Predicting Material Backorders in Inventory Management using Machine Learning

Rodrigo Barbosa de Santis
Graduate Program in Computational Modelling
Federal University of Juiz de Fora
Juiz de Fora, Brazil
rodrigo.santis@engenharia.ufjf.br

Eduardo Pestana de Aguiar
Department of Mechanic and Industrial Engineering
Federal University of Juiz de Fora
Juiz de Fora, Brazil
eduardo.aguiar@engenharia.ufjf.br

Leonardo Goliatt
Department of Computational and Applied Mechanics
Federal University of Juiz de Fora
Juiz de Fora, Brazil
leonardo.goliatt@ufjf.edu.br

***Abstract*—Material backorder is a common supply chain problem, impacting an inventory system service level and effectiveness. Identifying parts with the highest chances of shortage prior its occurrence can present a high opportunity to improve an overall company's performance. In this paper, machine learning classifiers are investigated in order to propose a predictive model for this imbalanced class problem, where the relative frequency of items that goes into backorder is rare when compared to items that do not. Specific metrics such as area under the Receiver Operator Characteristic and precision-recall curves, sampling techniques and ensemble learning are employed in this particular task.**

***Keywords*—Supply chain management, inventory planning and control, imbalanced learning, sampling methods, ensembles of classifiers**

I. Introduction

Artificial intelligence and big data has disruptively changed the industry, as the barriers of its implementation (cost, computing power, open-source platforms, etc) disappear. In this context, machine learning is applied on the design and development of predictive models which assess all areas of management, providing essential insights for companies to understand and react to changes in its operation.

A subject profoundly discussed in supply chain management is the inventory planning, which is an essential activity for any enterprise which tries to determine the decision about when to order and how much should order, considering different mechanisms of control [1]. Most of the approaches proposed so far formulate the problem as a multi-objective optimization one: ordering and storages costs must be held to a minimum, while service level is leverage as higher as possible.

A different approach for managing the inventory more efficiently - and complementary to the models developed in literature - is to identify the materials at risk of backorder before the event occurs, conferring the business a suitable time to react. A complication uprises in this particular kind of supervised learning application, since in regular inventory system the number of items which goes on backorder (positive or majority class) is utterly inferior to the amount of active items (negative or minority class). This case is known as the class imbalance problem [2] and it is common in many other real problems from telecommunications, web, finance-world, ecology, biology, medicine, among others, and requires appropriate techniques for handling the construction of the prediction model desired.

This paper proposes the application of a supervised learning model for backorder prediction in inventory control, based on the combination of sampling methods and ensemble of learning classifiers, and present results obtained in a real case study.

The major contributions of the paper are stated as follows:

- Comparison of different learning classifiers algorithms, based on specific techniques to tackle the class imbalanced problem, such as sampling and ensembles of classifiers;
- Providing a common framework for model development, testing and evaluation, in the considered detection system design;
- Achievement of 0.9482 ± 0.0025 AUC score, adopting *gradient tree boosting* model.

The major conclusions are as follows:

- Ensemble learning achieved greater scores than single classifiers, whilst sampling techniques usage did not generate benefits except when combined with ensembles;
- The proposed predictive machine exhibited high-potential of increasing service level in real inventory management systems;
- *Blagging*, a combination of under-sampling and tree ensemble, has shown more likelihood of being adopted in practice application considered its precision-recall curve and potential of enhancement.

This paper is organized into 5 sections. Following this Introduction, Section 2 provides a review on the state-of-art techniques for imbalanced learning. Section 3 provides a background of the dataset and methods applied in this study. Section 4 exhibits the results and discussions obtained from the methods application. Finally, Section 5 concludes this paper and includes some future works recommendations.

The current research is supported by Brazilian Federal Agency for Post-graduate Education (CAPES).

978-1-5386-3734-0/17/$31.00

II. Literature Review

A. The Imbalanced Classes Problem

In supervised learning, a dataset is said to be imbalanced when the number of instances of a given class of interest is rare when compared to the other (or others, in the case of multi-class problems). This is a problem of interest to research since there are many of classification problems of this nature in real-world, such as remote-sensing, pollution detection, risk management, fraud detection and medical diagnosis [2]–[4].

The balance ratio between the minority class and majority class in such applications may achieve distributions on the order of 1:100, 1:1,000 and 1:10,000. Standard learning classifiers trained using accuracy commonly perform poor results, ignoring minority classes which are treated as noise.

Several factors can increase the complexity of the imbalanced problem: the presence of small disjuncts groups of positive samples, classes overlapping hardening the induction of discriminative rules and the insufficiency of minority class examples. To deal with this particular circumstances, several techniques have been developed and categorized into three groups according to how they address the problem [3]. *Internal* approaches provide modifications to existing classifier learning algorithms to favor the learning of positive classes; *external* approaches are applied in data level to adjust classes distributions prior the application of the classifiers; *cost-sensitive learning* framework lies between internal and external approaches, since applies both data transformations (establishing misclassification costs to instances) and algorithm level adaptation by considering costs during the training process.

To evaluate the predictive learning systems developed adopting the proposed imbalanced methods framework, specific assessment metrics are necessary, further explored later in this section.

B. Assessment Metrics in Imbalanced Domains

Selecting the right evaluation metrics is a key determinant for guiding the construction of a predictive model. In a binary classification problem, the confusion matrix (shown in Table I) records the results of correctly and incorrectly recognized samples of each class.

TABLE I
Confusion Matrix in a Binary Classification Problem

	Positive prediction	Negative prediction
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

The accuracy rate (Eq. 1) has been the most usually applied empirical measure in classification. Nonetheless, in the framework of imbalanced datasets, accuracy is no longer a suitable metric, since it does not distinguish between the number of correctly classified examples of different classes. A typical example is an estimator which classifies all examples as negatives, leading to equivocated conclusions.

$$Acc = \frac{T_p + T_n}{T_p + F_n + F_p + T_n} \tag{1}$$

Several specific metrics are proposed within imbalanced problems domain in order to take into account the class distribution: precision, defined by (2), express the accuracy of an estimator when predicting the positive class, while recall (3), also known as true positive rate or sensitivity, indicates its ability of finding all the positive samples.

$$P = \frac{T_p}{T_p + F_p} \tag{2}$$

$$R = \frac{T_p}{T_p + F_n} \tag{3}$$

Precision-recall curves represent the conflict existing between both metrics and are commonly used in binary classification to understand the output of a classifier and aid the choice of the decision function threshold.

Another metric of interest obtained from the confusion matrix analysis is the fall-out (4), or the false positive rate, which is the number of false positives divided by the total number of negatives.

$$F = \frac{F_p}{F_p + T_n} \tag{4}$$

A standard approach used to evaluate classification models in imbalanced problems is to use the Receiver Operating Characteristic (ROC). Likewise precision-recall curve, this graphic allows the visualization of the trade-off between the precision and fall-out, as it evidences that any classifier cannot increase the number of true positives without also increasing the false positives. The Area Under the ROC Curve (AUC) corresponds to the probability of correctly identifying which one of the two stimuli is noise and which one is signal plus noise. AUC provides a single measure of a classifier's capability of evaluating which model is better on average and can be computed by:

$$AUC = \frac{1 + P - F}{2} \tag{5}$$

Other metrics can be used in classifiers evaluation, although AUC has been one of the most applied in literature for assessment and benchmark reference.

C. Sampling Methods and Ensembles in Imbalanced Datasets

Essentially, the idea of transforming an uneven set of classes into a balanced distribution may seem a reasonable solution for the imbalanced problem. This general idea led to the development of several techniques, known as sampling methods, which are grouped into two major groups: *under-sampling*, in which instances of the majority class are eliminated to adjust balance; or *over-sampling*, in which instances of the minority class are replicated to meet the majority one. The use of this method is justified by a verified improvement in overall classification performance in balanced datasets when

compared to imbalance datasets [4]. The major advantage of using these techniques is that they can be combined with any desired classifier.

Several sampling approaches have been employed so far: random replicating (or eliminating) instances from the classes, informed under-sampling intend to overcome the deficiency of information loss by deciding specific rules to determine what instances of majority class are going to be abandon, synthetic sampling seeks to create artificial data based on the similarities between the existing minority examples, data cleaning techniques are applied to remove classes overlapping prior the estimator fitting and cluster-based techniques creates synthetic instances for each class of the problem based on the cluster means of each.

The combination of sampling strategies with ensemble learning techniques has been broadly discussed in the community, given that the use of these techniques has presented higher quality results when compared to the application of the techniques apart.

III. Materials and Methods

A. Dataset

In this paper, we consider a real-world imbalanced dataset available on Kaggle's competition *Can You Predict Product Backorders?*[1]. Table II summarizes the properties of the dataset: the number of attributes, positive and negative classes, samples and imbalance ratio. The current service level of this inventory system is around 99,27%, and it is company's interest identifying parts with highest shortage risk prior the event, so short-term actions can be carry out to mitigate those risks and improve the general system performance.

TABLE II
Dataset Summary

Dataset	# Atts	# Pos.	# Neg.	# Total	Imb. Ratio
bopredict	22	13,981	1,915,954	1,929,936	1:137

The dataset contains the historical data for the 8 weeks prior to the week we are trying to predict, taken as a weekly snapshot at the start of the week. Attributes are defined as follows:

- x_1: Current inventory level of component;
- x_2: Registered transit time;
- x_3: In transit quantity;
- $x_{4,5,6}$: Forecast sales for the next 3, 6 and 9 months;
- $x_{7,8,9,10}$: Sales quantity for the prior 1, 3, 6, 9 months;
- x_{11}: Minimum recommended amount in stock;
- x_{12}: Parts overdue from source;
- $x_{13,14}$: Source performance in last 6 and 12 months;
- x_{15}: Amount of stock orders overdue;
- x_{16-21}: General risk flags;
- y: Product went on backorder.

[1]https://www.kaggle.com/tiredgeek/predict-bo-trial

Prior the application of the estimators, some basic transformations are performed such as binaries features encoding, quantity-related features normalization and missing values imputation.

Figure 1 shows a general 2D representation of a given random sample of the dataset, obtained by plotting two major components calculated by the Principal Component Analysis [5] method. The positive class occurrence is considered rare and it is outnumbered and overlapped by the negative class, increasing the complexity of the characteristic class imbalanced problem.

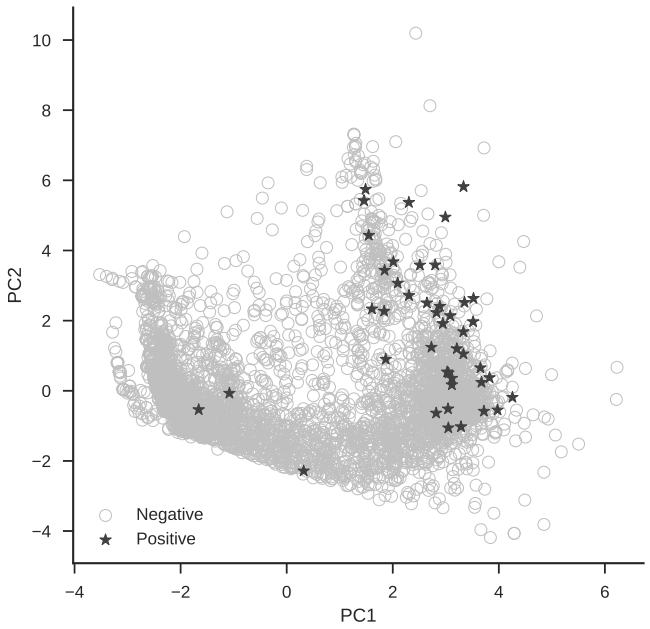

Fig. 1. Dataset random sub-sample with $n = 5,000$, represented by the 2 principal components – explained variance ratio equal to 39.95%. Positive classes (minority) are represented by stars, whilst the negative class by circles.

B. Learning Classifiers

Learning classifiers are unique algorithms able to generate models capable of performing complex tasks such as classification of events, images, among others. These models are developed through a supervised learning training process, where parameters are tuned – or rules are construct, depending on the nature of the algorithm – while error rate or other specific metric is minimized.

Some learning algorithms present internal modifications to handle imbalance datasets, such as the ones applied in this study. Another benefit of the followings models is that they provide a continuous output between 0 and 1, which can be interpreted as the fault probability.

1) Logistic Regression (LOGIST): Logistic Regression [6] is a popular and simple approach for binomial classification. The difference between logistic and linear regression is both in the choice of the parametric model and its assumptions. Because the model is trained by maximizing the binomial log-likelihood, it presents an improved sensitivity to imbalanced

learning. Thus, this algorithm is applied to provide a baseline score to the problem presented.

2) Classification Tree (CART): The Classification and Regression Tree [7] is a predictive model based on the construction of binary trees, in which a rule is associated with each branch created in a supervised training process. The adoption of gain or entropy as the split criterion has demonstrated improved insensitivity to changes in sample distributions. Although CART is considered a weak learner, the model presents some interest benefits such as legibility, outliers robustness and feature selection capability.

Parameters and search range adopted for the classifiers are exhibited in Table III. Highlighted parameters (in bold) presented higher scores in our dataset.

TABLE III
PARAMETERS AND RANGES ADOPTED FOR CLASSIFIERS

Method	Parameters	Range
LOGIST	Penalty	**L1**, L2
	Regularization - C	**1**, 10, 100, 1000
CART	Criterion	Gini
	Max. depth	5, 6, 7, **8**, 9
	Min. samples leaf	5

C. Sampling Techniques

Sampling is used to balance classes prior the application of the estimator, as the general performance of a learning classifier algorithm is better in an even dataset. Whereas sampling is considered an external approach that modifies the dataset previous the model fitting, a CART classifier is adopted along the methods.

1) Random Under-Sampling – RUS: Under-sampling method which removes data from the original data set by randomly selecting a set of majority class examples, while maintaining all examples of the minority classes [4]. A particular problem that uprises from this method application is that some relevant informations about majority classes may be lost, harming the model performance when applied in the whole set.

2) SMOTE: The Synthetic Minority Over-sampling Technique [8], also known as SMOTE, creates artificial data based on the feature space similarities between exiting minority examples, by considering the K-nearest neighbors in Euclidean space. To generate a synthetic sample, a random neighbor is selected and multiplied by the corresponding feature vector difference with a random number between [0,1].

Parameters defined for sampling methods are exhibited in Table IV.

D. Ensemble Learning

Ensembles methods are used to combine predictions of several base estimators built with a given learning algorithm in order to improve the generalization capability when compared to an individual estimator [5]. Two basic approaches are exploited: within *bagging*, estimators are trained independently and their prediction mean is assign to the ensemble's, while *boosting* sequentially build estimators that try to get higher scores in the domain that the previous has not performed satisfactorily. The motivation of using this method, specially combined with sampling, is to incorporate several weak estimators into robust ensembles with higher capabilities of distinguishing minority and majority classes.

TABLE IV
PARAMETERS ADOPTED FOR SAMPLING

Method	Parameters	Range
RUS	Ratio	1:1
SMOTE	Ratio	1:1
	Number of neighbors - K	5

1) Random Forest (FOREST): Random forest is a tree-based ensemble built using a bootstrap sample, where training set batches are drawn with replacement [5]. During the construction of the tree, the split selected is the best among a random subset of the attributes, leading to a randomness that leans the performance of the forest over a single non-random tree. The bias increase is offset by averaging variance decrease, achieved by the combination of the probabilistic prediction of the base classifiers.

2) Gradient Tree Boosting (GBOOST): Gradient Tree Boosting is a boosting-based ensemble which applies an arbitrary differentiable loss function and it is used in a variety of areas including web search ranking and ecology [5]. The algorithm is very capable of handling heterogeneous attributes and is robust to outliers, whilst its major drawback is that it can hardly be parallelized.

3) Blagging (BLAG): Blagging, also known by several others names in literature such as UnderBagging or EasyEnsemble, is a special ensemble which combines random under-sampling and ensemble learning [3]. Each base estimator is trained using a re-sampled batch containing all examples of the minority class and an equal size drawn of the majority class. The model provides higher generalization capability since more parcels of the majority classes are considered by the system then compared to a single under-sampled built tree, although it is probable that some useful negative instances may not be taken into consideration by the ensemble.

Parameters and search range adopted for the ensembles are exhibited in Table V. Highlighted parameters (in bold) presented higher scores in our dataset.

E. Model Selection

Training and test sets were split using a 0.85:0.15 proportion, generating 25 random instances of the problem preserving the original ratio between classes. A stratified 5-fold cross-validation scheme is adopted in training to avoid overfitting in training.

All possible parameters combination within the defined ranges are enumerated and models are tuned through exhaustive grid search procedure, using the AUC score for evaluation. Both estimator, scores and parameters settings are recorded.

TABLE V
PARAMETERS AND RANGES ADOPTED FOR ENSEMBLES

Method	Parameters	Range
FOREST	Number of estimators	10
	Max. depth	5, 6, 7, 8, **9**
	Min. samples leaf	5
GBOOST	Number of estimators	10
	Max. depth	5, 6, 7, 8, **9**
	Min. samples leaf	5
BLAG	Number of estimators	10

TABLE VI
MEAN AUC TRAIN AND TEST RESULTS – N=25

Model	AUC_{train}	AUC_{test}
LOGIST	0.9196 ± 0.0006	0.9206 ± 0.0032
CART	0.9443 ± 0.0010	0.9381 ± 0.0025
RUS	0.9412 ± 0.0008	0.9317 ± 0.0033
SMOTE	0.9406 ± 0.0006	0.9336 ± 0.0027
FOREST	0.9523 ± 0.0008	0.9441 ± 0.0022
GBOOST	0.9672 ± 0.0015	$\mathbf{0.9482 \pm 0.0025}$
BLAG	0.9834 ± 0.0002	$\mathbf{0.9478 \pm 0.0022}$

F. Software and Hardware

The routines were implemented in Python 3.6.1 programming language, using *Scikit-learn* 0.18.1 [5] and *Imbalanced-learn* 0.2.1 [9] machine learning libraries. Blagging model implementation is provided by G. Louppe and T. Fawcett from Silicon Valley Data Science[2] repository. All codes and data are made available by the authors upon request.

Computer specifications used to execute the algorithm and generate the learning models are given as follows: CPU AMD Opteron Processor 6272 (64 cores of 2.1GHz and cache memory of 2MB), RAM of 250GB and operational system Linux Ubuntu 14.04.4 LTS. Once constructed, each optimized model performs the prediction quickly, promptly allowing the analysis and parameter testing in the design cycles.

IV. RESULTS AND DISCUSSION

Table VI present the mean and standard deviation achieved by each family of algorithms in the 25 random instances of the problem. The simplest LOGIST model performed the lowest score (0.9206), followed by the sampling-based approaches RUS (0.9317) and SMOTE (0.9336). Although the performance of the under-sampling method slightly higher in training set, the AUC score of the synthetic over-sampling procedure has shown preferred results. In most cases than, the computational cost of models adopting over-sampling are significantly higher, as the matrix to be processed by the models is considerably larger when compared to a down-sampled one.

CART produced higher AUC scores solely (0.9381) than when combined with preprocessing techniques. Among ensembles approaches, FOREST has shown a significant increase by adopting a bagging ensemble of trees (0.9441), however boosting scored the highest score (0.9482). BLAG achieved approximate results compared to GBOOST (0.9478), forasmuch as a narrower hyper-parameter range was adopted for the model.

As stated, the AUC score can be interpreted as the probability of a given classifier rank a random positive example above a random negative example and is obtained calculating the area under the Receiver Operator Characteristic curve, exhibited in Figure 2. At the time that a random classifier is expected to produce a 0.50 AUC, all the models presented obtained scores higher than 0.90, which is classified as excellent for a diagnostic system. This fact evidences that both attributes, preprocessing and model selection scheme applied are appropriate for addressing the complex problem faced.

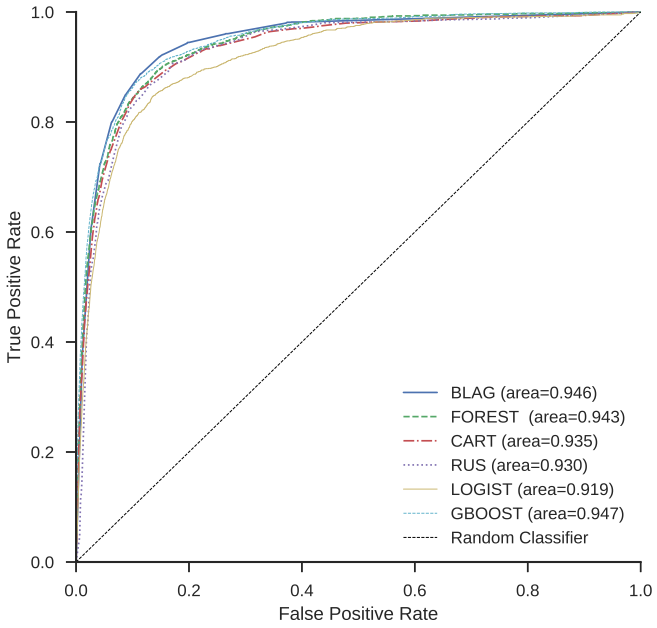

Fig. 2. Receiver Operator Characteristic curves of selected models for a particular instance of the problem. AUC for each model is obtained from the traced curves, where an area of 0.50 area is considered a worthless test, while an area of 1.00 is associated to a perfect test.

Lastly, Figure 3 exhibits the precision-recall curves for the selected models. In this fashion, it becomes simple to observe how precision and recall behaves when the decision function threshold varies. The area under the precision-recall can also be used as an evaluation metric, in which BLAG demonstrated a distinct leverage over all other models (area of 0.307).

LOGIST and CART suggest an upper limit of 0.20 precision, while ensemble learning generated dominant solutions over the single instance models. RUS presented an early peak in precision, which could contribute to the construction of the nearly linear aspect of BLAG curve, since the model can be understood as a bagging ensemble of RUS instances.

In practical application, a BLAG-based prediction system could be set with a recall of 0.20, detecting 20% of items which goes into backorder – in this particular example an

[2]https://svds.com/

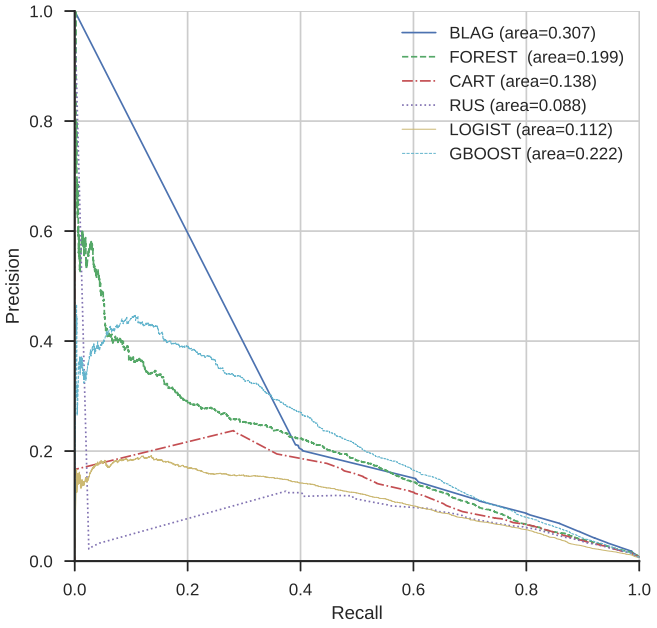

Fig. 3. Precision-recall curves of selected models for a particular instance of the problem. The area under the curve can also be understand as an evaluation metric, by indicating the operational metrics achievability of a given detection system.

approximated amount of 2,796 items - and correctly identifying 60% of the positive class (precision of ≈ 0.60). In this same hypothetic situation, considering all positive classes are addressed by operation's intervention, the inventory system service level is improved from 99,27% to 99,42%.

A more appropriate rule to set the decision function threshold would be leveraging the associated cost with the false-positive and false-negative prediction, interpreted as the material shortage cost and the follow-up/order amendment cost, respectively.

Not only a single sampling-ensemble model was tested between a large amount of existing models available, there's plenty of room for BLAG model improvement by exploring broader range in grid search, applying different base estimators such as tree boosting or further adopting informed approaches for under-sampling, counterpoising information loss deficiency in the random approach. Additionally, ensemble results can often be extent by increasing the number of estimators within the formed group.

V. Conclusion

The current paper presented the results of machine learning classifiers application within a predictive system design for inventory control, in extension of inventory planning models customarily discussed in literature. The current real case study has shown to be a feasible and interest mechanism of control, indicating materials with higher shortage likelihoods in short-period time and giving able time for a company to react. A company's overall service level can be extent by adopting a system such as this.

Since the items which goes on backorder (positive class) are rare compared to items which does not (negative class), some particular methods and metrics are employed either in design, development, and evaluation of the models in the considered imbalanced class problem. GBOOST has shown the best AUC score, although BLAG performed preferably when taking into account precision-recall curves, computational costs, and enhancement capability.

Future works include exploring further classification learning ensemble and sampling based algorithms, besides learning algorithms with different foundations such as support vector machines and neural networks, and verify potential performance improvements. In doing so, a cost-sensitive learning framework can be developed and misclassification costs incorporated, allowing cost curves analysis in model design.

Acknowledgment

The authors would like to thank the Federal University of Juiz de Fora and FAPEMIG (grant APQ 01606/15) for supporting this work.

References

[1] C. Tsou, "Multi-objective inventory planning using MOPSO and TOPSIS," *Expert Systems with Applications*, vol. 35, pp. 136-142, 2008.

[2] V. Lopez, A. Fernandez, S. Garcia, V. Palade, 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.

[3] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, "A review on ensembles for 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, Jul. 2012.

[4] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledgment and Data Engineering*, vol. 21, no. 9, pp.1263-1284, Sep. 2009.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, M. and E. Duchesnay, "Scikit-learn: machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[6] D. W. Hosmer Jr., S. Lemeshow and R. X. Sturdivant, "Applied logistic regression," John Wiley & Sons, vol. 398, 2013.

[7] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees," Wadsworth, Belmont, CA, 1984.

[8] N. V. Chawla, K. W. Bowyer, L. O.Hall and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, pp. 321-357, 2002.

[9] G. Lemaitre, F. Nogueira and C. K. Aridas, "Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol.18, no. 17, pp. 1-5, 2017.