Fast Non-Technical Losses Identification Through Optimum-Path Forest

Caio C. O. Ramos, André N. Souza
Department of Electrical Engineering
São Paulo State University
Bauru, São Paulo, Brazil
caioramos@gmail.com, andrejau@feb.unesp.br

João P. Papa, Alexandre X. Falcão
Institute of Computing
University of Campinas
Campinas, São Paulo, Brazil
papa.joaopaulo@gmail.com, afalcao@ic.unicamp.br

***Abstract*—Fraud detection in energy systems by illegal consumers is the most actively pursued study in non-technical losses by electric power companies. Commonly used supervised pattern recognition techniques, such as Artificial Neural Networks and Support Vector Machines have been applied for automatic commercial frauds identification, however they suffer from slow convergence and high computational burden. We introduced here the Optimum-Path Forest classifier for a fast non-technical losses recognition, which has been demonstrated to be superior than neural networks and similar to Support Vector Machines, but much faster. Comparisons among these classifiers are also presented.**

Index Terms: Non-Technical Losses, Optimum-Path Forest

I. Introduction

Recently, the electrical systems of several countries have been going through some changes, beginning with the privatization of electric power companies, introducing a competitive environment in the national scene. The investments made by companies have as main objective a significant improvement of their financial and technical performance, seeking higher productivity, efficiency and profitability. In order to get better management with regard to the energy losses, one of the ways to maximize the available energy for commercialization in the electric power companies is to reduce the theft or fraud of energy [1].

The losses of electric power are constituted by the difference between the generated/bought energy and the billed ones, and can be divided into two distinct types: (i) technical and (ii) non-technical losses. The former are related with problems in the system through the physical characteristics of the equipment, that is, the technical losses are the energy lost in the transport, the transformation and the equipment of measurement, becoming a very high cost to the electric power companies [2]. The commercial losses or non-technical losses are those associated with the commercialization of the supplied energy to the user and refer to the delivered and not billed energy, propitiating a loss in the profits. They also are defined as the difference between the total losses and the technical losses, been strongly related to illegal connections in the distribution system [3].

Theft and problems with power meters, with the purpose to modify the registration of electric power, are the main causes of commercial losses in national and international electric power companies, evidencing the energy frauds [4]. However, it is a hard task to calculate or measure the amount of the commercial losses, because in most part of the cases it is almost impossible to know where they occur [5].

The illegal connections of electric power are the reason of constant concern, both for the electric power companies and for regulatory agencies. In this context, several manners of frauds may affect the authorities, that will not collect the tributes and taxes [1], [6]. The minimizing of the losses caused for these frauds represents the guarantee of investments in the product quality, the maintenance of the patrimony of the concession and, mainly, the possibility of improvement in the public services with less cost [7].

Although the advances in this area can be evidenced in recent years, especially with several techniques of electric energy measurement, it becomes more necessary the research of alternative methods with the great flexibility and easy adaptation to the context of the problem, as the models of the computational techniques with intelligent algorithms. Nagi et al. [8] used Support Vector Machines - SVM [9] for detection of electricity theft, and Nizar et al. [7] applied data mining-based techniques for non-technical loss analysis. Monedero et al. [10] proposed to use Artificial Neural Networks - ANN [11] together with statistical analysis for fraud detection in electrical consumption. A hybrid approach between Genetic Algorithms - GA [12] and SVM was also applied for non-technical losses detection [13].

Despite the use of these artificial intelligence techniques have been increasing, some flaws need to be revisited. An ANN with multi-layer perceptrons (ANN-MLP), for example, can address linearly and non-linearly separable problems, but not non-separable situations with maximum effectiveness [11]. As an unstable classifier, collections of ANN-MLP [14] can improve its performance up to some unknown limit of classifiers [15]. Support vector machines have been proposed to overcome the problem, by assuming linearly separable classes in a higher-dimensional feature space [9]. Its computational cost rapidly increases with the training set size and the number of support vectors. As a binary classifier, multiple SVMs are required to solve a multi-class problem. Panda et al. [16] presented a method to reduce the training set size before computing the SVM algorithm. Their approach aims to identify and remove samples likely related to non-support vectors. However, in all SVM approaches, the assumption of separability may also not be valid in any space of finite dimension [17].

Recently, a novel framework for graph-based classifiers that reduce the pattern recognition problem as an optimum path forest computation (OPF) in the feature space induced by a graph were presented [18]. These kind of classifiers do not interpret the classification task as a hyperplanes optimization problem, but as a combinatorial optimum-path computation from some key samples (prototypes) to the remaining nodes. Each prototype becomes a root from its optimum-path tree and each node is classified according to its strongly connected prototype, that defines a discrete optimal partition (influence region) of the feature space. The OPF-based classifiers have some advantages with respect to the aforementioned classifiers: (i) one of them is free of parameters, (ii) they do not assume any shape/separability of the feature space and (iii) run training phase faster, which allows the development of real time applications for fraud detection in electricity systems.

This work presents a fast approach to identify commercial losses, i. e., whether a user is becoming an illegal consumer or not, by means of an OPF-based classifier, considering information from databases of a brazilian energy company. We are the first in introducing an Optimum-Path Forest classifier in this field of knowledge. Some comparisons among OPF, SVM with Radial Basis Function - RBF and linear kernels and ANN-MLP are also performed. The remainder of this paper is organized as follows. Section II describes the theory of Optimum-Path Forest and Section III presents the dataset and recognition features used. Finally, experimental results and conclusions are stated in Sections IV and V, respectively.

II. Optimum-Path Forest Classifier

Let Z_1 and Z_2 be the training and test sets with $|Z_1|$ and $|Z_2|$ samples such as points or image elements (e.g., feature vectors, pixels, voxels, shapes and texture information). Let $\lambda(s)$ be the function that assigns the correct label i, $i = 1, 2, \ldots, c$, from class i to any sample $s \in Z_1 \cup Z_2$. Z_1 is a labeled set used to the design of the classifier and Z_2 is used to assess the performance of classifier and it is kept unseen during the project.

Let $S \subset Z_1$ be a set of prototypes of all classes (i.e., key samples that best represent the classes). Let v be an algorithm which extracts n attributes (color, shape or texture properties) from any sample $s \in Z_1 \cup Z_2$ and returns a vector $\vec{v}(s) \in \Re^n$. The distance $d(s,t)$ between two samples, s and t, is the one between their feature vectors $\vec{v}(s)$ and $\vec{v}(t)$. One can use any valid metric (e.g., Euclidean) or a more elaborated distance algorithm. Our problem consists of using S, (v, d) and Z_1 to project an optimal classifier which can predict the correct label $\lambda(s)$ of any sample $s \in Z_2$. The OPF classifier creates a discrete optimal partition of the feature space such that any sample $s \in Z_2$ can be classified according to this partition. This partition is an optimum path forest (OPF) computed in $\Re^n$ by the image foresting transform (IFT) algorithm [19].

Let (Z_1, A) be a complete graph whose the nodes are the samples in Z_1 and any pair of samples defines an arc in $A = Z_1 \times Z_1$ (Fig. 1a) . The arcs do not need to be stored and so the graph does not need to be explicitly represented. A path is a sequence of distinct samples $\pi = \langle s_1, s_2, \ldots, s_k \rangle$, where $(s_i, s_{i+1}) \in A$ for $1 \le i \le k-1$. A path is said *trivial* if $\pi = \langle s_1 \rangle$. We assign to each path π a cost $f(\pi)$ given by a path-cost function f. A path π is said optimum if $f(\pi) \le f(\pi')$ for any other path π', where π and π' end at a same sample s_k. We also denote by $\pi \cdot \langle s, t \rangle$ the concatenation of a path π with terminus at s and an arc (s,t).

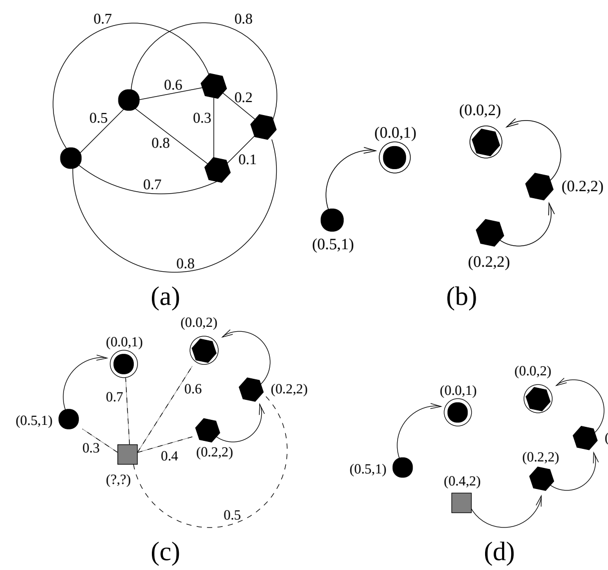

Fig. 1. (a) Complete weighted graph for a simple training set. (b) Resulting optimum-path forest for f_{max} and two given prototypes (circled nodes). The entries (x, y) over the nodes are, respectively, the cost and the label of the samples. The directed arcs indicate the predecessor nodes in the optimum path. (c) Test sample (gray square) and its connections (dashed lines) with the training nodes. (d) The optimum path from the most strongly connected prototype, its label 2, and classification cost 0.4 are assigned to the test sample.

The OPF algorithm may be used with any *smooth* path-cost function which can group samples with similar properties [19]. We are interested in prototypes that fall in the region between classes, which are generally overlapped regions. So, we will address the path-cost function f_{max}, because of its theoretical properties for estimating prototypes that have this behavior (Section II-A gives the details about this procedure):

$$\begin{aligned} f_{max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in S, \\ +\infty & \text{otherwise} \end{cases} \\ f_{max}(\pi \cdot \langle s, t \rangle) &= \max\{f_{max}(\pi), d(s,t)\}, \end{aligned} \tag{1}$$

such that $f_{max}(\pi)$ computes the maximum distance between adjacent samples in π, when π is not a trivial path.

The OPF algorithm assigns one optimum path $P^*(s)$ from S to every sample $s \in Z_1$, forming an optimum path forest P (a function with no cycles which assigns to each $s \in Z_1 \backslash S$ its predecessor $P(s)$ in $P^*(s)$ or a marker nil when $s \in S$. Let $R(s) \in S$ be the root of $P^*(s)$ which can be reached from $P(s)$. The OPF algorithm computes for each $s \in Z_1$, the cost

$C(s)$ of $P^*(s)$, the label $L(s) = \lambda(R(s))$, and the predecessor $P(s)$, as follows.

Algorithm 1: – OPF ALGORITHM

INPUT:	*A λ-labeled training set Z_1, prototypes $S \subset Z_1$ and the pair (v, d) for feature vector and distance computations.*
OUTPUT:	*Optimum-path forest P, cost map C and label map L.*
AUXILIARY:	*Priority queue Q and cost variable cst.*

```
1.  For each s ∈ Z1\S, set C(s) ← +∞.
2.  For each s ∈ S, do
3.     └ C(s) ← 0, P(s) ← nil, L(s) ← λ(s), and insert s in Q.
4.  While Q is not empty, do
5.     | Remove from Q a sample s such that C(s) is minimum.
6.     | For each t ∈ Z1 such that t ≠ s and C(t) > C(s), do
7.     |    | Compute cst ← max{C(s), d(s,t)}.
8.     |    | If cst < C(t), then
9.     |    |    | If C(t) ≠ +∞, then remove t from Q.
10.    |    |    | P(t) ← s, L(t) ← L(s), C(t) ← cst
11.    └    └    └ Insert t in Q.
```

Lines $1-3$ initialize maps and insert prototypes in Q. The main loop computes an optimum path from S to every sample s in a non-decreasing order of cost (Lines $4-10$). At each iteration, a path of minimum cost $C(s)$ is obtained in P when we remove its last node s from Q (Line 5). Ties are broken in Q using first-in-first-out policy. That is, when two optimum paths reach an ambiguous sample s with the same minimum cost, s is assigned to the first path that reached it. Note that $C(t) > C(s)$ in Line 6 is false when t has been removed from Q and, therefore, $C(t) \neq +\infty$ in Line 9 is true only when $t \in Q$. Lines $8-11$ evaluate if the path that reaches an adjacent node t through s is cheaper than the current path with terminus t and update the position of t in Q, $C(t)$, $L(t)$ and $P(t)$ accordingly.

A. Training

We say that S^* is an optimum set of prototypes when Algorithm 1 minimizes the classification errors for every $s \in Z_1$. S^* can be found by exploiting the theoretical relation between minimum-spanning tree (MST) and optimum-path tree for f_{max} [20]. The training essentially consists of finding S^* and an OPF classifier rooted at S^*.

By computing an MST in the complete graph (Z_1, A), we obtain a connected acyclic graph whose nodes are all samples of Z_1 and the arcs are undirected and weighted by the distances d between adjacent samples (Figure 2a). The spanning tree is optimum in the sense that the sum of its arc weights is minimum as compared to any other spanning tree in the complete graph. In the MST, every pair of samples is connected by a single path which is optimum according to f_{max}. That is, the minimum-spanning tree contains one optimum-path tree for any selected root node.

The optimum prototypes are the closest elements of the MST with different labels in Z_1 (i.e., elements that fall in the frontier of the classes). By removing the arcs between different classes, their adjacent samples become prototypes in S^* and Algorithm 1 can compute an optimum-path forest with minimum classification errors in Z_1 (Figure 1b). Note that, a given class may be represented by multiple prototypes (i.e., optimum-path trees) and there must exist at least one prototype per class.

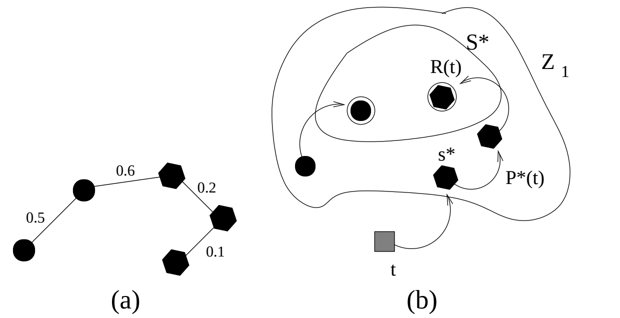

Fig. 2. (a) MST of the graph shown in Figure 1a where the optimum prototypes share the arc of weight 0.6. (b) The classification of a test sample (gray square) t as in Figure 1c assigns the optimum path $P^*(t)$ from $R(t) \in S^*$ to t passing through s^*.

B. Classification

For any sample $t \in Z_2$, we consider all arcs connecting t with samples $s \in Z_1$, as though t were part of the training graph (Figure 1c). Considering all possible paths from S^* to t, we find the optimum path $P^*(t)$ from S^* and label t with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t) \in S^*$ (Figure 2b). This path can be identified incrementally, by evaluating the optimum cost $C(t)$ as

$$C(t) = \min\{\max\{C(s), d(s,t)\}\}, \ \forall s \in Z_1. \qquad (2)$$

Let the node $s^* \in Z_1$ be the one that satisfies Equation 2 (i.e., the predecessor $P(t)$ in the optimum path $P^*(t)$). Given that $L(s^*) = \lambda(R(t))$, the classification simply assigns $L(s^*)$ as the class of t. An error occurs when $L(s^*) \neq \lambda(t)$.

III. MATERIAL AND METHODS

In this section we will describe the dataset used in the experiments, as well the features extracted from each consumer profile.

For the development of this work, we used a dataset obtained from a brazilian company of electric power composed by 736 profiles, divided into 116 illegal and 620 legal consumers. This dataset was previously labeled by technicians of the aforementioned company. Each consumer profile is represented by four features, as follows:

- Contracted Demand: the value of demand to be continuously available by the energy company and shall be paid likewise whether the electric power is used or not by the consumer, in kilowatts (kW);
- Measured Demand or Maximum Demand (D_{max}): the maximum demand for active power, verified by measurement, at intervals of fifteen minutes during the billing period, in kilowatts (kW);
- Load Factor (LF): the ratio between the average demand ($D_{average}$) and maximum demand (D_{max}) of the consumer unit, recorded in the same time period, as follows:

$$LF = \frac{D_{average}}{D_{max}}. \qquad (3)$$

The LF is an index factor that shows how the electric power is used in a rational way, and $D_{average}$ is defined as the ratio between the total energy (ϵ_T) and the period T, as described by

$$D_{average} = \frac{\epsilon_T}{T}, \tag{4}$$

and ϵ_T is given by

$$\int_0^T D(t)dt, \tag{5}$$

in which $D(t)$ is the demand curve (Figure 3).

- Installed Power (P_{inst}): the sum of the nominal power of electrical equipments installed and ready to operate at the consumer unit, in kilowatts (kW).

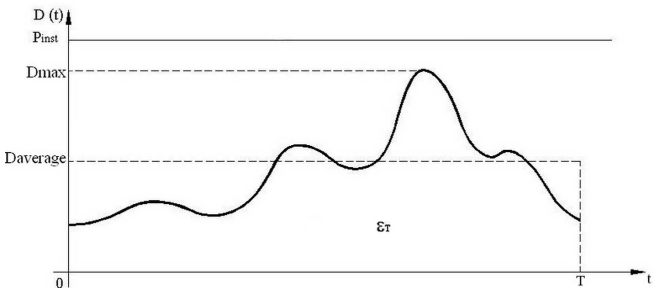

Fig. 3. Graphical interpretation of the features extracted from the consumers' profile.

IV. Experimental Results

We performed two series of experiments: in the former (Section IV-A) we used 50% of the whole dataset for training and the remaining 50% for testing classifiers, and in the last one (Section IV-B) we executed the experiments with different training and test set size percentages to allow a comparison about the robusteness of the classifiers with respect to variations on the sets size. For both experiments, we executed OPF, SVM-RBF (SVM with RBF as kernel function), SVM-LINEAR (SVM with a linear kernel function) and ANN-MLP (ANN-MLP trained by backpropagation algorithm) 10 times with randomly generated training and test sets, to compute the mean accuracy and its standard deviation, and the mean training and test execution times (in seconds).

For SVM-RBF, we used the latest version of the LibSVM package [21] with Radial Basis Function (RBF) kernel, parameter optimization and the one-versus-one strategy for the multi-class problem. With respect to SVM-LINEAR, we used the LibLINEAR package [22] with default $C = 1$ parameter. For OPF we used the LibOPF [23], which is a library for the design of optimum-path forest-based classifiers, and for ANN-MLP we used the Fast Artificial Neural Network Library (FANN) [24]. The network configuration is i:h_1:h_2:o, where $i = n$ (number of features), $h_1 = h_2 = |Z_1|/4$ and $o = c$ (number of classes) are the number of neurons in the input, hidden and output layers, respectively. Note that we used here two hidden layers, i. e., h_1 and h_2 with 25% of the training set size. The ANN-MLP was trained with a backpropagation algorithm, and its architecture was empirically chosen.

A. Classifiers evaluation

We evaluate here the OPF, SVM-RBF, SVM-LINEAR and ANN-MLP for non-technical losses detection using 50% for training and 50% for testing. Table I shows the mean accuracies and mean training and classification times (in seconds) after 10 runnings with randomly generated training and test sets.

TABLE I
MEAN ACCURACY AND MEAN TRAINING AND CLASSIFICATION TIMES OPF, SVM-RBF, SVM-LINEAR, AND ANN-MLP.

Classifier	Accuracy	Training time	Classification Time
OPF	90.21±2.93	0.0257	0.0223
SVM-RBF	88.93±3.07	13.4817	0.0222
SVM-LINEAR	45.40±6.31	2.4514	0.0048
ANN-MLP	53.01±6.95	1708.85	0.0078

We can see that OPF and SVM-RBF can produce similar results and also outperformed SVM-LINEAR and ANN-MLP. Even though, OPF was 524.57 times faster than SVM-RBF in the training phase. This point can make OPF able for real-time training depending systems, in which samples from known/unseen classes can be added to training set and re-training is needed. Similarly, one can have a real time detection system for non-technical losses in which new consumers' profiles can be added and the system need to be re-trained (reseted) as fast as possible, with minimum costs.

B. Classifiers robustness

We evaluate here the robustness of the classifiers with respect to variations on the training set size. We repeated the experiments shown in previous section with different training and test set sizes for OPF, SVM-RBF, SVM-LINEAR and ANN-MLP. Figure 4 displays the mean accuracies over the test set after 10 rounds of experiments for each training and test set size percentages configuration.

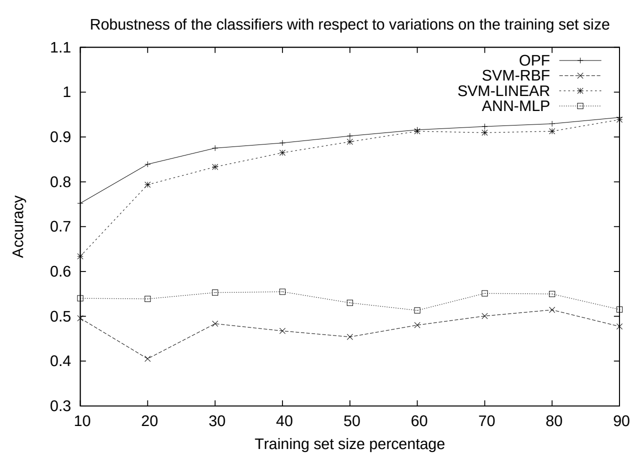

Fig. 4. Robustness of the classifiers with respect to different training and test size percentages.

We can see that only OPF and SVM-RBF have presented similar performances, increasing their mean accuracy with respect to bigger training set sizes, and ANN-MLP and SVM-LINEAR had also similar results, but with low recognition rates. This probably occurred due to (i) the choice of an ANN-MLP architecture is a hard task, making it inviable in large datasets and (ii) the SVM-LINEAR C parameter was not optimized. These situations make OPF a good classifier, because it is free of parameters, does not make assumption about shape and separability of the feature space and runs training phase faster.

V. Conclusions

We presented here a fast graph-based approach for automatic non-technical losses recognition. Nowadays, the fraud detection in power electric systems by illegal consumers is a hot topic, due to the non triviality of this problem to national and international electric power companies, and we are the first to introduce the OPF classifier in this context.

Experiments using OPF, SVM-RBF, SVM-LINEAR and ANN-MLP in a dataset with different training and test size percentages demonstrated that OPF and SVM-RBF are similar and outperformed the remaining ones, but OPF training phase is much faster, making it able for real time training-depending systems, in which new consumers' profiles can be added at moment and a fast system re-training will be necessary.

In the future works, we pretend to use with more electric power companies datasets, as well to apply more classifiers and also to implement SVM-LINEAR C parameter optimization.

References

[1] I. Davidson, "Evaluation and effective management of nontechnical losses in electrical power networks," in *Proceedings of the 6th IEEE Africon Conference in Africa*, vol. 1, 2002, pp. 473–477.

[2] A. Nizar, Z. Dong, and Y. Wang, "Power utility nontechnical loss analysis with extreme learning machine method," *IEEE Transactions on Power Systems*, vol. 23, pp. 946–955, 2008.

[3] M. de Oliveira, D. Boson, and A. Padilha-Feltrin, "A statistical analysis of loss factor to determine the energy losses," in *Proceedings of the IEEE/PES Transmission and Distribution Conference and Exposition: Latin America*, 2008, pp. 1–6.

[4] R. Alves, P. Casanova, E. Quirogas, O. Ravelo, and W. Gimenez, "Reduction of non-technical losses by modernization and updating of measurement systems," in *Proceedings of the IEEE/PES Transmission and Distribution Conference and Exposition: Latin America*, 2006, pp. 1–5.

[5] A. Nizar and Z. Dong, "Identification and detection of electricity customer behaviour irregularities," in *Proceedings of the IEEE/PES Power Systems Conference and Exposition*, 2009, pp. 1–10.

[6] A. Nizar, Z. Dong, M. Jalaluddin, and M. Raffles, "Load profiling method in detecting non-technical loss activities in a power utility," in *Proceedings opf the IEEE International Power and Energy Conference*, 2006, pp. 82–87.

[7] A. Nizar, Z. Dong, J. Zhao, and P. Zhang, "A data mining based ntl analysis method," in *Proceedings of the IEEE Power Engineering Society General Meeting*, 2007, pp. 1–8.

[8] J. Nagi, A. Mohammad, K. Yap, S. Tiong, and S. Ahmed, "Non-technical loss analysis for detection of electricity theft using support vector machines," in *Proceedings of the 2nd IEEE International Power and Energy Conference*, 2008, pp. 907–912.

[9] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Workshop on Computational Learning Theory*. New York, NY, USA: ACM Press, 1992, pp. 144–152.

[10] I. Monedero, F. Biscarri, C. León, J. Biscarri, and R. Millán, "Midas: Detection of non-technical losses in electrical consumption using neural networks and statistical techniques," in *Proceedings of the International Conference on Computational Science and Applications*, vol. 3984. Springer Berlin/Heidelberg: Lecture Notes in Computer Science, 2006.

[11] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall, 1994.

[12] J. Koza, *Genetic programming: on the programming of computers by means of natural selection*. Cambridge, MA: The MIT Press, 1992.

[13] J. Nagi, K. Yap, S. Tiong, S. Ahmed, and A. Mohammad, "Detection of abnormalities and electricity theft using genetic support vector machines," in *IEEE TENCON Region 10 Conference*, 2008, pp. 1–6.

[14] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[15] L. Reyzin and R. Schapire, "How boosting the margin can also boost classifier complexity," in *Proceedings of the 23th International Conference on Machine learning*. New York, NY, USA: ACM Press, 2006, pp. 753–760.

[16] N. Panda, E. Chang, and G. Wu, "Concept boundary detection for speeding up svms," in *Proceedings of the 23th International Conference on Machine learning*. New York, NY, USA: ACM Press, 2006, pp. 681–688.

[17] R. Collobert and S. Bengio, "Links between perceptrons, mlps and svms," in *Proceedings of the 21th International Conference on Machine learning*. New York, NY, USA: ACM Press, 2004, p. 23.

[18] J. Papa, A. Falcão, and C. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[19] A. Falcão, J. Stolfi, and R. Lotufo, "The image foresting transform theory, algorithms, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, 2004.

[20] C. Allène, J. Y. Audibert, M. Couprie, J. Cousty, and R. Keriven, "Some links between min-cuts, optimal spanning forests and watersheds," in *Mathematical Morphology and its Applications to Image and Signal Processing (ISMM'07)*. MCT/INPE, 2007, pp. 253–264.

[21] C. C. Chang and C. J. Lin, *LIBSVM: A Library for Support Vector Machines*, 2001, software available at url http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[22] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[23] J. Papa, C. Suzuki, and A. X. Falcão, *LibOPF: A library for the design of optimum-path forest classifiers*, 2008, software version 1.0 available at http://www.ic.unicamp.br/∼afalcao/LibOPF.

[24] S. Nissen, *Implementation of a Fast Artificial Neural Network Library (FANN)*, 2003, department of Computer Science University of Copenhagen (DIKU). Software available at *http://leenissen.dk/fann/*.