Detection of Non-Technical Losses Using Smart Meter Data and Supervised Learning

Madalina Mihaela Buzau, *Student Member, IEEE*, Javier Tejedor-Aguilera, Pedro Cruz-Romero, *Member, IEEE*, and Antonio Gómez-Expósito, *Fellow, IEEE*

***Abstract*—Non-technical electricity losses due to anomalies or frauds are accountable for important revenue losses in power utilities. Recent advances have been made in this area, fostered by the roll-out of smart meters. In this paper, we propose a methodology for non-technical loss detection using supervised learning. The methodology has been developed and tested on real smart meter data of all the industrial and commercial customers of Endesa. This methodology uses all the information the smart meters record (energy consumption, alarms and electrical magnitudes) to obtain an in-depth analysis of the customer's consumption behavior. It also uses auxiliary databases to provide additional information regarding the geographical location and technological characteristics of each smart meter. The model has been trained, validated and tested on the results of approximately 57 000 on-field inspections. It is currently in use in a non-technical loss detection campaign for big customers. Several state-of-the-art classifiers have been tested. The results show that extreme gradient boosted trees outperform the rest of the classifiers.**

***Index Terms*—Supervised learning, non-technical losses, smart meter, extreme gradient boosted trees.**

I. Introduction

NON-TECHNICAL electricity losses (NTL) due to any kind of anomaly (installation error, meter parametrization error, faulty meter or energy fraud) represent a major problem for the utilities. Not only do they cause significant revenue losses but they can also affect the power system operation as they provide uncertainty of the real consumption [1].

Reducing NTL is of major interest to the electricity providers as they represent a significant part of the total power losses [2]. Furthermore, detecting NTL in industrial and large commercial customers is of particular interest as their consumption is equal to approximately 55% of the total energy consumption (EC). Surely, detecting an anomaly in the meter of an industrial customer recovers significantly higher revenue losses than in the case of a residential customer. Moreover, large customers represent major interest for anomaly detection when the cost of the on-field inspection itself is also considered.

Attempting to detect NTL using a supervised approach can be quite challenging as this is an extremely imbalanced classification problem [3]. Naturally, in developed countries the number of electric supplies with any kind of detected anomaly is a tiny portion of the global amount. Moreover, as the customer samples are labeled manually by on-field inspections they are prone to human error. Introducing misclassified samples makes it more difficult for a machine learning (ML) algorithm to distinguish between classes.

Smart meters (SMs) allow utilities to devise new and innovative ways to detect NTL, a task perceived in the past as very difficult given the granularity of the data at that time. With the SM roll-out utilities have now access to frequent measurements of EC, giving them a better understanding of their customers' consumption behavior [4].

In this work, we propose a methodology which uses the SM data and auxiliary databases to formulate various characteristics of the customer's consumption behavior and also to provide additional information with regard to the geographical and technological characteristics of the SM. These characteristics are afterwards introduced into several supervised ML algorithms for model selection and evaluation. The models have been trained, validated and tested using real data from all the customers of the largest electricity utility in Spain (Endesa), with a contracted power higher than 50 kW.

II. Related Work

The current approaches for NTL detection found in the literature can be categorized either in hardware or non-hardware solutions. The non-hardware solutions can be based on state estimation, game theory or classification algorithms [5]. Our approach proposes a non-hardware solution based on classification. We will thus be focusing in this section on the recent advances made in this area.

Table I presents the main characteristics and performances of previous approaches (discussed in this section) as well as our approach. A common aspect is the building of a global model that can be used for all customers, though Jokar *et al.* [6] built a model on a customer-by-customer basis. Their approach uses Support Vector Machines (SVM) to distinguish between the normal and fraudulent pattern of the

Manuscript received July 7, 2017; revised December 1, 2017 and February 10, 2018; accepted February 13, 2018. Date of publication February 21, 2018; date of current version April 19, 2019. This work was supported by the European Community's Seventh Framework Programme FP7-PEOPLE-2013-ITN (ADVANTAGE Project) under Grant 607774. Paper no. TSG-00945-2017. *(Corresponding author: Madalina Mihaela Buzau.)*

M. M. Buzau, P. Cruz-Romero, and A. Gómez-Expósito are with the Department of Electrical Engineering, University of Seville, 41092 Seville, Spain (e-mail: madalina.buzau@gmail.com).

J. Tejedor-Aguilera is with Endesa-Enel, 41005 Seville, Spain.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2018.2807925

TABLE I
COMPARISON OF CURRENT APPROACH WITH PREVIOUS WORKS

Method	**Type of NTL detected**	**Data source for NTL cases**	**# of customers**	**Type of data**	**% samples with NTL**	**ML Algorithms**	**Results (best algorithm)**			
							TPR	FPR	PRC	AUC
[9]	abrupt changes	real on-field inspections	383	monthly EC & credit worthiness rating	13.83 %	SVM	-	-	77.41 %	-
[6]	changes EC pattern	synthetic	5K	half-hourly EC	50 %	SVM/customer	86 %	16 %	-	-
[12]	fraud	real on-field inspections	21583	monthly EC & auxiliary databases	14.85 %	NN	29.47 %	65.03 %	-	-
[15]	all	real on-field inspections	-	monthly EC & auxiliary databases	-	NB, KNN, DT, NN, SVM, RF, GBM, AB	-	-	-	0.84
[3]	all	real on-field inspections	≈ 100K	monthly EC	0 % - 100 %	Boolean, Fuzzy and SVM	-	-	-	0.56
[7]	fraud	synthetic	5600	half-hourly EC	-	NN	93.75 %	25.00 %	78.95 %	-
[8]	fraud	synthetic	5650	half-hourly EC	-	DT	-	-	-	-
[14]	all	real on-field inspections	3.5M	monthly EC & auxiliary databases	10 % - 90 %	K-Means, RF	-	-	-	0.74
[10]	abrupt changes	-	-	EC	-	ELM, OS-ELM, SVM	-	-	-	-
[13]	all	real on-field inspections	700K	monthly EC & auxiliary databases	1 % - 90 %	LR, KNN, SVM, RF	-	-	-	0.63
[16]	null EC	real on-field inspections	3510	monthly EC & auxiliary databases	4.67 %	Text mining, NN, DT, SOM-NN	-	-	14.75 %	-
[11]	contract diversion	synthetic	4245	hourly EC & weather data	10 % - 50 %	K-Means, LR, KNN, SVM	-	-	-	-
current	all	real on-field inspections	57304	SM data & auxiliary databases	5.38 % - 8.37 %	K-Means, KNN, LR, SVM, NN, XGBoost	-	-	-	0.91

customer. Rather than classifying the customers directly as having a NTL or not, Ford *et al.* [7] and Cody *et al.* [8] forecast the energy consumption of the customers. A neural network (NN) is used in [7], whilst Cody *et al.* [8] use a decision tree (DT). If the difference between the actual and forecasted consumption exceeds the limit imposed by the authors, the customer is considered to be committing fraud.

Nagi *et al.* [9] use SVM and the results of real on-field inspections to detect NTL in Malaysia. An Extreme Learning Machine (ELM), Online Sequential Extreme Learning Machine (OS-ELM) and SVM were used to detect electricity theft in [10]. The authors trained the algorithms with the results of real on-field inspections, though the performance of these algorithms has not been reported.

Han *et al.* [11] propose a solution to detect the NTL that occurs due to energy contract diversion with a cheaper contract. The authors use the k-means algorithm to cluster load profiles. A similarity and normality index is computed for each customer. These indexes are used as an input to several algorithms such as logistic regression (LR), k-nearest neighbors (KNN) and SVM.

A solution for fraud detection based on NN has been presented in [12]. The authors use monthly consumption data and auxiliary databases to train a NN with the results of real on-field inspections of a Brazilian electricity utility.

Similarly to our approach, the solutions proposed in [3], [13], [14], and [15] treat NTL as a black-box, aiming to detect all types of NTL. Glauner *et al.* [3] use boolean, fuzzy logic and SVM to detect NTL. The input features for the algorithms consist only of the last 12 monthly EC measurements. Glauner *et al.* [13] improved the approach in [3] by adding the geographical location of the customer to compute the inspection rate and the NTL rate in its neighborhood. The methodology was tested with LR, KNN, SVM and random forests (RF). Meira *et al.* [14] used RF for supervised learning and k-means clustering during feature engineering to create features with regards to the geographical location, transformers and consumption profiles. Coma-Puig *et al.* [15] used several ML algorithms to detect both electricity and gas NTL and discovered that a single gradient boosted machine (GBM) gave a better performance than any ensemble or any other classifier. The algorithms used were Naive Bayes (NB), AdaBoost (AB), KNN, DT, NN, SVM, RF and GBM.

Guerrero *et al.* [16] propose a methodology to increase the precision of NTL campaigns based on null consumption analysis. Text mining and NN are used for customer filtering whilst a second module creates rules devised from DT and self-organizing maps (SOM-NN).

As seen in Table I, the performance of the models is assessed using various metrics such as the true positive rate (TPR), known also as the recall (RCL), the false positive rate (FPR), the precision (PRC) and the AUC score. Due to the imbalanced nature of NTL detection, we believe that the AUC score provides more reliable results as it assesses the ranking quality of customers rather than their classification. The utility does not need a list with all the customers classified either with or without NTL but rather a ranked list of customers according to their probability of having an NTL. Thus, the performance of our model has been assessed using the AUC score.

Compared to previous approaches, our work distinguishes itself by:

- Using all the information the SMs record: EC, alarms and electrical magnitudes. We believe these additional data

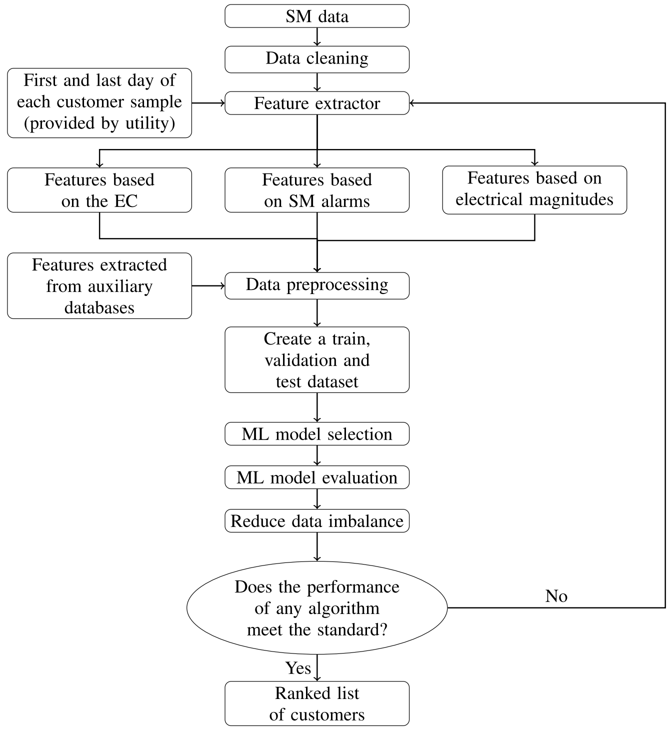

SM data
Data cleaning
First and last day of each customer sample (provided by utility)
Feature extractor
Features based on the EC
Features based on SM alarms
Features based on electrical magnitudes
Features extracted from auxiliary databases
Data preprocessing
Create a train, validation and test dataset
ML model selection
ML model evaluation
Reduce data imbalance
Does the performance of any algorithm meet the standard?
No
Yes
Ranked list of customers

Fig. 1. Methodology outline for NTL detection.

are vital for NTL detection as studying only the consumption behavior of the customer is not sufficient to detect a wide range of NTL.

- Applying both distance and density based outlier detection algorithms as well as the usage of the XGBoost classifier.
- Creating multiple training samples for customers with more than one inspection, as described in Section V.

III. Methodology Proposal

The main aim of the methodology described in this paper is to provide the utility with a ranked list of customers, according to their probability of having an anomaly in their electricity meter. The methodology uses mainly SM data for feature extraction (Figure 1). The features are based on SM alarms, EC and electrical magnitude measurements. It also uses features extracted from auxiliary databases which mainly provide geographical and technological characteristics of the customer. After preprocessing the datasets, the features are inserted as an input into several ML algorithms for model selection and evaluation. If the performance of the best model meets the desired standard required by the utility, its parameters are saved and used to make predictions on new customer samples obtaining a ranked list of customers as the final output.

IV. SM Data

The data used to train, validate and test the model were provided by Endesa. It included all the industrial and large commercial customers of the utility. Approximately 95% of these customers are equipped with meters capable of providing automatic reading. These meters register the EC every 15 minutes but due to the volume of data, the granularity was reduced to 5 measurements/day. This reduces also the privacy concerns that may arise with a higher data granularity.

TABLE II
SM Data

Daily measurements	Timestamp	
	Energy consumption	Daily Between 2 AM - 7 AM Between 8 AM - 1 PM Between 2 PM - 5 PM Between 6 PM - 8 PM Between 9 PM - 1 AM
	Quality byte	Intrusion Invalid lecture Synchronization Overflow Hourly verification Parameter modification Power fault Unit of measurement
Approx. 1-6 measure-ments/month	Timestamp	
	Active energy	Consumed Produced
	Reactive energy	Four-quadrant reactive energy
	Electrical magnitudes	Active power (R,S,T) Reactive power (R,S,T) Electric current (R,S,T) Voltage (R,S,T) Power factor (R,S,T)

Table II shows the measurements that were included in the SM data provided by the utility. Please note that the SM of these customers register the active and the four-quadrant reactive energy every 15 minutes/hour but we are collecting the total active and reactive energy consumption/production with the power snapshots.

V. Creating Customer Samples

The performance of the model was assessed on data from the last ten years, from 1st May 2007 until 30 December 2016. Nevertheless, the model will keep also updating with new data as it is aimed to detect anomalies that occur right at this moment. Therefore, the features which characterize the customer's consumption behavior will keep updating according to the latest data available.

The dataset contains customers who throughout our period of analysis either had none or at least one inspection. For customers who never had an inspection, their sample represents their entire consumption history. Customers with at least one inspection were divided in multiple samples (Figure 2).

The methodology presented in this paper uses a supervised approach to detect anomalies in the SMs by using the results of all the on-field inspections that have occurred for these type of customers. The training dataset has been created by selecting the customers with at least one inspection. This dataset has been used to train an ML algorithm in order to discover patterns in the characteristics of honest customers and customers detected with an anomaly in their meter.

The ranking list is created for customers who never had an inspection or whose last normalization date was more than

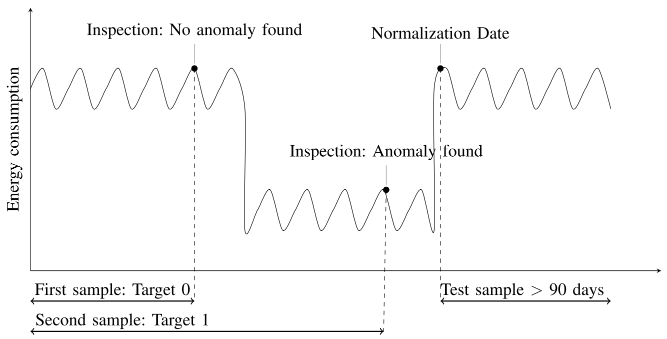


Fig. 2. Scenario of a customer with multiple training samples.

TABLE III
SIZE OF THE TRAINING DATASET AND THE RANKING LIST

First day analysis	01/05/2007
Last day analysis	30/12/2016
Unique customers in the training dataset	41571
Customer samples in the training dataset	57304
Customer samples in the ranking list	72489

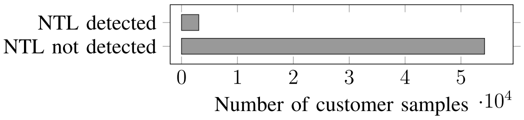


Fig. 3. Target distribution.

ninety days ago (Figure 2). This list is being obtained by using a trained model to make predictions on these unseen customer samples. The number of customers used during training and in the ranking list can be seen in Table III.

The main challenge of a ML model aimed to detect anomalies is given by the imbalance between customer classes. Figure 3 shows the number of samples of customers with and without an anomaly detected in the entire training dataset. This is an extremely imbalanced dataset as the number of customer samples with an anomaly detected represents $\approx 5\%$ of the entire training dataset. This will affect the learning process, as the model will be biased to predict the majority class.

VI. FEATURE EXTRACTION FROM SM DATA

Several types of features have been extracted using the SM data. Features developed using the quality byte (QB) measurement are aimed to detect meter faults or physical tampering. Features based on EC measurements aim to detect a drop in consumption or unusual consumption behaviors.

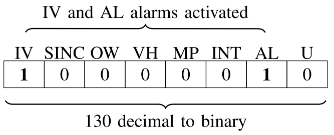
A. Features Extracted From QB Measurements

The QB measurement uses a 8-bit code to assess the quality of the measurement, as the IEC 870-5-102 protocol defines [17]. Table IV shows what type of alarms the SMs register.

In order to compute features related with alarms, each QB measurement, which was initially represented with the decimal numeration system has been converted to its binary representation. Furthermore, the binary value has been split into eight separate values, each value representing an alarm. If an alarm was triggered during the period of measurement (one day in our case) its value will be set to 1. Otherwise, its value will be zero.

TABLE IV
ALARMS REGISTERED BY THE QB MEASUREMENT [17]

Bit	Alarm	Description
7	IV	The measurement is valid (IV = 0)
6	SINC	Synchronized meter during the period of measurement (SINC = 1)
5	OW	Overflow (OW = 1)
4	VH	Hourly verification VH during the period of measurement (VH = 1)
3	MP	Parameter modification during the period of measurement (MP = 1)
2	INT	An intrusion has occurred during the period of measurement (INT = 1)
1	AL	Incomplete period due to power fault (AL = 1)
0	U	Unit of measurement. 0 for kWh/kvarh and 1 for MWh/Mvarh



IV	SINC	OW	VH	MP	INT	AL	U
1	0	0	0	0	0	**1**	0

Fig. 4. Example of a QB measurement.

Figure 4 shows an example of how a QB measurement was interpreted. When the value of a QB measurement is 130, its binary value will be 10000010, meaning that IV and AL were activated during the day when the measurement was taken.

Depending on the length of the contract, each customer will have a different number of QB measurements thus these indicators cannot be used in their raw state as a ML algorithm will require a fixed number of inputs. Instead of using the raw measurements, the features described in Table V have been computed for each customer. These features are generated for each *x* alarm (IV, SINC, OW, VH, MP, INT, AL) for different numbers of *n* days (15, 30, 60, 90, 180, 360, 720).

B. Features Extracted From EC Measurements

A sudden decrease in the EC can be noticed for most of the fraud and non-fraud related anomalies. Nevertheless, if the anomaly started before the period of analysis the decrease in the EC cannot be captured and clustering techniques must be introduced in order to capture unusual consumption behaviors.

1) Features Aimed to Detect Recent Anomalies: To detect anomalous measurements, the Z_{score} has been used. This score indicates how many standard deviations away from the mean is a new measurement.

$$Z_{score} = \frac{X_i - \bar{X}_i}{\sigma_{X_i}} \tag{1}$$

where X_i is an EC measurement of the customer i, $\bar{X}_i$ is the mean EC of the customer i and σ_{X_i} is the standard deviation of EC measurements of customer i.

To avoid erroneous results, the measurements have been divided into measurements taken on weekdays, Saturdays or Sundays. The measurements taken during a holiday have been removed. Table V shows the features computed using the Z_{score}. These features were computed for each type of day t (weekday, Saturday, Sunday), for each number of n days (15, 30, 45, 60, 90) and for measurements taken in different w time windows (as described in Table II).

The EC measurements can also be used to detect faults in the meter. The timestamp of each set of measurements can be used to compute the number of measurements received in the last n days. These data can inform a ML model of the number of missing measurements in the last n days for a certain SM.

SM data can also capture zero measurements. To make use of this knowledge, the number of days with 0 kWh consumption has been tracked in order to develop a new set of features. The slope of a linear model approximation of the EC measurements has also been used. Table V enumerates the features developed using the criteria described above. Multiple features were obtained by using different number of days (15, 30, 60, 90).

2) Features Aimed to Detect Old Anomalies: To detect anomalies that have started before the period of analysis, clustering techniques must be employed as for these cases a sudden drop in consumption cannot be observed.

Customer segments were created using the contracted power in each customer sample, to capture unusual behaviors. These segments were created using the k-means clustering algorithm proposed by Lloyd [18]. The optimal number of clusters was found at 25. Customer segments with less than 20 customers have been removed.

To identify abnormal customer profiles, two approaches have been used: distance based and density based measurements.

3) Base Models and Features Based on Distance Measurements: After obtaining customer segments using the contracted power of each customer, base consumption patterns have been created for each month of the analysis. The consumption patterns have been separated by weekdays and weekends. The base consumption patterns for each customer segment have been created using the EC of all non-anomalous customer samples belonging to that segment.

$$B^k_{i,j,t} = \left\{ \frac{1}{N}\sum_{z\in M} P^z_{I_t},\ \frac{1}{N}\sum_{z\in M} P^z_{II_t},\ \frac{1}{N}\sum_{z\in M} P^z_{III_t},\ \frac{1}{N}\sum_{z\in M} P^z_{IV_t},\ \frac{1}{N}\sum_{z\in M} P^z_{V_t} \right\}, \tag{2}$$

where $B^k_{i,j,t}$ is the base consumption pattern of month i, year j of the customer segment k for type of day t (weekday, Saturday, Sunday). M represents the set of customers belonging to the customer segment k that had an inspection without an anomaly detected whilst N is the number of these customers. P_{I_t}, P_{II_t}, P_{III_t}, P_{IV_t}, and P_{V_t} represent the average power consumption for type of day t during the time windows presented in Table II.

After creating base models for each customer segment, several features have been computed for each customer sample (regardless if they had an inspection or not) using the distance between the base model and the customers consumption pattern.

For each customer sample, two consumption patterns have been created by averaging the power consumption of the weekdays and weekends of the last month.

$$C_t = \left\{P_{I_t},\ P_{II_t},\ P_{III_t},\ P_{IV_t},\ P_{V_t}\right\}, \tag{3}$$

where C_t represents the consumption pattern and $P_{I_t}, P_{II_t}, P_{III_t}, P_{IV_t}, P_{V_t}$ are the average power consumptions for type of day t in the last month.

The features were developed by computing the Euclidean and Manhattan distances between each consumption pattern of a customer's sample and its base model. The Manhattan distance was computed for each individual time frame and also for the entire day, whilst the Euclidean distance was computed using all time windows.

$$M_{w_t} = \left| P_{w_t} - \frac{1}{N}\sum_{z\in M} P^z_{w_t} \right|, \tag{4}$$

$$M_{T_t} = \sum_{w=I}^{V} \left| P_{w_t} - \frac{1}{N}\sum_{z\in M} P^z_{w_t} \right|, \tag{5}$$

where M_{w_t} is the manhattan distance of a customer sample for time window w and for type of day t, and M_{T_t} is the total manhattan distance of all time windows.

The euclidean distance was computed using all time windows, and was defined as follows:

$$E_{T_t} = \sqrt{\sum_{w=I}^{V} \left(P_{w_t} - \frac{1}{N}\sum_{z\in M} P^z_{w_t} \right)^2}, \tag{6}$$

where E_{T_t} is the total euclidean distance of all time windows.

The features obtained using distance measurements are shown in Table V.

4) Features Based on Density Measurements: The second approach to detect an unusual customer behavior consisted on using the Local Outlier Factor (LOF) [19]. This metric assigns to each customer profile a degree of being an outlier by measuring how isolated is its consumption profile in comparison with the profiles in its neighborhood.

To compute the LOF for each customer involved, the last month's EC measurements of each customer were clustered together according to their customer segment. In Table V, the features computed using this metric are shown. The features were computed for each type of day t (weekday, Saturday, Sunday).

C. Features Extracted From Electrical Magnitudes

The features developed using the electrical magnitudes (EM) were aimed to detect mainly fraud such as phase inversions and shunts (three-phase customers). The snapshots were divided within three time frames (9AM to 6PM, 7PM to 10PM and 11PM to 8AM). The last snapshot within each time frame has been taken in order to compute the features. Table VI shows the features developed using EM.

TABLE V
FEATURES BASED ON SM DATA

Type of data	Input Features
QB	Number of days with alarm *x* in the last *n* days
	Number of days from last *x* alarm
Daily EC	Number of 0 kWh measurements in the last *n* days
	Slope of a linear model approximation
	Number of measurements received in the last *n* days
EC	Average Z_{score} of measurements taken during time window *w* on type of day *t* in the last *n* days
	Average Z_{score} of daily EC measurements taken on type of day *t* in the last *n* days
	Total euclidean distance for type of day *t*
	Total manhattan distance for type of day *t*
	Manhattan distance of time window *w* for type of day *t*
	LOF score of daily EC measurements for type of day *t*
	LOF score of EC daily profile for type of day *t*

TABLE VI
ELECTRICAL MAGNITUDE-RELATED FEATURES (THREE-PHASE CUSTOMERS)

Type of data	Detection aim	Input Features
Electrical magnitudes	Phase inversion	Phase voltage ≤ 0 (Yes/No)
		Phase imbalance $\Delta V = \frac{V_{max}-V_{min}}{V_{max}}$
		Phase electric current ≤ 0 (Yes/No)
		Phase active power ≤ 0 (Yes/No)
	Shunt	One power factor is 0 whilst the other two are different than 0 (Yes/No)
		Neutral current ratio $\frac{I_N}{I_{max}}$
		Neutral current angle

VII. FEATURES EXTRACTED FROM AUXILIARY DATABASES

The features described in Table VII have been provided by the utility. The majority of features come from the Tariff Summary (TS) database which uses the SM and the portable reading terminals data to compute the monthly EC and the maximum power in up to six different tariff periods.

The Geographic Information System (GIS) data provides information not only on the location of the customer but also on the rate of NTL in the neighborhood. Other auxiliary databases provide information with regards to the technological characteristics (TECH) of the SM such as the brand or whether the meter is located inside/outside. The contracts database offers information related to contract events as well as the activity type of the customer.

VIII. TRAINING, VALIDATION AND TESTING

To evaluate the performance of the methodology described in this paper, the original training dataset is split into a reduced training dataset, a validation dataset and a testing dataset. The validation dataset is used to tune the hyperparameters of our models whilst the testing dataset is used to assess how well the models generalize to new, unseen customer samples.

It is often encountered that the error obtained on the validation dataset is reported as the final error of the model [20]. However, this approach leads to biased error estimates as reported in [21]. In Figure 5, our approach for model selection and evaluation is presented. Given the scarcity of our anomalous samples, a nested cross-validation (NCV) has been chosen to make use of the available data as much as possible. The test fold is used only in the model evaluation stage.

TABLE VII
FEATURES BASED ON AUXILIARY DATABASES

Type of data	Type of Features
TS	Drop in monthly EC consumption using 1 year moving window
	Ratio between monthly ECs and contracted power
	Minimum, maximum, standard deviation and slope of a linear model for monthly ECs
GIS	Latitude, longitude, altitude, distance from shore, province, municipality population density and municipality surface
	% of NTL detected on a radius from 1-10 km
	Number of inspections, failed visits before last inspection and inspections with/without anomalies detected before last inspection
TECH	Type and model SM
	Location SM (inside/outside)
	SM date of fabrication
CONTRACTS	Business type, size, economic activity code
	Change of business activity, contracted power, tariff, SM
	Number of complaints 2-24 months

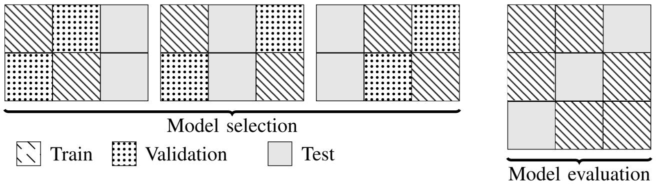


Fig. 5. 3-Fold Nested Cross-Validation example.

As it can be observed, a NCV is a computationally expensive approach compared to other traditional methods. However, its major advantage is that it provides an almost unbiased estimate of the true error [21]. This is extremely important for the utilities as they want to have a realistic assessment of how well the model will generalize to new customer samples.

IX. MODEL SELECTION AND EVALUATION

Before using the features described above in a ML algorithm, several preprocessing steps have been taken: (1) each feature has been standardized to have zero mean and unit variance; (2) the categorical variables have been converted to numerical ones using one-hot encoding; (3) the missing values of continuous features were replaced with the mean value whilst the missing values in discrete features were replaced with the most frequent value.

For model selection and evaluation, a 5-fold nested cross validation was used. Due to computational constraints, the model selection of hyperparameters was made using all the customers from Barcelona. The Scikit-learn library [22] has been used to fit the model using SVM, Logistic Regression (LR) and k-Nearest Neighbors (KNN). The model fitting with XGBoost [23] has been done using its Python API.

TABLE VIII
KNN GRID-SEARCH

Hyperparameter	Range of values
K	2, 4, 8, 16
p	2, 3

TABLE IX
LR GRID-SEARCH

Hyperparameter	Range of values
C	0.001, 0.01, 10, 100
R	L1 norm, L2 norm

A. Model Selection

During model selection, the inner loop of the NCV was used to select the hyperparameters that obtained the best results on the validation dataset. The hyperparameter optimization has been done using a grid-search approach.

1) K-Nearest Neighbors: KNN is one of the simplest classification algorithms. It uses the training data at test time to find the nearest neighbors. In our scenario, to get a probability estimate of having an anomaly for a new customer, the algorithm looks at the results of the on-field inspections. The results of the on-field inspections of the closest neighbors will be therefore averaged in order to compute a probability for the new customer.

Table VIII shows the hyperparameters used during grid-search. The best results were obtained using 16 neighbors (K) and a power parameter of 2 (p) which is equivalent to the euclidean distance.

2) Logistic Regression: The binary LR algorithm has also been used during model selection. This classification algorithm simply takes the matrix of input features X, multiplies it with a matrix of weights θ and passes it through the sigmoid function $g(z) = \frac{1}{1+e^{-z}}$, where $z = \theta^T X$ [24]. The classifier has been trained on a logarithmic loss function using the LIBLINEAR solver [25]. Table IX shows the hyperparameters used during grid-search for LR.

The C hyperparameter represents the inverse of the strength of regularization and it is used to control the overfitting of the model during training. The R hyperparameter represents the type of regularization, either L1 or L2. The best results on the validation folds were obtained using a C of 0.01 and a L2 regularization.

3) Support Vector Machines: As seen in the related work section, SVM are a very popular classifier for anomaly detection in the utilities. Unlike the previous algorithms, SVM do not predict probability estimates but rather decision values.

A SVM algorithm takes the input features into a high dimensional space and tries to find the optimal hyperplane that maximizes the margin between the vectors of the two classes [26]. This margin will be determined by the support vectors of the classes. The support vectors are customer samples from our training dataset that are the closest to the decision function.

Table X shows the hyperparameters used during grid-search for SVM. The hyperparameter C is similar to the LR parameter and represents the inverse of the strength of regularization. The kernel parameter is helpful if the customer classes are not linearly separable by a hyperplane in the high dimensional space. The best results on the validation folds were obtained using a C of 0.001 and a linear kernel.

TABLE X
SVM GRID-SEARCH

Hyperparameter	Range of values
C	0.001, 0.01, 10, 100
Kernel	Linear, Radial Basis Function

TABLE XI
XGBOOST GRID-SEARCH

Hyperparameter	Range of values
Number of trees	1000, 2000
Learning rate	0.01, 0.1
Maximum depth	7, 15
Minimum child weight	1, 10

4) Extreme Gradient Boosted Trees: XGBoost is one of the most popular ML algorithm in the data science community. In 2015, 17 out of 29 winning solutions on the Kaggle platform used XGBoost [23]. The algorithm uses gradient boosting [27] with a regularized cost-function. Gradient boosting builds an additive model by combining the predictions of many "weak" classifiers. The classifier in our case is a regression tree.

The model starts the training process with only one regression tree. This regression tree is looking to find a set of rules that separate customers with/without anomalies as best as possible. After building the first tree, the model adds a new regression tree with each training round. In each round, the model looks where the previous tree has predicted poorly and builds a new tree with a set of rules which will correct the mistakes of the previous one.

Table XI shows the hyperparameters used during grid-search for XGBoost. The best results for XGBoost were obtained using the following hyperparameters: a learning rate of 0.01, a maximum depth of 15, a minimum child weight of 1 and 2000 regression trees.

B. Model Evaluation

One of the most common metrics used for assessing the performance of a ML algorithm is the accuracy. However, the accuracy of an algorithm on a severely imbalanced dataset cannot provide a real assessment on its predictive power. Just by using a naive predictor which predicts that none of the customers has an anomaly in their meter we would achieve an accuracy of approximately 95%. A performance metric that has been proven to be reliable on imbalanced datasets is the AUC score [28], [29]. This metric assesses how fast the true positive rate increases with the increase of the false positive rate. By varying the decision threshold, the trade-off between the true and false positive rates can be observed on the Receiver Operating Characteristic (ROC) curve.

Figure 6 shows the results obtained during model evaluation for each classifier studied. The results were obtained by concatenating the predictions of all the test folds. This gives us a prediction score on the entire training dataset. As it can be observed XGBoost outperforms the rest of the classifiers

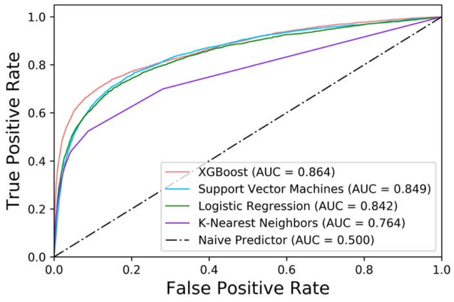


Fig. 6. Receiver Operating Characteristic curve.

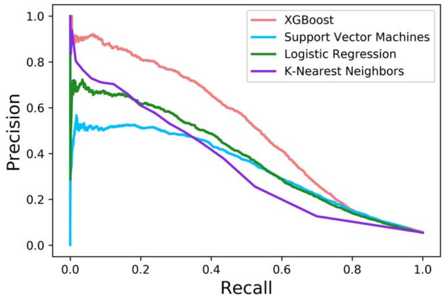


Fig. 7. Precision-Recall curves.

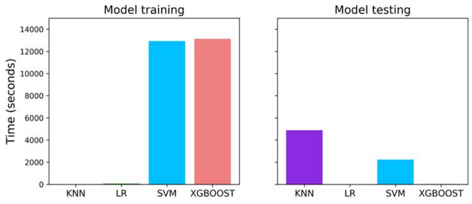


Fig. 8. Execution time.

whilst KNN obtains the lowest performance. The performance of a naive predictor, which predicts that none of the customers is fraudulent, has been added for benchmarking purposes.

Furthermore, the precision-recall curve [30] has been created for each classifier (Figure 7) in order to give a better overview on the performance of each algorithm. As with the ROC curve, the precision-recall curve has been obtained by varying the decision threshold for the probability estimates. When both the PRC and RCL of the model are taken into consideration, the performance of XGBoost is significantly better in comparison with the rest of classifiers. It can reach approximately 70% PRC at a 40% RCL.

Moreover, the execution time of each model during both training and testing can be seen in Figure 8. LR was the fastest algorithm during both training and testing. The experiments were run on a machine with a 3.9 GHz Intel Core i7 CPU.

X. Reducing Data Imbalance

The data imbalance has been reduced using undersampling techniques. With undersampling, some samples of honest customers are being removed during training. The selection of customers to be removed has been done with two methods. The first method removes the samples of customers who

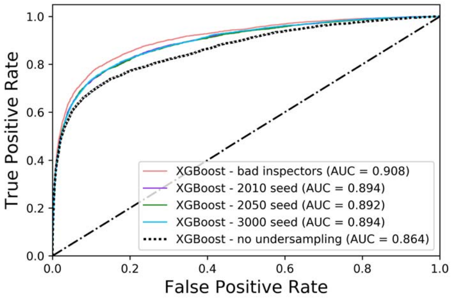


Fig. 9. ROC curve undersampling vs. no undersampling.

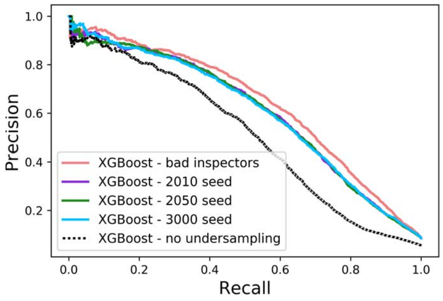


Fig. 10. Precision-Recall curves for undersampling.

were not identified with an anomaly in their meter but have been inspected by inspectors who might have misclassified fraudulent customers for more than 3 times. The misclassification has been assessed by looking at customers who had an inspection with no anomaly detected before an inspection with anomaly detected. The second method removes samples of honest customers using a different number for a random seed. The training dataset has been reduced from 57304 samples to 36806 samples.

Figure 9 shows the results obtained when removing the customer samples with a higher chance of being misclassified as honest customers. The figure shows also the results obtained when doing the undersampling randomly with different random seeds. Undersampling seems to improve the AUC score significantly. Nevertheless, the AUC score obtained using the first method is not much higher than the AUC scores obtained by randomly removing samples.

Figure 10 shows the precision-recall curves obtained with undersampling techniques. Undersampling obtains major improvements on the precision and recall performance. However, the difference between the two undersampling techniques is not conclusive.

XI. Discussion and Comparison

NTL detection is an extremely challenging task as it cannot be constrained solely to an anomaly detection problem. It is often encountered to find anomalous measurements or drastic changes in the customer's consumption pattern that are due to non-malicious factors. We have aimed to tackle these challenges by using all the available data the SMs provide.

Though the authors in [6]–[8] and [11] use SM data, they only use the EC measurements. Furthermore, the work

TABLE XII
COMPARISON WITH THE STATE-OF-THE-ART

Criteria	**[3]**	**[13]**	**[14]**	**[15]**	**Current**
Robustness to detect NTL occurring from beginning	low	low	high	medium	high
Adaptability to new types of NTL attacks	low	low	medium	medium	high
Data privacy	high	high	high	high	medium
Detection delay	12 months	12 months	12 months	-	90 days
AUC score	0.56	0.63	0.74	0.84	0.91

TABLE XIII
PERFORMANCE ANALYSIS ON TYPE OF DATA

Data source	**Type of data**	**AUC score**
SM data	EC	0.80
	EC+QB	0.85
	EC+QB+EM	**0.88**
Auxiliary databases	TS	0.76
	TS+GIS	0.84
	TS+GIS+TECH	0.85
	TS+GIS+TECH+CONTRACTS	**0.86**

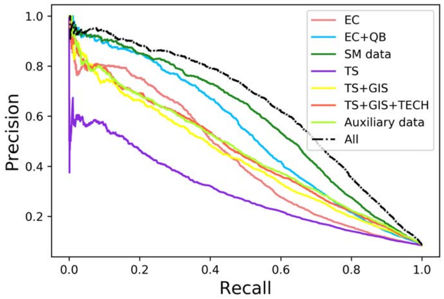


Fig. 11. Precision-Recall curves for different subsets of features.

presented in [16] improved the precision of campaigns that target customers with null consumption from 4.67% to 14.75%, in the same utility, Endesa. In comparison, our approach provides the utility a methodology to detect all types of NTL. Moreover, our best model achieves ≈ 21% precision for the new on-field inspections generated by our model.

Table XII shows a comparison between our methodology and the methodologies which report the AUC score. The robustness to detect NTL occurring from the beginning is assessed on whether the methodology compares the consumption behavior of the customer with similar customers as a descent in consumption cannot be observed. The methodologies presented in [3] and [13] do not make any comparison between the consumption behavior of similar customers. The approach in [15] compares the consumption of a customer with the average consumption, without using any clustering techniques. Meira *et al.* [14] make a thorough comparison between customers by using k-means clustering on geographical data, transformers and consumption profiles. The adaptability to new types of attacks is related with the diversity of type of features as well as the granularity of EC measurements. Approaches such as [3] and [13] use only the monthly EC and geographical data, making it harder to detect new types of NTL. The methodologies presented in [14] and [15] use a wider range of features. Nevertheless, the low granularity of monthly EC makes it more difficult to adapt to emergent NTL attacks such as intermittent fraud. Higher granularity of EC measurements reduces the privacy of customer but it increases the adaptability to new types of NTL and also shortens the detection delay.

We have attempted to replicate the methodologies presented in Table XII, using our dataset. For the experiment presented in [3], we have obtained an AUC score of 0.59. However, as this methodology requires consumption history of at least 12 months, we had to discard 20% of our training data. Training with a smaller batch of customers distorts the final result. The methodology presented in [13] introduces two major data leakages during training. The authors computed the inspection rate and the NTL rate of the neighborhood area of a customer, without removing the customer itself. In this case, if a customer had an inspection with NTL detected, the NTL detection rate in his/her area would be higher as the result of this inspection was taken into consideration. The other leakage is that they take information from the future to train the algorithm, as the date of the client inspection is not taken into account. To replicate the experiments in [14] and [15], we had to make assumptions on some of the parameters, (e.g., number of clusters used for the consumption profile, the time horizon of the analysis). Alas, a thorough comparison is not possible.

Furthermore, as none of the approaches presented in the table above used SM data, we have attempted to assess its impact on the AUC score by training the XGBoost model with different subsets of features (see Table XIII). A 0.88 AUC score was obtained only by using the features that the SM provide, without the use of auxiliary databases.

Figure 11 shows the precision-recall curves for all the subset features described in Table XII. The SM data features obtain much higher precision for the same recall obtained by the auxiliary data features.

XII. CONCLUSION

This paper presents a methodology for non-technical loss detection based on the use of smart meter data and auxiliary databases as raw data that feed a supervised machine learning algorithm (XGBoost). During training, the features of customers which had at least an inspection were used to train the algorithm.

The methodology has been tested on real data of the largest distribution company in Spain (Endesa), obtaining an AUC score of 0.91, higher than any previous approach as shown in the text. Moreover, the precision and recall for various decision thresholds on the probability estimates are also shown for different subsets of features, highlighting the advantages of using all the data.

This methodology is currently implemented in a real NTL campaign using the XGBoost classifier for training. It currently obtains a precision of ≈ 21% for new on-field inspections generated by our ranked list of customers.

ACKNOWLEDGMENT

The authors are grateful to Antonio Peralta-Sánchez, Daniel Capilla-Cerezo, José D. Carvajal-Valderrama, and

Lourdes Díaz-Mena, from the Endesa Distribución - Energy Recovery - Data Science team, for their invaluable help during the course of this project.

References

[1] R. R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on advanced metering infrastructure," *Int. J. Elect. Power Energy Syst.*, vol. 63, pp. 473–484, Dec. 2014.

[2] P. Glauner *et al.*, "The challenge of non-technical loss detection using artificial intelligence: A survey," *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, pp. 760–775, 2017.

[3] P. O. Glauner *et al.*, "Large-scale detection of non-technical losses in imbalanced data sets," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Minneapolis, MN, USA, 2016, pp. 1–5. [Online]. Available: http://arxiv.org/abs/1602.08350

[4] A. J. Nezhad, T. K. Wijaya, M. Vasirani, and K. Aberer, "SmartD: Smart meter data analytics dashboard," in *Proc. ACM 5th Int. Conf. Future Energy Syst.*, Cambridge, U.K., 2014, pp. 213–214.

[5] J. L. Viegas, S. M. Vieira, R. Melício, V. M. F. Mendes, and J. M. C. Sousa, "Classification of new electricity customers based on surveys and smart metering data," *Energy*, vol. 107, pp. 804–817, Jul. 2016.

[6] P. Jokar, N. Arianpoo, and V. C. M. Leung, "Electricity theft detection in AMI using customers' consumption patterns," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 216–226, Jan. 2016.

[7] V. Ford, A. Siraj, and W. Eberle, "Smart grid energy fraud detection using artificial neural networks," in *Proc. IEEE Symp. Comput. Intell. Appl. Smart Grid (CIASG)*, Orlando, FL, USA, 2014, pp. 1–5.

[8] C. Cody, V. Ford, and A. Siraj, "Decision tree learning for fraud detection in consumer energy consumption," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Miami, FL, USA, 2016, pp. 1175–1179.

[9] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and M. Mohamad, "Nontechnical loss detection for metered customers in power utility using support vector machines," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 1162–1171, Apr. 2010.

[10] B. Dangar and S. K. Joshi, "Electricity theft detection techniques for metered power consumer in GUVNL, GUJARAT, INDIA," in *Proc. Clemson Univ. Power Syst. Conf. (PSC)*, Clemson, SC, USA, 2015, pp. 1–6.

[11] S. Y. Han, J. No, J.-H. Shin, and Y. Joo, "Conditional abnormality detection based on AMI data mining," *IET Gener. Transm. Distrib.*, vol. 10, no. 12, pp. 3010–3016, Sep. 2016. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-gtd.2016.0048

[12] B. C. Costa *et al.*, "Fraud detection in electric power distribution networks using an ann-based knowledge-discovery process," *Int. J. Artif. Intell. Appl.*, vol. 4, no. 6, pp. 17–23, 2013.

[13] P. Glauner *et al.*, "Neighborhood features help detecting non-technical losses in big data sets," in *Proc. 3rd IEEE/ACM Int. Conf. Big Data Comput. Appl. Technol. (BDCAT)*, Shanghai, China, 2016, pp. 253–261. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3006299.3006310

[14] J. A. Meira *et al.*, "Distilling provider-independent data for general detection of non-technical losses," in *Proc. IEEE Power Energy Conf. Illinois (PECI)*, Champaign, IL, USA, 2017, pp. 1–5.

[15] B. Coma-Puig, J. Carmona, R. Gavaldà, S. Alcoverro, and V. Martin, "Fraud detection in energy consumption: A supervised approach," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Montreal, QC, Canada, 2016, pp. 120–129.

[16] J. I. Guerrero *et al.*, "Non-technical losses reduction by improving the inspections accuracy in a power utility," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1209–1218, Mar. 2018.

[17] R. E. de Espana, *Protocolo de Comunicaciones Entre Registradores y Concentradores de Medidas o Terminales Portatiles Lectura*, RED Eléctrica Deespaña, Madrid, Spain, 2002.

[18] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[19] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, Dallas, TX, USA, 2000, pp. 93–104.

[20] D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models," *J. Cheminformat.*, vol. 6, no. 1, pp. 1–15, 2014.

[21] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformat.*, vol. 7, no. 1, p. 91, 2006.

[22] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Feb. 2011.

[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Conf. Knowl. Disc. Data Min.*, San Francisco, CA, USA, 2016, pp. 785–794.

[24] A. Ng. *Machine Learning*. Accessed: Jan. 6, 2017. [Online]. Available: https://www.coursera.org/learn/machine-learning

[25] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[27] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[28] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[29] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," *Mach. Learn.*, vol. 31, no. 1, pp. 1–38, 2004.

[30] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, 2006, pp. 233–240.

Madalina Mihaela Buzau received the B.Eng. degree in power systems from the Politehnica University of Bucharest and the M.Res. degree in electrical engineering and sustainable development from the Lille University of Science and Technology. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, University of Seville. Her main research focus is on the usage of smart meter data and machine learning algorithms for non-technical loss detection in the utilities.

Javier Tejedor-Aguilera received the telecommunication engineering degree from the University of Seville, Spain. He is currently the Endesa Distribución responsible for non-technical losses detection. His primary areas of interest are data science, machine learning, and deep learning. He is an active participant in machine learning competitions.

Pedro Cruz-Romero (M'06) received the Ph.D. degree in electrical engineering from the University of Seville, Spain, in 2000. He is currently an Associate Professor. His primary areas of interest are magnetic-field mitigation and transmission and distribution operation and planning.

Antonio Gómez-Expósito (F'05) received the electrical engineering and doctor degrees from the University of Seville, Spain, where he is currently the Endesa Red Industrial Chair Professor. His primary areas of interest are optimal power system operation, state estimation, digital signal processing, and control of flexible ac transmission system devices.