IMPROVING ELECTRIC FRAUD DETECTION USING CLASS IMBALANCE STRATEGIES

Matías Di Martino, Federico Decia, Juan Molinelli and Alicia Fernández
Instituto de Ingeniería Eléctrica, Facultad de Ingeniería Universidad de la República Montevideo, Montevideo, Uruguay
{*alicia, matiasdm*}*@fing.edu.uy*, {*federicodecia, jmolinelli*}*@gmail.com*

Keywords: Electricity theft, Support vector machine, Optimum path forest, Unbalance class problem, Combining classifier, UTE.

Abstract: Improving nontechnical loss detection is a huge challenge for electric companies. The great number of clients and the diversity of the different types of fraud makes this a very complex task. In this paper we present a fraud detection strategy based on class imbalance research. An automatic detection tool combining classification strategies is proposed. Individual classifiers such as One Class SVM, Cost Sensitive SVM (CS-SVM), Optimum Path Forest (OPF) and C4.5 Tree, and combination functions are designed taken special care in the data's class imbalance nature. Analysis over consumers historical kWh load profile data from Uruguayan Electric Company (UTE) shows that using combination and balancing techniques improves automatic detection performance.

1 INTRODUCTION

Nontechnical losses represent a very high cost to power supply companies, who aims to improve fraud detection in order to reduce this losses. Research in pattern classification field has been made to tackle this problem (Ramos et al., 2010), (Nagi and Mohamad, 2010), (Muniz et al., 2009), (Jiang et al., 2000)

In Uruguay the national electric power company (henceforth call UTE) faces the problem by manually monitoring a group of customers. A group of experts looks at the monthly consumption curve of each customer and indicates those with some kind of suspicious behavior. This set of customers, initially classified as suspects are then analyzed taking into account other factors (such as fraud history, counter type etc.). Finally a subset of customers is selected to be inspected by an UTE employee, who confirms (or not) the irregularity. The procedure described before, has major drawbacks, mainly, the number of costumers that can be manually controlled is small compared with the total amount of costumer (around 500.000 only in Montevideo). To improve the efficiency of fraud detection and resource utilization, we implemented a tool that automatically detects suspicious behavior analyzing customers historical consumption curve. Thus, UTE's experts only need to look to a reduced number of costumers and then select those who need to be inspected.

Due to the applications nature there is a great imbalance between "normal" and "fraud/suspicious" classes. The class imbalance problem in general and fraud detection in particular have received considerable attention in recent years. Garcia et al. and Guo and Zhou review main topics in the field of the class imbalance problem (Garcia et al., 2007), (Guo and Zhou, 2008). These include: resampling methods for balancing data sets (Batista et al., 2004),(Barandela and Garcia, 2003), (Chawla et al., 2002), (Chawla et al., 2003), (Kolez et al., 2003), feature extraction and selection techniques -wrapper (Dash and Liu, 1997), and choose of F-value as performance measure.

In addition, it is generally accepted that combination of diverse classifiers can improve performance. A difficult task is to choose the combination strategy for a diverse set of classifiers. Kuncheva found the optimum set of weights for the majority weight vote combiner when the performance metrics is accuracy and with independent base classifiers (Kuncheva, 2004). Further analysis has been done on the relationship between diversity and the majority rules performance (Brown and Kuncheva, 2010), (Wang and Yao, 2009), (Chawla and Sylvester, 2007). In this paper we propose a combination function adapted to the imbalance between classes, using F-value as the performance measurement and some well-known pattern recognition techniques such as SVM (Support Vector Machine) (Vapnik, 1998), (Scholkopf and Smola, 2002), Tree classifiers and more recent algorithms such as Optimum Path Forest (Papa and Falcao, 2010),(Papa et al., 2007) as base classifiers.

Performance evaluation using test dataset shows very good results on suspicious profiles selection. Also, on field evaluation of fraud detection using our automatic system shows similar results to manual experts' method.

The paper is organized as follows. Section 2 describes general aspects of the class imbalance problem, section 3 describes different strategies proposed, section 4 presents the results obtained, and, finally, section 5 concludes the work.

2 THE CLASS IMBALANCE PROBLEM

When working on the fraud detection problem, one can not assume that the number of people who commit fraud are the same than those who do not, usually there are fewers elements from the class who commit fraud. This situation is known as the problem of class imbalance, and it is particularly important in real world applications where it is costly to misclassify examples from the minority class. In this cases, standard classifiers tend to be overwhelmed by the majority class and ignore the minority class, hence obtaining suboptimal classification performance. Having to confront this type of problem, we decided to use three different strategies on different levels, changing class distribution by resampling, manipulating classifiers, and on the ensemble of them.

The first consists mainly in resampling techniques such as under-sampling the majority class or oversampling the minority one. Random under-sampling aims at balancing the data set through random removal of majority class examples. The major problem of this technique is that it can discard potentially important data for the classification process. On the other hand, the simplest over-sampling method is to increase the size of the minority class by random replication of those samples. The main drawback of over-sampling is the likelihood of over-fitting, since it makes exact copies of the minority class instances As a way of facing the problems of resampling techniques discussed before, different proposals address the imbalance problem by adapting existing algorithms to the special characteristics of the imbalanced data sets. One approach is one-class classifiers, which tries to describe one class of objects (target class) and distinguish it from all other objects (outliers). In this paper, the performance of One-Class SVM, adaptation of the popular SVM algorithm, will be analyzed. Another technique is cost-sensitive learning, where the cost of a particular kind of error can be different from others, for example by assigning a high cost to mislabeling a sample from the minority class.

Another problem which arises when working with imbalanced classes is that the most widely used metrics for measuring the performance of learning systems, such as accuracy and error rate, are not appropriate because they do not take into account misclassification costs, since they are strongly biased to favor the majority class. In the past few years, several new metrics which measure the classification performance on majority and minority classes independently, hence taking into account the class imbalance, have been proposed (Manning et al., 2009).

- $Recall^p = \dfrac{TP}{TP+FN}$
- $Recall^n = \dfrac{TN}{TN+FP}$
- $Precision = \dfrac{TP}{TP+FP}$
- $F_{value} = \dfrac{(1+\beta^2)Recall^p \times Precision}{\beta^2 Recall^p + Precision}$

Table 1: Confusion matrix.

	Labeled as	
	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

$Recall^p$ is the percentage of correctly classified positive instances, in this case, the fraud samples. Precision is defined as the proportion of labeled as positive instances that are actually positive. The combination of this two measurements, the F-value, represents the geometric mean between them, weighted by the parameter β. Depending on the value of β we can prioritize Recall or Precision. For example, if we have few resources to perform inspections, it can be useful to prioritize Precision, so the set of samples labeled as positive has high density of true positive.

3 STRATEGY PROPOSED

The system presented consists of basically on three modules: Pre-Processing and Normalization, Feature selection and extraction and, finally, Classification. Figure 1 shows the system configuration. The system input corresponds to the last three years of the monthly consumption curve of each costumer, here

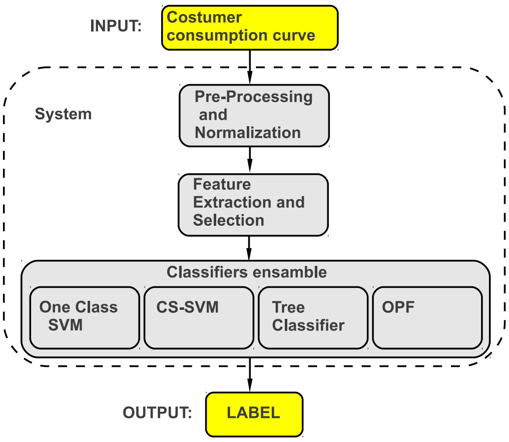
INPUT:
Costumer consumption curve
System
Pre-Processing and Normalization
Feature Extraction and Selection
Classifiers ensamble
One Class SVM
CS-SVM
Tree Classifier
OPF
OUTPUT:
LABEL

Figure 1: Block Diagram.

called $X^m = \{x_1^m, \quad \ldots \quad x_n^m\}$, where x_i^m is the consumption of the m costumer during the i-th month. The first module called Pre-Processing and Normalization, normalizes the input data so that they all have unitary mean and implements some filters to avoid peaks from billing errors.

The proposed methodology was developed as GUI software in Matlab using PRTOOLS (Duin, 2000), LibOPF (Papa et al., 2008) and LibSVM (Chang and Lin, 2001).

3.1 Attributes

A feature set was proposed taking into account UTEs technician experts in fraud detection by manual inspection and recent papers on non technical loss detection (Alcetegaray and Kosut, 2008), (Muniz et al., 2009), (Nagi and Mohamad, 2010). Below a list of some of the proposed features:

- Consumption ratio for the last 3, 6 and 12 months and the average consumption.
- Norm of the difference between the expected consumption and the actual consumption.
- Difference between Fourier coefficients from the last and previous years.
- Difference between Wavelet coefficients from the last and previous years.
- Difference in the coefficients of the polynomial that best fits the consumption curve.
- Variance of the consumption curve.
- Slope of the straight line that fits the consumption curve.

It is well known that when thinking about the features to use, large number of attributes do not imply better performances. The important thing is their relevance and the relationship between the number of these and the number of elements. This is why we implemented a feature selection stage. We implemented several algorithms for feature selection, and concluded that for each classifier algorithms it is best to use a different feature set.

3.2 Classifiers

SVM is an algorithm frequently used in pattern recognition and fraud detection. The main purpose of the binary SVM algorithm is to construct an optimal decision function $f(x)$ that predicts unseen data into two classes and minimizes the classification error. In order to obtain this, one looks to maximize the separation margin between the two classes and hence classify correctly unseen data (Nagi and Mohamad, 2010). This can be formulated as a quadratic programming optimization problem

$$\Phi(\omega,\zeta_i) = \min\left\{\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\zeta_i\right\} \qquad (1)$$

subjected to the constraint that all the training samples are correctly classified, that is

$$y_i(\langle\omega,x\rangle + b) \geq 1-\zeta_i,\;\; i = 1,2,\ldots,n \qquad (2)$$

where ζ_i for $i = 1,2,\ldots,n$ are nonnegative slack variables. C is a regularization parameter and is selected to be the tradeoff between the two terms in 1.

3.2.1 CS-SVM and One-class SVM

Two different approaches where introduced when describing the class imbalance problem, one-class classifiers and cost-sensitive learning. When applying this two approaches on SVM, we talk about One-Class SVM and CS-SVM.

In One-Class SVM equation 1 becomes,

$$\min_{\omega\in\mathcal{H},\zeta_i\in\mathbb{R},\rho\in\mathbb{R}} \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu l}\sum_{i=1}^{n}\zeta_i - \rho \qquad (3)$$

while in CS-SVM it becomes:

$$\Phi(\omega,\zeta_i) = \min\left\{\frac{1}{2}\|\omega\|^2 + \sum_{i/y_i=1} C^+\zeta_i + \sum_{i/y_i=-1} C^-\zeta_i\right\} \qquad (4)$$

Both the kernel parameter K and the values of C^+, C^- and ω are often chosen using cross validation. The method consists in splitting the data set into p parts of equal size, and perform p training runs. Each time, leaving out one of the p parts and use it as an independent validation set for optimizing the parameters.

Usually, the parameters which work best on average over the p runs are chosen. Finally, these average parameters are used to train the complete training set. There are some problems with this, as can be seen on (Scholkopf and Smola, 2002).

Having said this, the method used to determine the optimum parameters for CS-SVM was:

1. Determine sets $C = [C_1, C_2, ..., C_n]$ and $\gamma = [\gamma_1, \gamma_2, ..., \gamma_m]$.
2. Select $C_i \in C$ and $\gamma_j \in \gamma$, split the training set into p parts of equal size and perform p training runs. Each set is called B_i with $i = \{1, 2, ..., p\}$.
3. Use $B_{te} = B_1$ as the test set and $B_{tr} = B_2 \cup B_3 \cup \ldots \cup B_p$ as the training set.
4. Determine a classifier model for B_{tr}, C_i and γ_j. As the ratio between the two classes is unbalanced, when determining the CS-SVM classifier two parameters are defined, C^+ and C^- using class weights defined by calculating the sample ratio for each class. This was achieved by dividing the total number of classifier samples with the individual class samples. In addition, class weights were multiplied by a factor of 100 to achieve satisfactory weight ratios (Nagi and Mohamad, 2010).
5. Classify the samples from the training set B_{te} and compare the results with the labels predetermined. From these comparison, obtain the estimated F_{value} for C_i and γ_j called $F_{value_1}(C_i, \gamma_j)$.
6. Repeat these procedure for $B_{te} = B_2$ and the combination of the reaming sets as B_{tr} getting $e_2(C_i, \gamma_j)$, then for $B_{te} = B_3$ and so on until completing the p iterations.
7. For each pair of (C_i, γ_j) there's an estimation of the classification error for each cross validation. The classification error for this pair (C_i, γ_j) is the average value of the classification errors obtained in each cross validation, $e(C_i, \gamma_j) = \frac{1}{p} \sum e_l(C_i, \gamma_j)$.
8. This method is repeated combining all the values from the sets C and γ.
9. The values of C_{opt} and γ_{opt} are the ones for which the smallest classification error is obtained.

The metric used for measuring the classification error for this method was the F_{value}. For One-Class SVM, the method was the same but with the main objective of finding $\sigma \in S = \{\sigma_1, \sigma_2 .....\sigma_l\}$.

3.2.2 OPF

In (Ramos et al., 2010) a new approach, Optimum Path Forest (OPF), is applied to fraud detection in electricity consumption. The work shows good results in a problem similar to the targeted. OPF creates a graph with training dataset elements. A cost is associated to each path between two elements, based on the distance of the intermediate elements belonging to the path. It is assumed, that elements of the same class will have a lower path cost, than elements of different classes. The next step is to choose representatives from each class, called prototypes. Classifying a new element implies to find the prototype with lowest path cost. Since OPF is very sensitive to class imbalance, we under-sampled the majority class. Best performance was obtained while using a training data set with 40% of the elements from the minority class.

3.2.3 C4.5

The fourth classifier used is a decision tree proposed by Ross Quinlan: C4.5. Trees are a method widely used in pattern recognition problems due to its simplicity and good results. To classify, a sequence of simple questions is done. It begins with an initial question, and depending on the answer, the procedure continues until reaching a conclusion about the label to be assigned. The disadvantage of these methods is that they are very unstable and highly dependent on the training set. To fix this, in C4.5 a later stage of AdaBoost was implemented. It generates multiple instances of the tree with different portions of the training set and then combines them achieving a more robust result. As in OPF, sensitivity to class imbalance has led to sub-sampling the majority class. Again, we found that the best results was obtained while using a training data set with 40% of the elements from the minority class.

3.3 Combining Classifiers

The next step after selecting feature sets and adjusting classification algorithms to the training set, is to decide how to combine the information provided by each classifier. There are several reasons to combine classifiers, for example, to obtain a more robust and general solution and improve the final performance (Dietterich, 2000).

After labels have been assigned by each individual classifier, a decision rule is build as:

$$\begin{aligned} g_p(x) &= \lambda^p_{O-SVM} d^p_{O-SVM} + \lambda^p_{CS-SVM} d^p_{CS-SVM} \\ &\quad + \lambda^p_{OPF} d^p_{OPF} + \lambda^p_{Tree} d^p_{Tree} \end{aligned} \tag{5}$$

$$\begin{aligned} g_n(x) &= \lambda^n_{O-SVM} d^n_{O-SVM} + \lambda^n_{CS-SVM} d^n_{CS-SVM} \\ &\quad + \lambda^n_{OPF} d^n_{OPF} + \lambda^n_{Tree} d^n_{Tree} \end{aligned} \tag{6}$$

where $d_j^i(x) = 1$ if the classifier j labels the sample as i and 0 otherwise. Then if $g_p(x) > g_n(x)$ the sample is assigned to the positive class, if $g_n(x) > g_p(x)$ the sample is assigned to the negative class.

In (Kuncheva, 2004), the weighted majority vote rule is analyzed and optimum weights are found for maximum overall accuracy, assuming independence between classifiers: $\lambda_j^i = log\left(\frac{Accuracy_j}{1-Accuracy_j}\right)$, where $Accuracy_j$ represents the ratio of correctly classified samples for the classifier j, (in (Kuncheva, 2004) priors are also consider on the $g_{\{p,n\}}(x)$ construction adding $log(P(\omega_{\{p,n\}}))$)

Inspired in this result, but taking into account that we want to find a solution with good balance between Recall and Precision, several weights $\lambda_j^{p,n}$ were proposed:

- $\lambda_j^i = log\left(\frac{Recall_j^p+1}{Recall_j^p-1}\right)$
- $\lambda_j^i = log\left(\frac{F_{value_j}+1}{F_{value_j}-1}\right)$
- $\lambda_j^i = log\left(\frac{Accuracy_j}{1-Accuracy_j}\right)$
- $\lambda_j^p = Recall_j^n$ and $\lambda_j^n = Recall_j^p$

Also the *optimal* multipliers were found by exhaustive search over a predefined grid, looking for those which maximize the classification F_{value}. Search was made by looking for all the possibilities with $\lambda_j^i \in [0 : 0.05 : 1]$ and was evaluated with a 10-fold cross validation.

All of the proposed combined classifiers improved individual classifiers performance. In Table 2 we present the performance results using *optimal* multipliers, found by exhaustive search.

4 RESULTS

4.1 Data

For this paper we used a data set of 1504 industrial profiles (October 2004- September 2009) obtained from the Uruguayan electric power company (DATASET 1). Each profile is represented by the customers monthly consumption. UTE technicians make random profile selection and data labeling. Training and performance evaluation shown in Table 2 was done with DATASET 1. Another independent dataset (DATASET 2) of 3338 industrial profiles with contemporary data (January 2008-2011) was used for on field evaluation.

4.2 Labeling Results

Table 2 shows performance for individual classifiers and for the combination of them, results shown here were achieved by using a 10-fold cross validation using DATASET1. CS-SVM presented the best F_{value}, followed by One class SVM. We saw that combination improved performance achieving better results than those of the the best individual classifier.

Table 2: Data Set 1 labeling results.

Description	$Acc.$	$Rec^p.$	$Pre.$	$Fval.$
	(%)	(%)	(%)	(%)[$\beta = 1$]
O-SVM	84,9	54,9	50,8	52,8
CS-SVM	84,5	62,8	49,7	55,5
OPF	80,1	62,2	40,5	49
Tree (C4.5)	79	64,6	39	48,6
Combination	86,2	64	54,4	58,8

4.3 On Field Results

After all the proposed alternatives were evaluated (on DATASET 1), comparing automatic labelling with *manual labelling* performed by UTE's experts, we tested data labels with on field evaluation.

This test were done in the following way:

1. Train the classification algorithm using DATASET 1.
2. Classify samples from DATASET 2. Lets call DATASET 2P the samples of DATASET 2 labelled as positive (associated to abnormal consumption behaviour).
3. Inspect customers on DATASET 2P

560 samples of DATASET 2 were labelled as positive, from those, 340 were randomly selected (due to human resource issues) to perform inspections. The inspections yielded 11 irregular situations and 4 suspect situations (being analyzed). This results show that the automatic framework has a hit rate of between 3.3% and 4.4%. Manual fraud detection performed by UTE's experts during 2010 had a hit rate of about 4%, so results are promising. Specially taking into account that manual detection considers more information than just the consumption curve, such as fraud history, surface dimension and contracted power, among others.

Figures 2, 3 and 4 show some examples of customers classified as suspicious by our automatic system. Once inspected, illegal activities were detected in these cases.

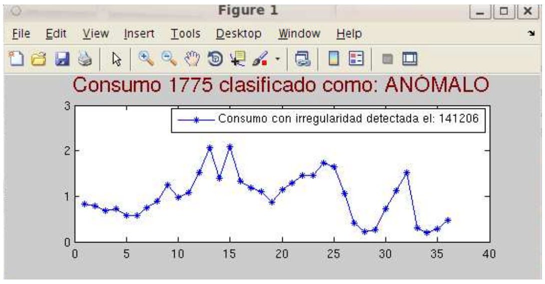
Figure 1
File Edit View Insert Tools Desktop Window Help
Consumo 1775 clasificado como: ANÓMALO
Consumo con irregularidad detectada el: 141206
0
1
2
3
0
5
10
15
20
25
30
35
40

Figure 2

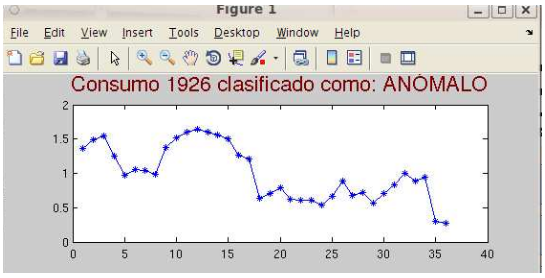
Figure 1
File Edit View Insert Tools Desktop Window Help
Consumo 1926 clasificado como: ANÓMALO
0
0.5
1
1.5
2
0
5
10
15
20
25
30
35
40

Figure 3

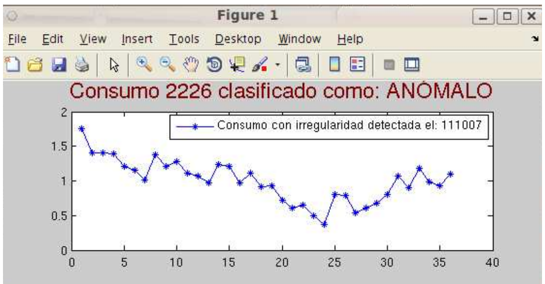
Figure 1
File Edit View Insert Tools Desktop Window Help
Consumo 2226 clasificado como: ANÓMALO
Consumo con irregularidad detectada el: 111007
0
0.5
1
1.5
2
0
5
10
15
20
25
30
35
40

Figure 4

5 CONCLUSIONS

We developed a framework able to detect customers whose consumption behaviour show some kind of irregularities. UTE is beginning to incorporate the system proposed and first results showed that it is useful and can lead to important savings, both time and money. We will continue working with UTE's collaboration, focusing our investigation on the lines of:

- Improving final performance and monitor bigger customer sets aiming to reach all customers in Montevideo (Uruguayan capital city).
- Analyze existence of data clusters.
- Add more features to our learning algorithm, such as: counter type (digital or analog), customer type (dwelling or industrial) and contracted power, among others.

We introduce different classifiers suitable for this type of problems (with unbalanced classes), comparing performance results for each of them. Innovative combination strategies are also proposed, all of them showing better results (using F-value as performance measurement) than the best individual classifier.

ACKNOWLEDGEMENTS

The authors would like to thank UTE, especially Juan Pablo Kosut, for providing datasets and share fraud detection expertise. We also want to thank Pablo Muse, Pablo Cancela and Martin Rocamora for useful advice.

REFERENCES

Alcetegaray, D. and Kosut, J. (2008). One class svm para la detección de fraudes en el uso de energía eléctrica. *Trabajo Final Curso de Reconocimiento de Patrones, Dictado por el IIE- Facultad de Ingeniería- UdelaR*.

Barandela, R. and Garcia, V. (2003). Strategies for learning in class imbalance problems. *Pattern Recognition*, pages 849–851.

Batista, G., Pratti, R., and Monard, M. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations 6*, pages 20–29.

Brown, G. and Kuncheva, L. (2010). ”good” and ”bad” diversity in majority vote ensembles. In *Multiple Classifier Systems*. Springer Berlin Heidelberg.

Chang, C. and Lin, C. (2001). *LIBSVM: a library for support vector machines*.

Chawla, N., Bowyer, K., and Hall, L. (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*.

Chawla, N., Lazarevic, A., and Hall, L. (2003). Smoteboost: impoving prediction of the minority class in boosting. *European Conf. ok Principles and Practice of Knowledge Discovery in Databases*.

Chawla, N. and Sylvester, J. (2007). Exploiting diversity in ensembles: Improving the performance on unbalanced datasets. *Departament of Computer Science and Engineering*.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1:131–156.

Dietterich, T. (2000). Ensemble methods in machine learning. *Multiple Classifier Systems, volume 1857 of Lecture Notes in Computer Science*.

Duin, R. (2000). *PRTools Version 3.0: A Matlab Toolbox for Pattern Recognition*.

Garcia, V., Sanchez, J., Mollineda, R., Alejo, R., and Sotoca, J. (2007). The class imbalance problem in pattern classification and learning. In *Congreso Espaol de Informtica*, Spain.

Guo, X. and Zhou, G. (2008). On the class imbalance problem. *IIE - Computer Society*, 1:192.

Jiang, R., Tagaris, H., and Laschusz, A. (2000). Wavelets based feature extraction and multiple cassifiers for electricity fraud detection.

Kolez, A., Chowdhury, A., and Alspector, J. (2003). Data duplication: an imbalance problem? *Proc. Proc. Intl. Conf. on Machine Learning, Workshop on Learning with Imbalanced Data Sets II*.

Kuncheva, L. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience.

Manning, C., Raghavan, P., and Schutze, H. (2009). *An Introduction to Information Retrival*. Cambridge University Press, Cambridge, England, 1 edition.

Muniz, C., Vellasco, M., Tanscheit, R., and Figueiredo, K. (2009). Ifsa-eusflat 2009 a neuro-fuzzy system for fraud detection in electricity distribution.

Nagi, J. and Mohamad, M. (2010). Nontechnical loss detection for metered customers in power utility using support vector machines. *IEEE TRANSACTIONS ON POWER DELIVERY, VOL. 25, NO. 2*.

Papa, J. and Falcao, A. (2010). Optimum-path forest: A novel and powerful framework for supervised graph-based pattern recognition techniques. *Institute of Computing University of Campinas*.

Papa, J., Falcao, A., and C.Suzuki (2008). *LibOPF: a library for Opthimum Path Forets*.

Papa, J., Falcao, A., Miranda, P., Suzuki, C., and Mascarenhas, N. (2007). Design of robust pattern classifiers based on optimum-path forests. *8th International Symposium on Mathematical Morphology Rio de Janeiro Brazil Oct*, pages 337–348.

Ramos, C., de Sousa, A. N., Papa, J., and Falcao, A. (2010). A new approach for nontechnical losses detection based on optimum-path forest. *IEEE TRANSACTIONS ON POWER SYSTEMS*.

Scholkopf, B. and Smola, A. (2002). *Learning with Kernels*. The MIT Press, London, 2. edition.

Vapnik, V. (1998). *Statistical Learning Theory*. New York: Wiley.

Wang, S. and Yao, X. (2009). Theoretical study of the relationship between diversity and single-class measures for class imbalance learning.