Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-023-00927-7

A case study of improving a non-technical losses detection system through explainability

Bernat Coma-Puig[1] · Albert Calvo[1] · Josep Carmona[1] · Ricard Gavaldà[1]

Received: 31 March 2021 / Accepted: 6 February 2023

Abstract

Detecting and reacting to non-technical losses (NTL) is a fundamental activity that energy providers need to face in their daily routines. This is known to be challenging since the phenomenon of NTL is multi-factored, dynamic and extremely contextual, which makes artificial intelligence (AI) and, in particular, machine learning, natural areas to bring effective and tailored solutions. If the human factor is disregarded in the process of detecting NTL, there is a high risk of performance degradation since typical problems like dataset shift and biases cannot be easily identified by an algorithm. This paper presents a case study on incorporating explainable AI (XAI) in a mature NTL detection system that has been in production in the last years both in electricity and gas. The experience shows that incorporating this capability brings interesting improvements to the initial system and especially serves as a common ground where domain experts, data scientists, and business analysts can meet.

Keywords Non-technical losses · Explainability · Expert system · Shapley values · LIME

Responsible editor: Martin Atzmüller, Johannes Fürnkranz, Tomas Kliegr, Ute Schmid

✉ Bernat Coma-Puig
bcoma@cs.upc.edu

Albert Calvo
albertc@cs.upc.edu

Josep Carmona
jcarmona@cs.upc.edu

Ricard Gavaldà
gavalda@cs.upc.edu

[1] Universitat Politècnica de Catalunya, Barcelona, Spain

Published online: 19 April 2023

1 Introduction

In the last decade, artificial intelligence (AI) has become a widely used discipline in the industry, expected to provide solutions in a broad range of applications ranging from Computer Vision, Market Analytics, or Fraud Detection. Generally speaking, AI helps companies optimise their processes, be more productive, and continuously work towards their goals. However, sometimes AI solutions (e.g., predictive models obtained through machine learning algorithms) are complex and hard to understand by a human, often requiring deep knowledge of the company goals and algorithmics. The process of empowering AI in the industry has opened a debate on trust and confidence, especially when it is used for automatic decision-making (Rudin 2019). Intending to provide a solution to these problems, the data science community has started to propose new tools to provide explanations to AI algorithms (Molnar 2020). The field of explainability is a relatively new field in AI that is currently booming.

This paper highlights the importance of explainability of predictive models, which are the dominant approach to Non-Technical Loss detection in utility companies. Non-Technical Losses (abbreviated as NTL) refers to all losses caused by intentional theft and meter malfunctions: Meter tampering, bypassing meters, faulty or broken meters, un-metered supply or technical and human errors in meter readings (Glauner et al. 2017). It contrasts with Technical Losses (TL), the energy losses during the distribution of the energy that can be technically explained, e.g., by the network impedance. According to a study by Northeast Group, NTL globally amounts to US$96billion per year (Northeast group 2017). Moreover, there are other consequences of fraudulent activity in energy consumption: Illegal connections and meter tampering are dangerous operations for those that carry them out, the people living in the building and the operators that maintain and repair them. NTL is not fair to the customers that do pay for what they consume. Ultimately, it affects public administrations that co-finance these utility services. In conclusion, reducing NTL is one of the top priorities of energy providers.

State-of-the-art frameworks to fight NTL use complex predictive models such as Gradient Boosting Tree Models, Deep Learning and Support Vector Machines. These techniques can learn very complex patterns but are challenging to interpret. This lack of interpretability can difficult the NTL detection's success, as human validation is almost mandatory to confirm the detected patterns. This is hard if no explanations are attached to the model's prediction, or to the model itself. Moreover, classical machine learning problems may produce hidden biases in the models learned, such as dataset-shift, as explained in Glauner et al. (2017). This problem is typically seen in the literature (i.e., the dilemma between accuracy vs interpretability), and recently different explanation methods have been proposed that aim provide interpretability to these complex algorithms.

This work is based on our experience of building an NTL detection system for the Spanish branch of the international utility company Naturgy, that distributes electricity and gas. We have described the architecture and results obtained by our system in Coma-Puig et al. (2016), Coma-Puig and Carmona (2019). This manuscript explains different approaches recently tried by us to provide interpretability to our system, starting from simple statistical analysis (i.e., odds-ratio, Pearson correlation and feature distribution), then using the feature importance implementation from the ensemble tree methods, to finally using the state-of-the-art explainability methods LIME (Ribeiro et al. 2016) (for tabular data) and SHAP (Lundberg and Lee 2017) [(specifically the TreeSHAP method for tree-based models (Lundberg et al. 2018, 2020)]. We compare the pros and cons of the methods, describe use cases and results obtained and finally explain the conclusions and remaining challenges in the field.

This paper is structured as follows: Sect. 2 summarises the related work in NTL and explainability in industry, including an explanation of our solution to detect NTL. Section 3 reports on the current alternatives available for understanding prediction models and Sect. 4 exemplifies them for the case of NTL detection. In Sect. 5 we draw conclusions and enumerate challenges for the present and future.

2 Background and related work in detecting non-technical losses

To contextualise our case study of explainability in the process of detecting NTL, this section provides an overview of the entire NTL detection and management process, the approaches used to build supervised NTL detection models, the difficulties of reporting it, how explainability can be useful in data science industrial projects, and related work.

2.1 Addressing the NTL problem

Companies are well aware that a fraction of customers commit fraud, either by meter tampering to reduce their reported consumption or by creating bypasses. Moreover, some NTLs occur because meters fail or malfunction without malicious intervention. The obvious solution, sending technicians to inspect every meter is, in general, not cost-effective. For this reason, companies generate *campaigns*, lists of customers that seem more likely to incur some form of NTL according to some criterion. The campaign can be defined by many different parameters (geography, type of contract or energy usage, intuition of the domain expert, measurement mismatch systems, prediction by some machine learning algorithm, …). The goal is, of course, that within a campaign, substantially more cases of NTL occur than by inspecting a randomly chosen subset of customers.

Once the campaign is designed, technicians are sent to visit customers and inspect meters. Each visit can be successful or not (the customer may be absent or may refuse access to the meter; threats to the inspectors to drive them away are not uncommon). When successful, the result of the visit (say, tampering, malfunction, or all normal) is reported. The company may proceed to ask for back payment or fines to the customer, often requiring months-long legal actions. The customer's payments are somewhat euphemistically called "recovered energy" and measured in energy units, e.g., kWh. The goals of a predictive model for NTL are to predict cases of NTL and predict expected recovered energy, as that figure goes into the cost-benefit equation of a potential campaign.

2.2 Related work in NTL detection

Formulating the detection of NTL cases using machine learning techniques is widely reported in the literature, especially in electricity.

A very common approach is to reduce the NTL detection problem to a binary supervised machine learning problem using a black-box algorithm. In Buzau et al. (2018) a framework is presented that shares many elements with our previous work; e.g., it also uses Gradient Boosting and is implemented for a utility company in Spain. In Nagi et al. (2009, 2011) two examples are described that use Support Vector Machines to detect NTL in India, whereas neural networks are proposed for NTL detection in Costa et al. (2013), Pereira et al. (2013), Ford et al. (2014).

There exist other examples in the literature that use other supervised, unsupervised, and statistical inference methods. In McLaughlin et al. (2013) and in Liu and Hu (2015) there are two examples of implementing Markov models (Hidden and Partially Observed Markov model respectively). In Monedero et al. (2012) the Pearson Correlation is used to detect abnormal consumption drops. In Badrinath Krishna et al. (2015), Angelos et al. (2011), Cabral et al. (2008) examples are included of using unsupervised methods (clustering and Self-Organised Maps respectively). Also, in the literature can be seen examples of detecting NTL using rule-based patterns from expert knowledge [(e.g., Guerrero et al. (2014)] or inferred by fuzzy logic techniques [e.g., Spirić et al. (2014)].

Our approach to detect NTL is a supervised method that relies on data to detect NTL. However, there exist other approaches that use network-oriented solutions to detect measurement mismatch and direct manipulation of the meter. These solutions require smart meters and infrastructure. For instance, it is common that the utility companies install Feeder Remote Terminal Units in their distribution networks that monitor the network distribution and consumption of a zone to detect consumption mismatch (Zhou et al. 2015).

Among the surveys that summarise the approaches in the literature to detect NTL in electricity, we highlight two: Glauner et al. (2017) provides a good summary of the challenges seen in detecting NTL cases such as the dataset-shift, the representation of reality through features and scalability. Survey (Messinis and Hatziargyriou 2018) is more enumerative and collects the different methods used to detect NTL. In gas and water, there are no studies, to our knowledge, of implementing supervised methods to detect NTL cases.

Our previous work in NTL is reported in a series of papers:

Coma-Puig et al. (2016) explains our binary supervised approach to detect NTL; Coma-Puig and Carmona (2018) is our first approach to using an explanatory algorithm (in this case LIME) in our system, and Coma-Puig and Carmona (2019) provides an extensive analysis of the benefits of using artificial intelligence techniques to detect NTL cases, as well as the problems detected such as the dataset-shift and other data-related biases; Coma-Puig and Carmona (2021) analyses the classical classification approach to detect NTL cases, and shows how a point-wise ranking regression approach provides better results in terms of energy recovered and in explanatory terms; Calvo et al. (2020) evidences the use of a bucket of models (i.e., use more specific mod-

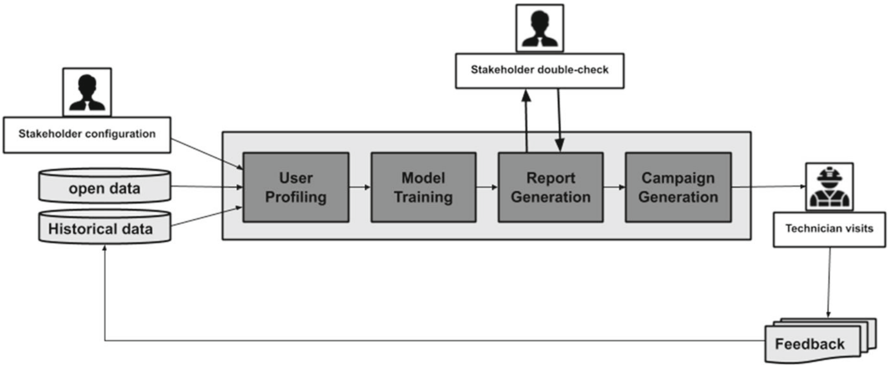

Fig. 1 The NTL detection Framework: When the stakeholder configures the campaign to be done, the system autonomously loads the data, trains the model, predicts the scores for each customer at the present, and finally the company generates the campaign. The results of the campaign is updated in the data sources

els for each type of NTL) increases the accuracy of our system (the models used are more specific for one type of NTL), and in interpretability (the explanations obtained from explanatory algorithms are easier to understand); finally, Coma-Puig and Carmona (2021) proposes a human-in-the-loop approach to involve and empower the stakeholders in charge of the campaign generation.

2.3 Supervised methods to detect non-technical losses

Let us describe our NTL detection system and the typical supervised machine learning approach to detect NTL. See also Fig. 1.

1. **Campaign configuration.** The stakeholder delimits the scope of the campaign (the type of utility, region and tariff) and extracts the required data from the company information systems.
2. **User profiling.** With this information, features are built to profile the customers visited in the past (which give the labelled training instances), and the current state of the target customers (which give the unlabelled instances for which a prediction is required). In general, the consumption features are the most important since they reflect the change of the consumption behaviour, but other complementary information such as where the customer lives, or the results of past inspections to the same customer, should be included.
3. **Model training.** With the historical NTL/Non-NTL profiles, a model is trained and a prediction if produced for each customer in the campaign. In general, the approaches seen in the literature build classification models (i.e., reduce the NTL detection as a binary classification problem). In our case, a prediction is both a score (to binary determine if the customer might be committing NTL) and an estimation of the recoverable energy. See Table 1).

Table 1 An example of the predictions generated by our NTL system. For each customer, it provides both a probability of the customer being committing NTL and also an estimation a prediction of the energy to recover for that customer

Customer	Score NTL	Score kWh
$Customer_4$	0.89	1231 kWh
$Customer_1$	0.66	6231 kWh
$Customer_3$	0.26	331 kWh
$Customer_0$	0.12	231 kWh
$Customer_2$	0.01	51 kWh

4. **Report generation and campaign generation.** The top-scored customers are included in a report, and the company decides which customer are included in a campaign.
5. **Feedback.** The result of the inspection (or the impossibility of it) for those customers visited in the campaign is included in the system, as feedback and labels for future campaigns.

2.4 Challenges in detecting NTL through machine learning

As seen in our previous work and other examples in the literature, the supervised approaches can provide good results in detecting NTL cases. However, many challenges hinder the system.

The main challenge faced when building NTL detection models is the use of observational data produced for other purposes. This reuse calls into question the quality of the data, both in qualitative and quantitative terms. To begin with, direct meter inspection by a trained technician is expensive and, for this reason, the companies pre-select those customers with energy losses to be verified on-site by the technician. Therefore, the historical NTL cases are customers who had abnormal consumption behaviour (e.g., a suspicion of fraud). Likewise, most of the customers that have normal consumption behaviour are not represented in the system. This lack of representativeness is especially true in regions with a very low fraud proportion, where the majority of the customers have never been included in a campaign. All this means that one of the maxims to guarantee a good result when using machine learning, which is that the training and test data should be i.i.d., i.e., independent and identically distributed, is not met. It is therefore a challenge to guarantee reliable results.

These representativeness issues are not easy for the company to correct, since it is difficult to reconcile the short-term interests of detecting as many frauds as possible with the idea of generating a training set representative of the whole population in the long term, i.e., the exploitation vs. exploration dilemma. For example, it makes perfect sense that if some customers are recidivist fraudsters (and therefore the company is sure that it can recover energy), the company will choose to visit these customers again rather than generate exploratory campaigns, despite this decision might generate biases in the trained models. Similarly, companies tend to generate more campaigns in those regions where, in general, the results are more efficient. We call efficient those campaigns where the company recovers more energy because it knows the learned patterns better (i.e., the proportion of NTL detected or energy recovered is usually higher than average) or where the campaigns are executed faster (because the outsourced company in charge of the campaigns works efficiently or the population lives in dense regions, e.g., large cities). This predilection for generating campaigns in regions where historically the best results have been obtained creates biases in the data. As a case in point, this is a real example from our NTL system: Since the company was much more successful in detecting NTL cases in one region than in the rest of the country, the system used to assign only high scores to customers from that region.

In many cases, we have been able to mitigate these biases by implementing simple solutions, for instance, by segmenting customers according to their characteristics (e.g., generating local-based regional campaigns). However, these simple logical solutions are useful in cases where the bias is easy to detect, but in general, the bias is not caused by one specific sensitive feature but as a result of combining different slightly-correlated features. The classical evaluation of a model through training-validation analysis is simple and inaccurate when dataset-shift occurs (Drummond and Japkowicz 2010).

2.5 Explainability in machine learning

The difficulty in detecting bias in predictive models is largely due to the tendency to use black-box algorithms (as in our case, that we use a Gradient Boosting Decision Tree model) which, in principle, guarantee high accuracy but pose a clear problem in terms of transparency. It is then difficult to determine whether the patterns detected by the predictive system are causal, robust and generalizable to unseen data (Pearl and Mackenzie 2018; Pearl 2009; Arrieta et al. 2020). Fortunately, the artificial intelligence researchers are becoming more and more aware that society needs transparent algorithms, and therefore several approaches have been developed to better understand what patterns the algorithms learn. In our case, as mentioned when describing our previous work, we have been pioneers introducing some of state-of-the-art explanatory solutions to improve report generation. We have found it crucial to address both the stakeholder and the data scientist concerns.

This work provides a global vision of this process, not sufficiently described in our previous works, analysing the benefits and disadvantages of using each approach tested (statistical analysis, Feature Importance, LIME Ribeiro et al. (2016) and Shapley (1953) Values from SHAP) in the NTL detection context. Other examples in the NTL literature that focus on understanding how the model learns to include (Salman Saeed et al. 2020), which implements feature selection through Pearson's Chi-square statistical test and plots the first trees of a Boosted C5.0 model, and Santos et al. (2021), which proposes a very similar approach of ours of combining state-of-the-art implementations of Gradient Boosting Trees and Shapley Values. This latter idea we introduced in Coma-Puig and Carmona (2019) and developed in subsequent papers.

LIME and SHAP are recent techniques, yet there are relevant examples in the literature of their implementation in real scenarios. A major example of implementing explanatory algorithms in medicine is Lundberg et al. (2018), where it is reported how the Shapley values help to prevent the hypoxaemia during surgery. Paper Galanti et al. (2020) describes a process management case, where the Shapley values are used to provide explainability to a process monitoring system, showing that this explanatory approach, combined with a Long-Short Term Memory Neural Network model, provides accurate and trustful explanations that are in line with the explanations from human stakeholders. Also in Rehse et al. (2019) is explained how global and local explanatory approaches are used in a deep learning process prediction system in industry, where the system assists the workers and provides better insights of the fully automated DFKI-Smart-Lego-Factory using mostly textual explanations, local rules and saliency maps. All these cases exemplify that it is possible to combine predictive black-box and explanatory algorithms to achieve accurate yet transparent algorithms in industry.

Despite these examples of use in industry, the issue is far from settled. Rudin (2019) makes a strong case to abandon the explained black-box approaches and go for interpretable models. And there are reports in the literature that these solutions may lack in stability (Alvarez-Melis and Jaakkola 2018) and trustworthiness (Slack et al. 2020).

2.6 Interface between data scientists, stakeholders and managers

When discussing the use of explainability in a machine learning project, the emphasis is often on understanding how good the predictive model is. This is, indeed, certainly accurate, and this project is no exception. For example, as we discussed in Coma-Puig and Carmona (2021), explainability allowed us to compare different models based on the learned patterns beyond benchmarking on a specific dataset. However, our experience with Naturgy also allowed us to see explainability as a tool to ensure the success of the project by improving communication between the actors involved in the project.

From a data scientist's perspective, implementing an NTL detection system in a company like Naturgy, with millions of customers in many different domains, is extremely challenging, as each domain has its own unique characteristics and biases derived from the use of observational data. In general, a good approach to bypass this tedious procedure is to rely on the knowledge of the stakeholder in charge of the campaigns, to understand what knowledge they have of the domains (e.g., which populations have more fraud). However, this collaboration can often be unfruitful, because on the one hand the objective of the data scientist is to replace the existing methods to detect NTL, detecting new patterns and improving their accuracy, and on the other hand, because the stakeholder often does not know how a predictive model works, and it is difficult for them to adapt his knowledge to the new approach to detect NTL.

Explainability helps to overcome this problem, as it allows to understand how certain data affect a model, simplifying the collaboration between stakeholder and data scientist. In other words, the data scientists move from asking the stakeholders in a generic way "what should we know about this domain" to asking "why the model has learned this pattern". With this fundamental change, the stakeholder can be much more assertive, as is clearer to them what is wanted to know about their knowledge. This allows them to take an active role, even if they do not have machine learning knowledge, in the NTL detection process using predictive models. The result is that they will not only be able to do a better double-checking analysis (as we have explained in Fig. 1), but also to propose new variables, correct biases and actively participate in the pre-processing of the data.

Finally, explainability is also extremely useful for the trustworthiness of the project in the company. One of the problems with applied data science solutions in industry is that, in some cases, machine learning or artificial intelligence are buzzwords associated with big promises and, therefore, the management stakeholders feel obligated to support these techniques in the company. However, the implementation of data science techniques in real scenarios is very challenging and, in many cases, the results are slow in coming and, in general, the system built is not the panacea that fulfils all the problems regarding NTL detection. Therefore, making the system transparent is mandatory in order to guarantee stakeholders' trust in management; if these stakeholders are aware of the system, understand what it does, and are aware that there is a learning process in which better and better predictive models are obtained, the commitment to the system will be consolidated, although there may be occasional unsatisfactory results.

3 Understanding a non-technical loss prediction model

This section provides an explanation of the explanatory approaches tested in our NTL detection system. More specifically, we explain three different statistical analysis (i.e., feature distribution, pearson correlation and odds-ratio) as well as three explanatory algorithms (i.e., Feature Importance, Local Surrogate Models through LIME and the Shapley Values from SHAP). Their use in our NTL detection system is explained in further Sect. 4, where their benefits and disadvantages is analysed.

3.1 Statistical analysis

A basic approach to understanding a predictive model is to analyse the training dataset statistically. This work highlights the following statistical measures:

- **Feature Distribution**: The distribution of the values of each feature in different domains and segments. For instance, this simple analysis is useful to identify regions where campaigns have been more successful at detecting NTL cases.
- **Pearson Correlation**: A measure of the linear correlation between two features, where 1 indicates a perfect positive correlation (i.e., for every increase in one feature, there is a positive increase of a fixed proportion in the other feature), and -1 indicates a perfect negative correlation (i.e., for every increase in one feature, there is a decrease of a fixed proportion in the other feature), where 0 indicates no linear relation. The coefficient (r) is defined as the ratio between the covariance Cov of the values of two features divided by the product of their standard deviation S, i.e.,:
$$-1 \leq r_{XY} = \frac{Cov(X,Y)}{S_X S_Y} \leq 1 \tag{1}$$

As explained in Sect. 2.2, in Monedero et al. (2012) the Pearson Correlation coefficient is used to detect an abrupt and gradual but constant decrease of consumption in customers, hence suspicious of NTL.

- **Odds-Ratio**: The Odds-Ratio OR statistic is usually used in medical reports [(as explained in Bland and Altman (2000)]. It quantifies the influence of a binary value to an outcome. In the NTL detection context, let $F_{x_i=1}$ be the number of NTL instances x with feature $x_i = 1$, $F_{x_i=0}$ be the NTL instances x with feature $x_i = 0$, $C_{x_i=1}$ be the non-NTL instances x with feature $x_i = 1$, and $C_{x_i=0}$ be the non-NTL instances x with feature $x_i = 0$; then the OR is:

$$OR = \frac{F_{x_i=1}/C_{x_i=1}}{F_{x_i=0}/C_{x_i=0}} = \frac{F_{x_i=1}/F_{x_i=0}}{C_{x_i=1}/C_{x_i=0}}. \tag{2}$$

Odds-Ratio values far from 1 indicate that customers with $x_i = 1$ and customers with $x_i = 0$ have a different proportion of NTL.

These statistical metrics are not often considered explanation methods but are useful to understand the data used to train the model and, in some cases, are enough to detect biases or undesired prediction rules (e.g., a preference for a region because campaigns there were more successful in the past).

3.2 Explainability

In machine learning, explainability refers to presenting textual, numerical or visual information that allows the human to understand predictions (Arrieta et al. 2020). It is therefore dependent on human judgement and so hard to mathematize uniquely.

We give two operational definitions of explainability in this paper: Model level and instance level. If M is a trained predictive model that receives instances $x = (v_1, \ldots, v_n)$ to predict, a model-level or global explanation of M is a vector $(w_1, \ldots, w_n)$ that describes how each feature x_i globally influences the predictions made by M, computed typically on the training instances used to build M. An instance-level or local prediction provides such a vector for a specific instance x, therefore how each feature influences M to produce its specific prediction $M(x)$.

Some simple models are considered self-explainable in one or both senses. Typical examples are Linear and Logistic Regression models (where the coefficients indicate how relevant each feature is for the model), Decision Trees (where one can follow a path in the tree to understand how the model scored an instance), or Decision Rules (where the predictive models are a set of if-else statements that can be easy to understand). Being the simplest, these models also tend to be the least accurate.

On the other hand, there are explainability methods that are model-agnostic, i.e., not related to a specific type of model, intended to be used for complex algorithms such as Gradient Boosting and Neural Networks. These include statistical analysis [(e.g., Partial Dependence Plot (Friedman 2001) or the Feature Interaction (Friedman and Popescu 2008)], Global Surrogate models (interpretable models whose values to predict are the values predicted by the complex model), Local Surrogate Models (interpretable models that aim to reproduce the behaviour of the complex model for one specific instance) or Shapley Values.

This work focuses on three representatives of the current state-of-the-art explanation methods: *feature importance* (from the ensemble tree methods), *local surrogates* from the LIME library, and the SHAP library that computes the *Shapley Values*.

3.2.1 Feature importance

In tree models (i.e., a decision tree, or an ensemble of trees), the different methods to feature importance can be divided into *prediction* and *occurrence* methods:

- Prediction methods: Analyse the influence of the feature values in the model's predictions. This naive definition includes, for instance, the Random Forest implementation from *Scikit-learn* (Pedregosa et al. 2011) (that evaluates the Gini impurity of the samples of the nodes decrease after a split using that feature), or the $LossFunctionChange$ from *Catboost* (Prokhorenkova 2017) that evaluates how the prediction changes if that feature is removed.
- Occurrence methods: Measure the importance of the feature by analysing their occurrences in the training process, i.e., how many times the feature has been used in the splitting process, usually referred as *weight* or *frequency*, or the number of instances in the node split by that feature, usually referred as *coverage*.

3.2.2 Local surrogate models

Local surrogate models are simple interpretable models that aim to replicate the prediction made by complex black-box models for one specific prediction: Let M be a predictive model that the surrogate model aims to explain, x the instance to be explained, and L_n an interpretable model (e.g., a linear Regression) trained on n instances chosen somehow, then we would like to have $L_n(x) \simeq M(x)$ while keeping the model complexity of L_n as low as possible, for example, using as few features as possible to provide a simple explanation. Different methods differ on the type of model L_n and the instances used to build it, which may be selected from the training set or synthetically generated.

A state-of-the-art approach to local surrogate models is LIME (Local interpretable model-agnostic explanation; Ribeiro et al. 2016). This library provides different approaches to obtain the instances to create the surrogate model. For the tabular data approach (our case), LIME perturbs each feature of x independently, using a normal distribution with the same mean and standard deviation. Then, it weights the perturbed instances according to their proximity to the original instance to be explained, and trains an interpretable model with these instances. Finally, it provides an explanation (i.e., if each feature value increases or decreases the prediction, as visually explained in Fig. 2). The sum of these values should correspond to the prediction made by the original model.

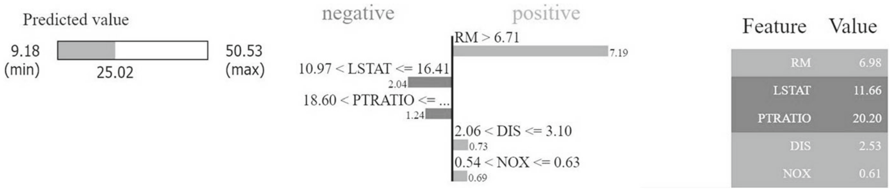

Fig. 2 Example of the output from LIME in the Boston Dataset regression prediction dataset from Scikit-Learn. On the left is shown the local prediction. At the center the five features that most influenced the prediction are provided. For instance, we can see that having the value of the RM feature (average number of rooms per dwelling) greater than 6.71 increases the prediction by 7.19. Finally, on the right there is a summary of the exact feature values for that instance

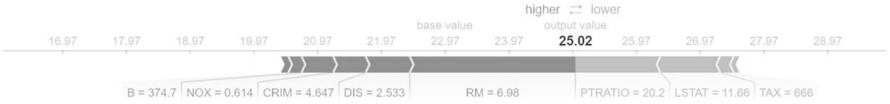

Fig. 3 Example of the output from SHAP in the Boston dataset regression prediction dataset from Scikit-Learn. We can see that the Base Value is 22.97. In the left-side are plotted the features that increase the prediction (e.g., the RM feature, the same one analysed in previous Fig. 2, also increases the prediction). In right-side are plotted the features that decrease the prediction. The sum of the Shapley Values and the Base Value is equal to the predicted value

3.2.3 Shapley values (SHAP)

Shapley (1953) Values is a method to analyse the importance of each player in a cooperative game, to reasonably determine the importance of each player for the payoff. SHAP adapts this idea to determine how much the value of each feature of x influences the prediction $M(x)$. From a Base Value that corresponds to the mean of the labelled instances in the training set, SHAP analyses how each feature in each instance increases or decreases this Base Value to achieve the final prediction from M.

The Shapley Values of a feature value in instance x is usually defined as:

$$\psi_i = \sum\nolimits_{S\subseteq\{x_1,\ldots,x_m\}\setminus\{x_i\}} \frac{|S|!(p-|S|-1)!}{p!} \left(val\left(S\cup\{x_i\}\right) - val(S)\right)$$

where p corresponds to the number of features, S a subset of the features from the instance and val corresponds to the function that indicates the payout for these features. In the equation, the difference between the val corresponds to the marginal value of adding the feature in the prediction for a particular subset of features S. The summand denotes all the possible subsets S that can be done without including the feature from which the Shapley Values is calculated, i.e., v_j. Finally, $\frac{|S|!(p-|S|-1)!}{p!}$ corresponds to the permutations that can be done with subset size $|S|$, to properly distribute the marginal values between all the features of the instance. All possible subsets of features are considered, and the effect in the prediction of including the feature to each subset is observed. Figure 3 shows an example of the feedback provided by SHAP for explaining the prediction of a particular instance from a public dataset.

SHAP offers different methods to compute the Shapley Values, depending on the supervised algorithm used. In our case, we would use the TreeSHAP method for Tree models, since we are using a Gradient Boosting Ensemble Tree model. Other methods are the KernelExplainer, which is a generic approach using weighted linear regressions, similarly to the LIME approach, or the Deep Explainer, an enhanced implementation of the DeepLift algorithm (Shrikumar et al. 2017), to compute the Shapley Values for deep learning algorithms.

3.3 Comparison

These four approaches have different characteristics that are summarised in Table 2.

The statistical analysis is the only approach that analyses the labelled data instead of the model learnt. Therefore, the analyses using this approach might be inconclusive (in many cases, one might not know exactly how the biases are reflected in the learning process), but as explained in Sect. 3.1, it can be very useful to understand better the data available.

The feature importance method from the tree algorithms offers a global vision of how the model was learnt by ranking the most important features. The problem with this approach is that it provides a *how much* vision of the importance of a feature, but does not provide a *how*: the method analyses how much the feature influences during the training process (e.g., how many times the feature is used), but it does not provide how the feature influenced the output.

The local surrogate approach from LIME solves the problems explained in the two previous approaches: it explains how the model was learnt, providing instance-level explanations. But, as explained in the next section, the method is unstable: explanations depend on the parameter settings and on the random sample chosen to create the surrogate model, which results in different explanations if the algorithm is run several times on the same instance.

SHAP also analyses how the model has learned and offers instance explanation, but it has the advantage that the local's explanation sum offers a global explanation of the model. Moreover, the SHAP for tree models is robust, always providing the same explanation for the same instance and model.

Next section explains how these methods performed in our NTL detection system.

4 Case study: explaining NTL prediction for electricity

4.1 Context and purpose of explainability in our NTL system

During the last years, we have been developing an NTL detection system for the Spanish branch of an international utility company. Our approach, explained in Coma-Puig and Carmona (2019), is a supervised learning system that builds a Gradient Boosting Decision Tree (GBDT) Model for NTL patterns from the past to score the customers at present. For each present customer, it returns a normalised value for the probability of NTL and predicts the recoverable energy. Based on these scores and the

Table 2 Comparison in terms of Coverage (what the method analyses), Model-Agnosticism (if the method can be used in any model), Robustness (if the methods always provide the same explanation for the same data) and Scope (if the method explains an instance or provides a global explanation of the model) between the four methods considered

	Statistical analysis	Feature importance	LIME (tabular data)	SHAP (TreeSHAP)
Coverage	Data	Model	Model	Model
Strategy	Ante-Hoc	Post-Hoc	Post-Hoc	Post-Hoc
Model Agnostic	Yes	No	Yes	Yes
Robustness	Stable	Stable	Unstable	Stable
Scope	Global	Global	Local	Local/Global

target segment, the company builds a campaign with the customers to be visited. A new model is trained every time the system generates campaigns, since we are constantly updating the system, including new labeled information. The domains are very varied: from highly populated regions with a lot of historical NTL/non-NTL information, to sparsely populated regions with very little information. This need for flexibility made us decide to use GBDTs models, since they offer state-of-the-art results for tabular data in all types of domains with little tuning (Shwartz-Ziv and Armon 2022).

The system achieves, in many cases, the desired results. For instance, two types of campaigns where the system worked particularly well were those that aim to detect NTL in customers with no contract[1] (achieving campaigns with a precision i.e., proportion of NTL cases among the customers visited in the campaign, higher than 50%), and campaigns to detect NTL in non-consuming customers, achieving a precision up to 36%.[2] These results were much better than generating campaigns by random choice of customers, and also better than the heuristics previously used by the company. However, it exhibited some of the problems mentioned in Sect. 2.4 of undesired biases and dataset-shift (e.g., to score higher the customers from a specific region) a known problem in the detection of NTL, as explained in Glauner et al. (2017). This resulted in some cases in campaigns that were worse than expected.

To better understand our system and detect its biases, we started to explore which explainability solutions could be introduced in our system to better understand it. The result of this analysis is this paper that explores the advantages and disadvantages of the tested methods. Introducing explainability into our system has been of great help to improve our system, make it more robust, and facilitate the interaction between the Stakeholders in charge of making the campaigns and us scientists at the University. Part of these satisfactory results are explained in Coma-Puig and Carmona (2021), where we analyse how explainability allowed us to compare models beyond benchmarking. Also, in Coma-Puig and Carmona (2021), we explain how we combined explainability and a human-in-the-loop method to empower the stakeholder, making them active participants in the generation of the campaigns, and thus be able to guarantee that the system learns generalisable patterns.

To illustrate the main ideas, this paper uses features from our dataset. Appendix A contains the list of features used in this paper, together with a short explanation. This analysis has been the seed to explore the solutions in our system based on explainability.

4.2 The starting point: limitations of statistical analysis

Our first approach to understand our system was to use statistical analysis as explained in Sect. 3. This approach was useful to detect several problems in our dataset, that influenced in the training process, for instance:

- **Recidivist customers**: Some of the customers that commit fraud are recidivist, that is, they commit fraud repeatedly again and again even after being detected as

[1] Customers that had contract but cancelled it. The wiring and other installation, in many cases, remain installed, so it is feasible to manipulate the installation and commit fraud.

[2] This results was extremely celebrated by the company, since the company was not able to discern when a house was empty or if it was not consuming record due to a meter manipulation.

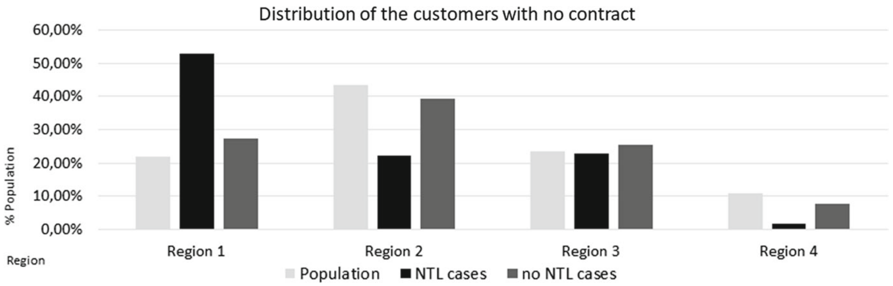

Fig. 4 Example of a bias in our dataset: The labelled distribution in the company for similar nearby regions is much different, and our model using this dataset as it is could be biased to high-score the customers from Region 1

such. Our model detected this pattern, and therefore it was highly biased towards the features related to previous visits (e.g., if a customer has a registered episode of NTL in the past, or if the number of visits by a technician is higher than the average). In itself, this pattern is correct, but it was not interesting in terms of knowledge discovery, as it was already known and used by the company.

- **Biased information**: The labelled information available had biases. By design, most available labelled instances corresponded to visits actually made, so highly biased towards the customers that had looked suspicious in the past, and also to geographical areas where inspections had been more frequent for whatever reason. Therefore, the available labelled data was not distributed as the whole population, and our models learned some erroneous patterns. This is exemplified in Fig. 4, a real case of bias in a segment of our labelled dataset.

Nevertheless, the statistical analysis could not provide satisfactory explanations for how our black-box model made predictions. For this reason, we started to introduce in our system different explanatory approaches (i.e., the methods explained in Sect. 3), to understand better the role of each feature in the prediction process.

This section explains our experience using these algorithms in our NTL detection system. Through the study, we fix a particular reference predictive model called M from now on. It is a CatBoost model trained with 3060 trees and almost 303,000 labelled instances (3.5% of NTL cases), tuned through a training-validation process (with a 90–10 proportion). The hardware used for this work is an Intel Core i7-8550U CPU, with 16GB of RAM and an SSD disk. We report on the nature of the information provided by each method, its helpfulness to discover biases, and its strengths and weakness.

4.3 Feature importance

We will analyse the feature importance method in Catboost, more specifically the $PredictionValuesChange$ method that evaluates how much on average the prediction changes if the value of that feature is changed[3] The result of the method is a

[3] The definition of the $PredictionValuesChange$ is available in the CatBoost's documentation, https://catboost.ai/docs/concepts/fstr.html#fstr__regular-feature-importance.

ranking of features by importance, where importance values are normalised so that their sum equals 100.

Figure 5 shows the top 10 features[4] of our reference model M. This information provides a global picture of the model learnt since the most important features to detect NTL cases are consumption features (and also, to a lesser extent, visit features). Overall, the fact that 8 out of 10 features are related to consumptions or visits (and not, for example, town) convinces the domain experts that the model is focusing on the right information. Hence, using feature importance is helpful as a first sanity check of the model.

However, feature importance is insufficient to analyse deeper how features influence the prediction. See for instance the most important feature, *Last Impossible 2* in Fig. 5; this feature indicates the last time the company was unable to do a “No Fraud" visit. A “No Fraud" visit is one whose main aim is not to detect Fraud, but some other general-purpose. Nevertheless, the impossibility of performing the visit can hide an abnormal behaviour from the customer (e.g., the customer avoids the technician visit because he knows the meter has been tampered with). Although the model has learned this fraudulent pattern, it can not be confirmed through feature importance since it only provides a global, very fast (it took less than 0.05 s) to compute the explanation of the importance of the model’s features but does not provide the reason behind a high relevance. Indeed, in the ideal case, a high relevance describes a learned pattern of NTL. But it can also be the result of a bias in the data, or an internal decision of the learning algorithm that is not always justified or understandable by the stakeholder.

In this situation, it is necessary to complement the Feature Analysis with, for instance, statistical analysis. Table 3 analyses the distribution of the *Last Impossible 2* feature. In general, this feature for most labelled instances is undefined, i.e., no *Last Impossible 2* has occurred, and is therefore profiled with a missing value (a value that CatBoost handles internally). However, this proportion is reduced when we focus our analysis on the NTL cases with more than 3500kWh[5] recovered (where more than 10% of the cases had a *Last Impossible 2*). Finally, 3 out of 4 cases in which the company recovered more than 35000kWh had a defined value for *Last Impossible 2* (i.e., the customer was visited by a technician but the visit was not possible to be carried out); this pattern might be the one learnt by our model.

In conclusion, according to our experience, the feature importance methods might not be a proper method to understand the NTL detection model’s patterns fully but can be a good baseline approach to detect clear undesired patterns.

4.4 LIME

This section discusses an example of the explanations obtained from LIME for tabular data. We fix one particular customer for the rest of the section. It is an NTL case for which a fraud of 3000kWh was reported; our model predicted an energy to recover

[4] This section only analyses the top n features of each method to facilitate the explanation.

[5] The consumption of a house or apartment in Spain is, on average, around 3500kWh. Therefore, this figure is informally used in the project as a delimiter of what would be a great NTL case, i.e., an NTL case in which the amount of energy recovered is remarkable.

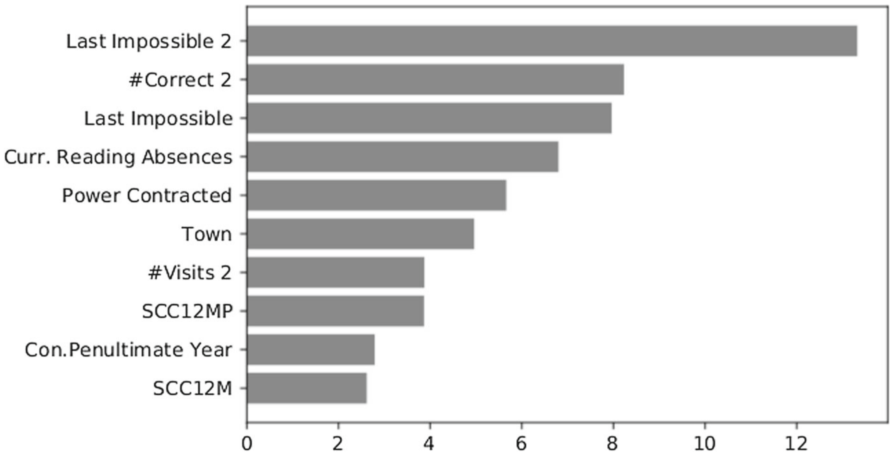

Fig. 5 Top 10 most important features according to the feature importance method from Catboost, i.e., PredictionValuesChange. It evaluates how much the prediction changes if the value of that feature is changed, on average

Table 3 Analysis of the value *Last Impossible 2* feature for the NTL labelled customers with recovered energy from 3500kWh to 35000kWh, more than 3500kWh and all the labelled customers

Top selection	Last impossible 2 undef.	Last impossible 2≥0
3500 to 35000 kWh	571	84
> 35000 kWh	1	3
Customers	295218	7324

This feature indicates the months passed since the last *Last Impossible 2* visit, where the value remains undefined (i.e., the missing values) in case of no visit. The proportion of *Last Impossible 2* $\geq$ 0 increases for the NTL cases, specially when the energy recovered is high

of around 2100kWh. The execution time to obtain the explanation was 38 s. Note that all features discussed are numerical because LIME requires re-encoding categorical variables as numerical ones. This example has been carefully selected because it exemplifies the problems we have had with LIME in our system.

Figure 6 shows an example of a subset (top-10) of the most important features for that customer according to LIME: the explanation indicates which feature values increase or decrease the specific prediction of 2100kWh; the sum of each prediction apportion should be the prediction done by the black-box model in that local region (or at least, a good approximation).

However, LIME seems to have an important problem of robustness, exemplified in Figs. 6 and 7: If the LIME algorithm is rerun on the same instance, a different random sample is generated each time to generate the local model, and this leads to different explanations of the same instance.

A second issue with LIME, reported in the literature Molnar (2020), is the high sensitivity of the output to the setting of certain parameters, particularly the kernel width. For instance, Fig. 8 shows the explanation of the same instance of Fig. 6, but

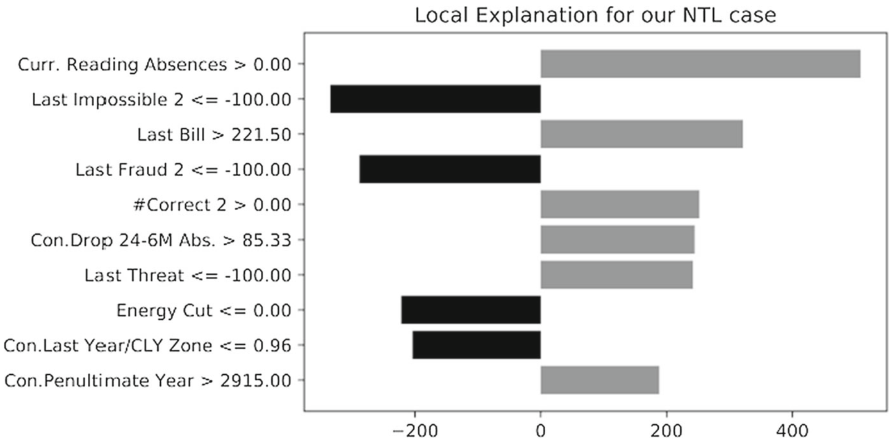

Fig. 6 Local Explanation of the NTL case, first run. The top-10 most important features from the LIME explanation of an NTL case with energy recovered of around 3000kWh. The most important features are the *Current Reading Absences* (i.e., that it has absences in readings) as an indicator of NTL and the *Last Impossible 2* feature (i.e., that has a negative value, indicating the absence of Impossible "No Fraud" visits as a non-NTL pattern)

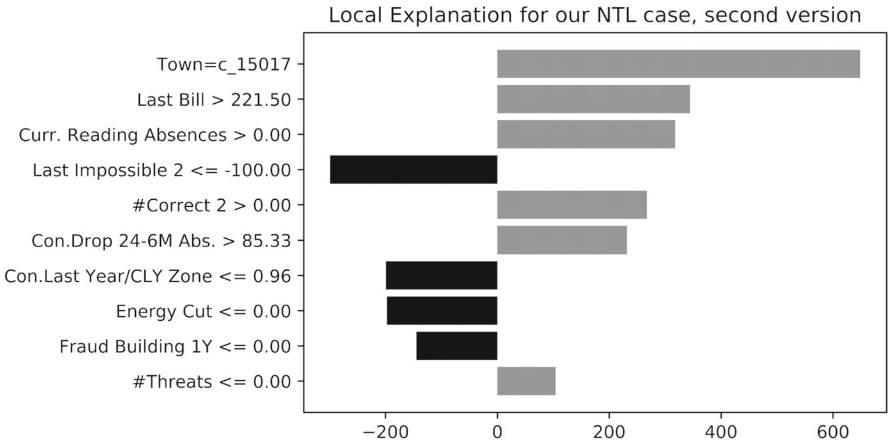

Fig. 7 Local Explanation of the NTL case, second run. The same instance of Fig. 6 is explained differently by LIME in a second run, due to sampling a different set of neighbours

now using a different $kernel\_width$ value. There is also little theoretical guidance for choosing appropriate values.

Finally, there is no guarantee that the explanation obtained from $L(x)$ is faithful to what $M(x)$ computes. This can also be seen in Fig. 7- The *#Threats* feature indicates if the company's technician has received threats when performing an installation or revision at that customer, and *Energy Cut* indicates whether energy has been cut to this user any time in the past. Nevertheless, upon close inspection, these features are not used in the computation of $M(x)$.

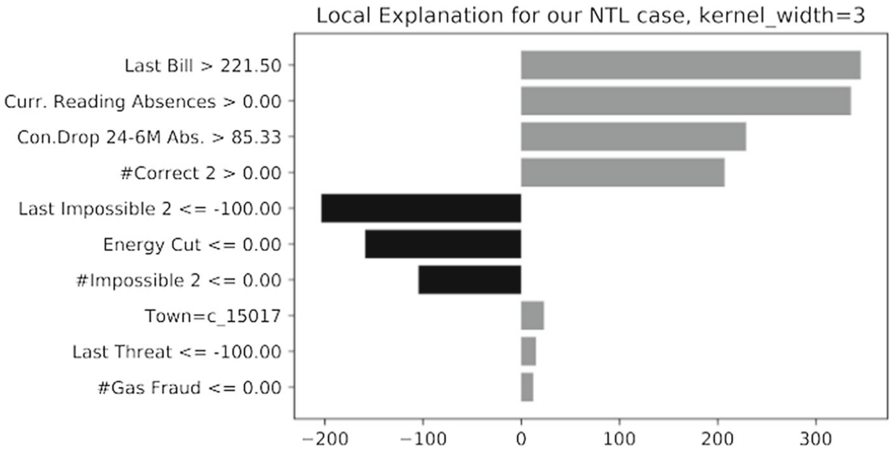

Fig. 8 Example of how the $kernel\_width$ highly influences the explanation process

Table 4 Results from the tests in Coma-Puig and Carmona (2018) where we used LIME as a post-process method to discard customers with an unjustifiedly high score. The post-process increases the precision (i.e., the proportion of NTL cases in our validation dataset that would be included in the simulated campaign) 13%

Dataset	%NTL	%Non-NTL	%Precision
Original campaign	72	28	72
LIME campaign	18	3	85

Despite these problems, based on the information provided by LIME, a methodology can be proposed. In Coma-Puig and Carmona (2018), we describe an approach for double-checking the predictions made by a model by implementing a rule system. This system would determine, based on the features that, according to LIME, mostly influenced the score for each instance, if the high prediction was trustful, discarding as NTL-cases those instances for which, according to human knowledge, a high score is not justified. The accuracy in our tests increased around 13% with this simple heuristic, as shown in Table 4.

4.5 SHAP

This section analyses the explanations from SHAP for trees (i.e., the Tree Explainer). We use the same reference instance used in the previous section, i.e., the positive NTL customer for which 3000kWh of fraud were reported and for which our model M predicted 2100kWh of recoverable energy.

Figure 9 shows the explanation of SHAP of a subset (top-10) of the most important features for our reference instance. Similarly to LIME, SHAP indicates how the feature values increase or decrease the specific prediction of the energy to recover. Moreover,

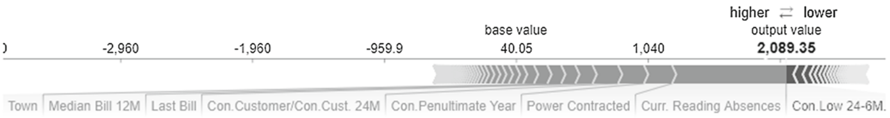

Fig. 9 Example of the Shapley Values for our reference NTL case of 3000kWh. The most important feature for this instance is the *Curr. Reading Absences* and the *Power Contracted*

it does not have the LIME's robustness problems since the TreeSHAP's explanation computation is deterministic and always provides the same explanation for a given model and instance. Also, the explanation is consistent with what the model has learnt, which was not always the case in LIME as described before with the *Energy Cut* feature.

SHAP for tree-based models (Lundberg et al. 2018) is, according to our experience, a very robust and rich method to provide interpretability to our system. The fast (in our case, the system computed the Shapley Values in around 260 s[6] implementation allows to obtain instance-level explanations, but also global model interpretation, e.g., the *summary_plot* shown in Fig. 10. This plot provides a summary of the model in terms of feature importance, similar to the feature importance reported by learning methods. Remarkably, this global explanation is consistent with the local explanations [as explained in Molnar (2020), "*the Shapley values are the 'atomic unit' of the global interpretations*"], and with the feature dependency of the predictive tree Model. Moreover, the theory of the Shapley (1953) Values guarantees the properties of *efficiency* (the feature contribution adds up the difference of prediction), *symmetry* (two features have the same Shapley Values if they contributed equally), *dummy* (a feature that is not used in the prediction model has a Shapley value of 0) and *additivity* (if the computation of the prediction can be divided into sub-processes, i.e., the boosting process in our model, the Shapley Values can be seen as the average of the Shapley values of each tree).

The improvements over LIME in robustness and the theory behind the explanatory algorithm are also evident when using SHAP in post-processing. In the Table we show a pre-processing system similar to the one implemented in Coma-Puig and Carmona (2018) with LIME. The two post-processings are not directly comparable (e.g., in Coma-Puig and Carmona (2018) we applied it in a classification model, while in Coma-Puig and Carmona (2021) we applied it in a regression model), but we did indeed detect that, as the explanations in SHAP are more robust, we could be more precise in our post-process. The lack of robustness in LIME is shown in Table 4, where most of the high-scored customers are discarded. This is due to the fact that LIME provided unrealistic explanations that made most of the explanations untrustful from the stakeholders' point of view view.

[6] In this paper we did not use the GPU accelerated version of TreeSHAP, that provides a faster computation but requires an Nvidia GPU.

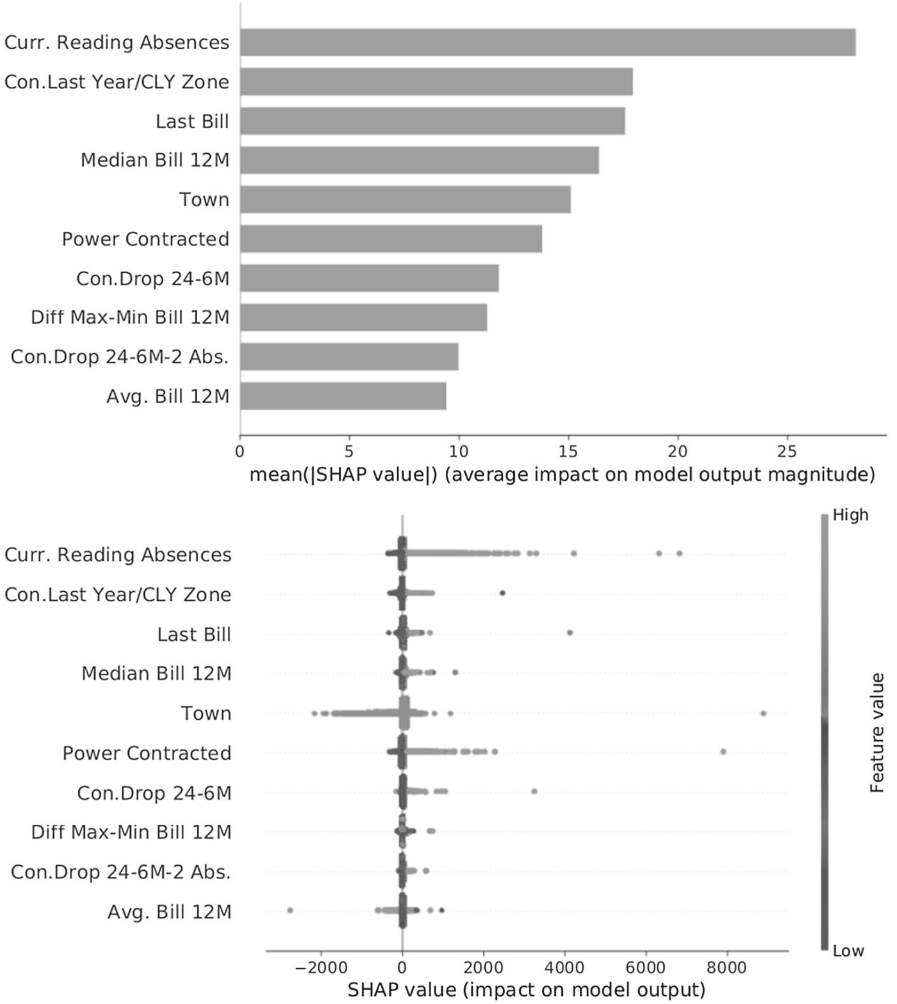

Fig. 10 Two versions of the *summary_plot* from SHAP: above, a box-plot diagram showing the average of the Shapley values for the most important features (similar to feature importance plots). Below, a distribution-like description (more fine-grained) of the same information. According to SHAP, the most important feature in the model is *Curr. Reading Absences*

4.6 Comparison and final remarks

Feature importance, the local approach from LIME for tabular data, and the SHAP method for tree-based trees are three very different approaches to providing explainability to our NTL detection system. In summary:

- **Depth**: The big difference in terms of depth is that Feature Importance provides a superficial and modular explanation of the influence of each variable on the predictions, while LIME and SHAP offer deeper explanations at the instance level. Therefore, as we have seen in our use case, Feature Importance can be interest-

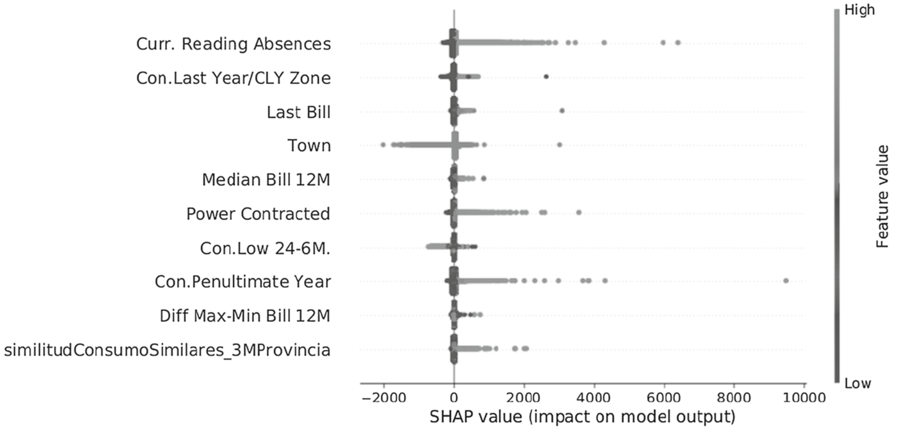

Fig. 11 The *summary_plot* obtained by SHAP after correcting the bias detected in Fig. 10

Table 5 Results from the tests in Coma-Puig and Carmona (2021) where we used the local explanations from SHAP as a post-process method to discard those top n high-scored customers whose most important fraudulent feature was not consumption related. This small post-process increased the kWh recovered per customer visited in the simulated campaign

KWh per visit	n=528	n=211	n=106	n=42
Regression	887.3	1266	1554.8	1740.3
Regression + Rule	944	1398.4	1741.5	2328.7

ing because of its effectiveness in getting a first sanity check of the model, but its superficiality would not allow us to implement the double-checking methods exemplified in Coma-Puig and Carmona (2018) (that uses LIME) or Coma-Puig and Carmona (2021) (that uses Shapley values).

- **Bias Detection**: Feature importance is a good approach to easily detect biases and other data-related problems. However, this can also be done with SHAP, which by complementing such information with local explanations, gives us a better insight into which values cause biases. In contrast, LIME's local approach makes it much more complicated when it comes to analyzing biases and unwanted patterns in the model due to the lack of global analysis.
- **Robustness**: LIME has the problems of robustness of explanations across runs due to its random component, which makes the whole approach look unreliable. In contrast, feature importance and SHAP (Tree SHAP) always give the same results for the same data.
- **Truthfulness**: Feature importance and SHAP compute the importance of the features by analyzing (with very different approaches and with a different focus) how the prediction changes when there is a modification in the feature. The local model

from LIME, on the other hand, can use features in the local explanation not used by the model (and therefore the explanation is not trustworthy).

- **Complexity**: Obtaining explanations for each method, in our case study, is fast:
  - The Feature Importance provides a superficial modular explanation in much less than a second.
  - The LIME method provides a local explanation in around 38 s.
  - SHAP provides local and global explanation in around 250 s.

  Thus, we could conclude that no approach can be discarded because it is computationally expensive. That said, it is worth noting that LIME offers local explanations (i.e., if we wanted a global explanation of the system, e.g., which variable might be relevant in general, we would have to compute the explanations multiple times). Regarding SHAP, we should also take into account the computational cost of obtaining the explanations with other SHAP's explainers since we use the implementation specific for tree-based models: TreeSHAP has a computational cost of $O(TLD^2)$, being T is the number of trees, L the maximum number of leaves in any tree and D the maximal depth of any tree, while the KernelSHAP (the model-agnostic approach that can be used for any type of algorithms such as neural networks, support vector machines or tree-based models) cost is $O(TL2^M)$ in tree models, being M the number of features.

5 Conclusions and challenges

5.1 Summary of this work

This work explains, based on our experience after several years of collaboration with the utility company Naturgy, the existing challenges to achieving robustness in an autonomous NTL detection system based on supervised methods. The use of observational data entails the existence of biases that, in general, are difficult to detect when using black-box algorithms. We explored different explanatory approaches to make our system transparent, with the aim of better understanding.

Section 4 covers a standard statistical analysis approach (i.e., using Pearson Correlation, Odds-Ratio and feature distribution), to feature importance method from the tree ensemble methods, to finally implement state-of-the-art solutions such as Local Surrogate models (i.e., LIME for tabular data) and Shapley Values (i.e., SHAP for tree models). According to our experience, the information provided by SHAP is the most complete, useful, and reliable. It provides both a complete global explanation of the model and also consistent instance-level explanations. Also, it is free from the stability issues encountered in practice when using LIME. Feature importance also provides a global explanation, but the information provided is generic and cannot be used to analyse specific instances.

Therefore, after the analysis explained in this manuscript, we chose to implement Shapley values to analyse our system.

5.2 Achievements through explainability

The decision of using SHAP as the algorithm to explain our system was a big quality step in our project. As expected, there was a clear improvement in terms of understanding our system. This of course meant a better system overall, since we implemented several improvements in our pre-processing and training stages that made our system better. But it is that in addition that we had many other benefits beyond the improvement in understanding our system. First, explainability meant that the research in our project (e.g., the process of testing improvements such as introducing weights in our labelled information) was much more agile, as with explainability we had a method for testing these methods and their consequences in the predictions, whereas before we had to wait months to see the campaign's results on exploratory campaigns. This agility was also seen in the interaction with the stakeholders. The fact that all the actors in the NTL detection process could be involved, as we explained in Sect. 2.6, meant that more and better improvements could be implemented in our system.

5.3 Challenges and future work

The inclusion of the explanatory algorithm allowed us to improve the system, but it also adds an extra layer to the system that slows down the generation of campaigns. From our point of view as data scientists, this is a minor problem as we prioritise the predictive capacity of the model; introducing the explanatory process, even if it means increasing the time needed to generate a campaign, also means avoiding generating campaigns that, based on the patterns learned, are bad (as they have learned bad patterns that do not generalise correctly) and should not be generated. But sometimes, based on our experience with Naturgy, the generation of campaigns in a specific period of time is mandatory to meet business objectives. Thus, the opinion of the data scientist (i.e., iterate as many times as necessary a model to ensure its correctness) and the need of the company (i.e., generate a campaign at a precise time) come into conflict.

A possible improvement for our system would be to make the whole process of campaign generation and validation more organic and natural to the company's tempo, making model generation and validation faster. Therefore, our future goal is to make the whole process of campaign generation more intuitive for the company's stakeholders.

Acknowledgements This work has been supported by MCIN/AEI funds under grant PID2020-112581GB-C21.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

A explanation of the features commented in the paper

This appendix provides an explanation of all the features mentioned in the plots or examples used along the paper. Unlike other work we have done for the company in gas, we do not deseasonalise the electricity consumption curve, as the differences in consumption between months are mild and do not justify it.

Last Impossible 2: This variable indicates the number of months elapsed since the last time the company could not perform a visit, e.g., installation inspection. For instance, a technician might visit an installation, but might not be able to carry out the inspection because the customer prevents it (e.g., the installation is on their property and does not allow the technician to access it). Suffix 2 indicates that the original purpose of the visit was not to detect an NTL case (referred to below as "no Fraud" visit), but to perform another technical visit.

# Impossible 2: Number of times the company could not perform a visit to the customer. The purpose of the visit was not to detect NTL.

# Correct 2: Number of “no Fraud" visits to the customer with result of “no NTL detected”.

Last Impossible: Number of elapsed months since the last time the company could not perform a visit. In this case we compute both the visits to detect NTL cases but also the other visits (referred in this analysis, as explained before, with the suffix 2).

Current Reading Absences: Number of months since the last meter reading.

Power Contracted: The power contracted by the customer.

Town: Town where the customer lives. This is a categorical variable.

# Visits 2: Number of “no Fraud" visits made to customer. We do not consider the results of the visit, but if the customer has been historically "controlled" by the company.

SCC12MP: Similarity (in terms of consumption curve, 12 months) between the customer and similar customers from the same province. More specifically, we compute the average consumption per month of the customers from the same province and Tariff, normalize the consumption curve, and compute how similar is this normalized consumption curve with the normalized consumption curve of the customer. A low value would indicate that the consumption is similar to the expected consumption curve, while a very high value indicates that there is no similarity. It is expected that a customer would have a consumption curve similar to the customer from the same region and, therefore, a high value shoould be an indicator of NTL.

SCC12M: Similarity (in terms of consumption curve, 12 months) between the customer and similar customers. This variable is computed as explained in **SCC12MP**, but the comparison is done with all the customers included in the campaign, i.e., the customers from the region.

Con.Penultimate Year: Consumption of the customer during the penultimate year. By itself, this information might not provide great information, but it is useful to understand the historical consumption behaviour of the customer and, therefore, can nuance the meaning of consumption behaviours at the present time (e.g., if the customer has a low consumption right now, this low consumption is more suspicious if we see that in the past the customer consumed large amounts of energy).

Last Bill: Amount of energy billed (i.e., kWh) to the customer in the last bill.

Last Fraud 2: Number of months since the last time the company performed a “no Fraud" visit with an NTL result.

Con.Drop 24–6 M Abs: Absolute consumption difference (i.e., kWh) between two consecutive 6-months period of time. It is checked for the last 24 months.

Last Threat: Number of months since the last time the customer threatened a technician from the company to avoid the technical revision of the meter.

#Threat: Number of times the customer has threatened a technician from the company to avoid their technical revision of the meter. A threat indicates that the customer violently prevents the technician from checking the installation, a clear indicator that the customer has tampered with the meter.

Last Threat: Number of months since the last time the customer threatened a technician from the company to avoid the technical revision of the meter.

Energy Cut: Number of times the company cut the energy supply to the customer.

Fraud Building 1Y: Number of times an NTL has been found in the building of the customer during the last year. This information is interesting since between neighbors can share information about how to implement fraud.

#Gas Fraud: Number of times the customer has had NTL cases in gas.

Con.Last Year/CLY Zone: Ratio between the customer consumption and the average consumption in the region (last 12 months). This information is straightforward, i.e., a much lower consumption should be an indicator of NTL.

Con.Low 24–6 M.: Ratio between the customer consumption and the average consumption of similar customers (i.e., customers from the same region and Tariff). The period of time considered is the last 24 months, and the consumption window is 6 months.

Con. Customer/Con. Cust. 24 M: Ratio between the customer consumption and the average consumption in the region (last 24 months).

Median Bill 12 M: Median Bill of the customer for the last 12 months.

References

Alvarez-Melis D, Jaakkola TS (2018) On the robustness of interpretability methods. arXiv preprint arXiv:1806.08049

Angelos EWS, Saavedra OR, Cortés OAC, de Souza AN (2011) Detection and identification of abnormalities in customer consumptions in power distribution systems. IEEE Trans Power Delivery 26(4):2436–2442

Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-López S, Molina D, Benjamins R et al (2020) Explainable artificial intelligence (xai): concepts, taxonomies, opportunities and challenges toward responsible ai. Inf Fusion 58:82–115

Badrinath Krishna V, Weaver GA, Sanders WH (2015) Pca-based method for detecting integrity attacks on advanced metering infrastructure. In: Campos J, Haverkort BR (eds) Quantitative evaluation of systems. Springer International Publishing, Cham, pp 70–85

Bland JM, Altman DG (2000) The odds ratio. BMJ 320(7247):1468

Buzau MM, Tejedor-Aguilera J, Cruz-Romero P, Gómez-Expósito A (2018) Detection of non-technical losses using smart meter data and supervised learning. IEEE Trans Smart Grid PP(99):1–1

Cabral JE, Pinto JO, Martins EM, Pinto AM (2008) Fraud detection in high voltage electricity consumers using data mining. In: IEEE/PES transmission and distribution conference and exposition. IEEE 2008:1–5

Calvo A, Coma-Puig B, Carmona J, Arias M (2020) Knowledge-based segmentation to improve accuracy and explainability in non-technical losses detection. Energies 13(21):5674

Coma-Puig B, Carmona J (2019) Bridging the gap between energy consumption and distribution through non-technical loss detection. Energies 12(9):1748

Coma-Puig B, Carmona J (2021) Non-technical losses detection in energy consumption focusing on energy recovery and explainability. Mach Learn 111:1–31

Coma-Puig B, Carmona J (2018) A quality control method for fraud detection on utility customers without an active contract. In: Proceedings of the 33rd annual ACM symposium on applied computing, ser. SAC '18. New York, NY, USA: ACM, 2018, pp 495–498. [Online]. https://doi.org/10.1145/3167132.3167384

Coma-Puig B, Carmona J (2021) A human-in-the-loop approach based on explainability to improve ntl detection. In: 2021 international conference on data mining workshops (ICDMW). IEEE, 2021, pp 943–950

Coma-Puig B, Carmona J, Gavalda R, Alcoverro S, Martin V (2016) Fraud detection in energy consumption: a supervised approach. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, pp 120–129

Costa BC, Alberto BL, Portela AM, Maduro W, Eler EO (2013) Fraud detection in electric power distribution networks using an ann-based knowledge-discovery process. Int J Artif Intell Appl 4(6):17

Drummond C, Japkowicz N (2010) Warning: statistical benchmarking is addictive. Kicking the habit in machine learning. J Exp Theor Artif Intell 22(1):67–80

Ford V, Siraj A, Eberle W (2014) Smart grid energy fraud detection using artificial neural networks. In: 2014 IEEE symposium on computational intelligence applications in smart grid (CIASG), pp 1–6

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29:1189–1232

Friedman JH, Popescu BE et al (2008) Predictive learning via rule ensembles. Ann Appl Stat 2(3):916–954

Galanti R, Coma-Puig B, de Leoni M, Carmona J, Navarin N (2020) Explainable predictive process monitoring. In: 2020 2nd international conference on process mining (ICPM). IEEE, 2020, pp 1–8

Glauner P, Meira JA, Valtchev P, State R, Bettinger F (2017) The challenge of non-technical loss detection using artificial intelligence: a survey. Int J Comput Intell Syst 10:760–775

Guerrero JI, León C, Monedero I, Biscarri F, Biscarri J (2014) Improving knowledge-based systems with statistical techniques, text mining, and neural networks for non-technical loss detection. Knowl Based Syst 71:376–388

Liu Y, Hu S (2015) Cyberthreat analysis and detection for energy theft in social networking of smart homes. IEEE Trans Comput Soc Syst 2(4):148–158

Lundberg SM, Erion GG, Lee S-I (2018) Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888

Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. Adv Neural Inf Process Syst, 30

Lundberg SM, Nair B, Vavilala MS, Horibe M, Eisses MJ, Adams T, Liston DE, Low DK-W, Newman S-F, Kim J et al (2018) Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nat Biomed Eng 2(10):749

Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee S-I (2020) From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2(1):2522–5839

McLaughlin S, Holbert B, Fawaz A, Berthier R, Zonouz S (2013) A multi-sensor energy theft detection framework for advanced metering infrastructures. IEEE J Sel Areas Commun 31(7):1319–1330

Messinis GM, Hatziargyriou ND (2018) Review of non-technical loss detection methods. Electric Power Syst Res 158:250–266

Molnar C (2020) *Interpretable machine learning*. Lulu. com

Monedero I, Biscarri F, León C, Guerrero JI, Biscarri J, Millán R (2012) Detection of frauds and other non-technical losses in a power utility using pearson coefficient, bayesian networks and decision trees. Int J Electr Power Energy Syst 34(1):90–98

Nagi J, Yap KS, Tiong SK, Ahmed SK, Mohamad M (2009) Nontechnical loss detection for metered customers in power utility using support vector machines. IEEE Trans Power Delivery 25(2):1162–1171

Nagi J, Yap KS, Tiong SK, Ahmed SK, Nagi F (2011) Improving svm-based nontechnical loss detection in power utility using the fuzzy inference system. IEEE Trans Power Delivery 26(2):1284–1285

Northeast group l (2017) Electricity theft and non-technical losses: global markets, solutions, and vendors. [Online]. http://www.northeast-group.com/reports/Brochure-ElectricityTheft&Non-TechnicalLosses-NortheastGroup.pdf

Pearl J (2009) Causality. Cambridge University Press, Cambridge

Pearl J, Mackenzie D (2018) The book of why: the new science of cause and effect. Basic Books

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830

Pereira LAM, Afonso LCS, Papa JP, Vale ZA, Ramos CCO, Gastaldello DS, Souza AN (2013) 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). April 2013, pp 1–6

Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A (2017) Catboost: unbiased boosting with categorical features. 2017

Rehse J-R, Mehdiyev N, Fettke P (2019) Towards explainable process predictions for industry 40 in the dfki-smart-lego-factory. KI - Künstliche Intell 33(2):181–187

Ribeiro MT, Singh S, Guestrin C (2016) “why should I trust you?”: explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, August 13-17, 2016, pp 1135–1144

Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell 1(5):206–215

Salman Saeed M, Mustafa MW, Sheikh UU, Jumani TA, Khan I, Atawneh S, Hamadneh NN (2020) An efficient boosted c5.0 decision-tree-based classification approach for detecting non-technical losses in power utilities. Energies 13(12):3242

Santos RN, Yamouni S, Albiero B, Vicente R, Silva JA, Souza FB, Freitas Souza M, Lei Z (2021) Gradient boosting and shapley additive explanations for fraud detection in electricity distribution grids. Int Trans Electr Energy Syst 31(9):e13046

Shapley LS (1953) A value for n-person games. Contrib Theory Games 2(28):307–317

Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. In: Proceedings of the 34th international conference on machine learning-volume 70. JMLR. org, pp 3145–3153

Shwartz-Ziv R, Armon A (2022) Tabular data: deep learning is not all you need. Inf Fusion 81:84–90

Slack D, Hilgard S, Jia E, Singh S, Lakkaraju H (2020) Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In: Proceedings of the AAAI/ACM conference on AI, ethics, and society, 2020, pp 180–186

Spirić JV, Stanković SS, Dočić MB, Popović TD (2014) Using the rough set theory to detect fraud committed by electricity customers. Int J Electr Power Energy Syst 62:727–734

Zhou Y, Chen X, Zomaya AY, Wang L, Hu S (2015) A dynamic programming algorithm for leveraging probabilistic detection of energy theft in smart home. IEEE Trans Emerg Top Comput 3(4):502–513