Tesis Doctoral
Ingeniería Eléctrica

Machine learning algorithms for the detection of non-technical losses in electrical distribution networks

Autor: Madalina-Mihaela Buzau
Directores: Pedro Cruz-Romero
Antonio Gómez-Expósito

Departamento de Ingeniería Eléctrica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Tesis Doctoral
Ingeniería Eléctrica

Machine learning algorithms for the detection of non-technical losses in electrical distribution networks

Autor:

Madalina-Mihaela Buzau

Directores:

Pedro Cruz-Romero
Antonio Gómez-Expósito

Departamento de Ingeniería Eléctrica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

2019

Tesis Doctoral: Machine learning algorithms for the detection of non-technical losses in electrical distribution networks

Autor: Madalina-Mihaela Buzau
Directores: Pedro Cruz-Romero
Antonio Gómez-Expósito

El tribunal nombrado para juzgar la Tesis arriba indicada, compuesto por los siguientes doctores:

Presidente:

Vocales:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

To my family
To my professors

Acknowledgements

This work would have never been possible if it wasn't for the help and support of several people and organizations. First and foremost, I want to say a big thank you to the directors of my thesis: Pedro Cruz-Romero and Antonio Gómez-Expósito. Thank you for your invaluable feedback and support throughout my thesis. I am forever grateful for the freedom that you gave me in exploring my own research ideas, allowing me to be more creative, as well as for the support that you gave me throughout my publishing journey.

Throughout this thesis, I have greatly benefited from the amazing collaboration between the Electrical Engineering Department of the University of Seville and the Energy Recovery - Data Science team of Endesa Distribución. I would like to thank Javier Tejedor-Aguilera, for his amazing support and insights and for instilling in me the passion for data science. As the non-technical losses manager of Endesa, Javier gave me a clear understanding of how NTL detection is handled by electricity utilities as well as guided me throughout my research ideas so that they can be useful in the real environment. I would like to thank as well the talented data scientists and engineers of this department: Antonio Peralta-Sánchez, Daniel Capilla-Cerezo, José D. Carvajal-Valderrama and Lourdes Díaz-Mena. I will forever cherish the great memories that I have gathered whilst working with you!

I am deeply grateful to the European Community's Seventh Framework Programme (FP7-PEOPLE-2013-ITN) for funding part of this thesis through the ADVANTAGE (Advanced Communications and Information processing in smart grid systems) project as well as to Endesa, for funding the last stage of my PhD research.

Many thanks to my office colleagues: Catalina Gómez-Quiles and Guadalupe Arcia-Garibaldi. Thank you for making my time enjoyable at the office as well as for the interesting discussions we've had during my time here.

Last but not least, a big thank you to my family and Skype. Though thousands of kilometers apart, my parents were always there for me, through the good and the hard times of this PhD.

Contents

List of Abbreviations

ANOVA	Analysis of variance
CNN	Convolutional neural network
CSS	Charged systems search
CWR	Credit worthiness rating
DSE	Distribution state estimation
DT	Decision trees
EC	Energy consumption
ELM	Extreme learning machines
EM	Electrical magnitudes
EMCs	Electro-mechanical meters
FPR	False positive rate
GBDT	Gradient boosting decision tree
GUI	Graphical user interface
HNN	Hybrid neural network
KEPCO	Korea electric power cooperation
KNN	K-nearest neighbors
LOF	Local outlier factor
LR	Logistic regression
LSTM	Long short-term memory
LV	Low voltage
ML	Machine learning
MLP	Multi-layer perceptrons
MV	Medium voltage
NB	Naive bayes
NCV	Nested cross-validation
NN	Neural networks
NTL	Non-technical losses
NTRU	Number theory research unit
OPF	Optimum-path forest

$OS-ELM$	Online sequential extreme learning machines
$PR-AUC$	Area under the precision-recall curve
PRC	Precision
PSO	Particle swarm optimization
QB	Quality byte
RCL	Recall
RD	Research and development
RF	Random forests
$RMSE$	Root mean square error
$ROC-AUC$	Area under the receiver operating characteristic curve
SAE	Stacked autoencoder
$SCADA$	Supervisory control and data acquisition
$SGCC$	State Grid Corporation of China
SMs	Smart electricity meters
$SMOTE$	Synthetic minority over-sampling technique
SOM	Self-organising maps
SVM	Support vector machines
TNB	Tenaga nasional berhad
TPR	True positive rate
TSR	Three sigma rule
VEE	Validation, estimation and editing
$WD-CNN$	Wide and deep convolutional neural networks
WLS	Weighted least squares

1 Introduction

1.1 Importance of non-technical losses detection

Non-technical losses (NTL) represent a serious concern for electricity utilities as they are responsible for significant revenue losses as well as affecting the power grid reliability. These losses are defined as the energy consumption (EC) of the clients that has not been billed by the utility [1]. Worldwide, a recent report has estimated that NTL are responsible for yearly revenue losses of $96 billion [2]. At the grid level, NTL can affect the power system operation by overloading transformers and causing voltage unbalances as well as providing uncertainty with regards to the real consumption [3, 4, 5].

NTL affects honest customers as well, as high NTL rates will reflect in the electricity price as well as in the reliability of the power grid. Moreover, fraudulent connections to the grid increase the risk of fire and electrocutions [6]. Consequently, the electricity utilities are increasingly investing more time and effort to reduce NTL losses given their negative technical and economic impact.

The current way to mitigate the effects of NTL is through on-field inspections based on customer data analysis. The objective of these inspections is to find customers with an anomaly or a fraud in the electricity meter in order to recover the revenue loss produced by these customers. In the past, these inspections were made based on simple rules derived from expert knowledge. With the roll-out of smart-grid technologies, the utilities have now access to a more in-depth understanding of their customers consumption behavior due to increased data granularity. The algorithms for NTL detection have thus shifted from simple rules to more complex models based on machine learning. The mix of increased data granularity and machine learning has pushed this field forward, as many advances have been reported in the recent literature.

Ultimately, the recovery of the revenue losses caused by NTL majorly depends on the precision of these on-field inspections. A low precision will further augment these losses rather than mitigate them. Besides the economic impact, a high precision of on-field inspections can further discourage recurrent fraud behavior or new frauds. Thus, it is extremely important to advance and increase the precision of future on-field inspections

for an efficient revenue loss recovery as well as to increase the security of the power grid and improve its operations.

1.2 Non-technical losses detection in the smart metering context

In this thesis, we will explore the capabilities of machine learning algorithms for NTL detection in a smart metering context, where electricity utilities have access to measurements provided by smart electricity meters (SMs). It is estimated that by 2020, 70% of the European Union customers will have their conventional meters replaced with SMs [7].

The SMs roll-out has a positive impact on both power grid planning and operations. As an example, the power grid planning can be improved by using the SMs data to correct topological errors. Furthermore, by integrating Supervisory Control and Data Acquisition (SCADA) systems with SMs data, the power systems operations and reliability can be improved by increasing network observability [8]. SMs also improve NTL detection in electricity grids, as they provide better security, control [9] and a better understanding of the customer's consumption behavior through increased data availability.

SMs are measuring devices that are able to provide high-granularity timestamped EC measurements as well as additional data in comparison with conventional electro-mechanical meters (EMCs) [10]. The additional data consists either of power quality measurements or events logs. The measurements related to power quality monitoring are: current, voltage, power factor, active and reactive power [5] whilst the events logs monitor the activation of different alarms in the SMs.

Table 1.1 shows a comparison between SMs and EMCs. An important advantage of SMs, besides increased data availability and granularity, is the capability of remote reading. This removes the possibility of human errors that might occur with EMCs readings. SM data also helps to detect zero or missing measurements. It gives a better overview of the EC pattern by being able to differentiate the EC by time and type of day.

Table 1.1 Smart meters versus electro-mechanical meters.

Smart meters	Electro-mechanical meters
15 minutes/hourly measurements	monthly measurements
energy consumption, power quality and events monitoring	energy consumption
remote reading	manual reading

1.3 Classification of non-technical losses

NTL can occur due to both malicious and non-malicious causes. NTL that occurs due to non-malicious cases can be attributed either to billing errors or meter malfunctioning due to a component fault. The malicious cause of NTL is energy theft. Energy theft can be done through various techniques. Though SMs bring a lot of advantages due to remote

reading and increased data granularity, they also introduce new methods for committing energy theft [11, 12, 13, 14]. Energy theft techniques have been classified in [13] as cyber attacks or physical attacks. Cyber attacks can occur through any sort of communication interference that aims to block, interrupt or alter the EC measurements [13]. Physical attacks occur either through meter tampering or double-tapping. Meter tampering can be done either by mechanically altering the SM components, manipulating the wires [15] or through high-frequency interference sources or strong magnets [16]. A common way of meter tampering is using a shunt device between the input and output terminals of the SM to divert the current. Figure 1.1 shows a shunt fraud scenario. With a shunt fraud, the EC pattern stays the same but at a lower amplitude.

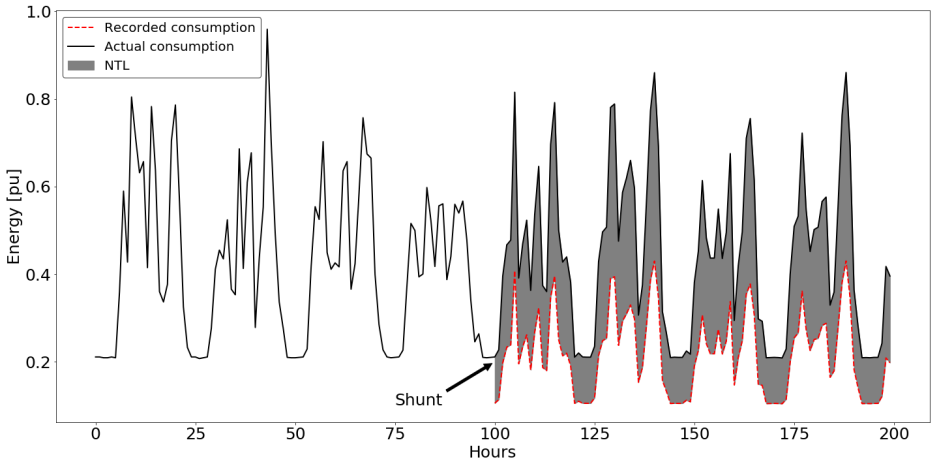


Figure 1.1 Shunt fraud scenario.

Another common method of energy theft is double tapping, where part of the consumption is connected directly to the grid, bypassing the SM. Usually, the customer will bypass the appliances with a higher EC [15]. Figure 1.2 shows the EC scenario of a double tapping case that happens from the beginning of the client's contract. Double tapping changes significantly the real EC pattern, unlike a shunt.

Last but not least, Figure 1.3 shows an example of an NTL that occurs due to a non-malicious cause: an electronic fault. In this case, the meter simply records 0 kWh or null measurements. Though this case might seem easy to detect at a first glance, it can be easily mistaken as the EC pattern of client who goes on a long holiday.

As it can be noticed in Figures 1.1, 1.2 and 1.3, NTL occurs whenever the billed EC is lower than the actual consumption.

This thesis will not focus on specific types of NTL or trying to understand the different techniques through which a customer can alter the readings of a SM. Rather, it will treat NTL as a black-box, providing a global solution for NTL detection in electricity utilities. This means that the methodologies developed in this thesis will aim to detect any type of NTL, regardless of the source. This is especially important since, as mentioned previously,

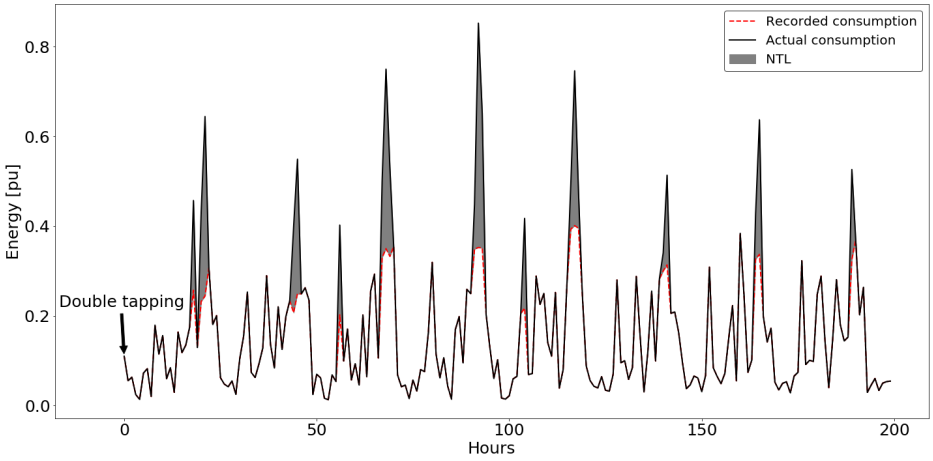


Figure 1.2 Double tapping scenario.

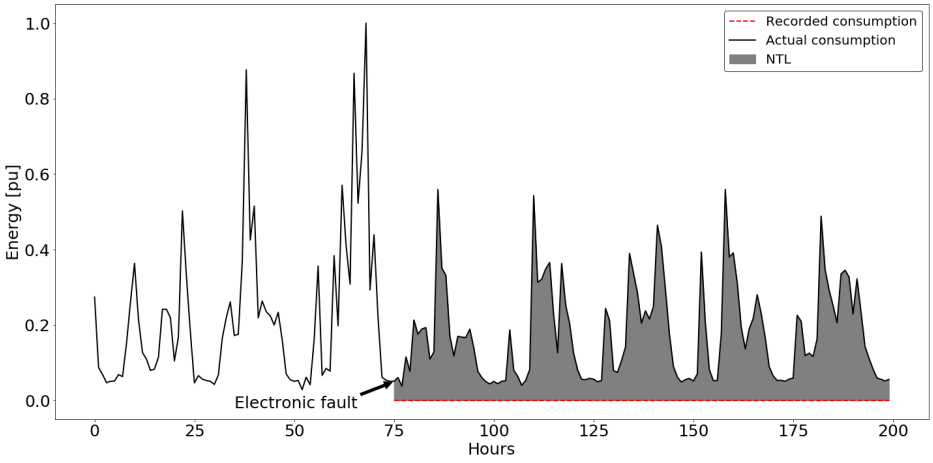


Figure 1.3 Electronic fault scenario.

SMs introduce new methods for energy theft and methodologies can be become easily outdated if they cannot adapt automatically to new types of NTL.

Hence, the following chapters will focus on providing a methodology to detect any type of mismatch between the recorded and actual consumption of a SM that results in an NTL.

1.4 Non-technical losses detection with machine learning

The current NTL detection algorithms fall into three categories: data-oriented, grid-oriented or hybrid-oriented which is a mix between these two [17]. This thesis will

investigate the capabilities for NTL detection of data-oriented approaches. Data-oriented approaches exploit the data available at the customer level, without the need of any additional hardware components. The following types of data are considered to be available: SM raw data, technical and geographical characteristics of the SM and the results of previous on-field inspections.

Supervised learning algorithms are used on datasets where each sample has its corresponding target [18]. Thus, the results of previous on-field inspections can be used to train a machine learning (ML) model in a supervised manner. Figure 1.4 shows an example of a training dataset for NTL detection. During training, the ML model will learn a function that is able to map the input data of each customer to its corresponding target. Once the model has been trained, it can be used to make predictions on new unseen customer samples, using the mapping function learned during training. NTL detection is a binary classification task as there are only two target values (0 - no NTL detected in the SM, 1 - NTL detected in the SM).

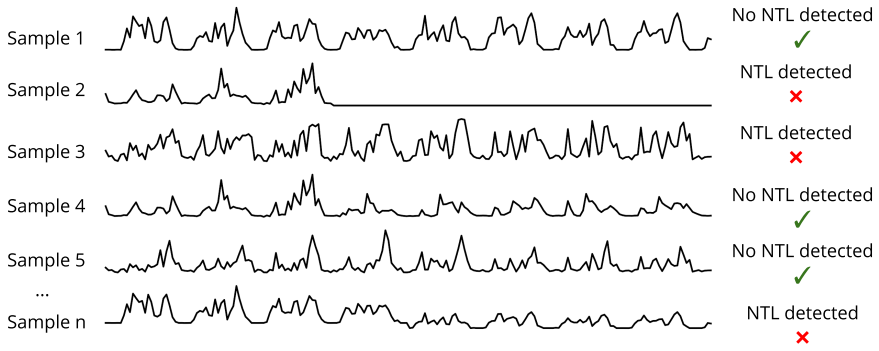


Figure 1.4 Example of an NTL dataset.

SM raw data is, however, unstructured, making it difficult for traditional supervised algorithms such as decision trees to discover useful patterns. In most cases, a structured dataset is created after extracting relevant features for NTL detection from the raw SM data. The feature extraction methodology is based either on expert knowledge or unsupervised learning algorithms. Unsupervised learning algorithms are used for clustering, density estimation or dimensionality reduction [18]. The feature extraction processing step can be omitted when using deep neural networks, which are able to learn by themselves the appropriate representation of data.

Nonetheless, NTL detection in electricity utilities is an extremely challenging case of anomaly detection. The main struggle of an ML algorithm built for this task is not merely detecting changes in the customer's consumption pattern but rather if these changes are due to malicious or non-malicious factors (e.g. holidays, change of lifestyle, appliances).

Furthermore, attempting to detect NTL using a supervised approach can be quite challenging as this is an extremely imbalanced classification problem [19]. Naturally, the number of customers who are not committing fraud or have a faulty meter is much higher than the number of customers detected with NTL. Moreover, as the customer samples are labeled manually by on-field inspectors they are prone to human error. Introducing misclassified samples makes it more difficult for a ML model to distinguish between classes.

1.5 Research objectives

The global objective of this thesis is to explore the capabilities of machine learning algorithms and SM data for NTL detection in electricity utilities. The goal of these algorithms is to detect any type of NTL, regardless of its source (excluding NTL which comes from billing errors). This research is focused on two types of customers: industrial/large commercial (contracted power > 50 kW) and residential/small commercial (contracted power < 15 kW).

The thesis has the following objectives:

- Research and develop feature extraction methodologies based on SM data availability.
- Investigate the performance of state-of-the-art machine learning algorithms using handcrafted feature engineering methodologies.
- Assess the impact of SM data availability for NTL detection.
- Investigate the performance of algorithms that are not based on expert knowledge, using deep neural networks.

1.6 Outline

This thesis has been written in the short format, based on the published work in [20] and [21]. The first part of Chapter 2 will give an overview of grid-oriented methodologies. Its second and third part will focus on describing the state-of-the-art in hybrid-oriented and data-oriented methodologies for NTL detection. Chapter 3 will provide a summary of the results obtained by the proposed methodologies. Its first part starts by giving an overview of feature engineering techniques for NTL detection, for multiple types of data: EC measurements, smart meter alarms and electrical magnitudes. The performance of several state-of-the-art ML algorithms is also investigated. This part of the chapter finishes by describing different techniques used for tackling the imbalance that occurs in NTL datasets as well as the impact of data availability on the performance of NTL detection models. The second part of Chapter 3 will describe a deep learning architecture that is capable to handle both sequential and non-sequential data, such as the EC history. It will also provide a comparison between the proposed deep learning model, traditional machine learning models and previously proposed deep learning models for NTL detection. Last

but not least, Chapter 4 and 5 will discuss the main insights obtained after experimenting with both methodologies as well as provide the final conclusion for this thesis. Both publications [20, 21] can be found in the Appendices A and B.

2 State of the art in non-technical losses detection

This chapter will present the state of the art in NTL detection methodologies. Although the work presented in this thesis focuses on data-oriented methodologies, this section will give a broader overview of all the methodology types that are used in this field. Thus, the next sections of this chapter will discuss the following types of methodologies: grid-oriented, data-oriented and hybrid-oriented, which is a mix between both [17].

2.1 Grid-oriented methodologies

Grid-oriented methodologies use energy balances, power flows or state estimation to detect where NTL occurs. Besides using the data recorded at the customer level, these methodologies rely on measurements made throughout the entire distribution grid. Most of these methodologies also need knowledge of the network topology and parameters.

In [22], the authors propose a methodology for NTL detection at the MV/LV transformer level, using distribution state estimation (DSE). This methodology needs a single main measurement at the beginning of the distribution substation. Technical and commercial data of the grid, such as the connectivity between the meter of the customer and the transformer, is also needed. The monthly billed energy at the MV/LV transformer level has been used in order to assign a ratio of the total power consumption provided by the main measurement, to each transformer. The main advantage of this methodology is that it has very low metering requirements. Its drawback is that it cannot detect NTL at the customer level.

A methodology for detecting NTL using SM data has been proposed in [23]. The method has been devised specifically for theft detection. An energy balance is performed in order to detect NTL at the MV/LV transformer level. If an NTL is detected at this level, the methodology will further detect the NTL location at the customer level, by comparing the estimated and measured voltages of the SMs. This methodology has been tested on a

simulated typical LV grid configuration from the Netherlands, with 240 customers, and was shown to be able to detect intermittent NTL cases as well.

The authors use DSE to detect NTL in [24], using the Weighted Least Squares (WLS) technique. The methodology assumes availability of network and measurement data. The distribution transformers found with bad data measurements, according to the DSE, can be shortlisted as locations for possible on-field inspections. The model has been tested on a 11-bus test system and has been shown to outperform traditional meter data validation, estimation and editing (VEE) methods for estimating meter load curve if the EC patterns are volatile.

In [25], the authors use the Energy Flow Problem Solution (based on state estimation theory) to detect NTL. The methodology is based as well on detecting bad data measurements, through the use of energy balances. It was tested on a very simple grid and showed a need of redundant measurements for an accurate NTL detection.

Using power flow computations, the authors in [26] estimated NTL at the distribution transformer level. Since it is extremely hard in practice to have an exact knowledge of the network topology and parameters, the authors assumed that the distribution circuit has a simplified topology and connectivity. This assumption allowed to compute the level of technical losses and indirectly the NTL. The methodology has been tested on a simulation with 10 customers, where one of the customers was committing theft. The simulation has shown that the methodology is able to detect NTL percentages as low as 10% at the transformer level.

An extension of the work presented in [26] can be found in [27]. For a better estimation of the technical losses, the authors considered the temperature dependency of the resistances within the test distribution circuit. The methodology was able to detect NTL percentages as low as 4%, at the transformer level. The main drawback of these two methodologies is that they need consumption data, where there is no theft, for an accurate estimation of the technical losses.

A two-stage methodology for NTL detection in LV networks using SM data has been proposed in [28]. The first stage of the methodology is detecting NTL at the distribution transformer level by checking the difference between the current measured on the secondary side of the distribution transformer and the sum of all the SMs current measurements. The second stage of the methodology detects NTL at the branch level in the LV grid, by using the network topology and the lines' impedances. If the impedances are unknown, they can be estimated using historical measurement data, where no NTL was present. The methodology has been tested on a typical Portuguese LV network and has been shown to obtain a success rate of 85%, even in cases with incomplete data of the network.

In [29], the authors studied the problem of power quality monitoring for NTL detection. Their methodology establishes the minimum number of power quality monitors needed for total system observability whilst taking into consideration the P-median model. The P-median model has been used to select the location of monitors, making sure that the most important loads are monitored. The methodology has been tested on a 40-bus power distribution branch of a Brazilian utility.

The authors in [30] propose a methodology to detect NTL that occurs due to cyber-attacks. To establish the location of NTL, a comparison is made between the measurements received from reliable devices (e.g. phasor measurements units, intelligent electronic

devices) and the estimated measurements obtained from state estimation. The exact location of the NTL is then found using the A-Star algorithm. The methodology has been evaluated on a real distribution network.

A methodology for NTL detection based on state estimation, that preserves the privacy of the consumers, has been proposed in [31]. The privacy of the consumers has been preserved by encrypting the data using the Number Theory Research Unit (NTRU) algorithm. The experiments show that the proposed methodology is able to accurately detect NTL whilst attaining data confidentiality and authentication.

As shown above, grid-oriented methodologies can detect NTL with high accuracy. However, most of them require knowledge of network topology and parameters as well as the installation of additional metering devices in order to increase the observability of the distribution system. Thus, these methodologies cannot yet be widely used by the utilities as their data and metering requirements are not easily attainable in practice. Grid-oriented methodologies that are focused on detecting NTL at the distribution transformer level have lower data and metering requirements, hence they can be faster adopted by the utilities in the future.

2.2 Hybrid-oriented methodologies

Hybrid-oriented methodologies use, in the first stage, network related data to detect NTL at the distribution transformer level or in several areas of the LV network. The second stage of these methodologies detect NTL at the customer level, either by employing the use of statistical methods on the energy consumption data of the customers or through the use of machine learning algorithms.

In [32], the authors propose a methodology for NTL detection using DSE and analysis of variance (ANOVA). The DSE is used to detect distribution transformers with anomalous usage using the normalized residual test. To perform the DSE, the following information was used: customer SM data, historical data from SCADA and network topology information from various sources such as outage management and customer information systems. After identifying transformers with anomalous usage, the NTL is detected at the customer level using ANOVA. This analysis is done by comparing the EC measurements of a customer with its baseline EC profile that has been previously validated.

Another hybrid NTL methodology is proposed in [33], which uses DSE and an Optimum-Path Forest (OPF) classifier. The methodology uses the monthly EC measurements of the last 12 months as well as the geographic location of the meter. The DSE is used to estimate the NTL in each month at the bus level. These NTL estimations are added to the input of the OPF classifier. The OPF classifier was trained in a supervised manner, on a dataset with NTL samples that have been created synthetically. The NTL was considered to occur either as partial or total load reduction. The methodology needs additional measurement equipment as well as knowledge of the network topology and parameters. The results show that using the NTL estimations provided by the DSE brings significant improvements in the performance compared to using the solely EC measurements alone.

The authors in [34] use DSE for NTL detection in LV networks. The NTL was considered to occur due to electricity theft. The authors used the semi-definite relaxation method,

instead of the standard Newton Raphson method, in order to obtain the global optimal solution for the state estimation. Suspicious users were further investigated with ANOVA. The methodology has been tested on a 8-bus distribution system and has been shown to detect successfully electricity theft.

In [35], the authors propose a methodology based on state estimation and pattern recognition, for a distribution system with advanced metering infrastructure. The first stage of the methodology uses the Weighted Least Squares State Estimator (WLS-SE) to identify NTL at the transformer level. The authors show that analyzing the gross errors identified during state estimation using a geometrical approach, yields better results than the residual approach. The second stage of the methodology uses a multivariate normal distribution to detect abnormal EC patterns among customers. The NTL cases have been synthetically generated and were considered to come either from total or partial bypassing of the customer's meter. The methodology has been tested on an IEEE 69-bus test feeder and showed that it outperforms previous approaches based on residual analysis of gross errors.

A two-stage methodology for NTL detection has been proposed in [36]. The first stage of this methodology identifies NTL at the transformer level. The NTL is detected at the customer level using the K-Means algorithm and a Support Vector Machines (SVM) classifier trained for each customer in part. The NTL samples are synthetically generated. The methodology has been tested on SM data of over 5000 Irish homes and businesses during 2009 and 2010 and showed high performance even for sampling rates as low as 4 samples/day.

The authors in [37] proposed a graphical user interface (GUI) platform to detect NTL losses. The methodology is based on three stages. The first stage detects NTL at the distribution transformer level by comparing the measured current on the secondary side of the transformer with the aggregated current measurements of the SMs connected to it. The second stage detects NTL at the customer level by using fuzzy logic and a SVM classifier. The last stage of the methodology checks for correlations between anomalous customers and their event logs. The developed methodology aims to be implemented in a pilot project side with real SM data of customers.

In [38], the authors propose a methodology which detects areas with high NTL, in the LV network, through the use of SM measurements and energy balances. The second stage of the methodology detects NTL at the customer level, in the areas identified with high NTL, using a SVM classifier. The classifier uses as an input energy consumption data, clients' registration data and socio-economic indices. The methodology has been tested as part of a research and development (RD) project of ANEEL (Brazilian Electricity Regulatory Agency). The methodology to detect areas with high level of NTL has not been developed and tested at that stage of the project.

Hybrid-oriented methodologies can be more easily adopted by the utilities as they have lower data requirements for network topology, parameters and measurements. Nevertheless, most utilities do not currently have the necessary network data availability to deploy these methodologies on a large scale.

2.3 Data-oriented methodologies

Data-oriented methodologies are a very popular area of research for NTL detection, due to their low data availability requirements. Generally, these methodologies are based only on the data collected at the customer level. The data collected at the customer level includes the measurements recorded by the electricity meter as well as some information of the customer (location, meter brand, meter location etc.) that usually resides in auxiliary databases. In the beginning, these methodologies were based on simple rules extracted from customer consumption data, often relying on expert knowledge. Nowadays, these methodologies are based exclusively on machine learning (ML) techniques. The ML algorithms that are used for NTL detection are mainly based on supervised learning. There are a few works based on ML algorithms that use unsupervised or semi-supervised learning, such as [39] or [40], but this area of research is not fully developed at this time.

Methodologies for NTL detection based on supervised learning use the results of previous on-field inspections as labels, in order to create a training dataset. The objective of these algorithms is to classify as accurately as possible whether a customer sample has NTL or not. To train the supervised ML classifier, the data of each customer that had an inspection is collected and used as an input during the training stage. Two types of methodologies for processing the input, can be found in the literature:

- Input processing based on feature engineering - these methodologies are using expert and domain knowledge to extract features from raw consumption data recorded by the meter and auxiliary data that provides additional information of the meter (e.g. meter brand, location, contract type).
- Input processing based on raw data - these methodologies use the raw data recorded by the meter and the auxiliary data without any further processing.

Methodologies based on feature engineering can achieve great performance as they rely on the insights gathered by on-field inspectors or utility employees, whilst the methodologies based on raw data have the advantage that they do not have to rely on such expertise and are not constrained to the expert knowledge for the NTL detection task. Table 2.1 shows the main characteristics of the methodologies that will be discussed further. As seen in the table, 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 area under the receiver operating characteristic curve (ROC-AUC). Due to the imbalanced nature of NTL detection, the ROC-AUC score provides more reliable results as it assesses the ranking quality of customers rather than their classification. Another metric suitable for imbalanced datasets, that takes into account also the precision of the model, is the area under the precision-recall curve (PR-AUC). However, this metric has not started to be used by the research community.

In [41], the authors propose a methodology for NTL detection using Extreme Learning Machines (ELM). The approach has been tested and developed with real data from Malaysia's Tenaga Nasional Berhad (TNB), the largest electricity utility in Malaysia. For each customer in part, a typical customer profile is created for weekdays, Saturdays, Sundays and public holidays. If outliers are detected on any new load curve, the load curve is

Table 2.1 Data-driven methodologies for NTL detection.

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	ROC-AUC
[41]	all	-	1500	half-hourly EC data	-	ELM, OS-ELM, SVM	-	-	-	-
[42]	abrupt changes	real on-field inspections	383	monthly EC & credit worthiness rating	13.83 %	SVM	-	-	77.41 %	-
[43]	fraud	-	440	EC & auxiliary databases	-	SVM, MLP	-	-	-	-
[44]	all	real on-field inspections	9131	EC & auxiliary databases	-	MLP (trained with BP, PSO and CSS)	-	-	-	-
[45]	fraud	real on-field inspections	21583	monthly EC & auxiliary databases	14.85 %	MLP	29.47 %	65.03 %	-	-
[46]	fraud	synthetic	5600	half-hourly EC	-	MLP	93.75 %	25.00 %	78.95 %	-
[47]	fraud	synthetic	5650	half-hourly EC	-	DT	-	-	-	-
[48]	all	real on-field inspections	-	monthly EC & auxiliary databases	-	NB, KNN, DT, MLP, SVM, RF, GBDT, AdaBoost	-	-	-	0.84
[19]	all	real on-field inspections	$\approx$ 100K	monthly EC	0 % - 100 %	Boolean, Fuzzy and SVM	-	-	-	0.56
[49]	contract diversion	synthetic	4245	hourly EC & weather data	10 % - 50 %	K-Means, LR, KNN, SVM	-	-	-	-
[50]	all	real on-field inspections	700K	monthly EC & auxiliary databases	1 % - 90 %	LR, KNN, SVM, RF	-	-	-	0.63
[51]	partial and total reduction in consumption	synthetic	12180	monthly EC	7%	CNN, LSTM, SAE, MLP, DT, RF	60 %	-	100 %	0.893
[52]	all	real on-field inspections	3.5M	monthly EC & auxiliary databases	10 % - 90 %	K-Means, RF	-	-	-	0.74
[53]	null EC	real on-field inspections	3510	monthly EC & auxiliary databases	4.67 %	Text mining, MLP, DT, SOM	-	-	14.75 %	-
[54]	fraud	real on-field inspections	42372	daily EC data	8.53 %	TSR, LR, SVM, RF, MLP, WD-CNN	-	-	-	0.78

further classified for NTL detection, with one of the following algorithms: ELM, Online Sequential ELM (OS-ELM) and Support Vector Machines (SVM). The results showed that the ELM was able to outperform the SVM model.

A methodology based on a SVM classifier has been proposed in [42], where the authors used the model for classifying customers as having/not having an NTL in their meter. The methodology has been tested in 3 towns from Malaysia, targeting NTL that occurs as an abrupt change in the customer's consumption pattern. The input for the SVM classifier consisted of 24 daily average EC values, aggregated from monthly measurements. Besides EC data, the classifier uses as an input the credit worthiness rating (CWR) corresponding to a specific customer. Though the input processing does not involve extensive feature

engineering, the system uses additional filtering using data from the Customer Information Billing System as well as High Risk data, in order to correlate this information with the output predictions of the SVM.

Another methodology for NTL detection, based on an SVM classifier, has been proposed in [43]. The SVM model uses as input the geographical location, season of the year, type of customer and the EC. The EC has been encoded to take values between 0 and 7, and reduced to groups of three consecutive measurements. A Multi-Layer Perceptrons (MLP) network has been used to estimate the hyperparameters of the SVM classifier in order to maximize the accuracy of its predictions. A MLP network is simply a feedforward neural network (NN). The cost function, the kernel type (linear, polynomial or radial basis function) and the γ parameters of the SVM were all selected using the neural network. The purpose of the neural network was to save time in selecting the best hyperparameters for the SVM.

In [44], the authors used a MLP network for NTL detection, using two labeled datasets of a Brazilian electricity utility. The customers were either industrial or commercial. The MLP network was trained in a supervised manner using the following data as an input: demand billed, demand contracted, maximum demand, reactive energy, power of the transformer, power factor, installed power and load factor. The innovation part of this work is that the authors trained the MLP network with evolutionary algorithms rather than backpropagation, a common way of training neural networks. They have shown that the MLP network trained with Charged Systems Search (CSS) and Particle Swarm Optimization (PSO) achieved lower error rates than the MLP network trained with backpropagation.

MLP models have also been used in [45], to detect fraud in electricity meters. The methodology has been implemented in a Brazilian electricity utility and has been shown to improve with more than 50 % the precision of previous approaches used in the utility. The MLP network has been used as the classifier for NTL detection. The input of the model was based on 13 input features that were providing information with regards to the recorded EC or of socio-economic attributes of the client. The model has been trained with backpropagation and it used as labels the results of previous on-field inspections that have been done in the past by the utility, to target customers with fraud in their meters.

Another methodology based on MLP has been presented in [46], where the goal was to detect energy fraud in SMs. The methodology is based on real SM data belonging to approximately 5000 residential customers and 600 businesses. These data have been collected by the Irish Social Science Data Archive Center. The MLP model's objective is to predict the energy consumption of a customer. Potential energy fraud was detected by checking if the Root Mean Square Error (RMSE) of the predicted energy consumption and the actual energy consumption was higher than 0.5 kWh. The energy fraud samples have been synthetically generated by introducing random noise in the EC profile. The results show that the MLP model is able to detect successfully energy fraud in 93.75% of the cases. A similar methodology has been presented in [47], where the authors used decision trees (DT) instead of MLP to detect energy fraud in smart meters. The threshold for the RMSE has been set to 0.4 kWh in this case.

In [48], the authors propose a methodology for NTL detection in both electricity and gas meters. The methodology has been developed and tested with real data of a Spanish

gas and electricity utility, Gas Natural Fenosa, and uses consumption data (meter readings and billing extractions), static profile data (e.g. tariff, address, age and model of the meter), historical fraud cases and external information such as the Koppen climate classification data. The following classifiers have been tested: MLP, SVM, Naive Bayes (NB), K-Nearest Neighbors (KNN), DT, Random Forests (RF), Gradient Boosting Decision Tree (GBDT) and AdaBoost. The results showed that the GBDT was able to outperform any other classifier or ensemble of classifiers.

The impact of the imbalance that naturally occurs in NTL datasets has been assessed in [19]. Naturally, the number of customers who have been identified with NTL is much higher compared to the one of customers with no NTL in their meter. The methodology has been developed using real data of a Brazilian electricity utility. The NTL dataset has been subsampled to contain different percentages of samples with NTL, from 0.1% to 90%. Three models were used in the comparison: boolean logic, fuzzy logic and a SVM classifier. The input features for the models consisted simply of the last 12 months of EC measurements.

A methodology to detect NTL that occurs due to violations to the energy contract has been proposed in [49]. The methodology has been tested using load profiles from two types of contract, collected from the advanced metering infrastructure of Korea Electric Power Cooperation (KEPCO). K-Means clustering has been used to group customers and create a prototype for each cluster in part. A normality and similarity score has been obtained for each customer in part, by computing the conditional probability of a prototype under certain conditions. The conditions used were: type of day, type of weather, type of temperature and type of humidity. The normality and similarity scores were afterwards used to train three machine learning models (LR, KNN and SVM) in a supervised manner. The results showed that the methodology proposed in the paper has been able to surpass the precision of traditional methodologies such as traditional best-fit prototype-based classification and average prototype-based classification.

Using geographical data can improve the performance of the model used in the classification of NTL, as shown in [50]. The methodology creates input features using neighborhood data, which allows to extract the inspection ratio and the NTL percentage in a certain area. Besides neighborhood information, the methodology also uses the daily average consumption of the last 12 months as well as categorical master data (type of customer, status of the contract, number of wires and voltage) as input features. The following models were used in the comparison: KNN, LR, SVM and RF. All models have been shown to obtain a better performance when adding the neighborhood and categorical master data to the features based solely on EC measurements. The results show that RF slightly outperformed the rest of the classifiers.

A comparison between a convolutional neural network (CNN), a long short-term memory (LSTM) network and a stacked autoencoder (SAE) for the task of NTL detection, has been done in [51]. The performance of these algorithms has also been compared with the ones of traditional ML models: MLP, DT and RF. The models have been trained on a synthetic dataset with two types of NTL: partial and total load reduction. The input to these models was the monthly EC of one year. The results showed that the CNN outperformed the rest of the classifiers.

In [52], the authors compared the performance of RF, logistic regression (LR) and SVM on the task of NTL detection in electricity meters. This work also assessed the impact of the input features' type on the performance of the models. The input features created from the EC data either were based on statistical analyses that took into account the temporal nature of these measurements or based on a similarity comparison using K-Means clustering among customers. Additional input features were created using geographical and transformers data. The results showed that the RF slightly outperformed LR and SVM classifiers, and that the model can achieve stable performance using only input features coming from raw EC data.

The authors in [53] describe a two-stage methodology for NTL detection in meters that have null consumption readings or a drop in consumption. A null consumption measurement could be an indicator of an NTL loss in the meter, however, the authors argue that the meter can record null measurements simply if a house is empty or there is a drop in demand in a certain business sector. The authors use additional information, as simply relying on EC measurements is not sufficient to detect such cases. The first stage of the methodology is filtering customers based on text mining and MLP. The text mining techniques have been used to create a dictionary of concepts from inspectors' commentaries from previous on-field inspections or technical interventions. The MLP network has been used to classify the concepts into 5 categories: closed, correct, incorrect, low consumption and unuseful. The second stage of the methodology generates rules for NTL detection using DT and Self-Organising Maps (SOM). The results showed that the methodology was able to increase 3 times the precision of these type of inspections.

A methodology based on a wide and deep CNN (WD-CNN) has been used in [54] to detect fraud in electricity meters. The wide model of the network is a simple MLP network that uses as an input the 1D electricity consumption measurements. The CNN model uses as an input the 2D energy consumption data, obtained by stacking the weekly consumption profiles. The methodology has been developed and tested using a dataset provided by the State Grid Corporation of China (SGCC), which contains the energy consumption measurements within approximately three years and of more than 40000 customers. The Three Sigma Rule (TSR), which is a simple anomaly detection technique, has been used as a benchmark. The performance of the WD-CNN model has also been compared with more sophisticated ML models such as LR, SVM, RF and MLP. The results showed that the WD-CNN model outperformed the rest of the models.

As shown above, data-oriented methodologies for NTL detection have been studied extensively by researchers, as they rely only on the data that it is already available in the electricity utilities. However, it is extremely difficult to have an honest comparison between the performance of these methodologies. This is due to the fact that there are major differences in these approaches: different datasets availability, they use either real or synthetic NTL datasets or they monitor for different metrics.

2.4 Challenges in data-oriented methodologies for non-technical losses detection

As noted in [55], there are several challenges that impede the progress in this research area as well as the performance of NTL algorithms in the real environment. From my perspective, these are the following challenges that can be encountered by researchers who are trying to push forward and make advances in this field:

- Lack of benchmark datasets - Although the authors in [56] provided a benchmark dataset for NTL detection, this dataset consists of only EC data. Its main disadvantage is that it creates synthetical NTL samples that consist either in partial or total load reduction, whilst the non-NTL samples have normal consumption patterns. However, in the real environment, customers can have partial or total reduction due to causes that are not related to NTL such as going on a holiday. This is why I believe it is vital to use auxiliary data for NTL detection, as we cannot rely simply on the EC data. A benchmark dataset provided by an electricity utility would be the best scenario, as it will provide realistic and more complex NTL cases as well as additional data besides the EC measurements. However, with increased data privacy regulations, it seems to be increasingly difficult for the electricity utilities to share their data with the research community.
- Different metrics of performance - It is difficult to compare the performance of different methodologies when there isn't a common metric that is reported by all researchers. Moreover, metrics that are inappropriate for imbalanced datasets such as the accuracy, are heavily used, making it difficult to really assess the true performance of one's methodology. Recent research works have started to acknowledge the importance of choosing the right metric for NTL datasets. However, there are plenty of methodologies proposed in the past that were not properly assessed making it difficult to have an overview of the real progress across time in this area of research.

Besides the research challenges, practitioners that implement NTL detection models in the utilities struggle with significant challenges as well. Here are a few challenges that can be encountered when trying to deploy these models in the real environment:

- Noisy data and labels - Often, the data collected by electricity meters, either come with missing values or anomalous ones. It is difficult for the practitioner to know whether these issues come from NTL/non-NTL reasons. The choice of whether to impute these values with normal estimates or to keep them as they are can be vital to the performance of the model. Another important challenge is the noise in the labels (results of previous on-field inspections). As the customer samples are labeled manually by on-field inspections they are prone to human error. Introducing misclassified samples makes it more difficult and decreases the performance of NTL models.
- Dataset imbalance - NTL datasets available in electricity utilities can be extremely imbalanced, especially in developed countries. Naturally, in these countries, the number of electric supplies with any kind of detected anomaly is a tiny portion of the

global amount. For supervised approaches, this can represent a significant challenge as the model will be biased to predict the majority class (non-NTL). This challenge can be mitigated by creating a ranked list of customers based on their probabilities of having an NTL in the meter, rather than directly classifying a customer based on a probability threshold.

3 Summary of results

This chapter will provide an overview of the methodologies proposed in [20] and [21], along with their results and their contributions. Both methodologies have been developed and tested with real data from the largest electricity utility in Spain, Endesa. They rely on customer level data, coming from SMs and auxiliary databases, and on the results of previous on-field inspections. The NTL was treated as a black-box, thus both models aim to detect all types of NTL in the meters, regardless of their source.

3.1 Feature engineering and supervised learning for non-technical losses detection

The first methodology was focused on using handcrafted feature engineering based on SM data and auxiliary databases. After the features were extracted, they were used as an input into several machine learning (ML) models. These models were trained in a supervised manner, using the results of previous on-field inspections. The methodology was developed using data of industrial and large commercial customers, with a contracted power higher than 50 kW.

Compared to the methodologies presented in Chapter 2, which are based on handcrafted feature engineering and supervised learning, the proposed methodology differentiates itself by:

- Using all the information the SMs record: EC, alarms and electrical magnitudes. These additional data are vital for NTL detection as studying only the consumption behavior of the customer is not sufficient to detect a wide range of NTL (considering that only customer level data is available).
- 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.

The best ML model has been used to create a ranked list of customers according to their probability of having an NTL in their meter. The methodology has been implemented

in a real NTL campaign, obtaining a precision of ≈ 21 % for new on-field inspections generated by the model.

3.1.1 Smart meter data availability

The SM data used for training, validating and testing the ML models was provided by Endesa. These SMs 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 3.1 shows the measurements that were included in the SM data.

Table 3.1 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 measurements/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)

The dataset is comprised of SM measurements from the last ten years, from 1st May 2007 until 30 December 2016. It contains customers who throughout the 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 into multiple samples. The methodology for creating multiple samples can be be seen in Figure 3.1.

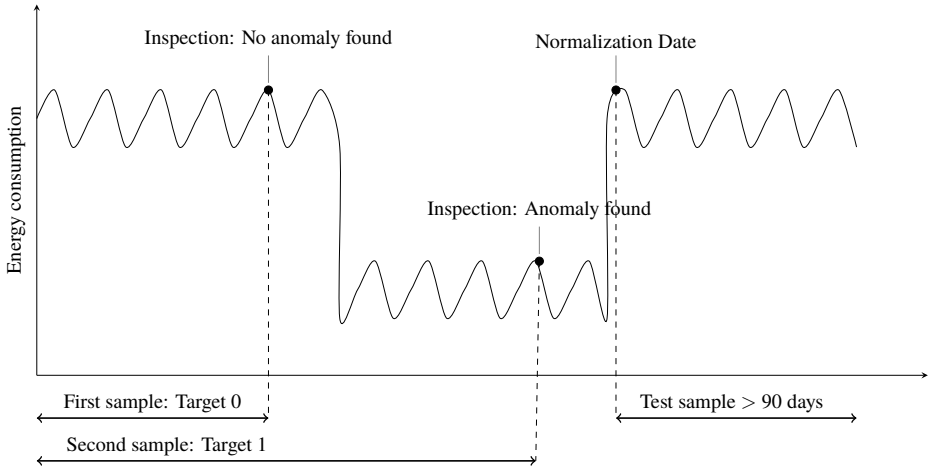


Figure 3.1 Scenario of a customer with multiple training samples.

By using the results of previous on-field inspections, two datasets have been created:

- Training dataset - this dataset has been created by selecting the customers who had at least one on-field inspection.
- Ranked list dataset - this dataset has no labels and has been created by using customers who never had an inspection or whose last normalization date was more than ninety days ago (See Figure 3.1).

The training dataset has been used to train in a supervised manner, using the labels of previous on-field inspections, several ML algorithms to assess their performance and select the best algorithm. 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 ranked list dataset has been used to generate new on-field inspections, by using the best ML algorithm trained with previous inspections data. This dataset gives the possibility to assess the capability of the ML model to generalize beyond its training data, on unseen customer samples.

Table 3.2 shows the number of customers used during training and in the ranking list. As mentioned in Chapter 2, NTL datasets are naturally imbalanced. Figure 3.2 shows the number of samples of customers with and without an NTL detected in the entire training dataset. This is an extremely imbalanced dataset as the number of customer samples with an NTL detected represents ≈ 5 % of the entire training dataset.

3.1.2 Methodology overview

The main aim of the proposed methodology was to provide an electricity utility with a ranked list of customers, according to their probability of having an NTL in their electricity meter. The methodology used mainly SM data for feature extraction (Figure 3.3). The

Table 3.2 Size of the training dataset and the ranking list.

First day analysis	01/05/2007
Last day analysis	31/12/2016
Unique customers in the training dataset	41571
Customer samples in the training dataset	57304
Customer samples in the ranking list	72489

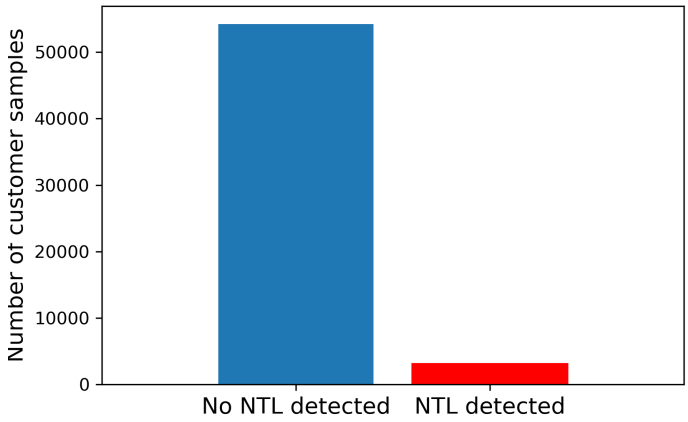


Figure 3.2 Target distribution.

features were based on SM alarms, EC and electrical magnitude measurements. It also used features extracted from auxiliary databases which mainly provide geographical and technological characteristics of the customer. The features extracted from auxiliary databases have been provided by the utility.

A training dataset has been created by concatenating all the features obtained from SM data and auxiliary databases. Several preprocessing techniques such as normalization and missing data imputation have been used, in order to prepare the data to be used as an input to the ML models. Feature normalization is a common ML preprocessing procedure, where each feature vector is normalized to be within the same range as the rest of the features. As most ML algorithms cannot work with missing values in the input, the missing data have been imputed using the statistics on the available data.

After preprocessing the dataset, the original training dataset has been split into a training, validation and test dataset. These datasets were further used as an input into several ML algorithms for model selection and evaluation. After the best ML model has been selected, its performance has been further improved by reducing the data imbalance. If the performance of the best model met the desired standard required by the utility, its parameters were saved and used to make predictions on new customer samples obtaining

a ranked list of customers as the final output.

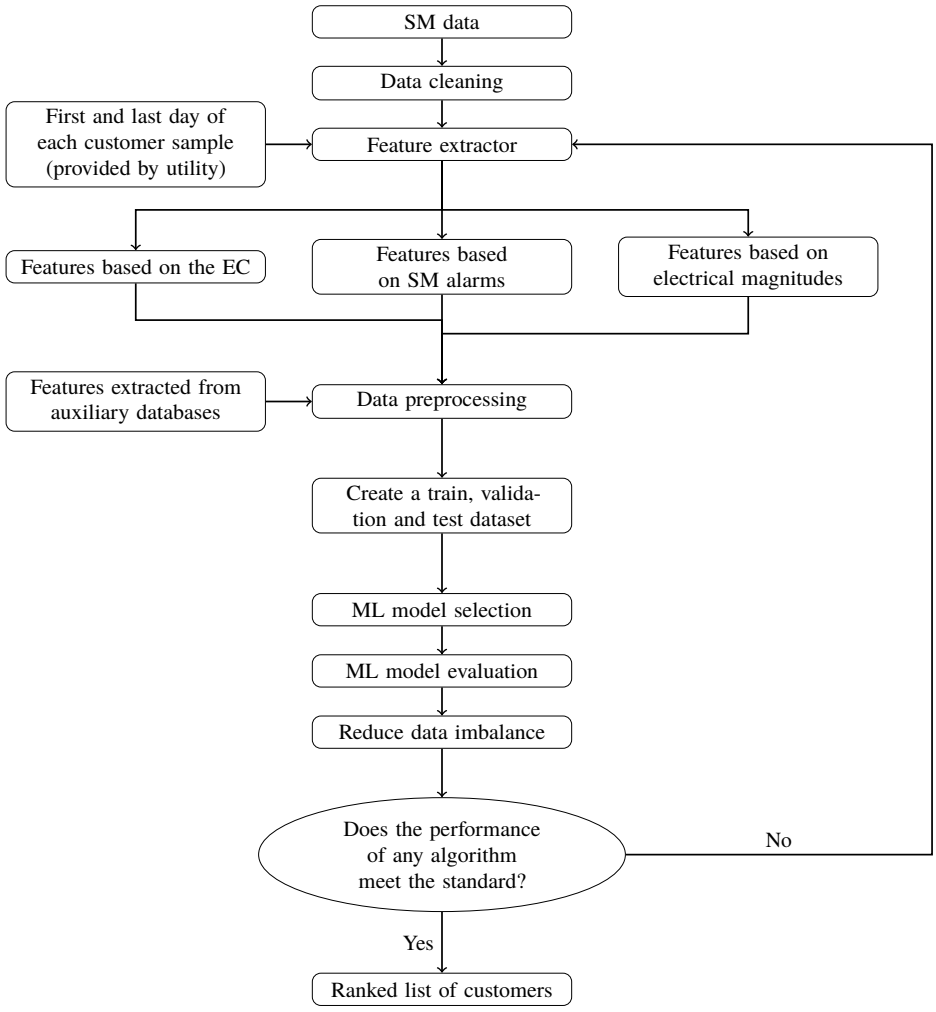


Figure 3.3 Methodology outline for NTL detection.

3.1.3 Feature engineering

Feature engineering is an important processing step in ML. The performance of a ML model is dependent on how relevant are the features extracted for the NTL detection task. These features are regularly used as an input to a ML algorithm. The following section will focus on the methodologies used for feature extraction from SM data. The extracted features are based either on EC measurements, smart meter alarms or electrical magnitudes. The smart meter alarms and EC measurements are considered to be recorded synchronously

whilst the electrical magnitudes are considered to be recorded asynchronously through power snapshots.

Features based on EC measurements aim to detect a drop in consumption or unusual consumption behaviors. They are divided into two types:

- Features that detect recent anomalies.
- Features that detect old anomalies or anomalies that start from the beginning of the contract.

3.1.3.1 Features aimed to detect recent anomalies

A sudden decrease in the EC can be noticed for most of the fraud and non-fraud related anomalies. To detect a recent drop in consumption, the Z_{score} can be used. This score indicates how many standard deviations away from the mean is a new measurement.

$$Z_{score} = \frac{X_i - \bar{X}_i}{\sigma_{X_i}} \tag{3.1}$$

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.

Given the data granularity of SMs, the EC consumption history can be separated by hours or type of day. Thus, a different set of Z_{scores} can be obtained for the first hour of the day, for weekdays, weekends etc. Figure 3.4 shows the procedure of feature extraction for these scores. The objective is to compute the average Z_{score} of the last n days. The size of this window is variable. As an example, a different set of features can be computed for a window size of 30, 60, 180 days. The mean and the standard deviation of the EC measurements are thus computed using the entire EC history except for the last n days. The entire set of features can be inserted as an input to a ML algorithm or as input to a feature selection algorithm in order to select the optimal window size given the dataset available for training. Though the Z_{score} is able to capture recent drops in consumption, it cannot capture drops in consumption that took place at the beginning of the contract as the mean and standard deviation of the EC will already be shifted towards the anomalous measurements.

In the case of a SM anomaly, in most cases, the SMs either stop sending measurements or record 0 kWh energy consumption. A set of features can be created to capture this behavior using a variable sized window. Figure 3.5 shows the methodology for feature extraction. The day of the last measurement can be easily retrieved as the EC measurements of SMs are timestamped.

Table 3.3 shows the features that have been computed to detect recent anomalies. Several types of features have been created, by:

- type of day (t): weekday, Saturday, Sunday.
- number of days (n): 15, 30, 45, 60, 90.
- time windows (w): as described in Table 3.1.

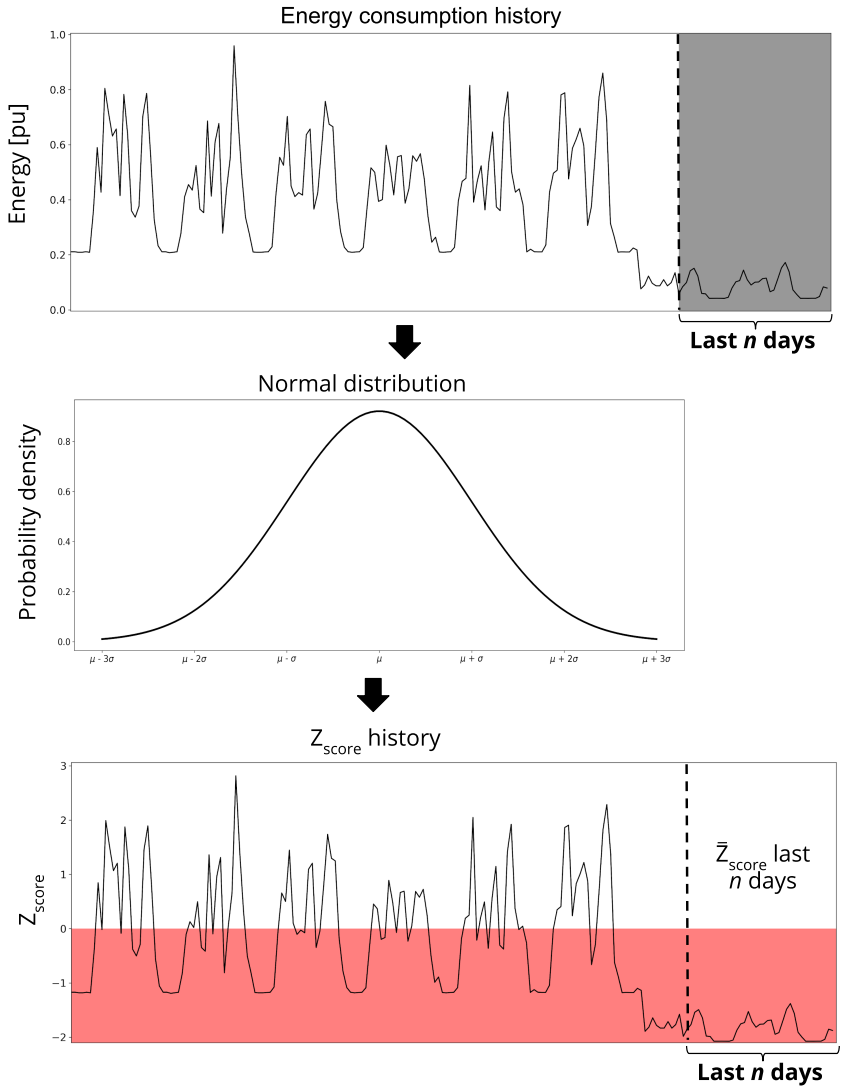


Figure 3.4 Z_{score} feature extraction.

3.1.3.2 Features aimed to detect old anomalies

To detect anomalies that have started before the period of analysis, or from the beginning of the contract, clustering techniques must be employed as for these cases a sudden drop in consumption cannot be observed.

Customer segments can be created by grouping customers by diverse characteristics

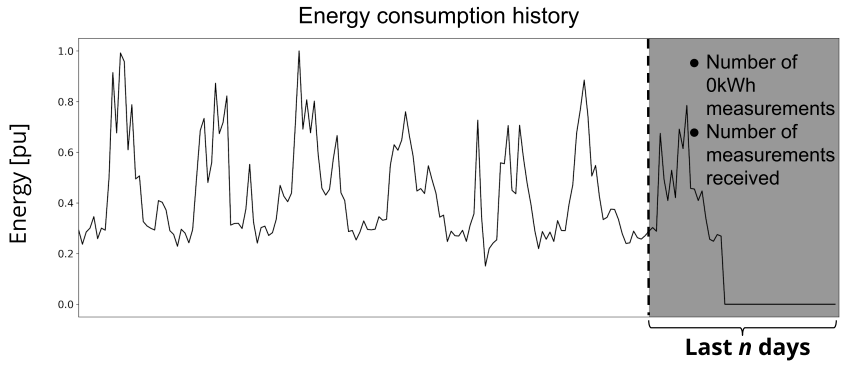


Figure 3.5 Feature extraction for anomaly detection in SMs.

Table 3.3 Features aimed to detect recent anomalies.

Type of data	Input Features
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
	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

of their contracts such as contracted power, tariff type or type of business in case of commercial customers. The K-Means clustering algorithm proposed by Lloyd in [57] can be used to create customer segments on continuous data such as the contracted power. For discrete data such as tariff type or type of business, the grouping of customers is done automatically.

Within these customer segments, there are several methods that can be employed to detect abnormal consumption patterns. In this thesis, we will explore both distance based as well as density based features.

a. Features based on distance metrics

If data of previous on-field inspections are available, base consumption profiles can be created using the consumption data of customers who weren't identified with an NTL in their meters. For each customer segment, a base profile can be created for each month of the analysis. Using the timestamp of EC measurements, the base profiles can be further improved by dividing the EC measurements by type of day (e.g. Monday, Tuesday, weekday,

weekends):

$$B^k_{i,j,t} = \{\frac{1}{N}\sum_{z\in M} P^z_{w_t} : 1 < w \leq L\}, \tag{3.2}$$

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. 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_{w_t} represents the average power consumption for type of day t during the time window w selected within a day. L represents the number of time windows selected within a day.

The maximum size of the base profile L is 24 if the hourly measurements are not aggregated further (e.g. morning, noon, evening).

After creating the base profiles for each customer segment part, a customer profile can be created for each customer in part, by using the last month of its EC history. These profiles are created for each customer, regardless if it had been previously inspected or the results of its inspection.

$$C_t = \{P^z_{w_t} : 1 < w \leq L\} \tag{3.3}$$

where C_t represents the consumption profile of the last month and P_{w_t} is the average power consumption for type of day t during the time window w selected within a day. L represents the number of time windows selected within a day.

Using the consumption profile of each customer, and the base profile assigned to its customer segment, two distance metrics can be computed: the Manhattan distance and the euclidean distance. The Manhattan distance can be computed for each time window in part and for the entire day, using the following mathematical formulations:

$$MHT_{w_t} = |P_{w_t} - \frac{1}{N}\sum_{z\in M} P^z_{w_t}|, \tag{3.4}$$

$$MHT_{T_t} = \sum_{w=1}^{L} |P_{w_t} - \frac{1}{N}\sum_{z\in M} P^z_{w_t}|, \tag{3.5}$$

where MHT_{w_t} is the Manhattan distance of a consumption profile for time window w and for type of day t, and MHT_{T_t} is the total Manhattan distance of all time windows.

The Euclidean distance can be computed on an entire day, by aggregating the distances within each time frame. It is defined as follows:

$$ECL_{T_t} = \sqrt{\sum_{w=1}^{L} (P_{w_t} - \frac{1}{N}\sum_{z\in M} P^z_{w_t})^2}, \tag{3.6}$$

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

Table 3.4 shows the features that have been computed using distance metrics.

b. Features based on density metrics

The second approach to detect an unusual customer behavior consisted on using the Local Outlier Factor (LOF) [58]. This metric assigns to each customer profile a degree of being

Table 3.4 Features aimed to detect old anomalies (distance metrics).

Type of data	Input Features
EC	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*

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. Table 3.5 describes the features that have been computed, using this metric.

Table 3.5 Features aimed to detect old anomalies (density metrics).

Type of data	Input Features
EC	LOF score of daily EC measurements for type of day *t*
	LOF score of EC daily profile for type of day *t*

3.1.3.3 Features based on smart meter alarms

Features developed using the quality byte (QB) measurement are aimed to detect meter faults or physical tampering. The QB measurement uses an 8-bit code to assess the quality of the measurement, as the IEC 870-5-102 protocol defines [59]. Table 3.6 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.

Figure 3.6 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.

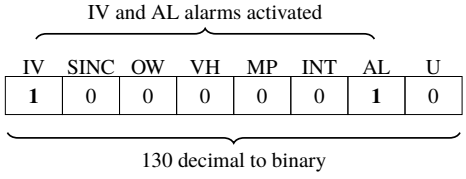


Figure 3.6 Example of a QB measurement..

Table 3.6 Alarms registered by the QB measurement [59].

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

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 3.7 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).

Table 3.7 Features based on SM alarms.

Type of data	Input Features
QB	Number of days with alarm x in the last n days
	Number of days from last x alarm

3.1.3.4 Features based on 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 3.8 shows the features developed using EM.

3.1.4 Supervised machine learning models

3.1.4.1 Evaluation and metrics

To evaluate the performance of the ML models, the original training dataset was split into a reduced training dataset, a validation dataset and a testing dataset. The validation dataset

Table 3.8 Features based on electrical magnitudes (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	Neutral current ratio $\frac{I_N}{I_{max}}$
		Neutral current angle

was used to tune the hyperparameters of the ML models whilst the testing dataset was used to assess how well the models generalize to new, unseen customer samples. This is a general practice in ML.

In Figure 3.7, the approach for model selection and evaluation is presented. Given the scarcity of NTL samples, a nested cross-validation (NCV) has been chosen to make use of the available data as much as possible. A 3-fold NCV has been chosen to use as example in Figure 3.7 for simplicity reasons. In practice, a 5-fold nested cross validation has been used. The test fold was used only in the model evaluation stage.

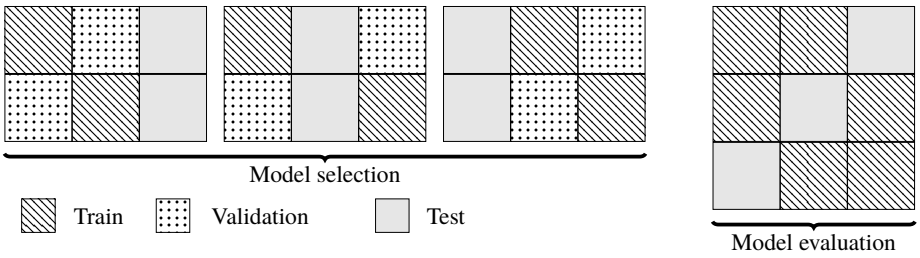


Figure 3.7 3-Fold Nested Cross-Validation example.

One of the most common metrics used for assessing the performance of a ML algorithm is accuracy. However, the accuracy of an algorithm on a severely imbalanced dataset cannot provide a real assessment of its predictive power. Just by using a naive predictor which predicts that none of the customers has an NTL 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 area under the receiver operating characteristic curve (ROC-AUC) [60], [61]. 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 ROC curve. The score ranges between 0 and 1, a score above 0.5 being obtained with better than random predictions. However, the ROC-AUC score

does not take into account the precision of the model, a metric that is extremely important for the NTL detection task. A metric suitable for imbalanced datasets that also takes into account the precision of the model is the area under the precision-recall curve (PR-AUC) [62, 63]. The PR-AUC score ranges as well between 0 and 1. The PR-AUC of a random classifier is ≈ 0.05, corresponding to the percentage of NTL samples in the dataset.

Both the ROC-AUC score and the PR-AUC score have been used to evaluate the performance of the ML models.

3.1.4.2 K-Nearest Neighbors

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 previous 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 3.9 shows the hyperparameters used during grid-search along with their optimal value found using the validation dataset.

Table 3.9 KNN Grid-Search.

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

Figures 3.8 and 3.9 show the performance of the KNN model on the test dataset. The KNN model obtains a ROC-AUC significantly higher compared to the ones obtained using random predictions. For the PR-AUC score, the KNN model performs ≈ 6 times better than random predictions.

3.1.4.3 Logistic Regression

The Logistic Regression (LR) algorithm has also been used in the comparison. 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 $z = \theta^T X$ [64]. The classifier has been trained on a logarithmic loss function using the LIBLINEAR solver [65]. Table 3.10 shows the optimal values found for the hyperparameters used during grid-search.

Table 3.10 LR Grid-Search.

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

Figures 3.10 and 3.11 show the performance of the LR model on the test dataset. The LR model obtains ROC-AUC score of 0.842, which is significantly higher than the one obtained by the KNN model as well as the one obtained by making random predictions.

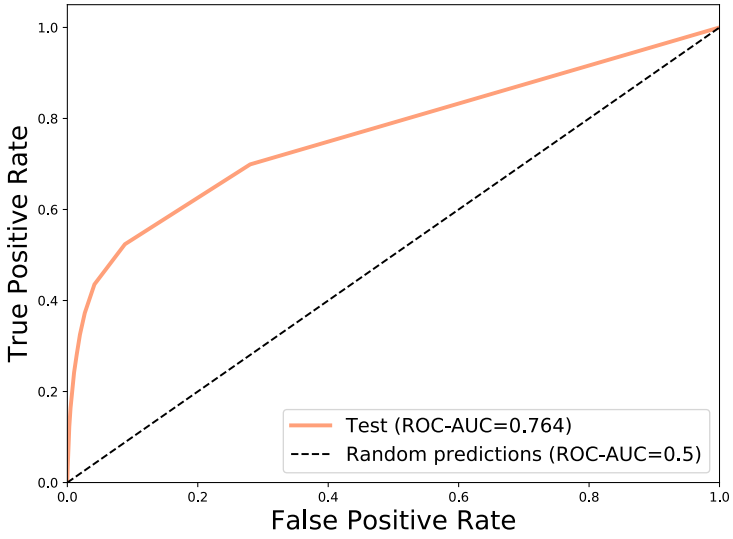


Figure 3.8 ROC curve for the KNN model.

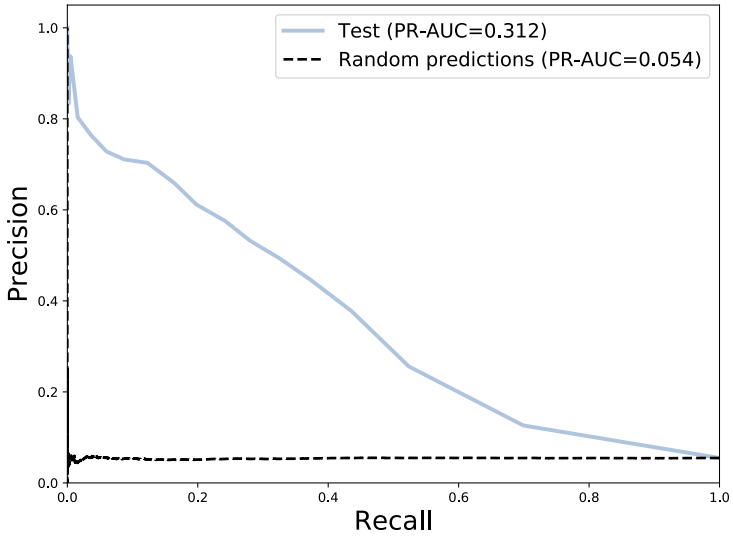


Figure 3.9 PR curve for the KNN model.

The model also obtains a PR-AUC score of 0.382, an improvement over random predictions of more than 7 times.

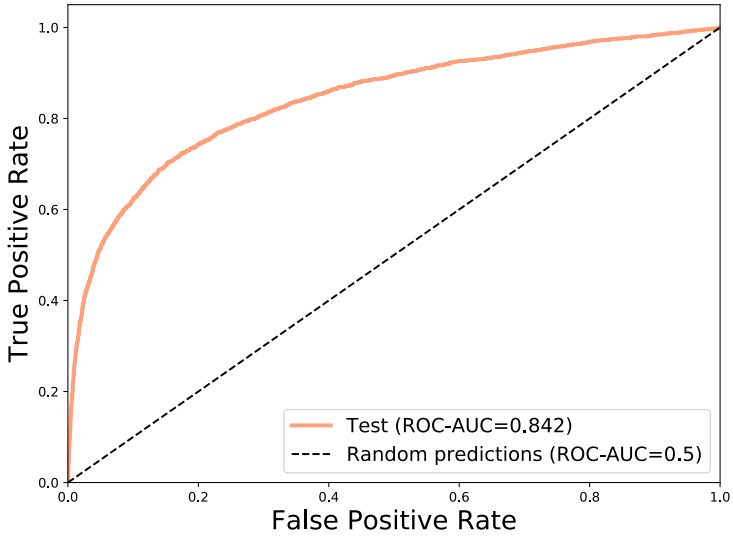


Figure 3.10 ROC curve for the LR model.

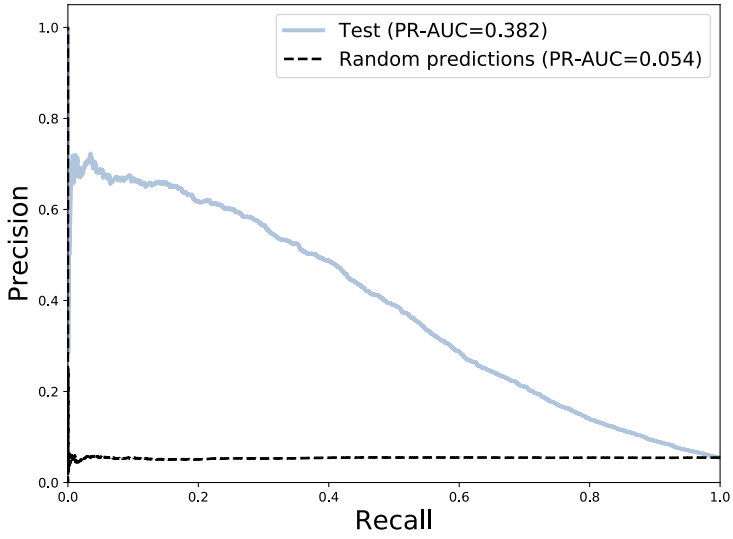


Figure 3.11 PR curve for the LR model.

3.1.4.4 Support Vector Machines

Support Vector Machines (SVM) are a very popular classifier for anomaly detection in the utilities. They not predict probability estimates but rather decision values. An 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 [66]. This margin will be determined by the support vectors of the classes. The support vectors are customer samples from the training dataset that are the closest to the decision function.

Table 3.11 shows the optimal hyperparameters found during grid-search. The kernel parameter is helpful if the customer classes are not linearly separable by a hyperplane in the high dimensional space.

Table 3.11 SVM Grid-Search.

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

As can be seen in Figure 3.12, the SVM obtains a slightly higher ROC-AUC score than the LR. However, when taking into consideration the PR-AUC score (see Figure 3.13), its performance is worse than the one of the LR model.

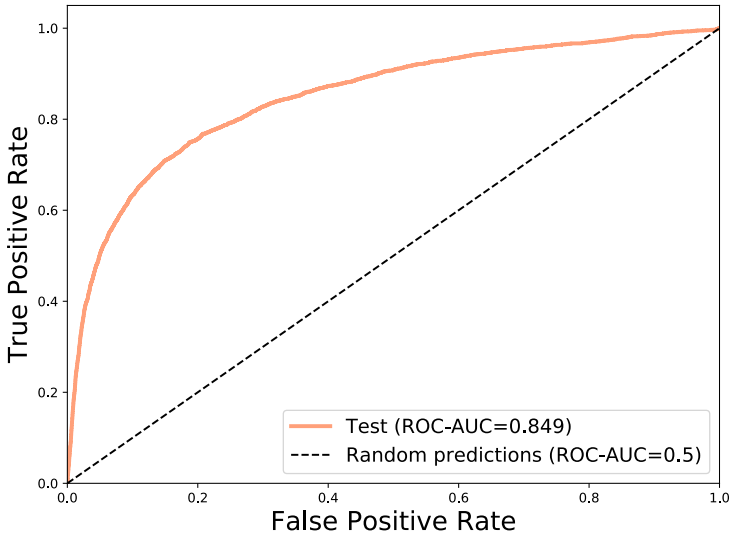


Figure 3.12 ROC curve for the SVM model.

3.1.4.5 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 [67]. The algorithm uses gradient boosting [68] with a regularized cost-function. Gradient boosting builds an additive model by combining the predictions of many "weak" classifiers. The classifier in this case is a regression tree.

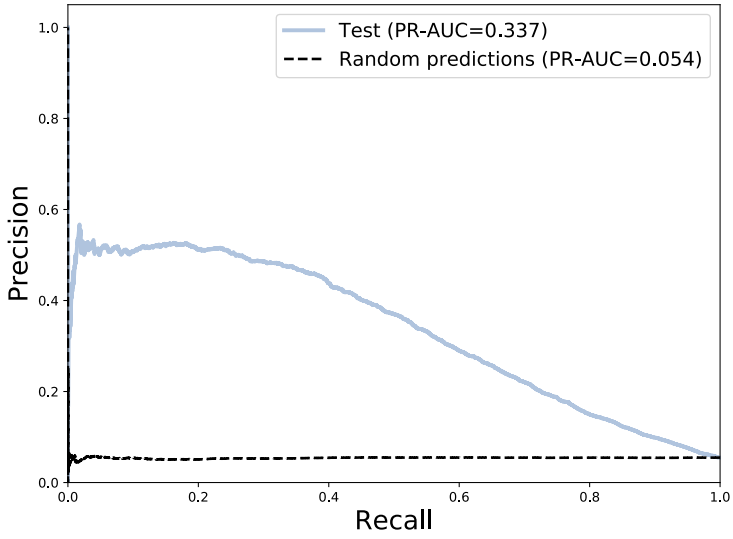


Figure 3.13 PR curve for the SVM model.

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 3.12 shows the hyperparameters used during grid-search for XGBoost.

Table 3.12 XGBoost Grid-Search.

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

As can be seen in Figures 3.14 and 3.15, XGBoost significantly outperforms all the previous ML models. It obtains a ROC-AUC score of 0.864 and a PR-AUC score of 0.508. The XGBoost model has a significant advantage over the rest of the models, especially when the precision of the model is also taken into consideration.

3.1.4.6 Comparison

The experiments done above show that the XGBoost model outperforms the rest of the classifiers whilst the KNN model obtains the lowest performance. When the PR-AUC of

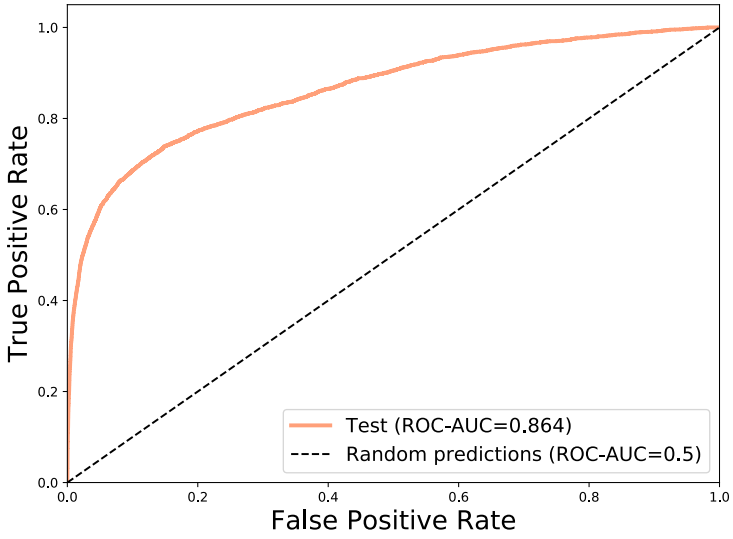


Figure 3.14 ROC curve for the XGBoost model.

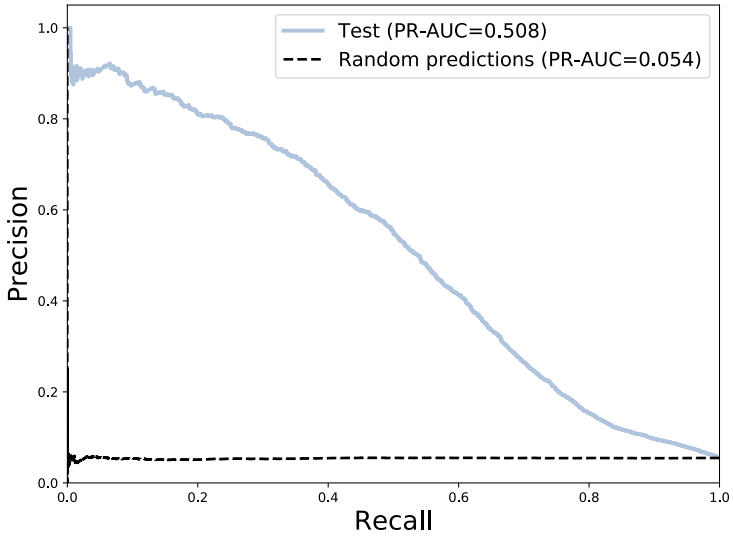


Figure 3.15 PR curve for the XGBoost model.

the models is taken into consideration, the performance of XGBoost is significantly better in comparison with the rest of classifiers.

Figure 3.16 shows the execution time for all the ML models used in the comparison, during both training and testing. As it can be seen, LR was extremely fast to train and to

test. The KNN model does not have any training step as it simply uses the entire training dataset to make a prediction, at test time. This can be a major issue as the dataset used to create the ranking list is often much larger than the training dataset. Finding the nearest neighbors requires going through the entire training dataset, for each customer in the ranking list in part. This can be a very computationally expensive process. The XGBoost model had the longest training time, but its test time was as fast as the one of the LR model.

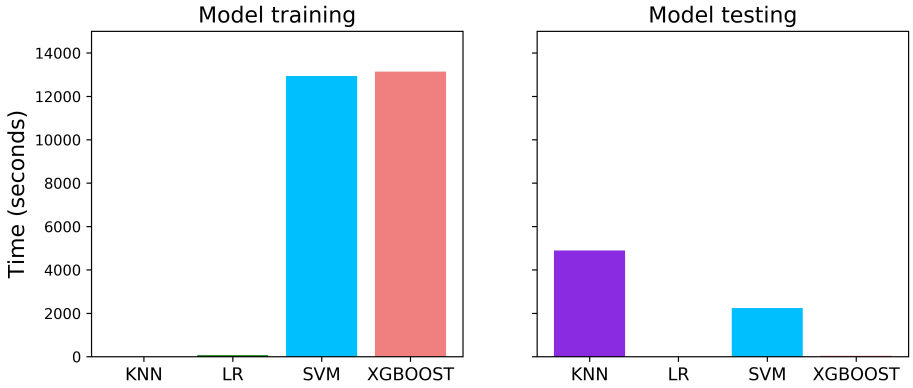


Figure 3.16 Execution time.

3.1.5 Reducing the data imbalance

As seen previously, only ≈ 5 % of the samples in the training dataset belong to SMs who had been identified with an NTL. This is an extremely imbalanced dataset. To mitigate this challenge, the dataset imbalance has been reduced using undersampling techniques. Undersampling simply removes samples that belong to the majority class. In this case, the majority class is represented by samples found with no NTL in the meter. To select which samples should be removed, two methods have been used:

- Undersampling by removing the samples of customers who 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.

- Undersampling by removing the samples of honest customers using a different number for the random seed.

The undersampling techniques have reduced the training dataset by ≈ 35 %. Their effects on the performance have been assessed by using the best ML model found in the comparison: XGBoost. Figures 3.17 and 3.18 show the results obtained after using both undersampling techniques.

As it can be seen, undersampling techniques significantly improve the performance of the XGBoost model. The improvement can be seen on both metrics (ROC-AUC and PR-AUC). The first undersampling method is slightly outperforming the second method

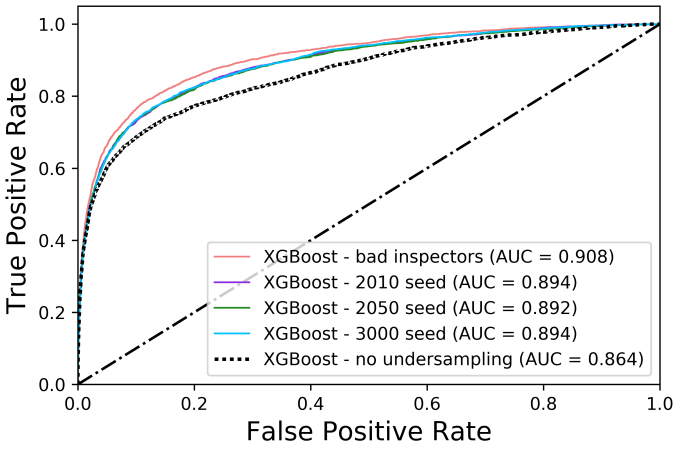


Figure 3.17 ROC curves for undersampling vs. no undersampling.

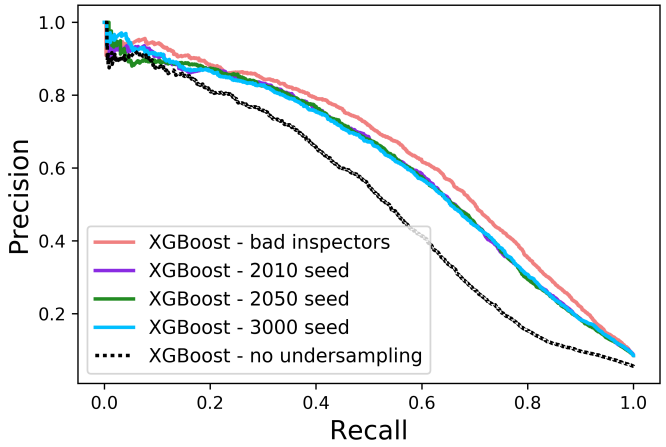


Figure 3.18 PR curves for undersampling vs. no undersampling.

which has been used with different random seeds. However, the boost in performance is not conclusive.

3.1.6 Performance analysis on type of data

The ML models used as input, features coming from two data sources: SM data and auxiliary databases. The features extracted from the SM data have been described in the previous section. The features extracted from the auxiliary databases have been provided

by the utility. Their description can be found in [20]. These features come from four types of databases:

- TS (Tariff Summary) - contains monthly EC and the maximum power in up to six different tariff periods.
- GIS (Geographic Information System) - provides information not only on the location of the customer but also on the rate of NTL in the neighborhood.
- TECH (Technological characteristics) - contains information such as the brand and model of the SM as well as whether the SM is located inside/outside.
- CONTRACTS - includes information related to contract events as well as the activity type of the customer.

Table 3.13 shows the ROC-AUC score obtained by using different subsets of features.

Table 3.13 Performance analysis on type of data.

Data source	Type of data	ROC-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**

Just by using the SM data, a ROC-AUC score of 0.88 is obtained. When it comes to the SM data, the highest boost in performance has been obtained by including the features extracted from SM alarms. The features extracted from electrical magnitudes add a boost in performance of 0.03 to the ROC-AUC score. For the features extracted from auxiliary databases, the highest impact is given by GIS features. This is not very surprising, especially as the authors in [50] already found that neighborhood features increase the predictive power of a ML model trained for NTL detection.

Figure 3.19 shows the PR curves for all the subset features described in Table 3.13. The SM data features obtain much higher precision for the same recall obtained by the auxiliary data features.

3.2 Raw smart meter data and hybrid deep neural networks for detection of non-technical losses

This section will describe the proposed methodology and the results obtained in [21]. The methodology has been developed and tested with real SM data of a spanish electricity utility. The data came from residential and small commercial customers with a contracted power smaller than 15 kW.

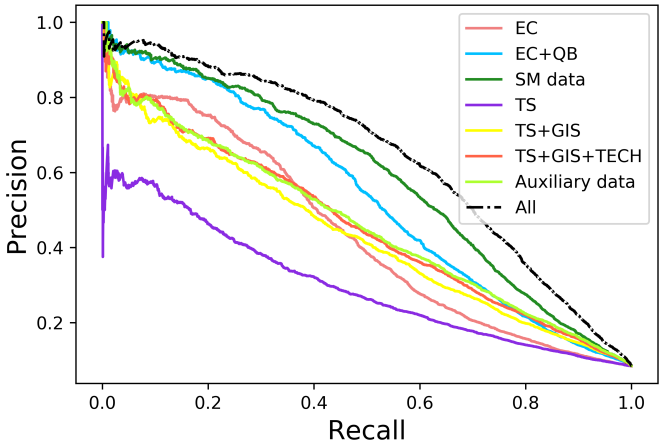


Figure 3.19 PR curves for different subsets of features.

The main aim of this methodology was to be able to self-learn the features relevant for detecting NTL in SMs, removing the need of handcrafted feature engineering. The proposed architecture consisted of a long short-term memory (LSTM) network and a multi-layer perceptrons (MLP) network, which used as an input simple raw data that come either from SMs or auxiliary databases. The first network analyses the raw daily EC history whilst the second one integrates non-sequential data such as the contracted power or geographical information.

This work differentiated itself from similar works presented in Chapter 2, by:

- Proposing a state-of-the-art methodology that can self-learn features that are relevant for NTL detection. This methodology was able to integrate both sequential and non-sequential data. To my knowledge, this was the first deep learning architecture for NTL detection that is able to accommodate both types of data.
- Investigating the boost in performance obtained by combining both types of data and showing that the hybrid network significantly outperforms a network that uses only EC data as an input.
- Showing that the proposed architecture vastly outperforms previous NTL detection models that were based on deep learning and raw data, as well as traditional ML classifiers.

The methodology was used as an NTL detection tool in the same electricity utility, achieving a precision of ≈ 47 % for new on-field inspections generated by the model.

3.2.1 Why use raw data?

The figures below show the EC history of specific SMs found with NTL in their meters. Figure 3.20 shows the EC profile of a SM that was directly tampered with a shunt device between the input and output terminals of the SM to divert the current.

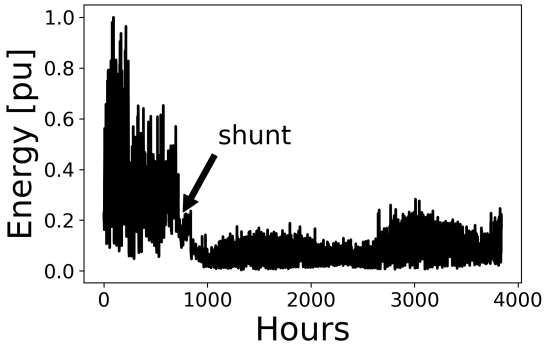


Figure 3.20 Shunt case (Source: Endesa).

An example of a double tapping fraud is shown in Figure 3.21. Double tapping is a typical fraud case, where part of the consumption is connected directly to the grid, bypassing the SM. In this particular case, the fraud occured from the beginning of the contract, so a descent in the consumption cannot be observed.

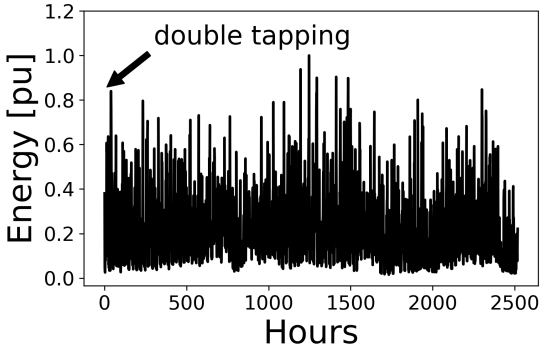


Figure 3.21 Double tapping case (Source: Endesa).

Figure 3.22 shows the case of a SM found with an electronic fault, a common cause of anomaly in SMs.

As it can be seen, though in all NTL cases the meter reports lower EC readings, the change in the consumption profile is manifested differently depending on the NTL source. Traditional approaches aim to capture the behavior of different types of NTL using hand-crafted feature engineering, as there is no mathematical formulation for the EC pattern of a SM with a shunt or double tapping. Their aim is to create a set of features that characterize each type of anomaly or fraud encountered through on-field inspections. As an example, features that detect a sudden drop in consumption are aimed to detect cases of shunts. In the case of electronic faults, features that monitor the number of 0 kWh measurements or number of missing measurements are employed. Unfortunately, these approaches rely

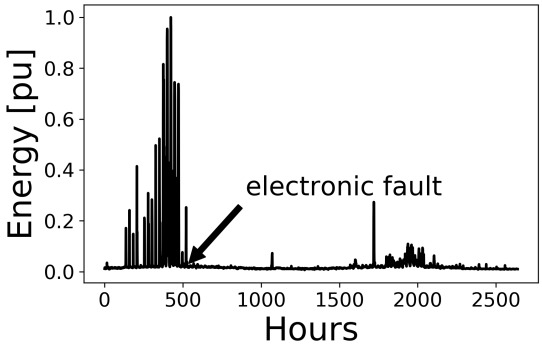


Figure 3.22 Electronic fault case (Source: Endesa).

heavily on expert knowledge which is very expensive and time-consuming. Moreover, they require experts to develop continuously new features in order to adapt to new types of NTL.

The deep learning architecture described in the next sections of this chapter has been created in order to mitigate the constraints of previous approaches, as it is able to self-learn the features relevant for NTL detection from raw EC measurements and can adapt automatically to new NTL behavior in the SM data.

3.2.2 Methodology

Figure 3.23 shows the methodology used to develop and test the deep learning model. As it can be seen, it needs as an input three types of data: EC history recorded by the SMs, auxiliary (geographical, contractual, technical and economic) data and the results of previous on-field inspections along with their dates. The SM data were used to create the LSTM input whilst the auxiliary data were used for the MLP input.

The same methodology described in Section 3.1.1 has been used to create customer samples. The processed dataset was split further based on whether a customer sample has been previously inspected or not. This created a labeled and unlabeled dataset. The unlabeled dataset consisted of samples belonging to customers who were never inspected or whose normalization date was more than 365 days ago. The labeled dataset was used to train the model in a supervised manner, using the results of previous on-field inspections. This dataset was split further into a training, validation and test dataset in order to assess the performance of the model on samples that have not been seen during training.

A thorough description of the data processing techniques used can be found in Section 3.2.3. During model selection, the best hyperparameters of the model were selected using the validation dataset. The final performance was assessed on the test dataset. Similarly with the previous approach, if the model performance on the test dataset was acceptable for the utility, the trained model was used to make predictions on the unlabeled dataset, creating a ranked list of customers to be inspected.

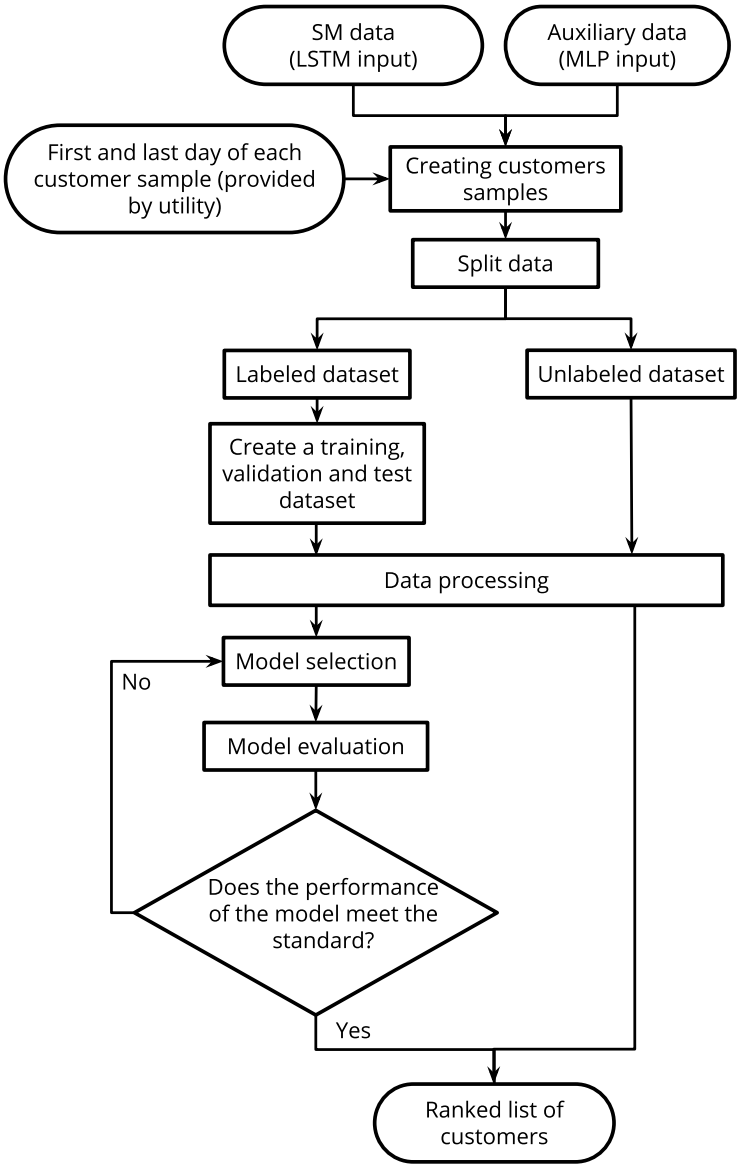


Figure 3.23 Methodology outline for NTL detection using raw data and hybrid neural networks.

3.2.3 Data availability and processing

Two types of data sources have been used in this methodology: SM data and auxiliary databases. The data has been provided by an electricity utility and contained meter

measurements and additional information of SMs that belong to residential and small-commercial customers, with a contracted power lower than 15 kW. Table 3.14 shows the data availability for this methodology. These data have been anonymized and a noise of approximately 5 km has been added to the geographical coordinates.

Table 3.14 Data availability.

Type of data	Data
Smart meter data	Hourly energy consumption Timestamp of energy consumption measurements
Geographical data	Latitude Longitude Altitude Municipality
Contractual data	Contracted power Contract type Voltage
SM technical data	SM model SM location SM firmware version SM production year
Economic data	Economic activity code

The labeled dataset that was used for training consisted of SMs which had at least one on-field inspection. To analyze the capability of the model to generalize beyond its training dataset, the original data have been split into a training, validation and test dataset. This is a different approach compared to the previous work, which used a nested cross-validation method for testing and selecting the best hyperparameters of each each model. However, using cross-validation techniques on deep learning algorithms can be very computationally expensive thus a single split of the original training dataset has been chosen.

The split has been done in a stratified manner, so that there is the same % of NTL samples in each dataset. The training dataset consisted of 80 % of the labeled dataset, whilst the validation and test datasets consisted of $\approx$ 10 % each. Table 3.15 shows the number of samples in each dataset as well as their % of NTL samples.

Table 3.15 Labeled dataset.

Dataset type	Number of samples	% of NTL samples
Training	85226	13.34%
Validation	10612	13.50%
Test	10701	13.23%

As it can be seen, the dataset is highly imbalanced which can make the model biased to probabilities closer to 0. This imbalance is however lower than the one in the previous

dataset.

Several data processing techniques have been used to process both the labeled and unlabeled datasets. Each type of sequential data (e.g. time series data such as the EC history) have been normalized separately, using their maximum value:

$$f(x) = \frac{x}{\max(x)} \tag{3.7}$$

In the case of non-sequential features, the missing values in non-categorical features were replaced with the mean. For categorical features, a special "Unknown" category has been created to replace missing data. After imputing the missing data, the non-sequential features were standardized to have 0 mean and unit variance using the following formula:

$$f(x) = \frac{x - \bar{x}}{s} \tag{3.8}$$

where $\bar{x}$ represents the mean of the input feature and s represents the standard deviation.

3.2.4 Hybrid deep neural network architecture

Figure 3.24 shows the architecture of the hybrid neural network (HNN-NTL), which is capable of integrating both sequential and non-sequential SM information. The network consists of three modules:

- LSTM module - uses the sequential data of the SM (e.g. EC).
- MLP module - uses the non-sequential data as as an input (e.g. SM location, model).
- Hybrid module - uses as input the outputs of the LSTM module and the MLP module and provides the final probability of having an NTL in the SM.

This type of architecture is very efficient as it permits joint training on both types of input. A detailed description of each module is presented in the following subsections.

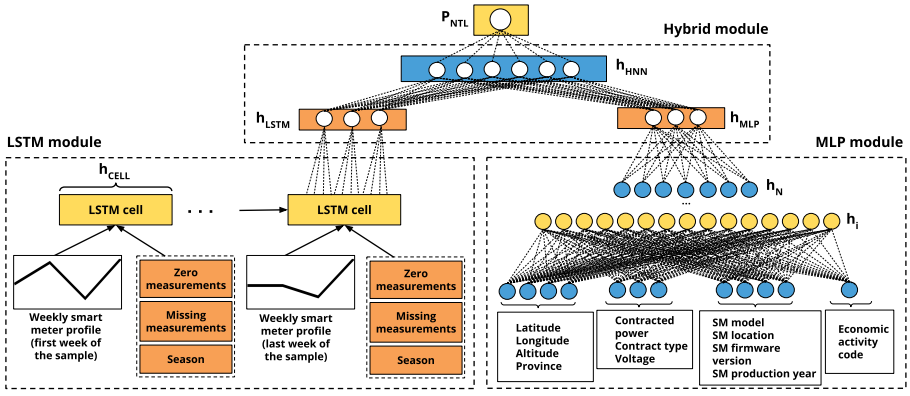


Figure 3.24 HNN-NTL model architecture.

3.2.4.1 Long short-term memory module for sequential data

Table 3.16 shows the input features that get fed into the LSTM model at each time step. The size represents the number of values that get fed into the input. Though the SMs record the EC hourly (See Section 3.2.3), the granularity has been reduced to the daily level as it has yielded better results than using the hourly measurements. The weekend ECs were removed using the timestamp of each SM measurement. Thus, the weekly profile consisted of 5 measurements of the daily average consumption at each time step.

Table 3.16 LSTM input at each time step.

Input	Description	Size
Weekly profile	Daily energy consumption of the weekdays.	5
Zero measurements	Number of 0 kWh measurements in each weekday.	5
Missing measurements	Number of null measurements in each weekday.	5
Season	The season of the week (spring, summer, autumn, winter).	4

Since the EC history recorded by the SMs can be years long and it is increasing day by day, a simple recurrent neural network [69] cannot be used as it would be very hard to train due to its vanishing and exploding gradient problems [70]. Thus, to capture the long-term dependencies in the variable-sized EC data an LSTM cell has been used [71]. The LSTM cell uses the sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$ and the hyperbolic tangent $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ as nonlinear activations and it has the following mathematical formulation:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{3.9}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3.10}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3.11}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3.12}$$

$$h_t = o_t \odot \tanh(C_t) \tag{3.13}$$

where i_t, f_t, o_t, C_t and h_t represent the activations of the input gate, forget gate, output gate, cell state and hidden state at time step t. W_i, W_f, W_o and W_c represent the weights of the input layer whilst U_i, U_f, U_o and U_c represent the recurrent weights of the LSTM. b_i, b_f, b_o, b_c are the biases of the network whilst x_t is the input feature vector at time step t and h_{t-1} represents the hidden state activation at the previous time step. $\odot$ represents the

element-wise multiplication (Hadamard product).

The LSTM module sees the entire EC history, week by week, and it provides a single final output, h_T, which is the hidden state of the LSTM cell at the final time step (last week of the sample).

3.2.4.2 Multi-layer perceptrons module for non-sequential data

The MLP network was used to analyze the non-sequential data. The input data that has been used for this module can be found in Table 3.17. Input features that are continuous have a size of 1 whilst categorical features have a higher dimension. Entity embeddings [72] have been used to encode the categorical variables, instead of one-hot-encoding, in order to reduce the input space of the MLP network.

Table 3.17 MLP input data.

Type of data	Input	Size
Geographical data	Latitude	1
	Longitude	1
	Altitude	1
	Municipality	5
Contractual data	Contracted power	1
	Contract type	2
	Voltage	1
SM technical data	SM model	3
	SM location	3
	SM firmware version	3
	SM production year	3
Economic data	Economic activity code	10

The MLP module has N hidden layers, where N is chosen using the validation dataset. Each hidden layer goes through an affine transformation ($n > 0$):

$$z_n = W_n h_{n-1} + b_n \tag{3.14}$$

where W_n represents the weights of layer n, h_{n-1} represents the hidden state of the previous layer and b_n represents the bias of the n_{th} layer.

To speed up the convergence of the network, a batch normalization layer [73] has been used on the affine transformation:

$$b_n = \gamma \hat{z}_n + \beta \tag{3.15}$$

where $\hat{z}_n$ represents the standardized affine activation with the mean and standard deviation of the batch sample and γ and β are trainable parameters that are tuned during optimization.

The final state of the hidden layer was obtained by using a rectified linear unit:

$$h_n = \max(0, b_n) \tag{3.16}$$

3.2.4.3 Hybrid module

The hidden state h_{LSTM} of the hybrid module has been obtained using as input the hidden state of the LSTM cell h_T at the final time step (T is the sequence length). Similarly, the hidden state h_{MLP} has been computed using as an input the hidden state activations of the last hidden layer in the MLP module h_N. Both h_{LSTM} and h_{MLP} have been computed using the transformations described in the equations (3.14), (3.15) and (3.16).

The hybrid module simply takes afterwards the hidden states h_{LSTM} and h_{MLP} and concatenates them in order to form a new hidden layer.

The final state of the HNN-NTL model is obtained as follows:

$$h_{HNN} = \max(0, \gamma \hat{z}_{HNN} + \beta) \tag{3.17}$$

where $\hat{z}_{HNN} = W_{HNN}[h_{LSTM}, h_{MLP}] + b_{HNN}$ and γ and β are trainable parameters of the model.

The outcome of the network was computed using the sigmoid activation, providing a score between 0 and 1. This score can be interpreted as the probability that there is an NTL in the SM, though its confidence strength depends on the strength of regularization [74]:

$$P_{NTL} = \frac{1}{1 + e^{-(W_{NTL} h_{HNN} + b_{NTL})}} \tag{3.18}$$

where P_{NTL} represents the probability that there is an NTL in the SM. W_{NTL} and b_{NTL} represent the trainable weights and bias of the output layer.

3.2.4.4 Learning and evaluation

The performance of the model has been evaluated with the logarithmic loss function, as this is a binary classification task:

$$L = \frac{1}{M} \sum_{i=1}^{M} -(y_i \log(P_{NTL}^i) + (1 - y_i) \log(1 - P_{NTL}^i)) \tag{3.19}$$

where M is the number of customer samples, y_i is the ground-truth label and P_{NTL}^i is the probability of NTL computed by the HNN-NTL model for the customer sample i.

The trainable parameters of the model have been initialized with a Xavier initialization [75] and optimized to minimize the loss function using the Adam optimizer [76], a first-order gradient-based optimization method.

The same metrics used in the previous approach have been used, as the dataset is as well very imbalanced and the goal of this methodology is the same as the previous one: to rank customers with NTL as high as possible. Thus, the ROC-AUC and PR-AUC scores are reported for the HNN-NTL model as well as for all the models used in the comparison. The ROC-AUC score of a dummy model which makes random predictions is 0.5 whilst its PR-AUC score is the percentage of NTL samples in the dataset, in this case ≈ 0.13.

3.2.4.5 Results

As it was seen in Figure 3.24, the size of the HNN-NTL network can be adjusted by controlling the size of various hidden layers such as h_{LSTM} and h_{HNN}. A grid-search has

been implemented in order to find the best hyperparameters. Table 3.18 shows the range of values searched as well as the optimal values found. The optimal values were found by monitoring the performance on the validation dataset. As a regularization method, a dropout layer has been used on the output of the h_{LSTM}, h_{MLP}, h_{HNN} and every h_i layer of the MLP module. No regularization has been used on the LSTM cell h_{CELL}, as it did not improve the performance of the model.

Table 3.18 HNN-NTL hyperparameters search.

Hyperparameter	Range of values	Optimal value
N	4, 6	4
Size h_i	256, 512	256
Size h_{CELL}	256, 512	256
Size h_{LSTM}	256, 512	512
Size h_{MLP}	256, 512	512
Size h_{HNN}	1024, 2048	1024
Dropout	0.3, 0.5	0.3

a. LSTM with the weekly profile

This experiment uses only the LSTM module from the HNN-NTL model, in order to assess its performance and prediction capabilities. It also uses only the weekly profile as an input, omitting the information related to the number of 0 kWh and null measurements as well as any temporal information of the EC history such as the season at each timestep.

Figure 3.25 shows the performance of the LSTM model when using as input only the weekly profile. As can be seen in Figures 3.26 and 3.27, a PR-AUC of 0.33 and a ROC-AUC score of 0.72 have been obtained on the test dataset. Even with such simple input, the model significantly outperforms random predictions.

b. LSTM with all data

The next experiment uses only the LSTM module, but with the entire input (weekly profile, zero measurements, missing measurements and season). Figures 3.28, 3.29 and 3.30 show the model performance of the LSTM model when using all input data. By using this additional data, the PR-AUC has increased from 0.33 to 0.41 on the test dataset.

c. HNN-NTL model

The performance of the HNN-NTL model can be seen in Figures 3.31, 3.32 and 3.33. As can be seen, the HNN-NTL model greatly outperforms the LSTM model, obtaining a PR-AUC score of 0.54 on the test dataset. As expected, using non-sequential features such as the contracted power or the SM model dramatically improves the performance for NTL detection.

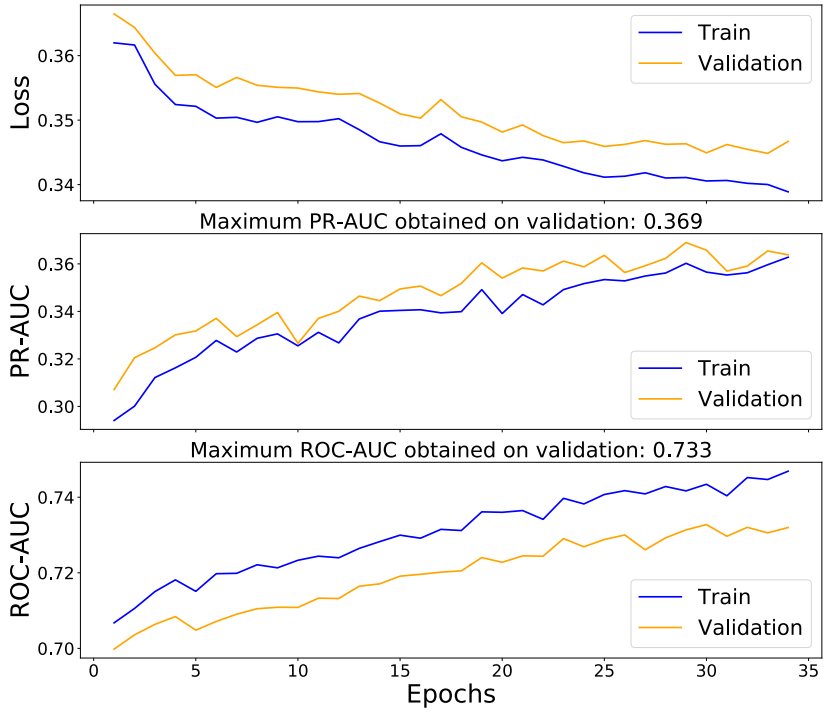


Figure 3.25 Simple LSTM model performance during training.

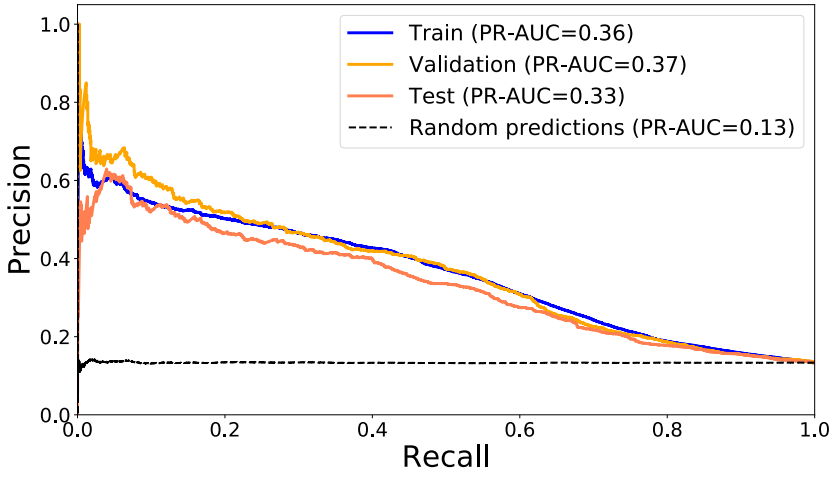


Figure 3.26 PR curve of simple LSTM model using the best trained model.

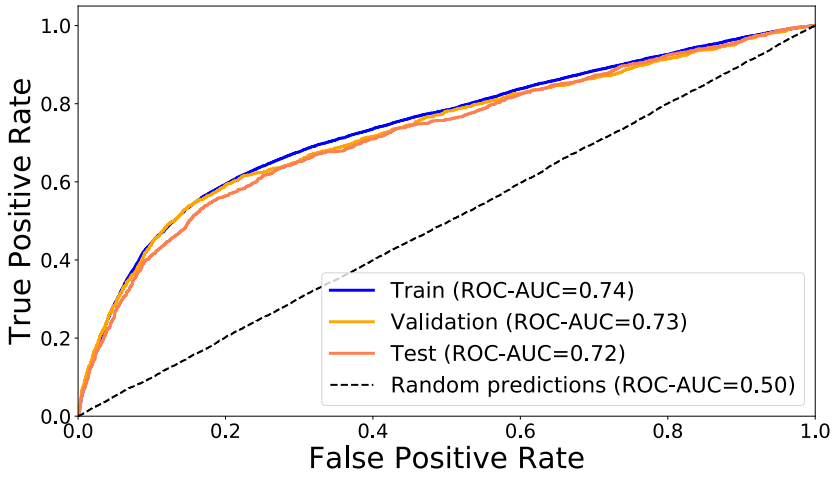


Figure 3.27 ROC curve of simple LSTM model using the best trained model.

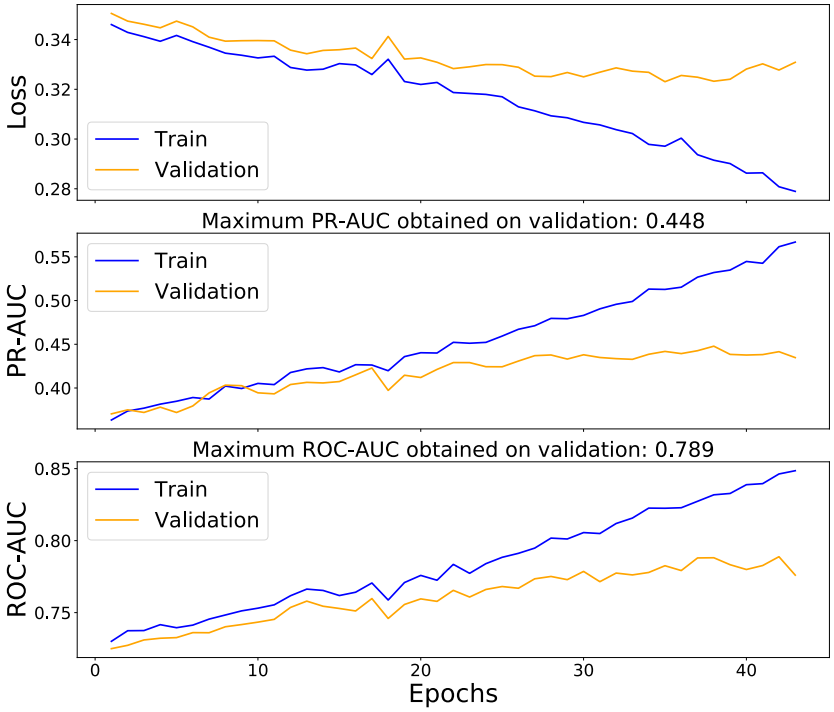


Figure 3.28 LSTM model performance during training.

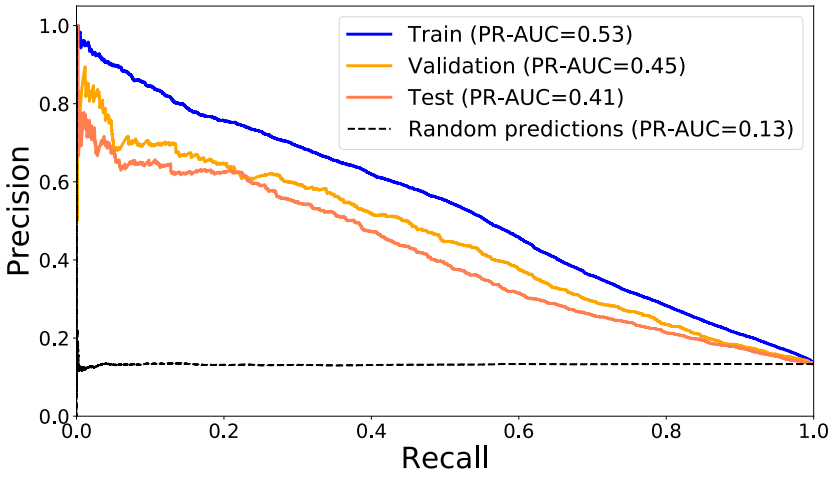


Figure 3.29 PR curve of LSTM model using the best trained model.

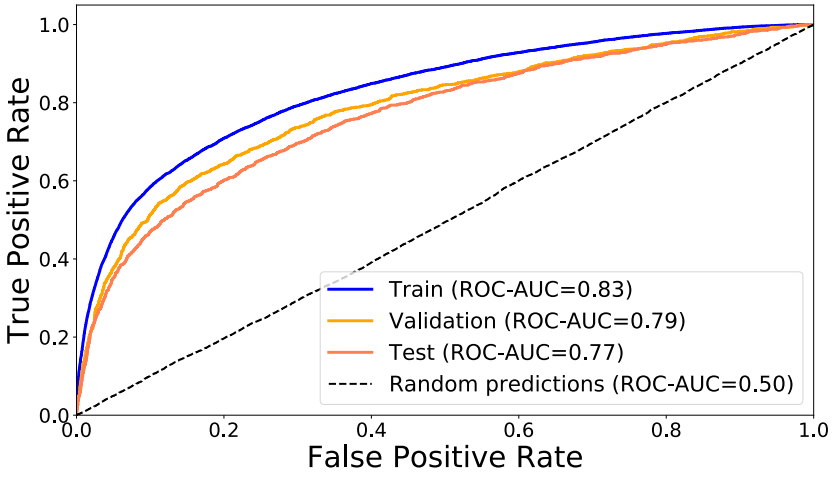


Figure 3.30 ROC curve of LSTM model using the best trained model.

3.2.5 Comparison with traditional machine learning models

In this section, a comparison between the performance of the HNN-NTL model and state-of-the-art classifiers has been made. The following algorithms have been used in the comparison:

- Support Vector Machines (SVM)
- Logistic Regression (LR)
- Random Forests (RF)

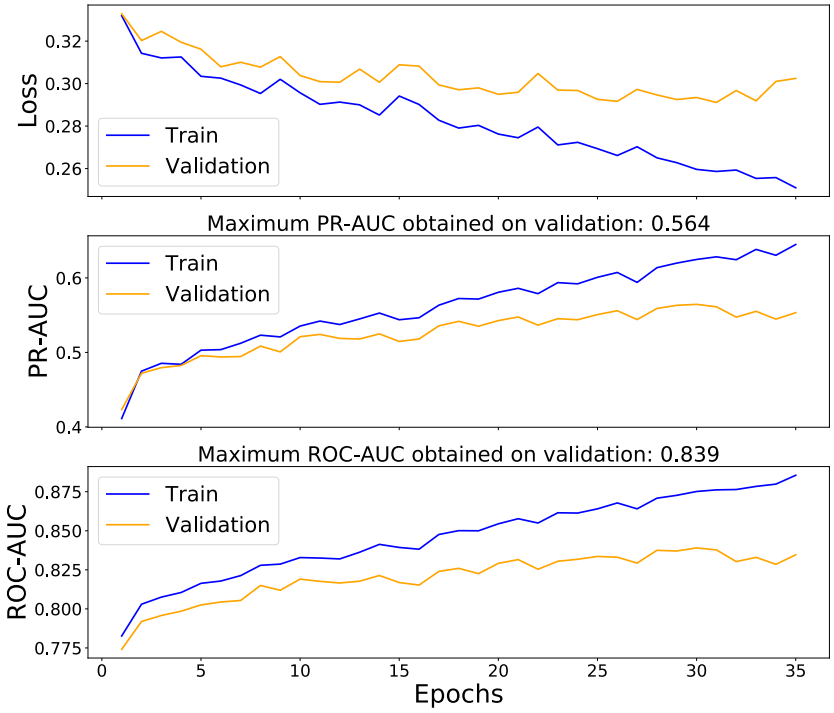


Figure 3.31 HNN-NTL model performance during training.

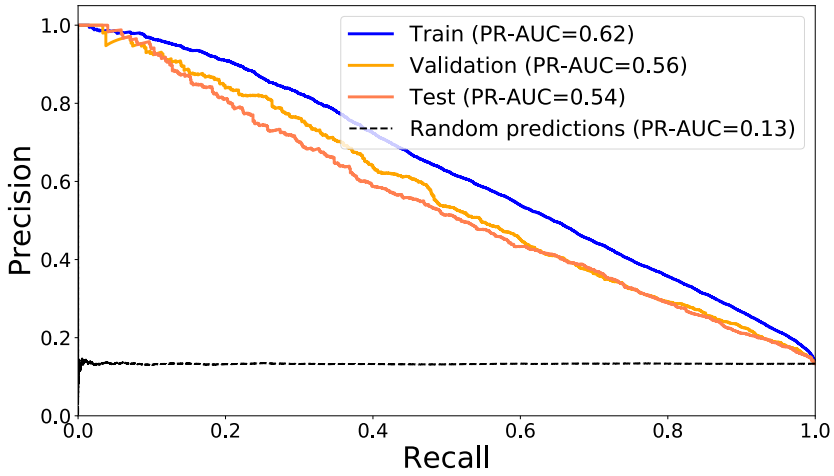


Figure 3.32 PR curve of HNN-NTL model using the best trained model.

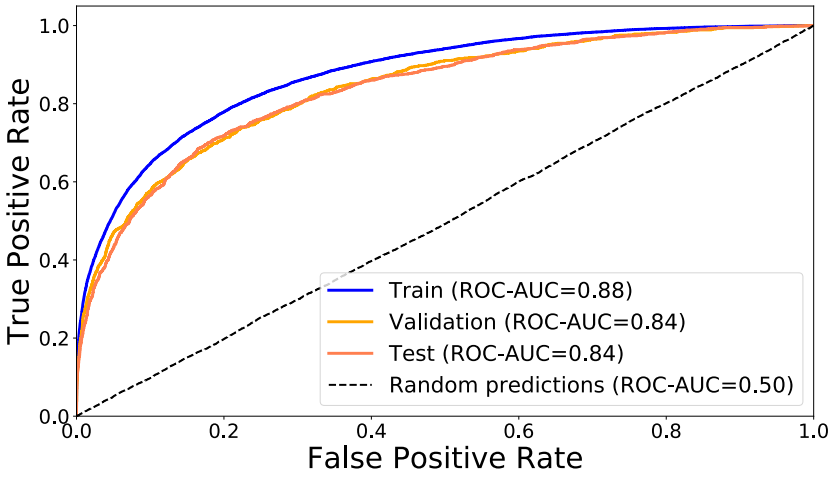


Figure 3.33 ROC curve of HNN-NTL model using the best trained model.

- Extreme Gradient Boosted Trees (XGBoost)
- Multi-Layer Perceptrons (MLP) Networks

All the models described above require a fixed size input, thus a fixed size window of one year on the sequential data has been used (daily EC of the past year). The same training, validation and test datasets described in Table 3.15 have been used for all the models in this comparison.

The input that was used for these models is shown in Table 3.19. The auxiliary data input is equivalent to the same input used in the MLP module of the HNN-NTL model. The same entity embeddings have been used for the categorical features. The EC input consists of the same information that has been used in the LSTM module, but restricted to the EC of the previous year.

3.2.5.1 Support Vector Machines

As seen in the previous methodology, as well as in Chapter 2, the SVM is a very popular classifier for NTL detection. Table 3.20 shows the range of hyperparameters used during grid-search as well as their optimal values found on the validation dataset. In this case, only a linear kernel has been used due to the size of the dataset.

Figures 3.34 and 3.35 show the performance of the SVM model on the training, validation and test dataset. The ROC-AUC and PR-AUC scores are much more lower than the ones obtained with the HNN-NTL model. However, the model is still performing considerably better than random predictions.

3.2.5.2 Logistic Regression

The LR model developed in the previous methodology has been used. Table 3.21 shows the hyperparameters used during grid-search for LR along with their correspondent optimal values.

Table 3.19 SVM, LR, RF, XGBoost and MLP inputs.

Input	Description	Size
Auxiliary data	Latitude	1
	Longitude	1
	Altitude	1
	Municipality	5
	Contracted power	1
	Contract type	2
	Voltage	1
	SM model	3
	SM location	3
	SM firmware version	3
	SM production year	3
	Economic activity code	10
EC input	Daily EC consumption	260
	Daily zero measurements	260
	Daily missing measurements	260

Table 3.20 SVM hyperparameters search.

Hyperparameter	Range of values	Optimal value
C	0.001, 0.01, 10, 100	0.001

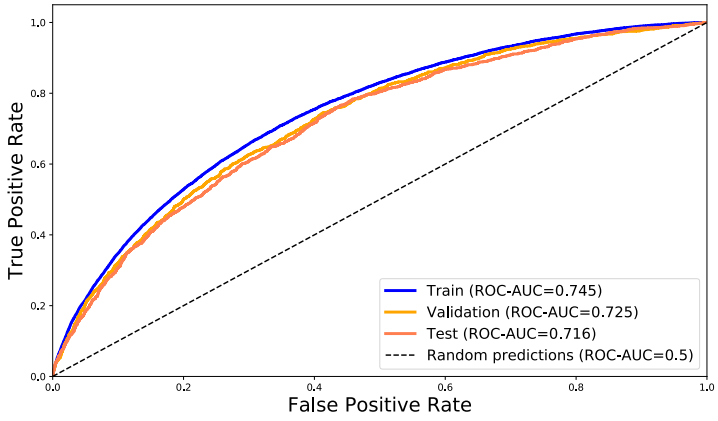


Figure 3.34 ROC curve of SVM model using the best trained model.

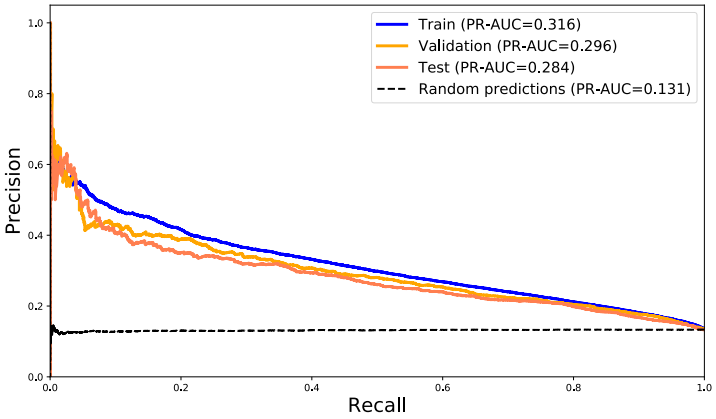


Figure 3.35 PR curve of SVM model using the best trained model.

Table 3.21 LR hyperparameters search.

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

Figures 3.36 and 3.37 show the performance of the LR model. As it can be seen, the LR model slightly outperforms the SVM. The performance on train, validation and test dataset is very similar which suggests that the model is not overfitted and that it has the ability to generalize beyond its training dataset.

3.2.5.3 Random Forests

RF fall into the category of ensemble models [77]. The algorithm combines several decision trees to create a collection of trees that can make more accurate predictions. The mode of the predictions of individual trees is used in order to output a final decision. Table 3.22 shows the hyperparameters used during grid-search for RF.

Table 3.22 RF hyperparameters search.

Hyperparameter	Range of values	Optimal value
Number of trees	1000, 2000	1000
Minimum samples split	5, 10, 15	10
Maximum depth	7, 15	15
Minimum samples leaf	5, 10, 15	15

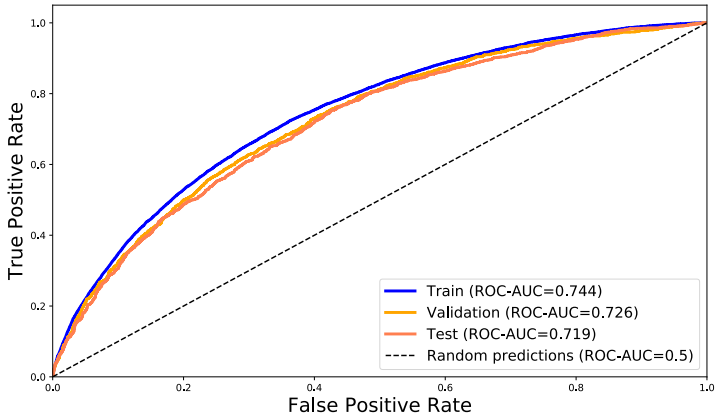


Figure 3.36 ROC curve of LR model using the best trained model.

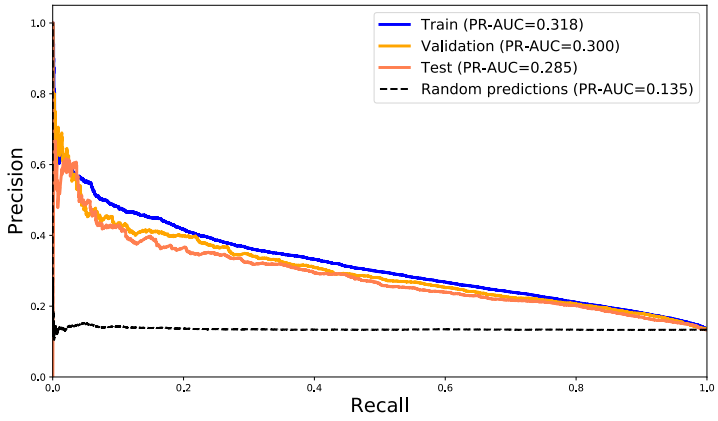


Figure 3.37 PR curve of LR model using the best trained model.

Figures 3.38 and 3.39 show the performance of the RF model. As it can be seen, the RF model significantly outperforms the SVM and LR models. However, the gap between the performance of the training, validation and test datasets is now considerable. This was expected as the RF model is using decision trees which are known to be prone to overfitting. Though the RF model outperforms the rest of the traditional classifiers, it still has not been able to surpass the performance of the HNN-NTL model.

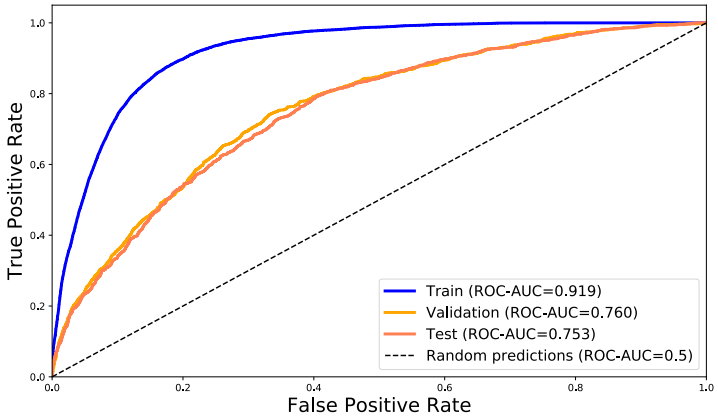


Figure 3.38 ROC curve of RF model using the best trained model.

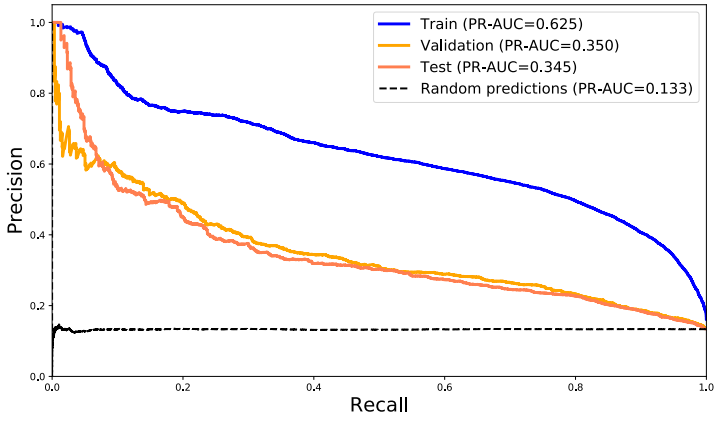


Figure 3.39 PR curve of RF model using the best trained model.

3.2.5.4 Extreme Gradient Boosted Trees

XGBoost has already been used successfully for NTL detection [20]. It is thus only fair to include it into the comparison. Table 3.23 shows the hyperparameters used during grid-search as well as the optimal values found.

Figures 3.40 and 3.41 show the performance of the XGBoost model on the training, validation and test datasets. As expected, XGBoost outperforms the SVM, LR and RF models. The XGBoost model is as well using decision trees so the model is still overfitting on the training dataset. This issue could be mitigated by searching for higher regularization

Table 3.23 XGBoost hyperparameters search.

Hyperparameter	Range of values	Optimal value
Number of trees	1000, 2000	1000
Learning rate	0.01, 0.1	0.01
Maximum depth	7, 15	7
Minimum child weight	1, 5, 10	10

rates during grid-search. The performance of the XGBoost model is still considerably lower than the one obtained with the HNN-NTL model.

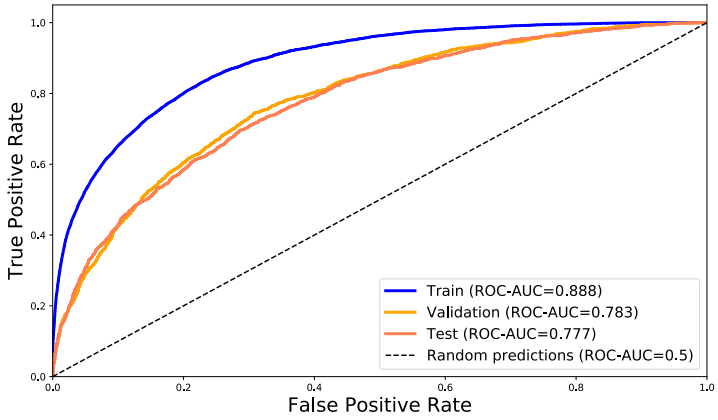


Figure 3.40 ROC curve of XGBoost model using the best trained model.

3.2.5.5 Multi-Layer Perceptrons Networks

MLP networks have been used successfully for NTL detection in [45] and [44]. In this comparison, the same architecture and hyperparameters of the MLP module from the HNN-NTL model has been used. This MLP network has in addition input features extracted from the raw EC data.

Figures 3.42 and 3.43 show the performance of the MLP model. The MLP model has not been able to surpass the performance of the best traditional classifier: XGBoost. A ROC-AUC score of 0.738 and a PR-AUC score of 0.314 has been obtained on the test dataset. The results show that restricting the input to the last year of the EC history dramatically affects the performance of the model. Neural networks by themselves are not able to surpass the performance of powerful classifiers such as XGBoost. The ability of the HNN-NTL model to surpass the performance of all traditional classifiers comes from the fact that it is not constrained to restrict the input size of the EC history.

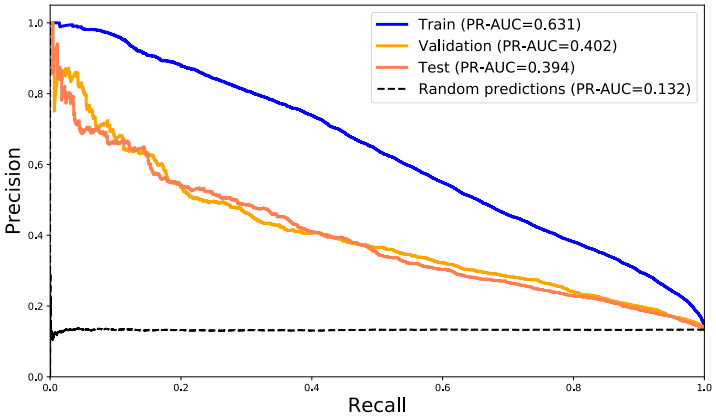


Figure 3.41 PR curve of XGBoost model using the best trained model.

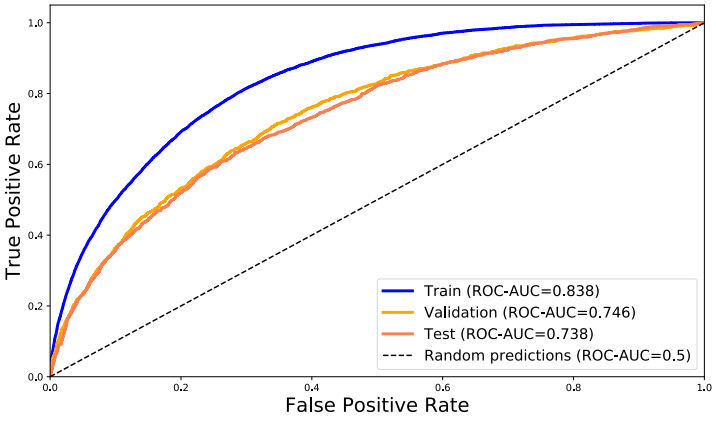


Figure 3.42 ROC curve of MLP model using the best trained model.

3.2.6 Comparison with other deep learning models

In this section, a comparison with the following deep learning approaches has been made:

- Convolutional Neural Networks, as in [51].
- Wide & Deep Convolutional Neural Networks, as in [54].

Both architectures do not allow for a variable sized input, thus the EC history has to be restricted to a fixed size input as well.

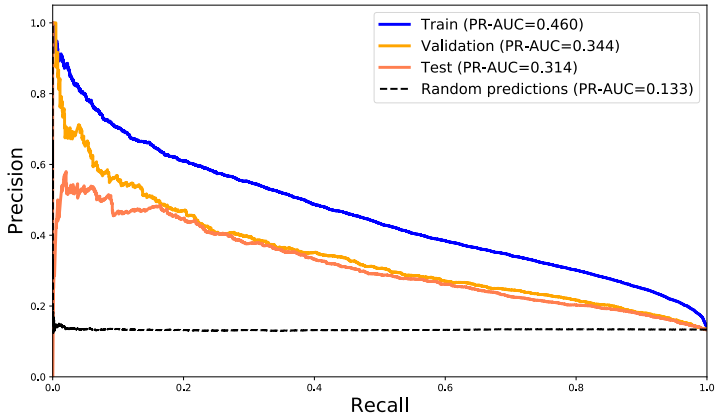


Figure 3.43 PR curve of MLP model using the best trained model.

3.2.6.1 Convolutional Neural Networks

CNN have been shown to outperform stacked autoencoders and LSTM networks in [51], on a dataset with synthetic NTL samples. The same architecture proposed in [51], as well as the same hyperparameters have been used. The original experiment used only monthly EC data as an input to the CNN network but the granularity has been increased to daily EC measurements given better data availability.

Furthermore, for fairness reasons, the experiments have been divided into two sections:

- First experiment: using only the EC data as an input.
- Second experiment: adding the MLP module from the HNN-NTL model to the CNN architecture, so that the network has access to the same information as the rest of the models.

a. Using only EC data as input

In this experiment, the CNN network uses as input only the 1D daily EC data, as can be seen in Table 3.24.

Table 3.24 CNN input.

Input	Description	Size
EC input	1D daily EC consumption of last year.	260

Figures 3.44 and 3.45 show the ROC-AUC and the PR-AUC scores obtained using the CNN network. As expected, the CNN performs very poorly compared to the HNN-NTL model as well as to the traditional classifiers. A ROC-AUC score of 0.689 and a

PR-AUC score of 0.244 have been obtained on the test dataset. The low performance can be attributed to the fact that the CNN network is restricted to a fixed size input as well to the fact that it uses only EC data.

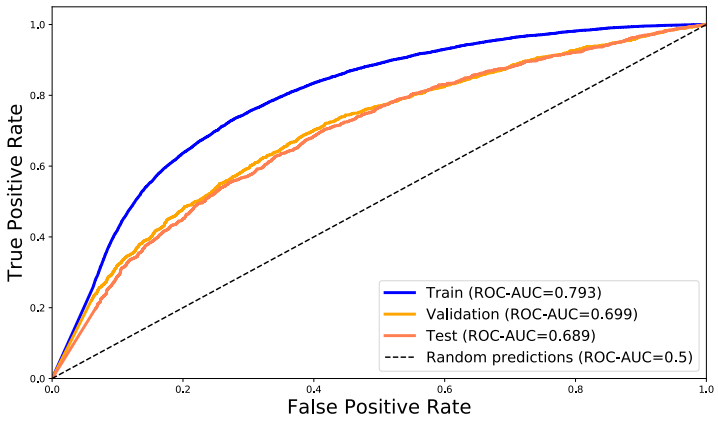


Figure 3.44 ROC curve of CNN model using the best trained model.

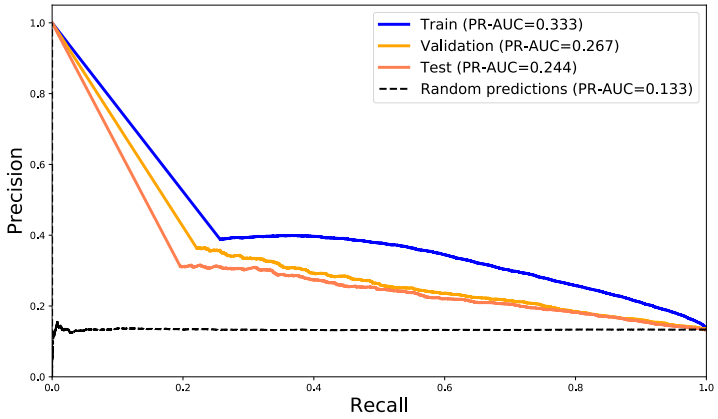


Figure 3.45 PR curve of CNN model using the best trained model.

b. Using both EC and auxiliary data as input

In this experiment, the output of the MLP module was simply concatenated with the output of the CNN module before making the final predictions. The CNN-MLP network uses as

input the 1D daily EC data for the CNN module and auxiliary data for the MLP module, as can be seen in Table 3.25.

Table 3.25 CNN-MLP input.

Input	Description	Size
CNN input	1D daily EC consumption of last year	260
MLP input	See Table 3.17	34

Figures 3.46 and 3.47 show the performance of the CNN-MLP model, using the PR and ROC curves. As expected, the CNN-MLP network outperforms significanty the simple CNN model. Having access to auxiliary data that gives the network additional information besides EC data, such as the location of the SM or its brand, seems to bring a significant boost in performance. The ROC-AUC score increased from 0.689 to 0.756 whilst the PR-AUC score increased from 0.244 to 0.327.

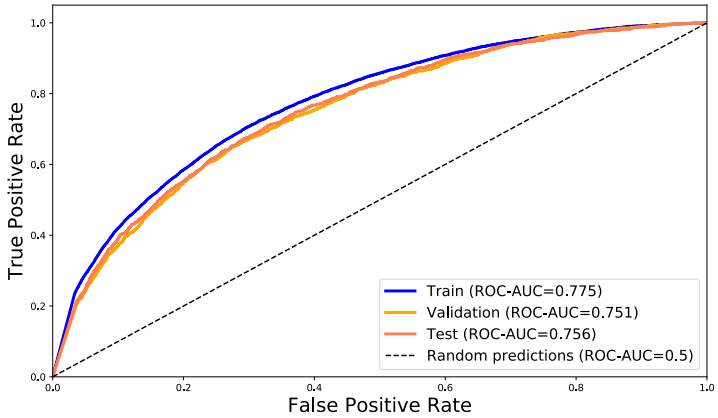


Figure 3.46 ROC curve of CNN-MLP model using the best trained model.

3.2.6.2 Wide & Deep Convolutional Neural Networks

The wide & deep convolutional neural network (WD-CNN) is a deep learning architecture for NTL detection proposed by the authors in [54]. The algorithm uses a wide network (equivalent to a MLP network) on the 1D daily EC data and a CNN on the 2D stacked weekly energy profiles. This experiment uses the same model architecture, therefore a grid-search has not been performed for this model. The special convolution kernel has also been implemented using a hyperbolic tangent activation function. Similarly to the CNN experiment, the MLP module has also been added as a separate component within the architecture so that the algorithm has access to the available auxiliary data.

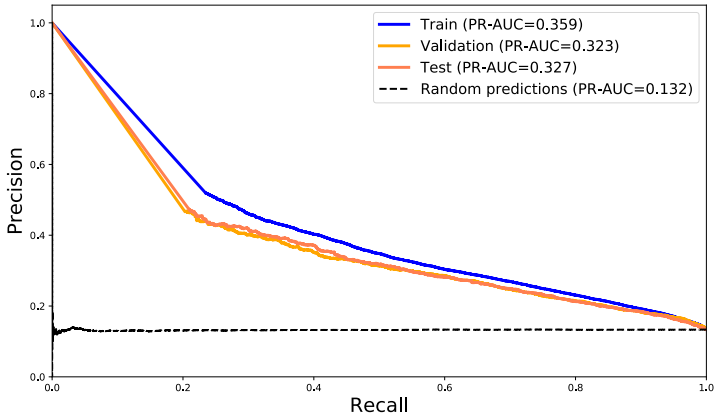


Figure 3.47 PR curve of CNN-MLP model using the best trained model.

a. Using only EC data as input

The input for the WD-CNN algorithm that uses only EC data as input can be found in Table 3.26. For the wide module, the input is simply the daily EC of the past year. To create the input for the CNN module, the daily EC consumption has been divided into weekly profiles and stacked into a 2D array. The weekly profiles were stacked starting with the first week of the year up until the last.

Table 3.26 WD-CNN input.

Input	Description	Size
Wide input	Daily EC consumption of last year.	260
CNN input	Weekly EC profiles of last year.	260

Figure 3.48 and Figure 3.49 show the performance of the WD-CNN model. The WD-CNN model obtains a slightly higher PR-AUC score compared to the CNN network but its ROC-AUC score is slightly lower. As mentioned, the network has access only to EC data thus its poor performance cannot be attributed solely to the architecture.

b. Using both EC and auxiliary data as input

The input for the WD-CNN-MLP algorithm can be found in Table 3.27. The WD-CNN-MLP architecture simply adds the MLP module used in the HNN-NTL model and concatenates its output with the WD-CNN module before making a final prediction.

Figures 3.50 and 3.51 show the performance of the WD-CNN-MLP model on the training, validation and test datasets. As with the CNN-MLP model, the performance fo the WD-CNN-MLP is significantly higher than the one of the simple WD-CNN net-

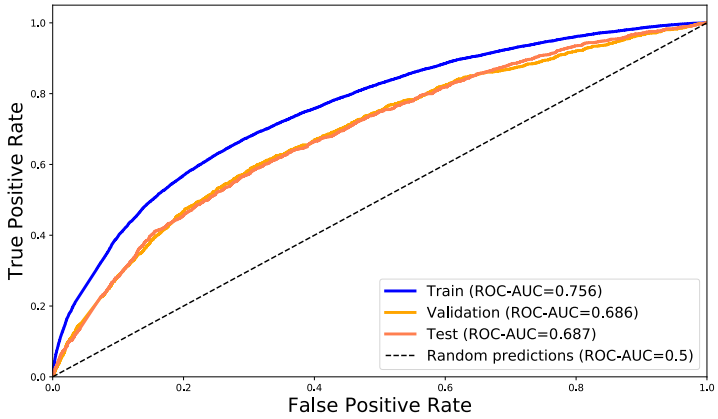


Figure 3.48 ROC curve of WD-CNN model using the best trained model.

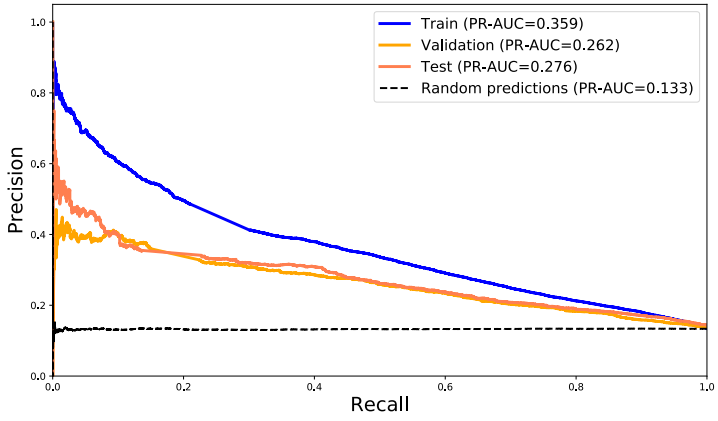


Figure 3.49 PR curve of WD-CNN model using the best trained model.

Table 3.27 WD-CNN-MLP input.

Input	Description	Size
Wide input	Daily EC consumption of last year	260
CNN input	Weekly EC profiles of last year	260
MLP input	See Table 3.17	34

work. Though the performance of the network has improved, the HNN-NTL model still outperforms it by a quite significant margin.

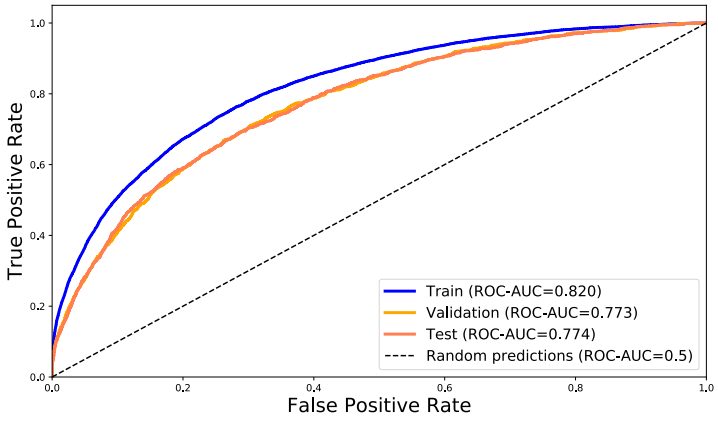


Figure 3.50 ROC curve of WD-CNN-MLP model using the best trained model.

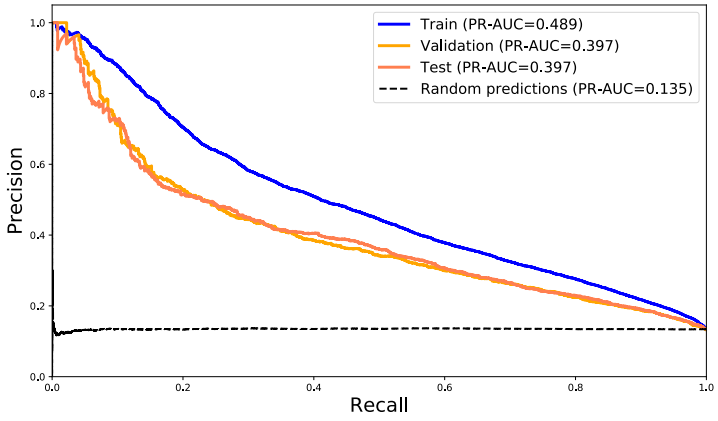


Figure 3.51 PR curve of WD-CNN-MLP model using the best trained model.

4 Discussion

The previous chapter described and showed the results of the two methodologies this thesis is based on. Each of these methodologies has been developed and tested on different datasets and targeted for different types of customers.

The first methodology has been targeted to detect NTL in industrial and large commercial customers, with a contracted power higher than 50 kW. Electricity utilities are particularly focused on these types of customers, as NTL in large customers can represent a large percentage of the total NTL losses. Thus, detecting an NTL in the meter of an industrial customer recovers significantly higher revenue losses than in the case of a residential customer. At the moment the first NTL model has been designed and developed, data-oriented methodologies were already starting to make use of SM data and handcrafted feature engineering based on domain knowledge. By comparing the existing literature, several gaps have been found in the following areas:

- Robustness to detect NTL occurring from the beginning.
- Adaptability to new types of NTL attacks.
- Data privacy.
- Detection delay.

Thus, the NTL methodology has been developed having in mind this set of criteria. To make sure that the NTL model is able to detect NTL that has occurred from the beginning, where no descent in the EC can be observed, clustering techniques have been used in order to compare customers among themselves rather than their individual consumption history. In order to improve the reliability of this technique, both distance and density metrics have been employed in order to asses how similar is a consumption curve compared to the ones belonging to the same group. Another important quality of an NTL methodology is the adaptability to new types of NTL attacks. To mitigate this challenge, the methodology extracts features from multiple types of SM data (e.g. alarms, electrical magnitudes) as well as auxiliary databases. Having access to diverse types of features, increases the adaptability of the model, as it has more complete information of the customer. Data granularity plays also an important role in the adaptability to new types of attacks as well

as in data privacy and detection delay. Intermittent fraud, where an NTL might occur only during some periods of a day, week or month can be detected much easier with a higher data granularity of EC measurements. Though the SMs of these customers record the EC every 15 minutes, the granularity has been reduced to 5 measurements a day in order to increase data privacy. The detection delay of an NTL methodology represented as well an important criterion in the design of the model. Very often, past methodologies proposed on NTL detection would rely on EC data of the past 12 months, creating a detection delay of one year. This represents a major issue for utilities. In theory, the methodology proposed has a very short detection delay (as little as one day), as it creates features from EC consumption histories of various lengths. In practice, a limit of 90 days on each customer sample has been put, in order to give the network more information on the consumption pattern. This makes the network more robust to the potential noise in the dataset.

Another important aspect of this work was training the XGBoost model with different subset features. This can represent an important step for any utility who is looking to collect data to create its own NTL detection algorithm. The results can give some information on whether the SM data or auxiliary data should be prioritized for data collection.

The second methodology was focused on developing an NTL model that does not rely on handcrafted feature engineering and domain knowledge. It was targeted to detect NTL on residential and small commercial customers with a contracted power lower than 15 kW. Naturally, this dataset was bigger than the previous one thus it was a great fit for methodologies based on deep learning, which require significant amounts of data to be trained on.

The most important limitation of traditional ML models is that they cannot function with a variable sized input. Since the length of a customer sample varies for each one in part, the EC history has to be either restricted to the last N measurements or to various features extracted from it, as it was done with the previous methodology. Thus, the second methodology has been designed so that it can process both input features with variable length as well as static features that do not change through time (e.g. SM location). This has been done by combining a LSTM network to process the variable sized input and a MLP network, that processes the non-sequential features. By combining both networks, it has been shown that a significant boost in performance is being obtained.

The same limitation of traditional ML models has been encountered in the previous deep learning methodologies proposed for NTL detection. Though the authors in [51] used a LSTM network to detect NTL, they have concluded that the CNN is outperforming it. However, their study has been done on a dataset with synthetically generated NTL cases, with oversimplifying assumptions on the types of NTL. The experiments presented in this thesis show that on real NTL datasets, LSTM is outperforming methodologies based on CNN. This was expected, as restricting the input of the EC history makes the model less reliable as it has incomplete information on the changes in the EC pattern (e.g. missing the descent in consumption). Moreover, none of the previous methodologies had the capability to integrate both sequential and non-sequential features.

A direct comparison between the two methodologies presented in this thesis is difficult as they were both developed and targeted to detect NTL in different types of customers.

5 Conclusion

This thesis explored the capabilities of machine learning algorithms and smart meter data for the detection of non-technical losses in electricity utilities. It has developed two major methodologies for NTL detection. The first one was based on handcrafted feature engineering whilst the second one was based on a simple end-to-end model that uses raw data as input. Both methodologies have been developed and tested on datasets with real non-technical losses cases, belonging to a Spanish electricity utility. The data came from smart meters' measurements as well as from auxiliary databases which contained additional information of the meters such as their location. Besides testing the performance of these models on previous on-field inspections, they have also been implemented as two separate non-technical losses campaigns in the same electricity utility. The first methodology has obtained a precision of ≈ 21 %, for new on-field inspections made on industrial and large commercial customers whilst the second methodology obtained a precision of ≈ 47 % on residential and small commercial customers. Chapter-wise summaries are given below.

- Chapter 1 started by emphasizing the importance of detecting non-technical losses in electricity utilities. It also provided a context for detecting these losses with smart meters and machine learning algorithms. A classification on the type of losses considered has also been made. Last but not least, this chapter finished by providing the research objectives of this thesis.
- Chapter 2 provided an overview of the existing methodologies proposed for NTL detection. These methodologies have been classified either as grid, hybrid or data-oriented. The challenges in non-technical losses detection have also been discussed in this chapter.
- Chapter 3 described the methodologies and the main results of the two journal articles this thesis was based on. The first part of this chapter focused on describing all the stages involved in creating a machine learning model using handcrafted feature engineering data as an input, to detect non-technical losses. The second part of the chapter described the end-to-end hybrid neural network model used to detect non-technical losses in customers with a contracted power lower than 15 kW.

- Chapter 4 emphasized the main insights gathered from developing and testing the methodologies proposed in this paper, for non-technical losses detection. A brief comparison of these methodologies with previous works has also been made.

5.1 Thesis contributions

This thesis can help both fellow researchers working in this field as well as electricity utilities around the world who want to migrate from models based on simple rules derived from expert knowledge to machine learning models that are capable to learn automatically from the data. The contributions of this thesis are outlined below.

- Data collection and processing pipelines of smart meter and auxiliary data - this thesis provided an in-depth description of the data sources and processing techniques necessary to build a machine learning model for NTL detection. Standardizing, handling categorical variables and missing data are all important stages in the pipeline of a machine learning model. Thus, it is very important to have an accurate description of the data sourcing and processing pipeline in order to be able to replicate and further advance these models.
- Handcrafted feature engineering and supervised learning models - the thesis described a comprehensive set of features that can be extracted from smart meter data, that are relevant for this task. They use all the information that the smart meters record: energy consumption, alarms and electrical magnitudes. It also used a very powerful classifier, extreme gradient boosted trees, which was able to outperform algorithms that have been very often used for non-technical losses detection (e.g. support vector machines).
- Impact of data type and undersampling techniques on non-technical losses detection performance - an experiment with different subsets of features sourced either from smart meter or auxiliary data has been made. These subset features have been used as an input to a machine learning model, to study their impact on the detection performance. This experiment can be interesting for the utilities and fellow researchers, as it provides an insight on where the efforts should be focused, to increase the performance of these models. To reduce the imbalance in the original dataset, an experiment has been made where the number of samples belonging to the majority class (samples that have not been found with a non-technical loss) has been reduced either randomly or by using simple heuristics. The results have shown that undersampling techniques are effective at improving the performance of non-technical losses detection models.
- Deep learning with raw smart meter and auxiliary data - this thesis described a methodology for non-technical losses detection that does not rely on handcrafted feature engineering and uses simple raw data as an input. The features that are relevant for non-technical losses detection are extracted automatically, from the data. To my knowledge, this is the first architecture proposed that is able to incorporate both sequential (e.g. energy consumption) and non-sequential (e.g. smart meter location) features.

5.2 Limitations and future work

Though the methodologies proposed in this thesis have been successfully implemented in the real environment, they still have limitations. As they are trained in a supervised manner, they require a labeled dataset built using the results of previous on-field inspections. Thus, they cannot be easily implemented by researchers or utilities who do not have access to such labels. These methodologies are also relying on the efficiency and the accuracy of on-field inspectors. It is not sufficient for the model to detect a non-technical loss in the meter. The on-field inspector has to be able to detect it as well. This limitation can be mitigated by introducing information on the percentage of non-technical losses at the distribution transformer level. This will increase the reliability of the predictions made by the machine learning model. In this context, a smart meter will be inspected only if a non-technical loss has been computed at the corresponding distribution transformer level. I believe that the number of false positives can be greatly reduced by combining machine learning algorithms with grid data. Thus, future work should be focused on developing hybrid methodologies.

Another interesting area for future work is the development of more advanced methodologies for reducing the natural imbalance that occurs in non-technical losses datasets. Though undersampling techniques have been proven to be very effective, they are quite simplistic. There are more sophisticated techniques such as Synthetic Minority Over-sampling Technique (SMOTE) [78], which is an oversampling methodology that can generate new samples belonging to the minority class (samples that were found with non-technical losses).

5.3 Dissemination

Besides the published work in two journals, my work has been disseminated at the following events, throughout the thesis:

- **Research Seminar in Smart Grid technology at Royal Society of London (October 2016)** - I had a poster presentation during this event and gave a brief talk describing my research. The title of the poster was "Data analytics for non-technical losses detection in power utilities".
- **Workshop on Smart Distribution Networks: Technologies and Business models, organized by SUNSEED, ADVANTAGE, PEER-2-PEER (April 2017)** - Gave a talk on "Smart meter data analytics for fraud/anomaly detection" where I have described the advances made in building non-technical losses detection models in the real environment, by collaborating with an electricity utility.
- **IEEE ICC 2017 Conference, Workshop on Integrating Communications, Control, and Computing Technologies for Smart Grid (May 2017)** - Poster presentation on "Data analytics for non-technical losses detection in power utilities".
- **12th IEEE PES PowerTech Conference (June 2017)** - Gave a presentation on “Non-Technical Losses Detection in Power Systems using Smart Meters Data”, within a special session held by the ADVANTAGE project.

- **Data Science Summer School, co-organised by the Data Science Initiative of École Polytechnique and DATAIA Institute (June 2018)** - Participated in the poster session held by this summer school. The title of my poster was "Machine learning algorithms for non-technical losses detection in electricity utilities".

Appendix A

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	$\approx$ 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

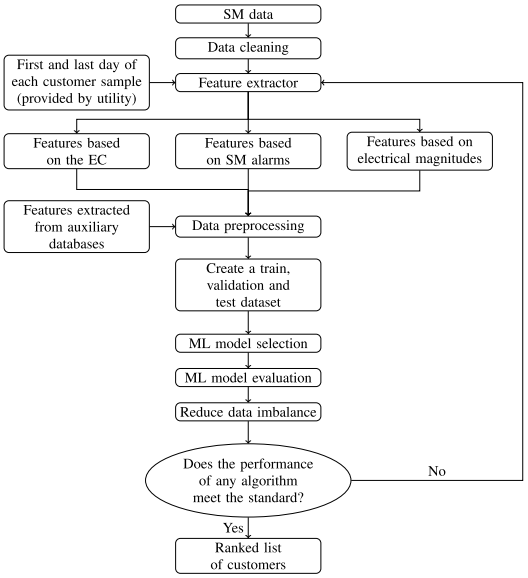


Fig. 1. Methodology outline for NTL detection.

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 measurements/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

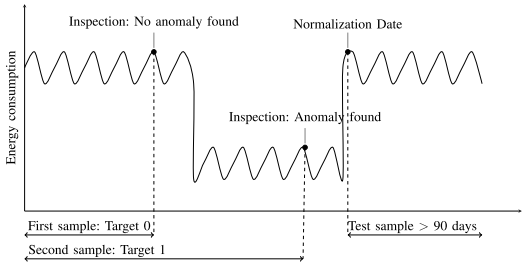


Fig. 2. Scenario of a customer with multiple training samples.

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

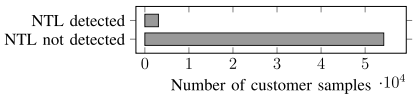


Fig. 3. Target distribution.

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

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

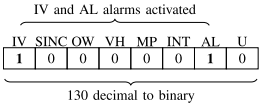


Fig. 4. Example of a QB measurement.

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.

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.

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

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

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:

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

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

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

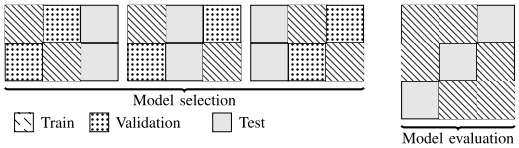


Fig. 5. 3-Fold Nested Cross-Validation example.

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.

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

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

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.

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

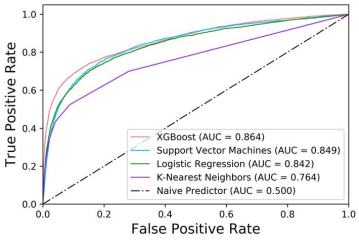


Fig. 6. Receiver Operating Characteristic curve.

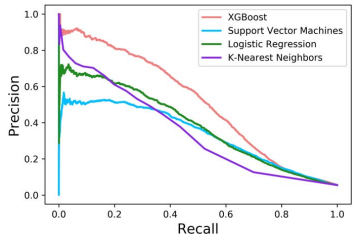


Fig. 7. Precision-Recall curves.

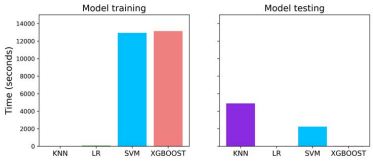


Fig. 8. Execution time.

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

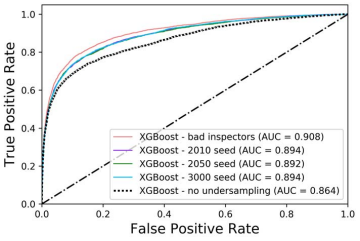


Fig. 9. ROC curve undersampling vs. no undersampling.

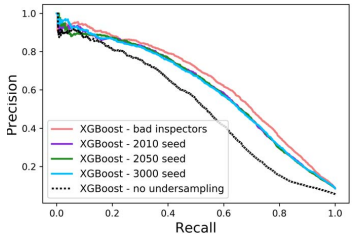


Fig. 10. Precision-Recall curves for undersampling.

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

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 $\approx$ 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.

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**

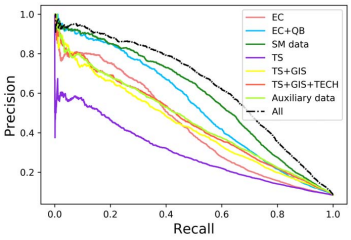


Fig. 11. Precision-Recall curves for different subsets of features.

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 $\approx$ 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.

Appendix B

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2943115, IEEE Transactions on Power Systems

Hybrid deep neural networks for detection of non-technical losses in electricity smart meters

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

***Abstract*—Non-technical losses in electricity utilities are responsible for major revenue losses. In this paper, we propose a novel end-to-end solution to self-learn the features for detecting anomalies and frauds in smart meters using a hybrid deep neural network. The network is fed with simple raw data, removing the need of handcrafted feature engineering. The proposed architecture consists of a long short-term memory network and a multi-layer perceptrons network. The first network analyses the raw daily energy consumption history whilst the second one integrates non-sequential data such as its contracted power or geographical information. The results show that the hybrid neural network significantly outperforms state-of-the-art classifiers as well as previous deep learning models used in non-technical losses detection. The model has been trained and tested with real smart meter data of Endesa, the largest electricity utility in Spain.**

***Index Terms*—Supervised learning, hybrid neural networks, non-technical losses, smart meter data.**

I. Introduction

NON-technical losses (NTL) in electricity utilities are defined as the energy consumed by the clients that has not been billed by the utility [1]. These losses, also known as commercial losses, can be caused either by theft, faults in the meters or billing irregularities [2], [3]. Regardless of their source, they are accountable for important revenue losses and have a negative impact on the grid reliability [4].

Worldwide, a recent report estimated that NTL are responsible for yearly revenue losses of $96 billion [5]. At the grid level, NTL can affect the power system operation by overloading transformers and causing voltage unbalances [4], [6]. Thus, reducing NTL will also reduce the physical losses of the grid [7]. Moreover, the costs of on-field inspections made to recover these losses, debt collection and even court costs in some cases, have to be taken into consideration as well. The low effectiveness of these inspections can further increase the NTL cost. Hence, it is extremely important for the utilities to advance in this field and increase the success rate of future on-field inspections for an efficient revenue loss recovery.

NTL detection is a complex anomaly detection task. Relying solely on outlier detection methods (e.g. k-means clustering, local outlier factor) is neither sufficient nor reliable for accurate predictions. Energy consumption (EC) patterns can change due to multiple factors and only a few of those changes are due to NTL. The main challenge of the algorithm is to recognize these NTL patterns among all of them.

In this paper, we propose an end-to-end solution for NTL detection using a hybrid neural network, requiring minimal input data and no domain knowledge of the problem. The architecture proposed in this paper outperforms state-of-the-art classifiers as well as previous deep learning approaches. The daily EC profile is analyzed through a long short-term memory network (LSTM) whilst the non-sequential data are passed through a multi-layer perceptrons (MLP) network. Throughout this paper, we refer to the EC, as the energy consumption that has been recorded by the meter. By combining the LSTM and MLP networks, major improvements in the performance have been obtained. The model has been trained and tested on real smart meter (SM) data of the largest electricity utility in Spain, Endesa.

A. Related work

The current NTL detection algorithms fall into three categories: data-oriented, network-oriented and a mix between both, hybrid-oriented [8]. Network-oriented approaches such as the methodology described in [9], use Distribution State Estimation (DSE) algorithms to search for irregular cases of EC. Hybrid-oriented methodologies use DSE algorithms combined with statistics or machine learning algorithms. A hybrid approach has been proposed in [10], where the authors use DSE and analysis of variance (ANOVA) for NTL detection. The DSE is used to detect distribution transformers with anomalous usage using the normalized residual test. After identifying transformers with anomalous usage, the NTL is detected at the customer level using ANOVA. Another hybrid NTL methodology is proposed in [11], which uses DSE and an optimum-path forest (OPF) classifier. The DSE is used to estimate the NTL in each month at the bus level. These NTL estimations are added to the input of the OPF classifier which was trained in a supervised manner, on a synthetic dataset. Network and hybrid approaches can detect NTL cases with a high precision but they cannot be widely implemented by most utilities as they require knowledge of the network topology and parameters, as well as the installation of additional metering devices.

Data-oriented approaches focus only on the data provided by the SMs, requiring no knowledge of the network topology and no additional hardware devices. The solution proposed in this paper is a data-driven approach based on supervised learning, using data from previous on-field NTL inspections.

Madalina-Mihaela Buzau, Pedro Cruz-Romero and Antonio Gómez-Expósito are with the Department of Electrical Engineering, University of Seville, Spain. Javier Tejedor Aguilera is with Endesa-Enel, Seville. Corresponding author: Madalina-Mihaela Buzau (email: madbuz@alum.us.es).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2943115, IEEE Transactions on Power Systems

Previous NTL detection methods that fall into this category [2], [12], [13], [14], [15], apart from being very expensive (they require the time-consuming engagement of experts), require specific human expert knowledge far from being complete in practical terms. Moreover, though the SMs roll-out improves the performance in NTL detection, they also introduce new methods for energy theft [16]. This makes it more difficult for detection algorithms based on expert knowledge to adapt to new types of frauds. Therefore, the problem of NTL detection is still not solved satisfactorily.

MLP networks have been previously implemented in NTL detection methodologies, but the majority of them have not been used with the purpose of replacing expert knowledge. As an example, the methodology described in [17] uses a MLP network to estimate the hyperparameters of a support vector machines (SVM) classifier, in order to maximize the accuracy of its predictions for NTL detection. The authors in [18] used statistical techniques and unsupervised neural networks (Kohonen networks), as two separate methodologies, for NTL detection. The Kohonen networks were used to cluster customers based on their EC pattern. In [19], the authors used text mining techniques to extract concepts from inspectors' comments of previous on-field inspections. After manually labeling 1% of the extracted concepts, a MLP network was used for training and for classifying the rest of the concepts. MLP networks have been used as end-to-end models for NTL detection in [20] and [21].

New methodologies based on deep learning have been proposed recently, such as the work described in [22], where the authors compared the performance of a convolutional neural network (CNN), a stacked autoencoder and a LSTM for the task of NTL detection. The results showed that the CNN outperformed the rest of the classifiers, on a synthetic dataset. Another deep learning methodology has been proposed in [23], where a CNN and a MLP network are used on raw EC data. However, CNN and MLP networks cannot work with sequential data such as the EC history, limiting the input to a fixed size window for all samples. For example, if the input is limited to EC data of the previous year, the model cannot capture a descent in the EC if it happened before the period of analysis. Furthermore, as we will show further, simply analyzing the EC on a real dataset, with real NTL samples, does not yield optimal results due to the nature of this problem.

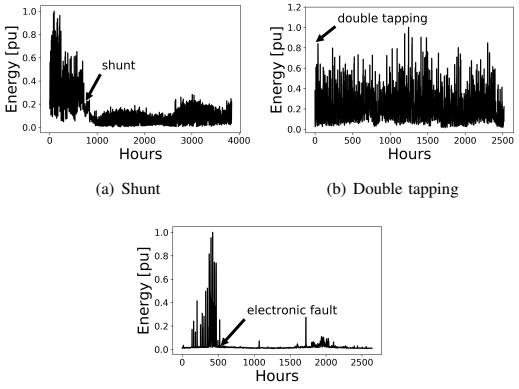
The main contributions of the paper are as follows:

- We propose a state-of-the-art methodology for NTL detection that can self-learn features that are relevant for NTL detection. This methodology can integrate both sequential and non-sequential data. To our knowledge, this is the first deep learning architecture for NTL detection that is able to accommodate both types of data.
- We investigate the boost in performance obtained by combining both types of data and show that the hybrid network significantly outperforms a network that uses only EC data as an input.
- We show that the proposed architecture vastly outperforms previous NTL detection models that were based on deep learning and raw data. We also compare its performance with state-of-the-art classifiers and their learning capabilities with raw SM data.

Contrary to [23] and [22], in this paper the precision of new on-field inspections guided by our proposed model is reported. The precision is the fraction of SMs that were actually identified with NTL, among the total number of SMs that have been inspected using the ranked list of customers provided by our model. This methodology is currently used as an NTL detection tool in the utility, achieving a precision of $\approx$ 47 % for new on-field inspections generated by our model. This represents a 3.5 times improvement over the precision of previous on-field inspections that have been generated through other tools and campaigns.

II. Motivation

Several types of fraudulent customer-related NTL are reported by utilities throughout the world [1]. Figure 1 shows the EC history of specific Spanish SMs found with NTL in their meters. Figure 1.a shows the EC profile of a SM that was directly tampered with a shunt device between the input and output terminals of the SM to divert the current. An example of a double tapping fraud is shown in Figure 1.b. Double tapping is a typical fraud case, where part of the consumption is connected directly to the grid, bypassing the SM. In this particular case, the fraud occured from the beginning of the contract, so a descent in the consumption cannot be observed. Figure 1.c shows the case of a SM found with an electronic fault, a common cause of anomaly in SMs.



(a) Shunt (b) Double tapping

(c) Electronic fault

Fig. 1. Real NTL cases (Source: Endesa)

Though in all NTL cases the meter reports lower EC readings, the change in the consumption profile is manifested differently depending on the NTL source. Traditional approaches aim to capture the behavior of different types of NTL using handcrafted feature engineering, as there is no mathematical formulation for the EC pattern of a SM with a shunt or double tapping. With this approach, a set of features that aim to characterize each type of anomaly or fraud is created. For example, features that detect a sudden drop in consumption are aimed to detect cases of shunts. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2943115, IEEE Transactions on Power Systems

the case of electronic faults, features that monitor the number of zero measurements or number of missing measurements are employed. Unfortunately, these approaches rely heavily on expert knowledge which is very expensive and time-consuming. Moreover, they require experts to develop continuously new features in order to adapt to new types of NTL. To mitigate the constraints of previous approaches, we propose a deep learning architecture that is able to self-learn these features from raw EC measurements and that can adapt automatically to new NTL behavior in the SM data.

III. Methodology

Figure 2 shows the methodology proposed in this paper for NTL detection. It has been developed and tested on a real dataset of an electricity utility. This methodology needs as an input three types of data: EC history recorded by the SMs, auxiliary (geographical, contractual, technical and economic) data and the results of previous on-field inspections along with their dates. The SM data are used to create the LSTM input whilst the auxiliary data are used for the MLP input.

The first step is to create customer samples. This methodology is the same as in [15]. By using the results of previous on-field inspections, the first and last day of each customer sample can be extracted following the rules presented in Figure 3. For customers who never had an inspection their sample represents their entire consumption history. The minimum length for a customer sample has been set to 365 days, as the traditional classifiers used in our comparison need at least an year of EC measurements. In practice, our HNN-NTL model can detect NTL on samples as short as 90 days. This is a major advantage as it significantly reduces the delay in detection. The processed dataset is split further based on whether a customer sample has been previously inspected or not. This creates a labeled and unlabeled dataset. The unlabeled dataset consists of samples belonging to customers who were never inspected or whose normalization date was more than 365 days ago. The labeled dataset is used to train the model in a supervised manner, using the results of previous on-field inspections. This dataset is split further into a training, validation and test dataset in order to assess the performance of the model on samples that have not been seen during training.

Data processing techniques have been used on all datasets. A thorough description of these techniques can be found in Section V-B. During model selection, the best hyperparameters of the model are selected using the validation dataset. The final performance is assessed on the test dataset. If the model performance on the test dataset is acceptable for the utility, the trained model is used to make predictions on the unlabeled dataset, creating a ranked list of customers to be inspected based on their probabilities of having NTL in their SM.

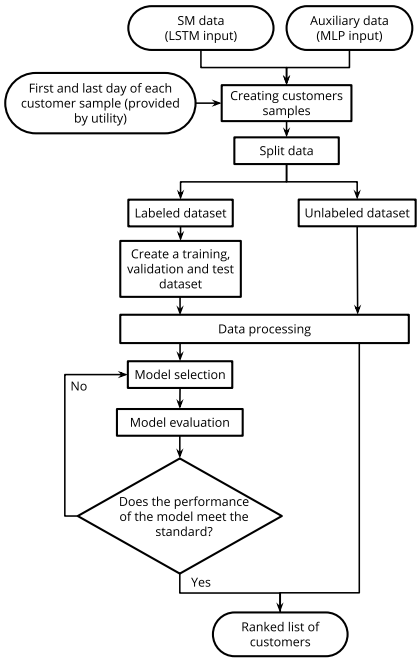


Fig. 2. Methodology outline for NTL detection

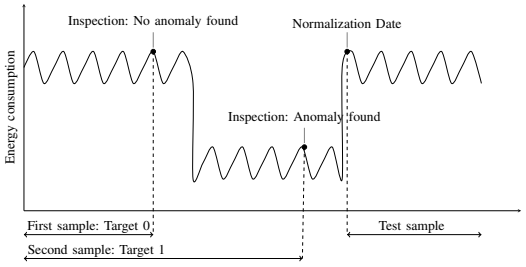


Fig. 3. Scenario of a customer with multiple training samples.

A. Model architecture of the hybrid neural network

The model proposed in this paper is a hybrid neural network (HNN-NTL), capable of integrating both sequential and non-sequential SM information (Figure 4). The network consists of three modules: the LSTM module, the MLP module and the hybrid module. The LSTM module uses the sequential data of the SM (e.g. EC), whilst the MLP module uses the non-sequential data as as an input (e.g. SM location, model). The outputs of the LSTM module and the MLP module are afterwards used as an input to the hybrid module which provides the final probability of having an NTL in the SM. This type of architecture is very efficient as it permits joint training on both types of input. A detailed description of each module is presented in the following subsections.

B. Long short-term memory module for sequential data

Table I shows the input features that get fed into the LSTM model at each time step. Though the SM records the hourly EC, we have reduced the granularity of measurements to the daily level as we have seen through our own experiments that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2943115, IEEE Transactions on Power Systems

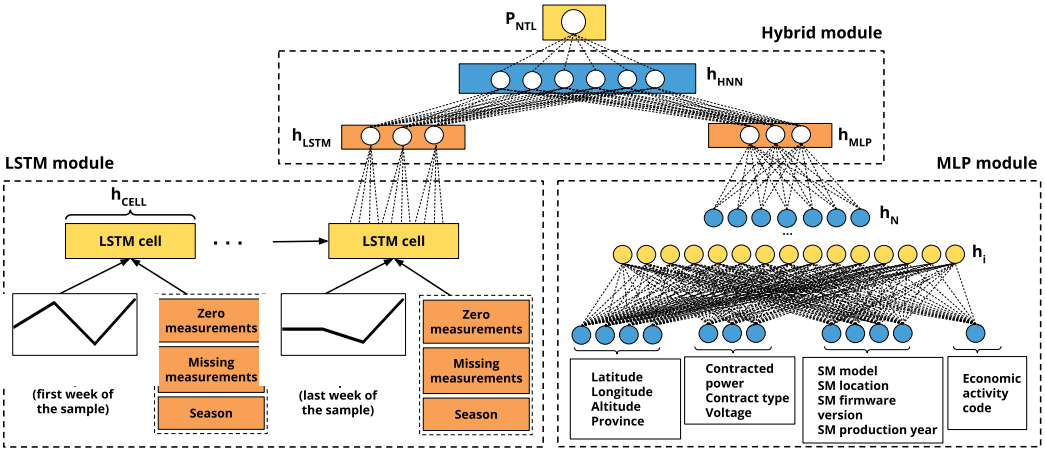


Fig. 4. HNN-NTL model architecture

it yields better results than hourly measurements. The daily EC has been obtained by simply averaging the hourly ECs in a day. The weekend ECs were removed using the time stamp of each SM measurement (weekend and weekday data are not comparable). Thus, the weekly profile consisted of 5 measurements of the daily average consumption at each time step.

TABLE I
LSTM INPUT AT EACH TIME STEP

Input	Description	Size
Weekly profile	Daily energy consumption of the weekdays.	5
Zero measurements	Number of 0 kWh measurements in each weekday.	5
Missing measurements	Number of null measurements in each weekday.	5
Season	The season of the week (spring, summer, autumn, winter).	4

Since the EC history recorded by the SMs can be years long and it is increasing day by day, a simple recurrent neural network [24] cannot be used as it would be very hard to train due to their vanishing and exploding gradient problems [25]. Thus, to capture the long-term dependencies in the variable-sized EC data an LSTM cell has been used [26]. The LSTM cell uses the sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$ and the hyperbolic tangent $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ as nonlinear activations and it has the following mathematical formulation:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t \odot \tanh(C_t) \quad (5)$$

where i_t , f_t, o_t, C_t and h_t represent the activations of the input gate, forget gate, output gate, cell state and hidden state at time step t. W_i, W_f, W_o and W_c represent the weights of the input layer whilst U_i, U_f, U_o and U_c represent the recurrent weights of the LSTM. b_i, b_f, b_o, b_c are the biases of the network whilst x_t is the input feature vector at time step t and h_{t-1} represents the hidden state activation at the previous time step. $\odot$ represents the element-wise multiplication (Hadamard product).

The LSTM module sees the entire EC history, week by week, and it provides a single final output, h_T, which is the hidden state of the LSTM cell at the final time step (last week of the sample). We decided to use the weekly profile as input to the LSTM, rather than the daily consumption, in order to reduce the number of time steps through the network. By reducing the number of time steps, the network is able to learn faster. Using the weekly profile gives the network the opportunity to detect more complex types of NTL, such as intermittent NTL, which occurs only on some certain days of the week. These cases would be much more difficult to detect when using only the monthly EC as an input.

C. Multi-layer perceptrons module for non-sequential data

The MLP network is used to analyze the non-sequential data. The metadata that has been used in this module can be found in Table II. Input features that are continuous have a size of 1 whilst categorical features have a higher dimension. We used entity embeddings [27] for the categorical variables, instead of one-hot-encoding, in order to reduce the input space of the MLP network. For example, only the economic activity code has 473 unique categories.

The MLP module has N hidden layers, where N is chosen using the validation dataset. Each hidden layer goes through an affine transformation ($n > 0$):

$$z_n = W_n h_{n-1} + b_n \quad (6)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2943115, IEEE Transactions on Power Systems

TABLE II
MLP INPUT DATA

Type of data	Input	Size
Geographical data	Latitude	1
	Longitude	1
	Altitude	1
	Municipality	5
Contractual data	Contracted power	1
	Contract type	2
	Voltage	1
SM technical data	SM model	3
	SM location	3
	SM firmware version	3
	SM production year	3
Economic data	Economic activity code	10

where W_n represents the weights of layer n, h_{n-1} represents the hidden state of the previous layer and b_n represents the bias of the n_{th} layer.

To speed up the convergence of the network, a batch normalization layer [28] has been used on the affine transformation:

$$b_n = \gamma \hat{z}_n + \beta \tag{7}$$

where $\hat{z}_n$ represents the standardized affine activation with the mean and standard deviation of the batch sample and γ and β are trainable parameters that are tuned during optimization.

The final state of the hidden layer is obtained by using a rectified linear unit:

$$h_n = \max(0, b_n) \tag{8}$$

D. Hybrid module

The hidden state h_{LSTM} of the hybrid module has been obtained using as an input the hidden state of the LSTM cell h_T at the final time step (T is the sequence length). Similarly, the hidden state h_{MLP} has been computed using as an input the hidden state activations of the last hidden layer in the MLP module h_N. Both h_{LSTM} and h_{MLP} have been computed using transformations described in the equations (6), (7) and (8).

The hybrid module simply takes afterwards the hidden states h_{LSTM} and h_{MLP} and concatenates them in order to form a new hidden layer.

The final state of the HNN-NTL model is obtained as follows:

$$h_{HNN} = \max(0, \gamma \hat{z}_{HNN} + \beta) \tag{9}$$

where $\hat{z}_{HNN} = W_{HNN}[h_{LSTM}, h_{MLP}] + b_{HNN}$ and γ and β are trainable parameters of the model.

The outcome of the network is computed using the sigmoid activation, providing a score between 0 and 1. This score can be interpreted as the probability that there is an NTL in the SM, though its confidence strength depends on the strength of regularization [29]:

$$P_{NTL} = \frac{1}{1 + e^{-(W_{NTL} h_{HNN} + b_{NTL})}} \tag{10}$$

where P_{NTL} represents the probability that there is an NTL in the SM. W_{NTL} and b_{NTL} represent the trainable weights and bias of the output layer.

IV. LEARNING AND EVALUATION

A. Loss function and optimization

The performance of the model has been evaluated with the logarithmic loss function, as this is a binary classification task:

$$L = \frac{1}{M} \sum_{i=1}^{M} -(y_i \log(P^i_{NTL}) + (1 - y_i) \log(1 - P^i_{NTL})) \tag{11}$$

where M is the number of customer samples, y_i is the ground-truth label and P^i_{NTL} is the probability of NTL computed by the HNN-NTL model for the customer sample i.

The trainable parameters of the model have been initialized with a Xavier initialization [30] and optimized to minimize the loss function using the Adam optimizer [31], a first-order gradient-based optimization method.

B. Metric for evaluation

One of the main challenges of tackling the NTL detection problem with a supervised approach is the imbalance between classes. A suitable metric for NTL detection is the area under the receiver operating characteristic curve (ROC-AUC) [32]. The ROC curve is obtained by plotting the true positive rate (TPR), also known as recall, against the false positive rate (FPR) by varying the decision threshold on the predictions. The score ranges between 0 and 1, a score above 0.5 being obtained with better than random predictions. Though the TPR and FPR are valuable metrics when assessing the performance of a model for NTL detection, they do not account for the precision of the model. A metric suitable for imbalanced datasets that takes into account the precision of the model is the area under the precision-recall curve (PR-AUC) [33], [34]. We therefore decided to use the PR-AUC metric for the model selection on the validation dataset, as taking into account the success rate of the on-field inspections is extremely important for the utilities, for economic reasons (a limited budget for on-field inspections leads to a limited number of inspections per year, for which the success rate is required to be maximized).

V. EXPERIMENTS AND RESULTS

A. Data availability

All the models have been trained and tested on real SM data of Endesa, the largest electricity utility in Spain. The data provided by the utility have been anonymized and a noise of approximately 5 km has been added to the geographical coordinates. All samples in the dataset have a contracted power below 15 kW.

The labeled dataset that is used for training consists of SMs which had at least one on-field inspection. To analyze the capability of the model to generalize beyond its training dataset, the original data have been split into a training, validation and test dataset. The split has been done in a stratified manner, so that there is the same % of NTL samples in each dataset. The training dataset consists of 80 % of the labeled dataset, whilst the validation and test datasets consist of $\approx$ 10 % each. Table III shows the number of samples in each dataset as well as their % of NTL samples.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2943115, IEEE Transactions on Power Systems

TABLE III
LABELED DATASET

Dataset type	Number of samples	% of NTL samples
Training	85226	13.34%
Validation	10612	13.50%
Test	10701	13.23%

As can be seen in Table III, the dataset is highly imbalanced which can make the model biased to probabilities closer to 0. However, this doesn't have a strong negative impact on our model as the metrics employed assess the ranking of samples rather than the confidence strength of their probabilities.

B. Data processing

The weekly profile, zero measurements and missing measurements have been normalized separately, using their maximum value:

$$f(x) = \frac{x}{\max(x)} \tag{12}$$

The season did not require any normalization, as it was one-hot-encoded.

In the case of non-sequential features, the missing values in non-categorical features were replaced with the mean. For categorical features, a special "Unknown" category has been created to replace missing data. After imputing the missing data, the non-sequential features were standardized to have 0 mean and unit variance using the following formula:

$$f(x) = \frac{x - \bar{x}}{s} \tag{13}$$

where $\bar{x}$ represents the mean of the input feature and s represents the standard deviation.

C. Implementation

Given the large size of the dataset, all the data exploration and processing have been done with open source software PySpark [35], taking advantage of distributed computing using a cluster of machines. All neural networks were built and trained using TensorFlow [36], an open-source deep learning framework. For the comparison with state-of-the-art ML algorithms, the Scikit-learn [37] library has been used to fit the models. For extreme gradient boosted trees [38], the model has been fitted using its Python API.

D. Experiment hyperparameters

The performance of a machine learning algorithm depends strongly on its hyperparameters. As it was seen in Figure 4, the size of the HNN-NTL network can be adjusted by controlling the size of various hidden layers such as h_{LSTM} and h_{HNN}. A grid-search has been implemented in order to find the best hyperparameters. Table IV shows the range of values searched as well as the optimal value found. The optimal value was found by monitoring the performance of the validation dataset. As a regularization method, a dropout layer has been used on the output of the h_{LSTM}, h_{MLP}, h_{HNN} and every h_i layer of the MLP module. No regularization has been used on the LSTM cell h_{CELL}, as it did not improve the performance of the model.

TABLE IV
HNN-NTL HYPERPARAMETERS SEARCH

Hyperparameter	Range of values	Optimal value
N	4, 6	4
Size h_i	256, 512	256
Size h_{CELL}	256, 512	256
Size h_{LSTM}	256, 512	512
Size h_{MLP}	256, 512	512
Size h_{HNN}	1024, 2048	1024
Dropout	0.3, 0.5	0.3

E. LSTM module results

In order to assess the performance in particular of the LSTM module (only sequential data), an experiment was performed with a simplified network where the MLP module and the h_{HNN} layer of Figure 4 were omitted.

1) LSTM with the weekly profile: Figure 5 shows the performance of the LSTM model when using as input only the weekly profile. As can be seen in Figures 6 and 7, a PR-AUC of 0.33 and a ROC-AUC score of 0.72 have been obtained on the test dataset. Even with such simple input, the model significantly outperforms random predictions.

2) LSTM with all data: Figures 8, 9 and 10 show the model performance of the LSTM model when using all input data (weekly profile, zero measurements, missing measurements and season). By using this additional data, the PR-AUC has increased from 0.33 to 0.41 on the test dataset.

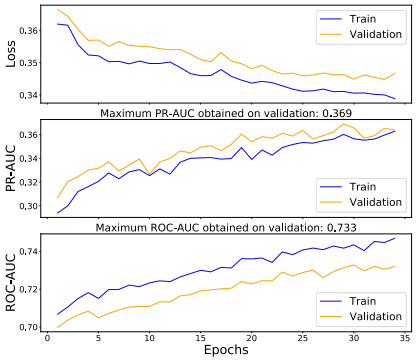


Fig. 5. Simple LSTM model performance during training

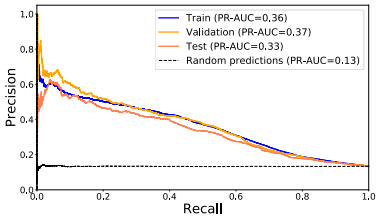


Fig. 6. PR curve of simple LSTM model using the best trained model

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2943115, IEEE Transactions on Power Systems

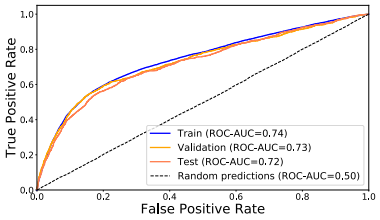


Fig. 7. ROC curve of simple LSTM model using the best trained model

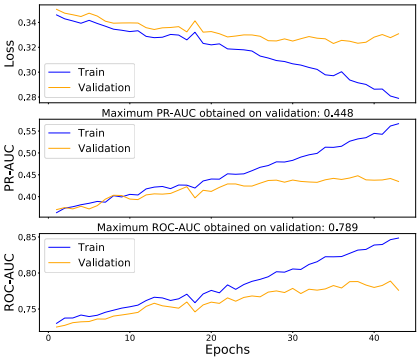


Fig. 8. LSTM model performance during training

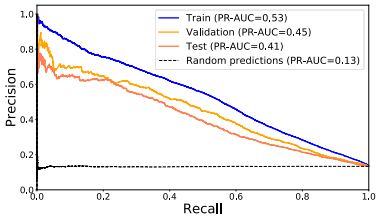


Fig. 9. PR curve of LSTM model using the best trained model

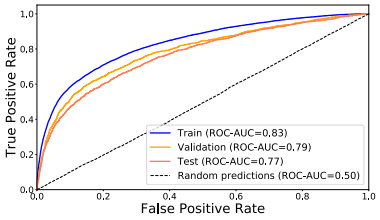


Fig. 10. ROC curve of LSTM model using the best trained model

F. HNN-NTL model results

The performance of the HNN-NTL model can be seen in Figures 11, 12 and 13. As can be seen, the HNN-NTL model greatly outperforms the LSTM model, obtaining a PR-AUC score of 0.54 on the test dataset. As expected, using non-sequential features such as the contracted power or the SM model dramatically improves the performance for NTL detection.

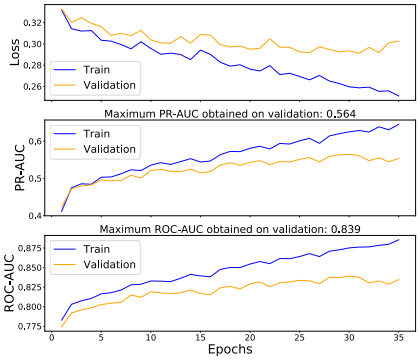


Fig. 11. HNN-NTL model performance during training

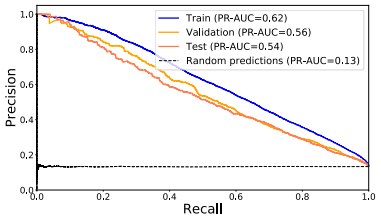


Fig. 12. PR curve of HNN-NTL model using the best trained model

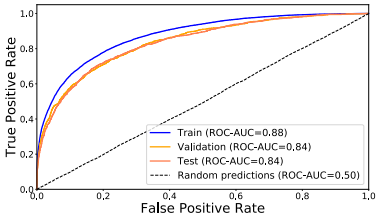


Fig. 13. ROC curve of HNN-NTL model using the best trained model

G. Comparison

In this section, we will compare the performance of the HNN-NTL model with state-of-the-art classifiers as well as previous deep learning approaches. As all the models described below require a fixed size input, we have used a fixed size window of one year on the sequential data.

1) Support Vector Machines: The support vector machine (SVM) is a very popular classifier for NTL detection. The objective of this algorithm is to find an optimal hyperplane that maximizes the margin between the vectors of the distinct classes [39].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2943115, IEEE Transactions on Power Systems

Table V shows the range of hyperparameters used during grid-search as well as their optimal value found on the validation dataset. A linear kernel has been used due to the size of the dataset.

TABLE V
SVM HYPERPARAMETERS SEARCH

Hyperparameter	Range of values	Optimal value
C	0.001, 0.01, 10, 100	0.001

2) Logistic Regression: Logistic regression (LR) is a popular supervised algorithm for binary classification tasks, that uses the same principles as neural networks. To obtain the probability of NTL for a sample, the algorithm multiplies the input features with a matrix of trained weights and passes the output through the sigmoid function [40]. The optimal weights were found using the logarithmic loss function and the LIBLINEAR solver [41]. Table VI shows the hyperparameters used during grid-search for LR.

TABLE VI
LR HYPERPARAMETERS SEARCH

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

3) Random Forests: Random forests (RF) fall into the category of ensemble models [42]. The algorithm combines several decision trees (DT) to create a collection of trees that can make more accurate predictions. The mode of the predictions of individual trees is used in order to output a final decision. Table VII shows the hyperparameters used during grid-search for RF.

TABLE VII
RF HYPERPARAMETERS SEARCH

Hyperparameter	Range of values	Optimal value
Number of trees	1000, 2000	1000
Minimum samples split	5, 10, 15	10
Maximum depth	7, 15	15
Minimum samples leaf	5, 10, 15	15

4) Extreme Gradient Boosted Trees: Extreme gradient boosted trees (XGB) is a very popular algorithm in the data science community, winning many competitions on the data science platform Kaggle [38]. It has already been used successfully for NTL detection [15]. The algorithm uses gradient boosting [43] with a regularized loss function which makes the model less prone to overfitting and increases its generalization capabilities on new samples. Gradient boosted learning is a very powerful machine learning technique which combines several DT to create a collection of trees that can make more accurate predictions. Table VIII shows the hyperparameters used during grid-search as well as the optimal value found.

5) Multi-Layer Perceptrons Networks: MLP networks have already been used successfully in NTL detection [20], [21]. In this comparison, we use the same architecture and hyperparameters of the MLP module from our HNN-NTL model. This MLP network has in addition input features extracted from the raw EC data.

TABLE VIII
XGB HYPERPARAMETERS SEARCH

Hyperparameter	Range of values	Optimal value
Number of trees	1000, 2000	1000
Learning rate	0.01, 0.1	0.01
Maximum depth	7, 15	7
Minimum child weight	1, 5, 10	10

6) Convolutional Neural Networks: CNN have been shown to outperform stacked autoencoders and LSTM networks, in [22], on a dataset with synthetic NTL samples. We used the same architecture proposed in [22], as well as the same hyperparameters that have been used in their experiment. Furthermore, for the sake of completeness, the MLP module has been added to the CNN architecture so that the network has access to the same information as the rest of the models. The output of the MLP module was simply concatenated with the output of the CNN module before making the final predictions. The original experiment used monthly EC data as an input to the CNN network but we have increased the granularity to daily EC measurements given our data availability.

7) Wide & Deep Convolutional Neural Networks: The wide & deep convolutional neural network (WD-CNN) is a deep learning architecture for NTL detection proposed by the authors in [23]. The algorithm uses a wide network (equivalent to a MLP network) on the 1D daily EC data and a CNN on the 2D stacked weekly energy profiles. We used the same hyperparameters that were used in their experiments, therefore a grid-search has not been performed for this model. The special convolution kernel has also been implemented using a hyperbolic tangent activation function. Similarly to the CNN-MLP experiment, the MLP module has also been added as a separate component within the architecture so that the algorithm has access to the available auxiliary data.

8) Results: This section shows the results obtained for the comparison of the HNN-NTL algorithm described above with other state-of-the-art classifiers. The same training, validation and test datasets described in Table III have been used for all the models in this comparison. The input that was used for the SVM, LR, RF, XGB and MLP models is shown in Table IX. As mentioned previously, these algorithms require a fixed size input, thus we have retrieved from each customer sample the daily EC of the past year. The auxiliary data input is equivalent to the same input used in the MLP module of the HNN-NTL model. The same entity embeddings have been used for the categorical features. The EC input consists of the same information that has been used in the LSTM module, but restricted to the EC of the previous year.

The CNN-MLP network uses as input the 1D daily EC data for the CNN module and auxiliary data for the MLP module, as can be seen in Table X.

The input for the WD-CNN algorithm can be found in Table XI. For the wide module, the input is simply the daily energy consumption of the past year. To create the input for the CNN module, the daily EC consumption has been divided

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2943115, IEEE Transactions on Power Systems

TABLE IX
SVM, LR, RF, XGB AND MLP INPUTS

Input	Description	Size
Auxiliary data	Latitude	1
	Longitude	1
	Altitude	1
	Municipality	5
	Contracted power	1
	Contract type	2
	Voltage	1
	SM model	3
	SM location	3
	SM firmware version	3
	SM production year	3
	Economic activity code	10
EC input	Daily EC consumption	260
	Daily zero measurements	260
	Daily missing measurements	260

TABLE X
CNN-MLP INPUT

Input	Description	Size
CNN input	1D daily EC consumption of last year	260
MLP input	See Table II	34

into weekly profiles and stacked into a 2D array. The weekly profiles were stacked starting with the first week of the year up until the last.

TABLE XI
WD-CNN INPUT

Input	Description	Size
Wide input	Daily EC consumption of last year	260
CNN input	Weekly EC profiles of last year	260
MLP input	See Table II	34

Table XII shows the results of the comparison, for various sizes of the training dataset. Deep learning models are known to be sensitive to the size of the dataset used during training. Therefore, this analysis shows whether the HNN-NTL model maintains its superiority when less training samples are available. As can been seen, the HNN-NTL model vastly outperforms any other algorithm, for both PR-AUC and ROC-AUC metrics and for all sizes of the training dataset.

TABLE XII
FINAL PR-AUC AND ROC-AUC RESULTS ON THE TEST DATASET

	Training = 40%		Training = 60%		Training = 80%	
Methods	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
Random	0.5	0.133	0.5	0.133	0.5	0.133
SVM	0.710	0.277	0.714	0.281	0.716	0.284
LR	0.715	0.282	0.718	0.285	0.719	0.285
RF	0.744	0.331	0.751	0.340	0.753	0.345
XGB	0.767	0.377	0.776	0.391	0.777	0.394
MLP	0.685	0.246	0.723	0.263	0.738	0.314
CNN-MLP	0.748	0.328	0.755	0.321	0.756	0.327
WD-CNN	0.768	0.381	0.770	0.385	0.774	0.397
HNN-NTL	**0.822**	**0.520**	**0.813**	**0.499**	**0.836**	**0.545**

For traditional classifiers such as SVM and RF the performance always increases with increased data availability. For the CNN-MLP and HNN-NTL models, the performance is either increasing or decreasing which suggests that a new grid-search on hyperparameters should be performed on each size of the training dataset for optimal results.

VI. CONCLUSION

In this paper, we have proposed a hybrid model for non-technical losses detection. To our knowledge, this is the first deep learning approach that is able to incorporate both sequential and non-sequential data. We have shown that by integrating auxiliary data, significant improvements in the performance of the model are achieved. The model has obtained a PR-AUC of 0.545 and a ROC-AUC score of 0.836 on the test dataset. The results show that it significantly outperforms previous deep learning approaches. Furthermore, the comparison with other state-of-the-art classifiers has shown that the proposed hybrid neural network model is able to surpass the performance of very powerful classifiers such as extreme gradient boosted trees. The methodology has been developed and tested with real smart meter data of Endesa, the largest electricity utility in Spain. It is currently used as a non-technical losses detection tool in the utility, obtaining a precision of $\approx$ 47 % for new on-field inspections generated by our ranked list of customers.

VII. ACKNOWLEDGEMENTS

The authors would like to thank Endesa for funding this research. They are also 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] International Conference on Electricity Distribution (CIRED), "Reduction of Technical and Non-Technical Losses in Distribution Networks," 2017. [Online]. Available: http://www.cired.net/files/download/188

[2] C. León, F. Biscarri, I. Monedero, J. I. Guerrero, J. Biscarri, and R. Millán, "Variability and trend-based generalized rule induction model to NTL detection in power companies," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 1798–1807, 2011.

[3] P. Glauner, J. A. Meira, P. Valtchev, R. State, and F. Bettinger, "The Challenge of Non-Technical Loss Detection Using Artificial Intelligence: A Survey," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 760–775, 2017.

[4] S. S. S. R. Depuru, L. Wang, and V. Devabhaktuni, "Electricity theft: Overview, issues, prevention and a smart meter based approach to control theft," *Energy Policy*, vol. 39, no. 2, pp. 1007–1015, 2011.

[5] Northeast Group LLC, "Electricity Theft and Non-Technical Losses: Global Markets, Solutions, and Vendors," 2017. [Online]. Available: http://www.northeast-group.com

[6] L. Arango, E. Deccache, B. D. Bonatto, H. Arango, P. Ribeiro, and P. M. Silveira, "Impact of electricity theft on power quality," in *2016 17th International Conference on Harmonics and Quality of Power (ICHQP)*. IEEE, 2016, pp. 557–562.

[7] A. Fragkioudaki, P. Cruz-Romero, A. Gómez-Expósito, J. Biscarri, M. J. de Tellechea, and A. Arcos, "Detection of Non-technical Losses in Smart Distribution Networks: A Review," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, Cham, 2016, pp. 43–54.

[8] G. M. Messinis and N. D. Hatziargyriou, "Review of non-technical loss detection methods," *Electric Power Systems Research*, vol. 158, pp. 250–266, 2018.

[9] Y. L. Lo, S. C. Huang, and C. N. Lu, "Non-technical loss detection using smart distribution network measurement data," in *IEEE PES Innovative Smart Grid Technologies*. IEEE, 2012, pp. 1–5.

[10] S. C. Huang, Y. L. Lo, and C. N. Lu, "Non-Technical Loss Detection Using State Estimation and Analysis of Variance," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2959–2966, 2013.

[11] R. D. Trevizan, A. Rossoni, A. S. Bretas, D. da Silva Gazzana, R. de Podestá Martin, N. G. Bretas, A. L. Bettiol, A. Carniato, and L. F. do Nascimento Passos, "Non-Technical Losses Identification Using Optimum-Path Forest and State Estimation," in *2015 IEEE Eindhoven PowerTech*. IEEE, 2015, pp. 1–6.

[12] A. H. Nizar, Z. Y. Dong, and Y. Wang, "Power utility nontechnical loss analysis with extreme learning machine method," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 946–955, 2008.

[13] N. F. Avila, G. Figueroa, and C. C. Chu, "NTL Detection in electric distribution systems using the maximal overlap discrete wavelet-packet transform and random undersampling boosting," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7171–7180, 2018.

[14] J. I. Guerrero, I. Monedero, F. Biscarri, J. Biscarri, R. Millan, and C. Leon, "Non-Technical Losses Reduction by Improving the Inspections Accuracy in a Power Utility," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1209–1218, 2018.

[15] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gomez-Exposito, "Detection of Non-Technical Losses Using Smart Meter Data and Supervised Learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2661–2670, 2019.

[16] T. J. Bihl and S. Hajjar, "Electricity theft concerns within advanced energy technologies," in *2017 IEEE National Aerospace and Electronics Conference (NAECON)*. IEEE, 2017, pp. 271–278.

[17] S. S. S. R. Depuru, L. Wang, V. Devabhaktuni, and P. Nelapati, "A hybrid neural network model and encoding technique for enhanced classification of energy consumption data," in *2011 IEEE Power and Energy Society General Meeting*. IEEE, 2011, pp. 1–8.

[18] I. Monedero, F. Biscarri, C. Léon, J. Biscarri, and R. Millán, "MIDAS: Detection of non-technical losses in electrical consumption using neural networks and statistical techniques," in *International Conference on Computational Science and Its Applications*. Springer, Berlin, Heidelberg, 2006, pp. 725–734.

[19] J. I. Guerrero, C. León, I. Monedero, F. Biscarri, and J. Biscarri, "Improving Knowledge-Based Systems with statistical techniques, text mining, and neural networks for non-technical loss detection," *Knowledge-Based Systems*, vol. 71, pp. 376–388, 2014.

[20] B. C. Costa, B. L. Alberto, A. M. Portela, W. Maduro, and E. O. Eler, "Fraud Detection in Electric Power Distribution Networks Using an ANN-Based Knowledge-Discovery Process," *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. 4, no. 6, pp. 17–23, 2013.

[21] L. A. Pereira, L. C. Afonso, J. P. Papa, Z. A. Vale, C. C. Ramos, D. S. Gastaldello, and A. N. Souza, "Multilayer perceptron neural networks training through charged system search and its application for non-technical losses detection," in *2013 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)*, 2013, pp. 1–6.

[22] R. R. Bhat, R. D. Trevizan, R. Sengupta, X. Li, and A. Bretas, "Identifying Nontechnical Power Loss via Spatial and Temporal Deep Learning," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 272–279.

[23] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2018.

[24] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[25] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[26] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] C. Guo and F. Berkhahn, "Entity Embeddings of Categorical Variables," *arXiv preprint arXiv:1604.06737*, 2016.

[28] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv preprint arXiv:1502.03167*, 2015.

[29] A. Karpathy, L. Fei-Fei, and J. Johnson, "CS231n: Convolutional Neural Networks for Visual Recognition," 2016. [Online]. Available: http://cs231n.stanford.edu/2016

[30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[31] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] P. O. Glauner, A. Boechat, L. Dolberg, R. State, F. Bettinger, Y. Rangoni, and D. Duarte, "Large-Scale Detection of Non-Technical Losses in Imbalanced Data Sets," in *2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2016, pp. 1–5.

[33] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[34] V. Raghavan, P. Bollmann, and G. S. Jung, "A critical investigation of recall and precision as measures of retrieval system performance," *ACM Transactions on Information Systems*, vol. 7, no. 3, pp. 205–229, 1989.

[35] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: cluster computing with working sets," in *2nd USENIX Workshop on Hot Topics in Cloud Computing*, vol. 10, 2010, p. 95.

[36] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, and G. Research, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv preprint arXiv:1603.04467*, 2016.

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

[38] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[39] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[40] A. Ng, "Machine Learning. Coursera." 2016. [Online]. Available: https://www.coursera.org/learn/machine-learning

[41] 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.

[42] L. Breiman and Leo, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[43] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

Madalina-Mihaela Buzau received her BEng in Power Systems from Politehnica University of Bucharest and her MRes in Electrical Engineering and Sustainable Development from Lille University of Science and Technology. She is now pursuing a Ph.D. degree in the Department of Electrical Engineering of the 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 (M06) received the Ph.D. degree in electrical engineering from the University of Seville, Spain, in 2000. Currently, he is 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.

List of Figures

List of Tables

Bibliography

[1] International Conference on Electricity Distribution (CIRED), "Reduction of Technical and Non-Technical Losses in Distribution Networks," 2017. [Online]. Available: http://www.cired.net/files/download/188

[2] Northeast Group LLC, "Electricity Theft and Non-Technical Losses: Global Markets, Solutions, and Vendors," 2017. [Online]. Available: http://www.northeast-group.com

[3] S. S. S. R. Depuru, L. Wang, and V. Devabhaktuni, "Electricity theft: Overview, issues, prevention and a smart meter based approach to control theft," *Energy Policy*, vol. 39, no. 2, pp. 1007–1015, 2011.

[4] L. Arango, E. Deccache, B. D. Bonatto, H. Arango, P. Ribeiro, and P. M. Silveira, "Impact of electricity theft on power quality," in *17th International Conference on Harmonics and Quality of Power (ICHQP)*. IEEE, 2016, pp. 557–562.

[5] R. Rashed Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on Advanced Metering Infrastructure," *International Journal of Electrical Power and Energy Systems*, vol. 63, pp. 473–484, 2014.

[6] E. Villar-Rodriguez, J. Del Ser, I. Oregi, M. N. Bilbao, and S. Gil-Lopez, "Detection of non-technical losses in smart meter data based on load curve profiling and time series analysis," *Energy*, vol. 137, pp. 118–128, 2017.

[7] European Commission, "Smart grids and meters," 2014. [Online]. Available: https://ec.europa.eu/energy/en/topics/market-and-consumers/smart-grids-and-meters

[8] F. C. L. Trindade, L. F. Ochoa, and W. Freitas, "Data analytics in smart distribution networks: Applications and challenges," in *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*. IEEE, 2016, pp. 574–579.

[9] M. Anas, N. Javaid, A. Mahmood, S. Raza, U. Qasim, and Z. Khan, "Minimizing Electricity Theft Using Smart Meters in AMI," in *Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. IEEE, 2012, pp. 176–182.

[10] S. S. S. R. Depuru, L. Wang, and V. Devabhaktuni, "Smart meters for power grid: Challenges, issues, advantages and status," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 6, pp. 2736–2742, 2011.

[11] S. McLaughlin, D. Podkuiko, and P. McDaniel, "Energy Theft in the Advanced Metering Infrastructure," in *International Workshop on Critical Information Infrastructures Security*. Springer, Berlin, Heidelberg, 2009, pp. 176–187.

[12] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka, A. Delozier, and P. McDaniel, "Multi-vendor penetration testing in the advanced metering infrastructure," in *Proceedings of the 26th Annual Computer Security Applications Conference*. ACM, 2010, pp. 107–116.

[13] S. McLaughlin, B. Holbert, A. Fawaz, R. Berthier, and S. Zonouz, "A Multi-Sensor Energy Theft Detection Framework for Advanced Metering Infrastructures," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1319–1330, 2013.

[14] T. J. Bihl and S. Hajjar, "Electricity theft concerns within advanced energy technologies," in *2017 IEEE National Aerospace and Electronics Conference (NAECON)*. IEEE, 2017, pp. 271–278.

[15] D. Warudkar, P. Chandel, and B. A. Sawale, "Anti-Tamper Features in Electronic Energy Meters," *International Journal of Electrical, Electronics and Data Communication*, vol. 2, no. 5, pp. 81–84, 2014.

[16] S. Yan, X. Liu, Y. Jiang, L. Ma, X. Wang, and Z. Li, "Analysis of Electricity Stealing and Research of Anti-stealing Measures," in *2015 3rd International Conference on Machinery, Materials and Information Technology Applications*. Atlantis Press, 2015, pp. 911–918.

[17] G. M. Messinis and N. D. Hatziargyriou, "Review of non-technical loss detection methods," *Electric Power Systems Research*, vol. 158, pp. 250–266, 2018.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2006.

[19] P. O. Glauner, A. Boechat, L. Dolberg, R. State, F. Bettinger, Y. Rangoni, and D. Duarte, "Large-Scale Detection of Non-Technical Losses in Imbalanced Data Sets," in *2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2016, pp. 1–5.

[20] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Detection of Non-Technical Losses Using Smart Meter Data and Supervised Learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2661–2670, 2019.

[21] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Hybrid deep neural networks for detection of non-technical losses in electricity smart meters," *IEEE Transactions on Power Systems*, 2019.

[22] R. V. Cruz, C. V. Quintero, and F. Pérez, "Detecting non-technical losses in radial distribution system transformation point through the real time state estimation method," in *2006 IEEE/PES Transmission & Distribution Conference and Exposition: Latin America*, 2006, pp. 1–5.

[23] P. Kadurek, J. Blom, J. F. Cobben, and W. L. Kling, "Theft detection and smart metering practices and expectations in the Netherlands," in *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*. IEEE, 2010, pp. 1–6.

[24] Y. L. Lo, S. C. Huang, and C. N. Lu, "Non-technical loss detection using smart distribution network measurement data," in *IEEE PES Innovative Smart Grid Technologies*. IEEE, 2012, pp. 1–5.

[25] A. V. Pazderin and V. O. Samoylenko, "Localization of Non-Technical Energy Losses based on the Energy Flow Problem Solution," in *Proceedings of the 6th IASTED Asian Conference on Power and Energy Systems (AsiaPES)*. ACTA Press, 2013, pp. 100–103.

[26] D. N. Nikovski, Z. Wang, A. Esenther, H. Sun, K. Sugiura, T. Muso, and K. Tsuru, "Smart meter data analysis for power theft detection," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, Berlin, Heidelberg, 2013, pp. 379–389.

[27] S. Sahoo, D. Nikovski, T. Muso, and K. Tsuru, "Electricity theft detection using smart meter data," in *2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2015, pp. 1–5.

[28] L. Marques, N. Silva, I. Miranda, E. Rodriges, and H. Leite, "Detection and localisation of nontechnical losses in low voltage distribution networks," in *Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MedPower 2016)*. IEEE, 2016.

[29] L. G. De O Silva, A. A. Da Silva, and A. T. De Almeida-Filho, "Allocation of power-quality monitors using the p-median to identify nontechnical losses," *IEEE Transactions on Power Delivery*, vol. 31, no. 5, pp. 2242–2249, 2016.

[30] J. B. Leite and J. R. S. Mantovani, "Detecting and locating non-technical losses in modern distribution networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1023–1032, 2018.

[31] M. Wen, D. Yao, B. Li, and R. Lu, "State Estimation Based Energy Theft Detection Scheme with Privacy Preservation in Smart Grid," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[32] S. C. Huang, Y. L. Lo, and C. N. Lu, "Non-Technical Loss Detection Using State Estimation and Analysis of Variance," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2959–2966, 2013.

[33] R. D. Trevizan, A. Rossoni, A. S. Bretas, D. da Silva Gazzana, R. de Podestá Martin, N. G. Bretas, A. L. Bettiol, A. Carniato, and L. F. do Nascimento Passos, "Non-Technical Losses Identification Using Optimum-Path Forest and State Estimation," in *2015 IEEE Eindhoven PowerTech*. IEEE, 2015, pp. 1–6.

[34] C. L. Su, W. H. Lee, and C. K. Wen, "Electricity theft detection in low voltage networks with smart meters using state estimation," in *2016 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2016, pp. 493–498.

[35] A. Rossoni, S. H. Braunstein, R. D. Trevizan, A. S. Bretas, and N. G. Bretas, "Smart distribution power losses estimation: A hybrid state estimation approach," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE, 2016, pp. 1–5.

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

[37] K. Kee, S. Shahab, and C. Loh, "Design and development of an innovative smart metering system with GUI-based NTL detection platform," in *4th IET Clean Energy and Technology Conference (CEAT 2016)*. IEEE, 2016.

[38] J. Pulz, R. B. Muller, F. Romero, A. Meffe, F. Garcez Neto, and A. S. Jesus, "Fraud detection in low-voltage electricity consumers using socio-economic indicators and billing profile in smart grids," *CIRED - Open Access Proceedings Journal*, vol. 2017, no. 1, pp. 2300–2303, 2017.

[39] I. Monedero, F. Biscarri, C. Léon, J. Biscarri, and R. Millán, "MIDAS: Detection of non-technical losses in electrical consumption using neural networks and statistical techniques," in *International Conference on Computational Science and Its Applications*. Springer, Berlin, Heidelberg, 2006, pp. 725–734.

[40] J. Tacón, D. Melgarejo, F. Rodríguez, F. Lecumberry, and A. Fernández, "Semisupervised Approach to Non Technical Losses Detection," in *Iberoamerican Congress on Pattern Recognition*. Springer, Cham, 2014, pp. 698–705.

[41] A. H. Nizar, Z. Y. Dong, and Y. Wang, "Power utility nontechnical loss analysis with extreme learning machine method," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 946–955, 2008.

[42] 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 Transactions on Power Delivery*, vol. 25, no. 2, pp. 1162–1171, 2010.

[43] S. S. S. R. Depuru, L. Wang, V. Devabhaktuni, and P. Nelapati, "A hybrid neural network model and encoding technique for enhanced classification of energy consumption data," in *2011 IEEE Power and Energy Society General Meeting*. IEEE, 2011, pp. 1–8.

[44] L. A. Pereira, L. C. Afonso, J. P. Papa, Z. A. Vale, C. C. Ramos, D. S. Gastaldello, and A. N. Souza, "Multilayer perceptron neural networks training through charged system search and its application for non-technical losses detection," in *2013 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)*. IEEE, 2013, pp. 1–6.

[45] B. C. Costa, B. L. Alberto, A. M. Portela, W. Maduro, and E. O. Eler, "Fraud Detection in Electric Power Distribution Networks Using an ANN-Based Knowledge-Discovery Process," *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. 4, no. 6, pp. 17–23, 2013.

[46] V. Ford, A. Siraj, and W. Eberle, "Smart Grid Energy Fraud Detection Using Artificial Neural Networks," in *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG)*. IEEE, 2014, pp. 1–6.

[47] C. Cody, V. Ford, and A. Siraj, "Decision tree learning for fraud detection in consumer energy consumption," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 1175–1179.

[48] B. Coma-Puig, J. Carmona, R. Gavalda, S. Alcoverro, and V. Martin, "Fraud Detection in Energy Consumption : A Supervised Approach," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 120–129.

[49] J. No, S. Y. Han, Y. Joo, and J.-H. Shin, "Conditional abnormality detection based on AMI data mining," *IET Generation, Transmission & Distribution*, vol. 10, no. 12, pp. 3010–3016, 2016.

[50] P. Glauner, J. A. Meira, L. Dolberg, R. State, F. Bettinger, and Y. Rangoni, "Neighborhood features help detecting non-technical losses in big data sets," in *2016 IEEE/ACM 3rd International Conference on Big Data Computing Applications and Technologies (BDCAT)*. IEEE, 2016, pp. 253–261.

[51] R. R. Bhat, R. D. Trevizan, R. Sengupta, X. Li, and A. Bretas, "Identifying Non-technical Power Loss via Spatial and Temporal Deep Learning," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 272–279.

[52] J. A. Meira, P. Glauner, R. State, P. Valtchev, L. Dolberg, F. Bettinger, and D. Duarte, "Distilling provider-independent data for general detection of non-technical losses," in *2017 IEEE Power and Energy Conference at Illinois (PECI)*. IEEE, 2017, pp. 1–5.

[53] J. I. Guerrero, I. Monedero, F. Biscarri, J. Biscarri, R. Millan, and C. Leon, "Non-Technical Losses Reduction by Improving the Inspections Accuracy in a Power Utility," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1209–1218, 2018.

[54] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2018.

[55] P. Glauner, J. A. Meira, P. Valtchev, R. State, and F. Bettinger, "The Challenge of Non-Technical Loss Detection Using Artificial Intelligence: A Survey," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 760–775, 2017.

[56] R. D. Trevizan, A. S. Bretas, and A. Rossoni, "Distribution Test System for Nontechnical Loss Detection," in *2018 North American Power Symposium (NAPS)*. IEEE, 2018, pp. 1–6.

[57] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[58] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," in *SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. ACM, 2000, pp. 93–104.

[59] R. E. de Espana. (2002) Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portatiles lectura (in spanish).

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

[61] T. Fawcett, "ROC Graphs : Notes and Practical Considerations for Researchers," *Machine learning*, vol. 31, no. 1, pp. 1–38, 2004.

[62] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[63] V. Raghavan, P. Bollmann, and G. S. Jung, "A critical investigation of recall and precision as measures of retrieval system performance," *ACM Transactions on Information Systems (TOIS)*, vol. 7, no. 3, pp. 205–229, 1989.

[64] A. Ng. Machine learning. Course Lecture. [Online]. Available: https://www.coursera.org/learn/machine-learning

[65] 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.

[66] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[67] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794.

[68] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[69] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[70] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on International Conference on Machine Learning (ICML'13)*. ACM, 2013, pp. 1310–1318.

[71] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[72] C. Guo and F. Berkhahn, "Entity Embeddings of Categorical Variables," *arXiv preprint arXiv:1604.06737*, 2016.

[73] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv preprint arXiv:1502.03167*, 2015.

[74] A. Karpathy, L. Fei-Fei, and J. Johnson, "CS231n: Convolutional Neural Networks for Visual Recognition," 2016. [Online]. Available: http://cs231n.stanford.edu/2016

[75] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[76] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[77] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[78] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.