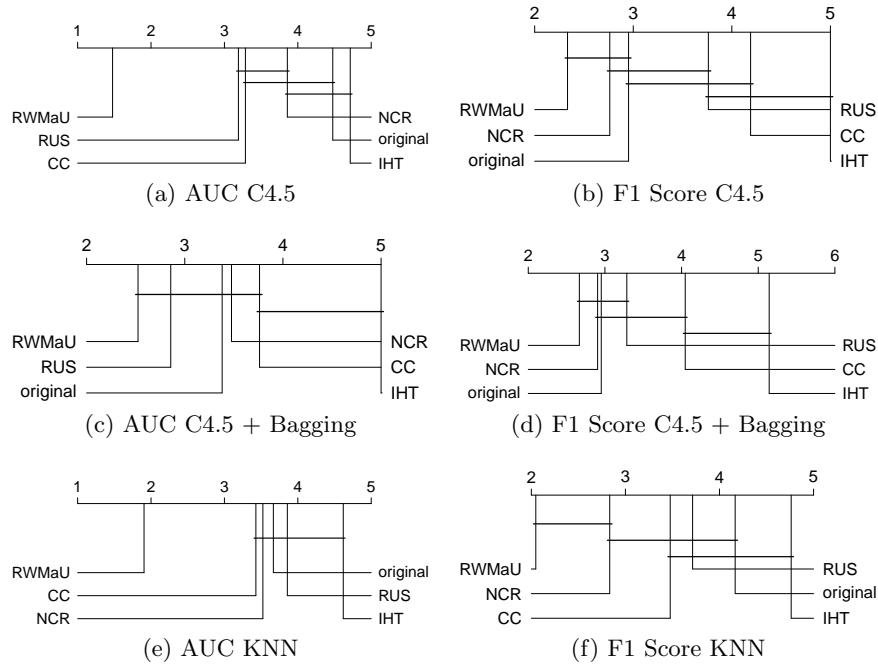


Fig. 1: Critical difference plots of the post-hoc Friedman test with significance level  $\alpha = 0.05$  for AUC and F1 Score from the KNN, C4.5 and C4.5 + Bagging experiments.

To further investigate, we have performed a statistical significance tests to understand the pairwise differences between the methods. The Friedman test with Finner  $p$ -value correction was performed to do a multiple classifier test on each combination of sampling method, used algorithm, on both F1-Score and



AUC scores for each sampling and classifier combination. The summary results are represented as critical difference plots in Figure 1<sup>4</sup>.

With respect to AUC scores, from the critical difference plots, it is clear that RWMaU performed significantly better than all the methods with used with C4.5 in Figure 1a and KNN in Figure 1e, with a significance level of  $\alpha = 0.05$ . Although the null hypothesis could not be rejected, except for IHT for C4.5 + Bagging case as can be seen in Figure 1c.

When we consider the same for F1 Score, RWMaU’s performance was found to be significantly better than RUS, CC and IHT with C4.5 (Figure 1b) and all other method except NCR with KNN (Figure 1f). On the other hand, for C4.5 + Bagging (Figure 1d), RWMaU was found to be significantly better than RUS, CC and IHT. In general, we may conclude that, RWMaU would improve the learning of the class-imbalanced datasets over the competing methods.

## 6 Conclusion

In this paper, we address the class imbalance problem by proposing an under-sampling method, Random Walk-steered Majority Undersampling (RWMaU). Our scheme re-balances the dataset by removing datapoints from the majority class. The main objective is to remove the majority points which are relatively closer to the minority class. The novelty of our method lies in the use of random walk visits to perceive the nearness of the points in a dataset. The majority class points which lie close to the minority class are subsequently undersampled. The AUC scores and the minority class F1 scores obtained from our empirical study show that RWMaU delivers improved performance over existing methods. RWMaU + KNN and RWMaU + C4.5 performed significantly better than all the other methods with respect to AUC scores, whereas they performed significantly better than most of the methods with respect to F1 score. Overall, RWMaU has attained the best rank in all cases with respect to both AUC and F1 scores. In future, we would like to design a minority-oversampling scheme which is built upon the random walks over the instances of a class-imbalanced dataset. It would also be interesting to integrate random walk based oversampling and undersampling in a single framework.

## References

1. Barua, S., Islam, M.M., Yao, X., Murase, K.: MWMOTE—majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering* **26**(2), 405–425 (Feb 2014)
2. Chakraborty, T., Chakraborty, A.K.: Hellinger net: A hybrid imbalance learning model to improve software defect prediction. *IEEE Transactions on Reliability* **70**(2), 481–494 (2021)

<sup>4</sup> The full result tables in supplementary material: [https://github.com/phoxis/rwmau/blob/main/RWmaU\\_ICONIP2021\\_Paper\\_Supplementary\\_Material.pdf](https://github.com/phoxis/rwmau/blob/main/RWmaU_ICONIP2021_Paper_Supplementary_Material.pdf).

3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.* **16**(1), 321–357 (Jun 2002)
4. Devi, D., Purkayastha, B., et al.: Redundancy-driven modified tomek-link based undersampling: A solution to class imbalance. *Pattern Recognition Letters* **93**, 3–12 (2017)
5. Elreedy, D., Atiya, A.F.: A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. *Information Sciences* **505**, 32–64 (2019)
6. Fernández, A., del Jesus, M.J., Herrera, F.: Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. *International Journal of Approximate Reasoning* **50**(3), 561–577 (2009), special Section on Bayesian Modelling
7. Fernández, A., García, S., del Jesus, M.J., Herrera, F.: A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems* **159**(18), 2378–2398 (2008), theme: Information Processing
8. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: 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)* **42**(4), 463–484 (2011)
9. Guo, H., Viktor, H.L.: Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *ACM Sigkdd Explorations Newsletter* **6**(1), 30–39 (2004)
10. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications* **73**, 220–239 (2017)
11. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). pp. 1322–1328. IEEE (2008)
12. Jamali, M., Ester, M.: Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 397–406 (2009)
13. Katzir, L., Hardiman, S.J.: Estimating clustering coefficients and size of social networks via random walk. *ACM Transactions on the Web (TWEB)* **9**(4), 1–20 (2015)
14. Kaur, H., Pannu, H.S., Malhi, A.K.: A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)* **52**(4), 1–36 (2019)
15. Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A., Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems* **29**(8), 3573–3587 (2017)
16. Krawczyk, B., Galar, M., Woźniak, M., Bustince, H., Herrera, F.: Dynamic ensemble selection for multi-class classification with one-class classifiers. *Pattern Recognition* **83**, 34–51 (2018)
17. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Conference on Artificial Intelligence in Medicine in Europe. pp. 63–66. Springer (2001)
18. Lin, W.C., Tsai, C.F., Hu, Y.H., Jhang, J.S.: Clustering-based undersampling in class-imbalanced data. *Information Sciences* **409**, 17–26 (2017)

19. Ling, C.X., Sheng, V.S.: Cost-sensitive learning and the class imbalance problem. *Encyclopedia of machine learning* **2011**, 231–235 (2008)
20. Lovász, L.: Random walks on graphs. *Combinatorics, Paul erdos is eighty* **2**(1-46), 4 (1993)
21. Maratea, A., Petrosino, A., Manzo, M.: Adjusted f-measure and kernel scaling for imbalanced data learning. *Information Sciences* **257**, 331–341 (2014)
22. Mena, L.J., Gonzalez, J.A.: Machine learning for imbalanced datasets: Application in medical diagnostic. In: *Flairs Conference*. pp. 574–579 (2006)
23. Mohammed, R., Rawashdeh, J., Abdullah, M.: Machine learning with oversampling and undersampling techniques: overview study and experimental results. In: *2020 11th International Conference on Information and Communication Systems (ICICS)*. pp. 243–248. IEEE (2020)
24. Ofek, N., Rokach, L., Stern, R., Shabtai, A.: Fast-cbus: A fast clustering-based undersampling method for addressing the class imbalance problem. *Neurocomputing* **243**, 88–102 (2017)
25. Peng, Y., Yao, J.: AdaOUBoost: adaptive over-sampling and under-sampling to boost the concept learning in large scale imbalanced data sets. In: *Proceedings of the international conference on Multimedia information retrieval*. pp. 111–118 (2010)
26. Ramentol, E., Caballero, Y., Bello, R., Herrera, F.: SMOTE-RS B\*: a hybrid pre-processing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. *Knowledge and information systems* **33**(2), 245–265 (2012)
27. Ribeiro, V.H.A., Reynoso-Meza, G.: Ensemble learning by means of a multi-objective optimization design approach for dealing with imbalanced data sets. *Expert Systems with Applications* **147**, 113232 (2020)
28. Sadhukhan, P.: Learning minority class prior to minority oversampling. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. pp. 1–8 (2019)
29. Smith, M.R., Martinez, T., Giraud-Carrier, C.: An instance level analysis of data complexity. *Machine learning* **95**(2), 225–256 (2014)
30. Tang, B., He, H.: Kerneladasyn: Kernel based adaptive synthetic data generation for imbalanced learning. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*. pp. 664–671. IEEE (2015)
31. Wheelus, C., Bou-Harb, E., Zhu, X.: Tackling class imbalance in cyber security datasets. In: *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. pp. 229–232. IEEE (2018)
32. Wong, M.L., Seng, K., Wong, P.K.: Cost-sensitive ensemble of stacked denoising autoencoders for class imbalance problems in business domain. *Expert Systems with Applications* **141**, 112918 (2020)
33. Zhang, C., Tavanapong, W., Kijkul, G., Wong, J., De Groen, P.C., Oh, J.: Similarity-based active learning for image classification under class imbalance. In: *2018 IEEE International Conference on Data Mining (ICDM)*. pp. 1422–1427. IEEE (2018)
34. Zhang, H., Li, M.: Rwo-sampling: A random walk over-sampling approach to imbalanced data classification. *Information Fusion* **20**, 99–116 (2014)
35. Zhang, Y.P., Zhang, L.N., Wang, Y.C.: Cluster-based majority under-sampling approaches for class imbalance learning. In: *2010 2nd IEEE International Conference on Information and Financial Engineering*. pp. 400–404. IEEE (2010)
36. Zheng, Z., Wu, X., Srihari, R.: Feature selection for text categorization on imbalanced data. *SIGKDD Explor. Newsl.* **6**(1), 80–89 (Jun 2004)

