

# Methods of Handling Unbalanced Datasets in Credit Card Fraud Detection

Elena-Adriana MÎNĂSTIREANU<sup>1</sup>,  
Gabriela MEȘNIȚĂ<sup>2</sup>

<sup>1</sup>PhD student, "Alexandru Ioan Cuza"  
University of Iași, Doctoral School of  
Economics and Business Administration, Iași,  
700057, Romania, [adrianastan3@gmail.com](mailto:adrianastan3@gmail.com)

<sup>2</sup>"Alexandru Ioan Cuza" University of Iași,  
Faculty of Economics and Business  
Administration, Business Information Systems  
Department, Iași, Romania,  
[gabriela.mesnita@feaa.uaic.ro](mailto:gabriela.mesnita@feaa.uaic.ro)

**Abstract:** *Nowadays fraudulent transactions of every type represent a major concern in the financial industry due to the total amount of money that are lost every year. Manually analyzing fraudulent transactions is unfeasible if we think at the huge amount of data and the complexity of bank fraud in the digitization era. In this context, the problem to detect the fraud can be achieved by machine-learning algorithms due to their ability of detecting small anomalies in very large datasets. The problem that arise here is that the datasets are highly unbalanced meaning that the non-fraudulent cases heavily dominates the fraudulent ones. In this paper, we are going to present three ways of handling unbalanced datasets by: resampling methods (undersampling and oversampling), cost-sensitive training and tree algorithms (decision tree, random forest and Naïve Bayes), emphasizing the idea of why the Receiver Operating Characteristics curve (ROC) should not be used on this type of datasets when measuring the performance of the algorithm. The experimental test was applied on a number of 890,977 banking transactions in order to observe the performance metrics of all the three methods mentioned above.*

**Keywords:** *bank fraud; machine-learning algorithms; resampling; cost-sensitive training; unbalanced dataset.*

**How to cite:** Mînăstireanu, E.-A., & Meșniță, G. (2020). Methods of Handling Unbalanced Datasets in Credit Card Fraud Detection. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 11(1), 131-143.  
<https://doi.org/10.18662/brain/11.1/19>

## 1. Introduction

During the last decades, fraudulent transactions brought losses of billions of dollars every year, forcing in this way financial institutions to continuously improving their systems for loss reduction and as a consequence to this, combating fraud became a popular topic to explore. The actions against bank frauds remain divided into fraud prevention actions and fraud detection actions. Fraud prevention actions consist of a set of principles, procedures and rules developed in order to stop fraud from occurring. On the other hand, the dynamics and the emergence of the new typologies of fraud require to identify new fraud detection action. This happens since delinquents are always looking for new ways and schemes to commit fraud. Thus, the problem of combating fraud by developing complex decision-making systems remains critical and complex, taking into consideration that financial institutions are collecting daily huge information from o series of sources. This action raises another issue, that of detecting a rare but important case from a huge amount of data. In real-world domains this refers to high unbalance problem which got more and more emphasis in the last couple of years. In order to resolve this problem, different authors have been found different solutions both for data and algorithm. At the data level (Chawla et al., 2003), these solutions include techniques like oversampling with replacement, random undersampling, directed oversampling and undersampling, oversampling with informed generation of new samples. At the algorithmic level (Provost & Fawcett, 2001), these include techniques of adjusting as follows: the costs of the various classes, the probabilistic estimate at the tree leaf, the decision threshold and recognition-based rather than discrimination-based learning. In this paper we are going to describe in a detailed manner three ways of handling unbalanced data by resampling, cost-sensitive training and tree algorithms.

The paper is structured as follows. The first part will analyse the background of high unbalanced data based on literature review. In the second part we will present the methodology of the research and the results from the test concerning the performance of the tree algorithms.

The main goal of the paper is to present different types of methods to deal with highly unbalanced data and some performance metrics regarding the tree algorithms used in fraud detection.

## 2. Background and literature review

Handling the class unbalanced problem has become a common issue whereas implementing machine-learning algorithms to the actual problems. A data set is unbalanced when there is a considerable disparity in the numbers of positive and negative instances, frequently with the positive instances being more numerous than the negative instances (Chawla et al., 2004; Chawla et al., 2002; Rao et al., 2006; Kubat et al., 1998). Major studies made around this problem concentrated especially on evaluation metrics and classification techniques. In literature the common measures applied to assess the performance of a classification method are as follow:

- Accuracy and error rate: these measure the general efficiency of the algorithm. This is made by assessing the proportion of correctly (accuracy) instances and those incorrectly (error rate). They are not appropriate to unbalanced datasets because they are focused more on the majority class.

- Precision, Recall and F-measure: The first one determines how good the classifier is in detecting the fraudulent cases, as it takes into account the proportion between the cases with true positive attribute and the sum between those true and false positive. The second one evaluates the quality of a qualifier in order to not omit instances that should framed into the label. The last one mixes the first two measure to qualify the quality of a classifier for the occasional classes (Van Rijsbergen, 1979).

- Gmean (Geometric mean): this type of measure is used to evaluate the performance of a classifier to create a balance between the minority and majority classes.

However, the measure technique used, the principal characteristic of the algorithm consists in getting a high percentage of correct samples detected in the minority class and a small error percentage in the majority class. The Receiver Operating Characteristic (ROC) curve represents a standard technique used for evaluating the tradeoffs between true positive and false positive error rates in the case of classification algorithms. While the Area Under the Curve (AUC) represents the area that exists under a ROC curve. In the opinion of Provost and Fawcett, the ROC convex hull can be used as a method of “identifying potentially optimal classifiers”. As stated by the authors, the significance of this consists in the fact that “if a line passes through a point on the convex hull, then there is no other line with the same slope passing through another point with a larger true positive intercept. Thus, the classification algorithm at that point is optimal under any distribution presumption in tandem with the slope” (Provost & Fawcett, 2001).

In order to handle the unbalanced problem several methods have been proposed. In this context we find in the literature many studies including that of Chawla and colab., who proposed a Synthetic Minority Oversampling technique or SMOTE, for short, by generating synthetic data at random taking into account the similarities that exists between the minority samples and the K-nearest neighbors of each minority sample. As stated by the authors, the advantage SMOTE technique is that “it maximizes the performance of the classifier and the learning biased as against the minority class”. However, this technique has some drawbacks, among which we can underline the fact that this technique is “applicable only for binary class problems” (Chawla et al., 2002).

Fernandez-Navarro et al. (2011) suggested two types of oversampling techniques: “a static SMOTE radial basis function method and a dynamic SMOTE radial basis function procedure” that was integrated into an algorithm of the mimetic type in order to optimize the radial basis functions neural networks. The experiments highlighted an improvement of the sensitivity in the generalization set and a high level of accuracy regarding the class classification.

Kerdprasop and Kerdprasop (2012) proposed a combination between random oversampling, SMOTE techniques and the following algorithms SVM, neural network, decision tree induction, regression analysis to get an improvement regarding the performance of the results obtained by the learned model. Furthermore, in order to get an improvement in the predicting accuracy they made use of a technique “based on a cluster feature selection”.

Seiffert et al. (2014), in their paper regarding classification performance in the imbalanced problems, used distinct classifiers including neural networks, decision tree, K-nearest neighbors, and Naïve Bayes. In their experiment they reviewed "the relationship between data sampling, classification performance, learner selection, and class imbalance and noise". Their conclusion was that less noise can have a significant impact on the performance of the sampling technique.

Hulse and Khoshgoftaar (2009) stated that the impact of noise is highly determined by the complexity of algorithm whilst simple classification algorithms like "Naïve Bayes and KNN are often more robust than more complex classification algorithms like random forests or SVM". Moreover, they emphasized the fact that the technique increases the “performance of class imbalance and noise classifiers”.

Oversampling and undersampling represents effective techniques of dealing with unbalanced data sets. Undersampling technique has as goal to

equilibrate class distribution through the random rejection of majority class samples, while oversampling aims to balance the distribution of classes by random replication of minority class samples. Chawla et al. (2002) state that oversampling “can increase the likelihood of occurring overfitting, since it makes exact copies of the minority class examples”. However, undersampling offers better results than oversampling when used on large domains. In a study made by Liu et al. (2010) results showed that oversampling techniques performs better than undersampling in the case of local classifiers whilst some undersampling techniques outperform oversampling in the case of classifiers that make use of global learning. Kotsiantis and Pintelas (2003) developed an “Agent-based Knowledge Discovery (ABKD) method” that combines three entities called agents (the first agent is used to learn using Naive Bayes, the second one learns using C4.5 and the third one learns using 5NN) on a cleaned version of training data. The agent’s predictions are then combined according to a certain voting scheme. The main objective of the method is to achieve different results for the detected errors through using different types of algorithms.

In many cases of unbalanced, both the distribution of data is modified, and the cost of misclassification errors is variable. “The cost sensitive learning considers the misclassification cost through assigning higher cost of misclassification to the positive class and provides the model with lowest cost” (Sun et al., 2007). However, the misclassification errors costs are often hidden and in this case cost sensitive learning may cause the appearance of overfitting (Biodgloi & Parsa, 2012). Another cost sensitive proposed in the literature (Uyar et al., 2010) is to adjust the “decision threshold of the machine learning techniques where the selection of threshold can be considered as an effective factor that influences the performance of the learning algorithms”. In the study of Weiss et al. (2007), results obtained concluded that cost sensitive learning technique performs more better than the sampling methods. The literature (Nguyen et al., 2009; Haibo & Edwardo, 2009; Chris & Robert, 2000; Charles et al., 2004) presents several ways of incorporating cost into decision tree classification, like: one “cost can be used in order to tune the decision threshold, another one can be applied in splitting attribute selection in the construction process of the decision tree, and another technique that can be considered consists in applying to the tree the cost sensitive pruning schemes”. Charles et al. (2004) proposed a method that can be used for building and testing decision trees that can minimize “the total sum of the misclassification and test costs”. The algorithm used is based on a splitting attribute that “minimizes the total cost, the sum of the test cost and the misclassification cost”.

### 3. Research methodology

For this experiment we used a public database Kaggle that contains information about transaction made by the European owners of credit cards in September 2013 (Kaggle, 2003). The chosen data set presents two-day transactions with 492 frauds.

The data set contains numeric variables that are the result of the Principal Component Analysis (PCA) algorithm used as normalization technique. Due to confidentiality issues, the original information about this data cannot be provided, thus these features are labeled with V1 to V21. In this public data set, 'Time' (transaction time) and 'Amount' (transaction amount) are the features that have not been converted by the Principal Component Analysis (PCA) algorithm. Also, there is a Class property which represents the response variable and takes 1 for fraud cases and 0 for genuine transactions. Due to this response variable, the data is extremely unbalanced, with only 0.172% of transactions having Class = 1. For handling this unbalanced issue, we will apply over the public data sets three methods:

- resampling where we are going to undersample the majority class and oversample the minority class through undersampling and oversampling;
- cost-sensitive learning where we are going to use penalized random forest;
- tree algorithms where we will use AUC precision recall curve as a performance metric:

$$AUC = \frac{TPrate + TNrate}{2}$$

In this step we will analyze all the three models (decision tree, random forest and Naïve Bayes loaded from Scikit-learn) with their respective:

- recall score ( $Recall = \frac{TP}{TP+FN}$  also called True Positive Rate (TPR), sensitivity or hit rate) refers to the amount of fraud cases our model is able to detect

- precision score ( $Precision = \frac{TP}{TP+FP}$  also called Positive Predicted Value (PPV)) refers to how precise is the model in detecting fraud transactions

- $F_\beta$  score ( $F_\beta = (1 + \beta^2) \frac{Precision \times Recall}{(\beta^2 \times precision) + Recall}$ ) =  $\frac{(1 + \beta^2) \times TP}{(1 + \beta^2) \times TP + \beta^2 \times FN + FP}$  ; the  $\beta$  parameter determines the weight of the

precision in the combined score,  $\beta < 1$  means more weight to precision,  $\beta > 1$  favors recall. For this experiment  $\beta = 0.5$  in order to not misclassify the normal cluster as fraud and to favor precision;

Where:

- TP = true positive referring to the number of positive cases which are predicted positive – meaning correctly classified fraud transactions

- TN = true negative referring to the number of negative cases which are predicted negative – meaning correctly classified non-fraud transactions

- FP = false positive referring to the number of negative cases which are predicted positive – meaning incorrectly classified fraud transactions

- FN = false negative referring to the number of positive cases which are predicted negative – incorrectly classified non-fraud transactions

And choose the model based upon the  $F_\beta$  score. The chosen model will then be optimized and used as final model in which we will plot the AUC precision recall curve.

In order to apply the resampling methods – undersampling and oversampling – we first needed to prepare our data. For this we applied a logarithmic transformation on the data in order to handle the highly skewed feature distributions. This logarithmic transformation ensures that the very large and very small values do not negatively affect the performance of the learning algorithms. Also significantly reduces the range of values caused by outliers. After this we normalized the Amount feature within 0 to 1 range and applied the oversampling method.

Oversampling represents a sampling method which “balances the data set through the replication of the samples of minority class”. The advantage is that no useful information will be lost as we will see in the undersampling technique and the disadvantage is that it may lead to “overfitting and high computational cost if the data set is already very large and unbalanced” (Guo et al., 2008; Kotsiantis et al., 2006). In the experiment all data points from the majority and minority training sets were used. Instances were randomly selected and replaced with data from the minority training set until we reached the expected balance of data. The results obtained are as follow:

- Recall = 0.91
- Precision = 0.97
- $F_\beta = 0.92$

The oversampling using SMOTE (Synthetic Minority Oversampling Technique) technique lead to the following results:

- Recall = 0.91

- Precision = 0.97
- $F_{\beta} = 0.92$

The SMOTE technique is based on finding the nearest neighbor of minority samples, taking their difference and multiplying this by a random number. Thus, it helps to increase the model accuracy.

Undersampling eliminates samples from the majority class in order to obtain a balanced dataset. The advantage is that the method can be used with efficiency in the case of large-scale applications, due to the numerous majority class samples. The technique has an important weakness because it can remove some information with potentially that that would be relevant to the classifiers (Nguyen et al., 2009; Kotsiantis et al., 2006). In the experiment, for this method we used all the training data points from the minority class. Additionally, samples were removed based on random process from the majority training set. This process have been repeated until the needed balance was achieved. The results obtained are as follow:

- Recall = 0.89
- Precision = 0.95
- $F_{\beta} = 0.90$

Unbalanced datasets can be handled by ensemble algorithms, penalized algorithms and tree algorithms separately. In this experiment we combined all these three algorithms in a single algorithm using Random Forest Classifier. This has decision tree as the base learner and has a parameter called 'class-weight'. Setting this parameter to 'balanced', weights inversely proportional to the class sizes are used to multiply the loss function. This modification uses cost sensitive learning, meaning that a penalty towards classifying accurately the majority class is added, so correct predictions from the minority class have a higher weight. For this algorithm the results obtained are as follow:

- Recall = 0.71
- Precision = 0.94
- $F_{\beta} = 0.75$

The results obtained for the decision tree algorithm without resampling the data are as follow:

- Recall = 0.76
- Precision = 0.82
- $F_{\beta} = 0.77$

The results obtained for the Naïve Bayes algorithm are as follow:

- Recall = 0.83
- Precision = 0.06
- $F_{\beta} = 0.07$

To summarize the results obtained it can be stated that the classifier that uses oversampling with SMOTE techniques has given the best performance metrics. Also, from the tree algorithms, the random forest classifier has given the best precision in detecting frauds with a precision of 94%.

To assess the overall classification performance, we made use of the area under the curve metric (AUC). AUC precision recall curve is not biased against the minority class meaning that it does not focus on the use of one class than the other one. It represents the existent compromise between precision and recall for different threshold. Average accuracy states that this “plot acts as the weighted mean of precision obtained at each threshold, with an increase in the recall from the previous threshold used as weight”. In our experiment the best threshold for the classifier should be around 0.85.

For our study we achieved 93% of area for AUC PR curve. A high value for the area under the curve presents both low false negative rate (FN rate) or high recall and low false positive rate (FP rate) or high precision.

High recall or low false negative results and high precision or low false positive results indicate that the classification algorithm returns accurate results. To sum up, we can say that a high performing system with both high metrics (FN rate and FP rate) will predict a large number of fraudulent transactions with very high precision and accuracy.

#### 4. Results and discussion

In this study we presented three ways of handling unbalanced data: resampling methods (undersampling and oversampling), cost-sensitive training and tree algorithms (decision tree, random forest and Naïve Bayes). The resampling methods and the tree algorithms have been loaded from the Scikit-learn and analysed based on the results obtained in the  $F_{\beta}$  score.

Out of the three methods that were used in the experiment, only the oversampling with SMOTE techniques has given the best performance metrics. This appear in literature as being the method of choice among the many available methods (Abdellatif et al., 2018; Ramentol et al., 2012; Mi, 2013) when it comes to handling unbalance data. Also, the literature states (Gaoa et al., 2011; Apurva & Patankar, 2015) that this method presents as major advantages the following: independent on underlying classifier and

very easy to implement and the following limitations: time consuming by introducing additional computational cost and overfitting. When it comes to the other methods used in the experiment, the:

- undersampling presented as an advantage the fact that this method is suitable for large scale applications and as a disadvantage the loss of some useful information through the process of removing significant patterns;
- cost-sensitive presented as an advantage the “minimization of the misclassification cost through affecting the classifier as against the minority class”, and as a disadvantage the fact that the misclassification costs are often unknown;
- tree algorithms presented as advantages the fact that working together offers high performing classification results and high resistance to noise, and as disadvantages time consuming and overfitting.

The overall classification performance was based on the results offered by AUC PR curve, which represents a convenient method to compare the performance of multiple classifiers. The results obtained in the experiment shows that the AUC PR curve measures correct ratio of FP to TP, whereas AUC of ROC does not measure the true output in high unbalance ratios. ROC curve is not a good visual illustration for highly unbalanced data, because the false positive rate ( $FPR = \frac{FP}{TRN}$ ) does not decreases drastically when the total of real negative cases is huge. Whereas precision score is highly sensitive to false positives. Also, the literature (Swamidass et al., 2010) highlights that ROC curve can offer inappropriate results and requires special attention when the dataset is highly unbalanced and there are two ROC curves that are crossing one another. In another study (Saito & Rehmsmeier, 2015) we find out that the AUC metric is much better than an original ROC curve because there can be some data points that can be missed from the ROC curve.

As a future work the research direction is to build a new classifier which will perform better in this data unbalanced problem as the existing classifier.

## 5. Conclusions and future direction

Data unbalance represents an important topic that has been investigated over the time by machine-learning researchers. In this way several approaches have been proposed. However, there is no general solution for this issue since every method comes with its own advantages and disadvantages.

With regards to the future researches, it is necessary to explore and implement a new classifier that will outperform the existing one, moving to hybrid algorithms.

---

## References

---

- Abdellatif, S., Ben Hassine, M. A., Ben Yahia, S., & Bouzeghoub, A. (2018) ARCID: A New Approach to Deal with Imbalanced Datasets Classification. In: Tjoa A., Bellatreche L., Biffl S., van Leeuwen J., Wiedermann J. (Eds.) SOFSEM 2018. Lecture Notes in Computer Science, vol 10706. *SOFSEM 2018: Theory and Practice of Computer Science* (pp. 569-580). Edizioni della Normale, Cham.
- Apurva, S., & Patankar, R. A. (2015). A survey on methods to handle imbalance dataset. *International Journal of Computer Science and Mobile Computing*, 4(11), 338-343. Retrieved from <https://ijcsmc.com/docs/papers/November2015/V4I11201573.pdf>
- Biodgloi, A. M., & Parsa, M. N. (2012). A hybrid feature selection by resampling, Chi squared and consistency evaluation techniques. *World Academy of Science, Engineering and Technology, International Journal of Computer and Information Engineering*, 6(8), 957-966. Retrieved from <https://zenodo.org/record/1060641#.Xlj1FKgzaM8>
- Charles, X. L., Qiang, Y., Jianning, W., & Schichao, Z. (2004). Decision trees with minimal costs. In *Proceedings of the 21<sup>st</sup> International Conference on Machine Learning (ICML 2004)*. Banff, Canada. Retrieved from <https://icml.cc/Conferences/2004/proceedings/papers/136.pdf>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1), 321–357. <https://doi.org/10.1613/jair.953>
- Chawla, N. V., Kolcz, A., & Japkowicz, N. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1), 1-6. <https://doi.org/10.1145/1007730.1007733>
- Chawla, N.V., Lazarevic, A., Hall, L.O., & Bowyer, K. W. (2003). Smoteboost: Improving prediction of the minority class in boosting. In *Proceedings of the 7th European Conference on Principles and Practices of Knowledge Discovery in Databases* (pp. 107-119). Cavtat-Dubrovnic, Croatia: Academic Press.
- Chris, D., & Robert, C. H. (2000). Exploiting the Cost(In) sensitivity of Decision Tree Splitting Criteria. ICML. Retrieved from [https://www.researchgate.net/publication/2626981\\_Exploiting\\_the\\_Cost\\_Insensitivity\\_of\\_Decision\\_Tree\\_Splitting\\_Criteria](https://www.researchgate.net/publication/2626981_Exploiting_the_Cost_Insensitivity_of_Decision_Tree_Splitting_Criteria)
- Fernandez-Navarro, F., Hervas-Martinez, C. and Gutierrez, P. A. (2011). A dynamic over-sampling procedure based on sensitivity for multi-class

- problems. *Pattern Recognition*, 44(8), 1821-1833. Retrieved from <http://ccc.inaoep.mx/~ariel/2012/A%20dynamic%20over-sampling%20procedure%20based%20on%20sensitivity%20for%20multi-class%20problems.pdf>
- Gaoa, M., Hong, X., Chen, S. & Harris, C. J. (2011). A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems, *Neurocomputing*, 74(17), 3456–3466. <https://doi.org/10.1016/j.neucom.2011.06.010>
- Guo, X., Yilong, Y., Cailing, D., Gongping, Y., & Yang, Z. (2008). On the class imbalance problem. *Fourth International Conference on Natural Computation, ICNC '08*, volume 4. <https://doi.org/10.1109/icnc.2008.871>
- Haibo, H., & Edwardo, A. G. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9). Retrieved from <https://www.cs.utah.edu/~piyush/teaching/ImbalancedLearning.pdf>
- Hulse, J. V., & Khoshgoftaar, T. (2009). Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, 68(12), 1513-1542. <https://doi.org/10.1016/j.datak.2009.08.005>
- Kaggle. (2003). The Home of Data Science & Machine Learning. Retrieved from <https://www.kaggle.com/agpickersgill/credit-card-fraud-detection/data>
- Kerdprasop, N. & Kerdprasop, K. (2012). On the generation of accurate predictive model from highly imbalanced data with heuristics and replication technologies. *International Journal of Bio-Science and Bio-Technology*, 4(1), 49-64. Retrieved from <https://www.earticle.net/Article/A207028>
- Kotsiantis, S. B. & Pintelas, P. E. (2003). Mixture of expert agents for handling imbalanced data sets. *Annals of Mathematics, Computing & TeleInformatics*, 1(1), 46-55. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.2997&rep=rep1&type=pdf>
- Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30, 25-36.
- Kubat, M., Holte, R.C. & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30, 195–215. Retrieved from <https://link.springer.com/article/10.1023/A:1007452223027>
- Liu, W., Chawla, S., Cieslak, D. A. & Chawla, N. V. (2010). A robust decision tree algorithm for imbalanced data sets. *Proceedings of the SIAM International Conference on Data Mining, SDM 2010*, April 29 - May 1, 2010, Columbus, Ohio, USA. <https://doi.org/10.1137/1.9781611972801.67>
- Mi, Y. (2013). Imbalanced classification based on active learning SMOTE. *Research Journal of Applied Science, Engineering and Technology*, 5(3), 944-949. <https://doi.org/10.19026/rjaset.5.5044>

- Nguyen, G. H., Bouzerdoum, A., & Phung, S.L. (2009). Learning pattern classification tasks with imbalanced data sets. In P. Yin (Ed.), *Pattern recognition* (pp. 193-208). Vukovar, Croatia: In-Teh.
- Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, 42, 203-231.
- Ramentol, E., Verbiest, N., Bello, R., Caballero, Y., Cornelis, C., & Herrera, F. (2012). SMOTE-FRST: A new resampling method sing fuzzy Rough Set Theory. In *World Scientific Proceedings Series on Computer Engineering and Information Science. Uncertainty Modelling in Knowledge Engineering and Decision Making* (pp. 800-805). [https://doi.org/10.1142/9789814417747\\_0128](https://doi.org/10.1142/9789814417747_0128)
- Rao, R. B., Krishnan, S. & Niculescu, R. S. (2006). Data mining for improved cardiac care. *ACM SIGKDD Explorations Newsletter*, 8(1), 3–10. <https://doi.org/10.1145/1147234.1147236>
- Saito, T. & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3). <https://doi.org/10.1371/journal.pone.0118432>
- Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J. & Folleco, A. (2014). An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences*, 259, 571-595. <https://doi.org/10.1016/j.ins.2010.12.016>
- Sun, Y., Kamel, M. S., Wong, A. K. C. & Wang, Y. (2007). Cost sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12), 3358-3378. <https://doi.org/10.1016/j.patcog.2007.04.009>
- Swamidass, S. J., Azencott, C. A., Daily, K. & Baldi, P. (2010). A CROC stronger than ROC: Measuring, visualizing and optimizing early retrieval. *Bioinformatics*, 26(10), 1348–1356. <https://doi.org/10.1093/bioinformatics/btq140>
- Uyar, A., Bener, A., Ciracy, H. N., & Bahceci, M. (2010). Handling the imbalance problem of IVF implantation prediction. *IAENG International Journal of Computer Science*, 37(2). Retrieved from <https://pdfs.semanticscholar.org/7c3e/7b7fcfb7c1246a4cd7f0a401a60d9479a22a.pdf>
- Van Rijsbergen, C. J. (1979). *Information Retrieval* (2<sup>nd</sup> ed.). Butterworths-Heinemann.
- Weiss, G. M., McCarthy, K., & Zabar, B. (2007). Cost-sensitive learning vs sampling. Which is best for handling unbalanced classes with unequal error costs? *Proceedings of the 2007 International Conference on Data Mining, DMIN 2007*, June 25-28, 2007, Las Vegas, Nevada, USA.