An Iterative Approach based on Explainability to Improve the Learning of Fraud Detection Models

Bernat Coma-Puig and Josep Carmona *

September 29, 2020

Abstract

Implementing predictive models in utility companies to detect Non-Technical Losses (i.e. fraud and other meter problems) is challenging: the data available is biased, and the algorithms usually used are black-boxes that can not be either easily trusted or understood by the stakeholders. In this work, we explain our approach to mitigate these problems in a real supervised system to detect non-technical losses for an international utility company from Spain. This approach exploits human knowledge (e.g. from the data scientists or the company's stakeholders), and the information provided by explanatory methods to implement smart feature engineering. This simple, efficient method that can be easily implemented in other industrial projects is tested in a real dataset and the results evidence that the derived prediction model is better in terms of accuracy, interpretability, robustness and flexibility.

1 Introduction

During the last years, utility companies started to use machine learning techniques to detect Non-Technical Losses (NTL)[1] between their customers, a huge problem that has a very high economic cost for these companies[2]. Many of the examples seen in the literature are supervised approaches and the algorithms usually chosen to build the models are Gradient Boosting Ensemble Trees, Deep Learning and Support Vector Machine, non-interpretable black-box algorithms that, in general, might provide higher accuracy than the interpretable algorithms such as Linear Regression or Decision Tree.

An example of the use of a black-box algorithm (in this case, a Gradient Boosting Decision Tree) to detect NTL is the system that the Universitat Politecnica de Catalunya has built for an international utility company from Spain. Our approach [1] has achieved good results, especially considering that it is implemented in a European region with a very low ratio of NTL cases. However, the system has problems, as we detail in Section 3, in terms of fairness and robustness, since the labelled dataset that provides the company is biased; The data-related problems (e.g. dataset-shift) is well-known in the NTL detection literature [2]. The different approaches that we tested to automatically mitigate the bias problems (e.g. weighting the customers to over-represent under-represented customers) had inconclusive results, achieving similar (or even worse) NTL detection in campaigns[3]. Moreover, we also had the typical problems of interpretability in black-box models: we could neither fully understand how these biases affected the model, nor properly report to the com-

*B. Coma-Puig (bcoma@cs.upc.edu) and J. Carmona (jcarmona@cs.upc.edu) are with the Universitat Politècnica de Catalunya, Barcelona, Spain.

[1]NTL refers to the energy loss as a result of meter tampering, other meter inaccuracies or vandalism, as opposed to the Technical Losses that refers to the energy loss due to the transmission or distribution of the energy.

[2]According to a study done by Northeast Group in 2017, the utility companies losses 96 billions of dollars in technical losses only in electricity.

[3]We refer to campaign the selection of customers to be visited

pany the patterns learnt.

To understand better how our model trained, we started to explore the inclusion of explanatory algorithms[4] in our system [1, 3]. Thanks to these algorithms, we have been able to analyse the quality of our models besides the benchmarking, an approach that we consider that has many limitations, especially with biased datasets [2]. Moreover, the explanatory algorithms allowed us to improve the reports that we provide to the company: Reporting how the values of the features influenced in the model's predictions have been useful to understand the company's point of view of the correctness of the models built.

In this work, we present our approach to exploit the information provided by the explanatory algorithm (in our case, the Shapley Values from SHAP [4]) and systematise the system-stakeholder interaction to increase the model success: to convert the process of building the model into an iterative process controlled by the stakeholder in charge of the NTL detection process. This specialist analyses, in each iteration, what the model has learnt and, in case it detects an undesired pattern, a bias or an unused feature, implements feature engineering to improve the model. In section 4 we test this approach in real dataset from the utility company, offering evidence that the resulting model is better in terms of **accuracy**, **robustness**, **interpretability**, **flexibility** and **simplicity**. In Section 5 we analyse this approach from a technical point of view (e.g. why we use the Gradient Boosting and SHAP). Finally, we conclude this work with Section 6, summarising the benefits of our approach and introducing possible future work.

2 Related Work and Preliminaries

2.1 Related Work in NTL detection

In the literature, there existed different examples of supervised systems to detect NTL. In [5], a similar approach to detect NTL cases in Spain is presented that uses XGBoost models; in [6, 7] there are two examples of using Support Vector Machines to detect NTL cases; in [8, 9] and [10], there are three examples of using artificial neural networks to detect NTL, and in [11, 12] two examples of using the Optimal-Path Forest Classifier.

In contrast to the aforementioned supervised techniques, there are also other different unsupervised approaches to detect NTL in the literature. In [13, 14] there are two examples of using detecting NTL using a clustering method; in [15], we can see another approach that uses unsupervised neural networks (Self-Organizing Maps). From a more industrial process control point of view, [16] and [17] are two examples of using a statistical process control method in the detection of anomalies. Other non-supervised approaches are [18] (an example of an expert system) and [19], an approach for analysing the load flow. To obtain a global vision of the NTL detection problem, we highlight [20] that analyses the technical challenges of detecting NTL and [21], a more classical work of summarising the existing approaches in the literature.

Unlike most of the related work in the literature that provides theoretical solutions or experimental analysis with synthetic or static data (i.e. results from one specific campaign done in regions with a high proportion of fraud), our work analyses the long-term problems of implementing an autonomous NTL system that works in different types of customers and regions, allowing us to detect the robustness and interpretability problems that we mitigate with this method.

2.2 Accuracy vs Interpretability and Human Knowledge

There exist in the literature a discussion about the trade-off between accuracy vs interpretability, i.e. the necessity of deciding between using very complex algorithms (e.g. Deep Learning, Ensemble Trees or Support Vector Machines) or more interpretable algorithms like Linear Regression or Decision Trees. In general, this idea of the accuracy vs interpretability is globally accepted (see for instance the DARPA'S XAI program document [22]), but there is still relevant

[4] Algorithms that humanly explain the influence of the features into the predictions made by the predictive models

work (e.g. [23]) that advocates to use interpretable algorithms or, at least, to change the approach of how we use the predictive algorithms, focusing on its interpretation.

In [24] there is a deep analysis of the term *intelligence* in artificial systems, considering insufficient the approach of benchmarking an intelligent system as the *skill* of correctly doing a specific task, e.g. a predictive model that assigns a label. This definition of intelligence masks what the author considers that should define intelligence in artificial intelligence, e.g. the ability of generalisation what the system learns, and how it has to be benchmarked, i.e. against human intelligence.

These two works, the papers from SHAP [4] and LIME [25], combined with other more classical machine learning techniques (e.g. feature selection [26] and active learning [27]), inspired us in the development of our proposal.

3 Our approach in the NTL detection system

3.1 The Supervised Approach

Our supervised approach can be summarised as follows:

1. **Campaign configuration** The stakeholder delimits the segmentation of the campaign (the type of utility, region and tariff), and extracts the data from the company.

2. **User profiling** The features are built to profile the visited customers in the past (which constitutes the labelled information, i.e. the NTL and non-NTL cases), and in the present (the customers to be predicted). In general, the consumption features are the most important, since they reflect the change in consumption behaviour.

3. **Model training and prediction** With the historical profiles, a model is trained, and a prediction is made: Each customer has an estimation of the amount of energy to recover, where a value close to 0 corresponds to a non-NTL case.

4. **Report generation and campaign generation** The top-scored customers are included in a report, and the company decides which customer are visited.

In general, the approaches seen in the literature build classification models (i.e. reduce the NTL detection as a binary classification problem). However, we detected that the binary approach in our system was able to detect NTL cases with very few energy to recover (e.g. close to 0 kWh) by finding patterns that were not related to the consumption of the customer (e.g. where the customer lives); Our work in [28] evidence that the use of regression helps to recover more energy for NTL detection.

3.2 Biases and Lack of Interpretability

Despite the successful campaigns that we have achieved both in electricity and gas [1], our supervised system has always had the following problems:

3.2.1 Data Problems, Biases and Dataset-Shift

As explained in [20], the NTL systems usually face data-related problems, i.e. biases and dataset-shift[5]. This is an intrinsic problem in any system that aims to detect NTL cases for a utility company: In general, the companies build their campaigns including the customers that are suspicious of NTL, intending to maximise the detection of energy to recover. Therefore, most of the customers (i.e. those that are not suspicious) are not properly represented in the system. This is especially true in our system, that aims to NTL in rich regions with very low NTL percentage (both due to the low fraud rate in the population but also because of the meters installed, that are newer and therefore less likely to fail). Moreover,

[5]Dataset-shift occurs when the distribution of the training dataset and the test dataset differs, making it difficult to train robust models.

there are many different company-related decisions in the process of controlling the customers that influence in the decision of visiting a customer.Two examples that contextualise this problem are the following:

- There are regions in which the company is much more successful in detecting NTL cases (because the technicians in that region are better, among other things) and, therefore, those customers are over-represented.
- The company can decide to over-control specific types of customers. For instance, the recidivist customers (i.e. the customers that are constantly committing NTL).

As explained in [1], we mitigated in our system some biases by implementing specific campaigns for the type of customers that were biasing the model. Other more technical approaches were tested (e.g. to apply weights to the under-represented customers) with inconclusive results: either the implementation of these solutions were complicated and required an undesired temporal cost or the campaigns done with these changes achieved worse results.

3.2.2 Interpretability problems and the Black-box Algorithms

The consequence of using a black-box algorithm to detect NTL (in our system, an Ensemble Tree Model) is that we were not able to properly analyse the correctness of our model beyond the typical benchmarking approach (i.e. training-validation-test dataset), an analysis that we consider insufficient, with many limitations [2]. For this reason, we explored the possibilities that explainability offered in our system: In [28] we analyse how the Shapley Values [29] from SHAP [4] can be a proper tool to analyse the fairness and robustness of a model beyond the benchmarking.

Despite this step forward in terms of understanding our system, we have still problems both to fully understand the models built but also to provide a simple explanation to the company's stakeholders. In other words, the Shapley Values provide us explainability, but the number of features used in our system (we currently have 154 features, but we have had several hundred) and the complexity of the patterns learnt makes that the explanation obtained is not often interpretable. Moreover, the fact that the stakeholders are not involved in the training process (i.e. they ask a campaign, and the system generates it without human interaction) difficult their interaction with the model. Two examples of the problems derived from the lack of interpretability are the following:

- The stakeholder's role in the process of generating campaigns is passive: They receive the list of top-scored customers with a report and analyse them. Not being involved in the process of building the model makes that they do not have a global vision of what the model learnt.
- Not being involved in the process of building the model also makes the stakeholders dependant of the data scientists. For instance, if the stakeholders detect a bias, they cannot directly correct it and restart the process of building the model to generate a new campaign.

3.3 Our Proposal: Building NTL Model

3.3.1 The Building Process

To mitigate the bias and interpretability problems in our NTL detection system, we propose to build our Regression models through an iterative structure, similar to a feature selection process. In each iteration, a human expert (in our case, a stakeholder specialist in detecting NTL) analyses through an explanatory algorithm the patterns learnt. In case the human expert detects an undesired pattern, it implements feature engineering to mitigate it. The result of this process explained in Figure 1), is a model that is more fair and robust in real scenarios.

The explanatory algorithm that according to our experience, provides better explanations are the Shapley Values [29] from SHAP [4]. The Shapley Values is a game theory approach to fairly distribute the payout among the players that have collaborated in a cooperative game. The SHAP library adapts this

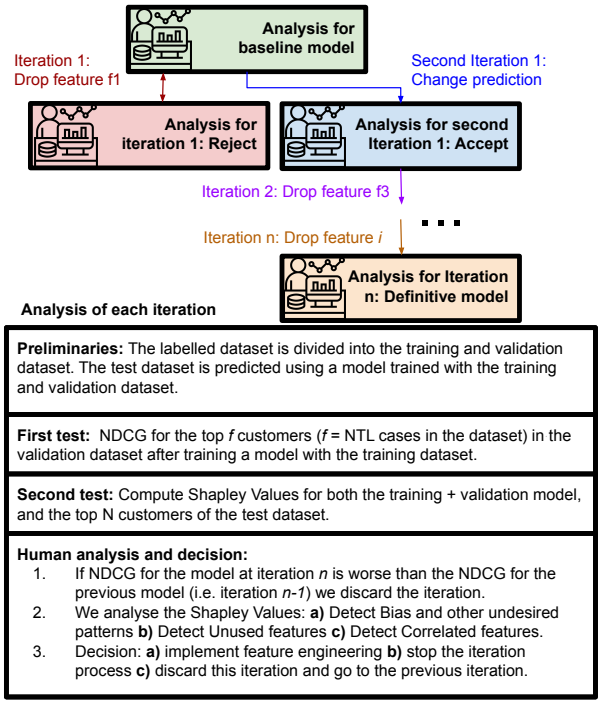

Figure 1: The building process is an iterative process similar to the classical feature selection to exploit human knowledge from the stakeholder. Using the Shapley Values as the explanatory method, it achieves a resulting model that is better in terms of fairness and robustness (it has fewer biases) but also in terms of interpretability (the stakeholder understand better what the model learnt).

idea[6] to determine how the values of the features of an instance x influenced in the prediction made for the supervised model $M(x)$. It is usually defined as follows:

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

In the equation, variable S runs over all possible subsets of feature values, the term $val\left(S \cup \{x_i\}\right) - val(S)$ corresponds to the marginal value of adding x_i in the prediction using only the set of feature values in S, and the term $\frac{|S|!(p-|S|-1)!}{p!}$ corresponds to the permutations that can be done with subset size $|S|$, to weight different sets differently in the formula. This way, all possible subsets of attributes are considered, and the corresponding effect is used to compute the Shapley Value of x_i. In our system, we use the Tree Explainer, the specific method to extract the Shapley Values from Tree Models [30].

In addition to the Shapley Values, we also use the Normalized Discounted Cumulative Gain ($NDCG$, [31]) to obtain a global vision of the quality of the predictions make by a model,

$$NDCG_t = \frac{DCG_t}{iDCG_t}$$

where DCG_t is defined as

$$DCG_t = \frac{\sum_{i=1}^{t} energy_i - 1}{log_2(i+1)}$$

being $energy_i$ the amount of energy recovered in the visit done to the customer ranked at position i, and $iDCG$ corresponds to the maximum DCG possible (i.e. a perfect prediction in terms of order). The $NDCG$, as explained in Figure 1, is used to compare two models directly.

The use of this ranking metric focusing on the energy to recover, as well as the use of the energy to recover as a metric, is justified in **??**, and can be summarised as follows:

- The customers to be visited are unknown (e.g. sometimes the campaigns can go from several dozens to hundreds). Therefore, the $NDCG$ provide a more generic vision of the correction of the model.
- A more classical approach of using (for example) $precision@k$ (i.e. precision at the top k instances) to evaluate a model can tend to exploit the existing biases in the data.
- Although the company evaluates the campaigns according to the NTL cases detected, the final metric used to consider a campaign successful is

[6] SHAP considers the payoff as the prediction and the values of the features the players of the cooperative game.

the amount of energy recovered. That is, is preferred a campaign with only one NTL detected with 10000kWh detected that a campaign with 10 NTL detected with a total of 3000kWh recovered.

3.3.2 The Human Analysis in the Building Process

With the information provided by the Shapley Values and the NDCG metric, the specialist has to analyse in each iteration n the correctness of the model trained in comparison to the previous iteration $n-1$. Also, she has to analyse how the model can be improved in iteration $n+1$ by implementing feature engineering, to mitigate the existing biases, and improve interpretability (i.e. the problems explained in Section 3). The guidelines of this iterative process are explained in Figure 1, and can be summarised as follows:

- The $NDCG_n$ of the validation dataset should not be worse than $NDCG_{n-1}$[7] since this would mean that the model in iteration $n-1$ is better than the model in n. If $NDCG_n << NDCG_{n-1}$, then we should discard the modifications done in that iteration (e.g. the *Drop feature f1* iteration from Figure 1)
- A Shapley Value from a high-scored instance that stands out in comparison to the rest of the Shapley Values can be a consequence of an outlier in the prediction labels (more specifically, an NTL case with a much higher value of kWh recovered than the rest of the NTL cases). In this case, the specialist can opt to reduce the highest prediction value from the training dataset.
- We should remove the instances that are not useful in the model (as the classical feature selection) to increase the interpretability of the model by the stakeholder.
- We should remove correlated features that contribute similarly according to the Shapley Values to guarantee a better model (e.g. avoiding overfitting and the curse of dimensionality) but also increase the interpretability of the model.
- We should remove those features that have unexpected Shapley Values (i.e. undesired patterns). For instance, we should consider removing the feature that profiles how many months the customer has had no consumption if there is a negative correlation between the value of the feature and the Shapley Values[8].

According to our experience, the correction of bias has priority over removing a feature: a bias highly influences in how a model is learnt and, therefore, its correction can cause that a feature with no importance in the biased model to gain relevance in the new model.

4 A case study with a real dataset

In this section, we analyse the benefits of implementing the building process in our current NTL system.

4.1 Preliminaries

4.1.1 The Dataset used

For the case study, we use a real dataset[9] from the utility company with more than 1.000.000 customers[10]. The labelled instances include around 10500 NTL cases, and almost 300000 non-NTL cases and the dataset is split into three sub-datasets: the training (80% of the labelled instances), the validation (10% of the instances) and the test dataset (the remaining 10%). Each partition is stratified (i.e. we keep the positive/negative ratio in each partition). There is no timestamp consideration (i.e. we do not use the last 10% of NTL cases as the test dataset)

[7] We would accept some margin in this description, i.e. we consider that a model is worse in terms of $NDCG$ when the value is significantly lower (at least 0.1 lower).

[8] Because one should consider that a customer that is not consuming anything is suspicious of having NTL.

[9] further information like the region, and the typology of the customers is anonymised to protect the privacy of the data.

[10] The customers are apartments and small houses from the same region of Spain

to guarantee diversity and reduce the differences between the datasets, mitigating the consequence of the decisions made by the stakeholder during the campaign building[11].

4.1.2 The Algorithm, Loss Function and Metric Used

The Gradient Boosting Model trained is a Root Mean Square Error Catboost Regressor, i.e. we consider the problem of detecting NTL as a point-wise ranking problem where we predict the amount of energy to recover for each customer. The methods used to analyse the correctness of our model are the $energy_{200}$, $NDCG$ and the Shapley Values:

- $energy_{200}$: This is a straightforward metric to analyse the amount of energy recovered in a simulated campaign of 200 customers for the test dataset[12]. It is only used this metric to compare the amount of energy recovered for the baseline model and the resulting mode after the process of building the model.

- $NDCG$: The normalised Discounted Cumulative Gain metric is used, as explained in Figure 1, to obtain a generic vision of how well the model orders the validation customers according to the energy to recover.

- We use the summary plot method from SHAP, that provides a global vision of how the values of each instance have influenced in the prediction. In Figure 2, there is an example of how these plots should be read and interpreted. It is used, as explained in Figure 1, to analyse both the patterns learnt in the training dataset but also to analyse the Shapley Values from the top 200 customers in the test dataset.

[11] For instance, if the stakeholder that builds the campaigns decide to visit recidivist customer in July and August, and in September we split the data considering the timestamp, in the test dataset we would have an over-representation of the recidivist customers. Splitting the data randomly would distribute the customers from the July and August between the train, validation and test dataset.

[12] 200 customers corresponds to a normal campaign size.

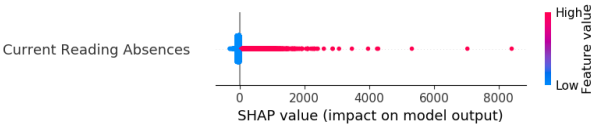

Figure 2: In red there are the high values of the features and, in blue, the low values. In this specific case we can see that having a high value in *Current Reading Absences* (i.e. that the company has several months with no new meter readings) increases the $\hat{y}$ value of the instance.

4.1.3 Semantic Grouping of Features and Evaluation

In our system, we have 154 features that can be summarised as follows:

Consumption Features From the consumption data available, we build the features that should define the consumption behaviour of the customers. These features are numeric.

Raw Consumption: These features refer to the kWh consumed by the customer during a period of time. We include consumption-related information extracted from the difference of meter readings (e.g. the consumption of the customer during the last three months), similar information from the company (i.e. features that are already computed by the company in their databases, such as the consumption of the customer during the last year) or the customer billing.

Consumption changes (the customer against itself): We include several features that compare, through a ratio, the customer consumption in two distinct periods of time, with the aim of detecting a modification in its consumption behaviour.

Consumption anomalies (the customer against other customers): As with the previous type of features, this group of features consists on the ratio between the consumption of the customer against other similar customers in the same period of time.

Consumption curve: These features aim to represent if the customer's consumption curve and the average consumption curve in the same period of time

are similar.

Visit features Most of the features that can be extracted from the visits done by the technicians to the customers are very important for deriving a supervised problem.

Labelled instances: When we profile the customers in month m, all the visits done in that month are the labelled instances for the supervised training stage.

Visit information: The same information used to extract labelled instances are used to build the visit features (e.g., a fraudulent visit in the month m, when we profile the customer at the month m that visit is a label, but at month m+1 becomes a feature).

Static features The static information is used both to segment the customers (e.g., the tariff) and to generate features. Some of these features are categorical (LightGBM and Catboost support this type of feature, but for XGBoost, it would be necessary to one-hot encode them).

Sociological features The aim of including sociological and geographical information is to nuance the final score of the customer; for instance, if we accept the premise that in poorer regions the people may commit more fraud, the system should prioritise the abnormal behaviours from lower incomes.

To facilitate the explanation and readability of this document, we extend our analysis for the visits group, including plots for the Shapley Values in the training and test dataset, before and after the iteration. A brief description of the visit features is the following:

- There is a big group of features that refers to the visit done to the customer: The *Fraud* features (that refers to the detection of NTL in the customer), the *Correct visit* (that refers to the non-NTL cases of the customer), the *Impossible* visit (that corresponds to the visits that had no conclusive result, i.e. neither fraud nor non-fraud), and the *Visit* (that corresponds to all the visits without NTL/Non-NTL distinction):
  - The # prefix refers to the occurrences of that type of visit (e.g. *#Visit* refers to the number of visits the company has done to the customer). Other features indicate the last occurrence of a specific type of visit (e.g. *LastVisit* refers to how many months has passed since the last visit).
  - Some features have different versions of the same idea (e.g. *LastFraud1*, *LastFraud2* and *LastFraud*). Suffix 1 refers to the visits done in campaigns that aimed to detect NTL cases, suffix 2 refers to the visits done with no aim to detect NTL (i.e. generic visits from the company) and the features with no suffix corresponds to the features that groups both types of features.
- There are also features related to the density of fraud around the customer:
  - *#FraudZone*: This feature indicates the number of NTL cases in a customer's zone. A zone is established by the company, and corresponds to a technical term regarding the distribution of the electricity: nearby towns or neighbourhoods in a big city share a zone. A derivative of this information is the *#FraudZone1Year*, that indicates the NTL cases in the last year.
  - *#FraudStreet*: It is the same information than the *#FraudZone* but focused specifically in the street where the customer lives.
  - *#FraudInBuilding*: Similarly, it counts the NTL cases in the building where the customer lives.
- There is a third group that refers to the threats of the customer to the technician, i.e. if the customer violently prevents the installation revision from being executed.

4.2 Tests

In this section, we exemplify the process of building a model by implementing the modifications explained in 3.3: to remove a feature due to its irrelevance, to remove a correlated feature and to unbias the model by correcting an outlier. We compare the baseline

model and the resulting model in terms of $energy_{200}$ to see if, in addition to the improvement in terms of interpretability and bias reduction (that would help to increase the robustness in real campaigns), the resulting model also recovers more energy in the test dataset.

First Model (baseline)

Our first model corresponds to the baseline, i.e. the model that would be used in a campaign before introducing the building process.

- $NDCG$: 0.44 in the validation dataset.
- $energy_{200}$: 249242.9kWh.
- Shapley Values: Figure 3 (training+validation model).

As we can see in Figure 3, we have outliers. This is a consequence of an NTL case with more than 260000kWh recovered, an extremely abnormal case of NTL due to the large amount of energy recovered[13]. To build a more fair model, we reduce the value to predict in this NTL case 4 times (i.e. from 260000 to 66000kWh).

Second Model (First iteration)

This model corresponds to the baseline model + correction of the bias.

- $NDCG$: 0.43 in the validation dataset.
- Shapley Values: Figure 4 (training+validation model).

First of all, we can see that we achieve a similar $NDCG$ value in the validation dataset, i.e. it seems that the unbias does not reduce the prediction capacity of our model. Then, the Shapley Values from Figure 4 seem to indicate that the model learnt is better: there are no outliers (the higher Shapley value is reduced from around 30000 to 5000).

[13]The second NTL case in the dataset is a case in which the company recovered 50000kWh. The typical customer consumption is close to 3500kWh per year

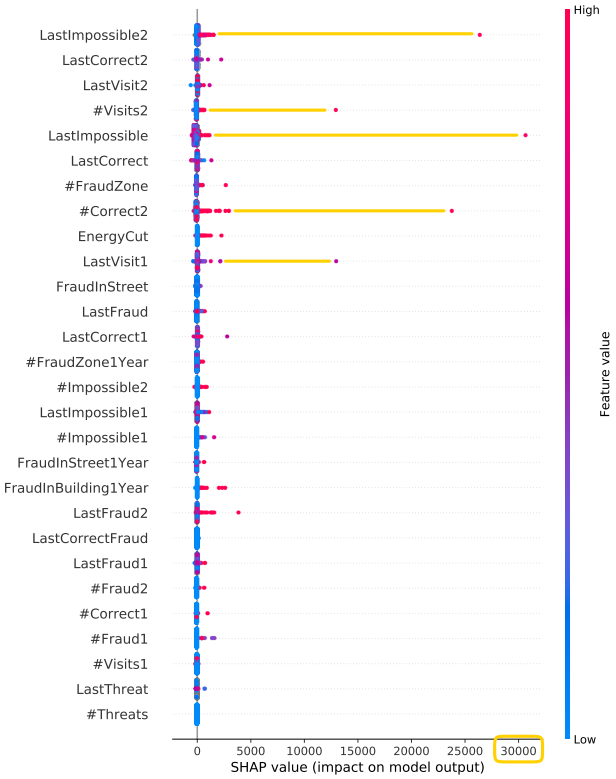

Figure 3: The outliers seen in the image (in yellow) are a consequence of an NTL case in which the amount of energy to recover is higher than 250000kWh when the second higher NTL case goes around 50000kWh. In this situation, the stakeholder in charge of the model building would consider to reduce this prediction value, to build a more unbiased model.

For the next iteration, we opt to drop the less important feature in the model: *#Threats*. This should not modify the model trained, but would simplify the explanation provided to the stakeholders.

Third Model (Second Iteration)

This third model corresponds to the baseline model + correction of the bias + *#threats* drop due to its low relevance.

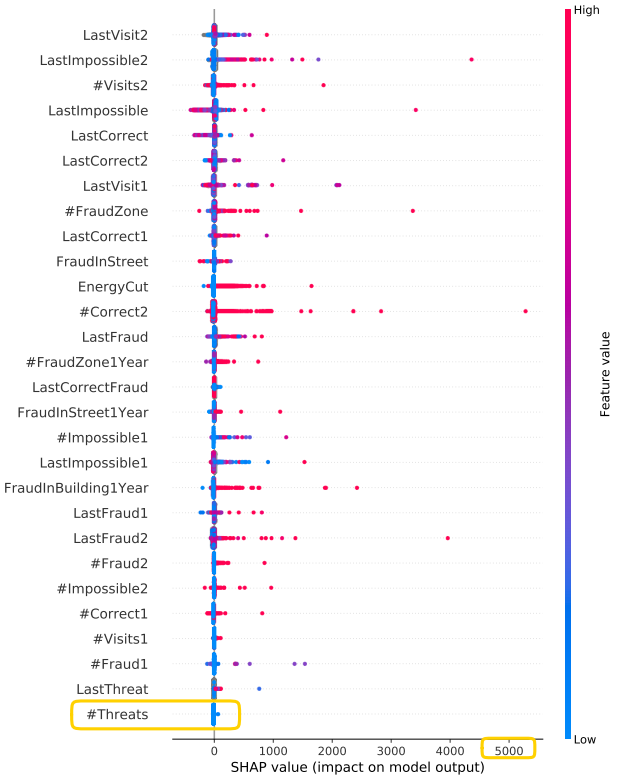

Figure 4: The Shapley Values for the trained model indicates the non-relevance of the *#Threats* feature. Therefore, with the aim of facilitating the interpretation of the model by the stakeholders, we drop this feature from the training process.

- $NDCG$: 0.42 in the validation dataset.
- Shapley Values: Figure 5 (training+validation model and top-scored customers from the test dataset).

Dropping the *#threats* has not changed much what the model has learnt (i.e. the plot from Figure 4 and the left plot from figure 5 are similar), but removing features with low relevancy helps to avoid overfitting and increases the interpretability of the model.

For this third iteration, we exemplify the process of removing a correlated feature from the model. As we can see in the Figure 5, the features *#FraudZone* and *#FraudZone1Year* provide similar information to the learning process globally: A high number of NTL cases in the zone is an indicator of NTL. If we focus on the Shapley Values from the top-scored 200 customers, we can see that the patterns learnt from the *#FraudZone* feature are unclear[14] and, for this case, we would opt to remove the *#FraudZone* feature.

Resulting Model

The resulting model corresponds to the baseline model + correction of the bias + *#threats* drop due to its low relevance + *#FraudZone* drop (correlated with *#FraudZone1Year*).

- $NDCG$: 0.44
- $energy_{200}$: 257038.7kWh

The resulting model exemplifies the benefits of building a model provides in our NTL detection system. We have easily detected and corrected a bias in our labelled dataset, and therefore the resulting model is fairer and should perform better in real-world scenarios. Moreover, we have seen that with only three iterations in our building process, we have increased around 8000kWh the energy recovered. This case study demonstrates that this process guarantees an improvement of the model in terms of **accuracy** and **robustness**.

Regarding the problems derived from using a black-box algorithm (Section 3.2.2), we have seen that the Shapley Values give us a global vision of the patterns learnt by the model, providing **interpretability** to the model learnt by the system. With this information and the iteration structure, we provide a **flexibility** to our NTL detection system, since we can adapt the training process.

Finally, it is necessary to highlight that all these advantages are a consequence of the building process, a very **simple** iteration process that empowers the

[14] From the stakeholder's point of view, it is simpler to explain the *#FraudZone1Year* pattern "high values is an indicator of NTL" than the patterns from *#FraudZone*, that are unclear, where sometimes a high value has positive Shapley Values, and in other cases, it has negative Shapley Values.

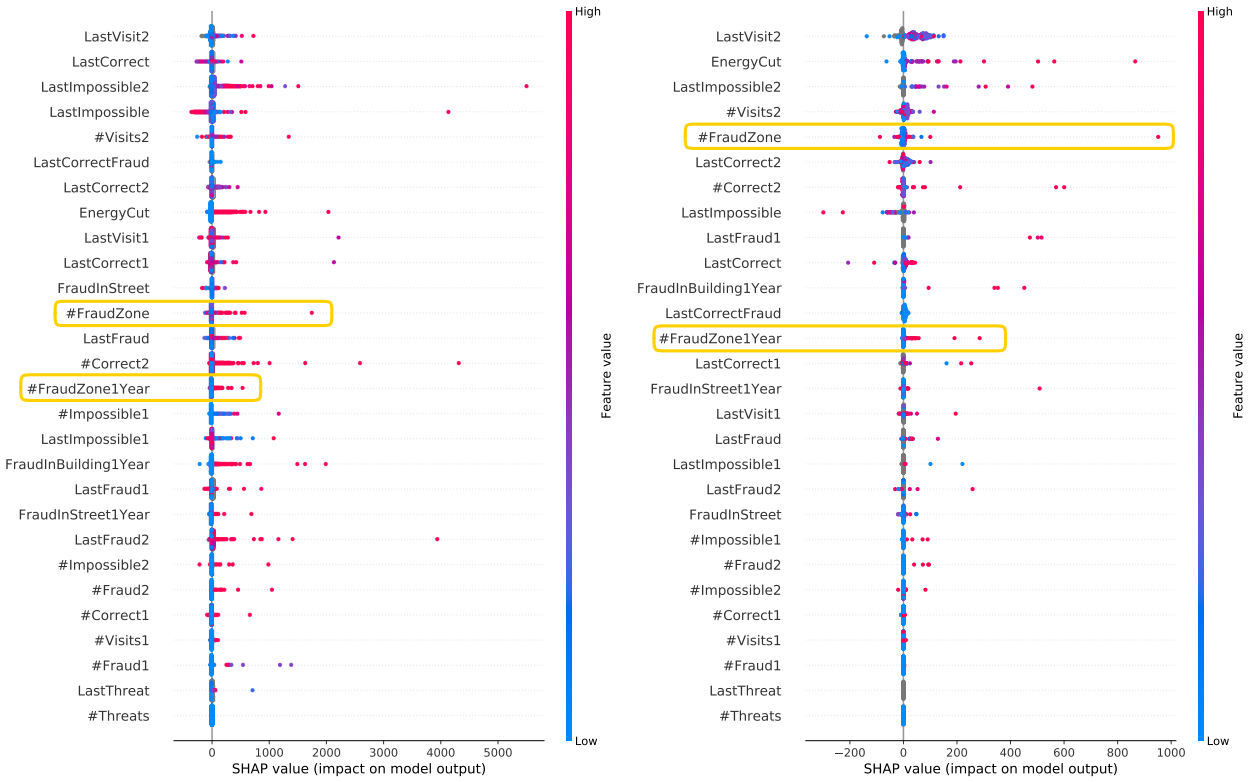

Figure 5: On the left, the Shapley Values for the training dataset and on the right the Shapley Values for the top-scored 200 instances from the test dataset. According to the first image, both features are correlated, therefore might want to remove one of the features to increase interpretability and reduce the curse of dimensionality. If we focus on the Shapley Values in the top-scored 200 customers test dataset, we see that how *#FraudZone1Year* influenced in the prediction is much clearer than in the *#FraudZone* feature and, therefore, we would drop the latter feature from the dataset.

stakeholders to improve a predictive system without the needs of a data scientist.

5 Analysis of this approach

5.1 Correction of bias and assertiveness increase

5.2 Technical Considerations

In this work, we have exemplified the implementation of the building approach on a regression model, but it can also be implemented for a classification (modifying the weights of the instances) and a ranking approach (modifying the priority between the instances).

Regarding the explanatory algorithm, SHAP provides the proper information to successfully implement the building process, since the Shapley Values can be used to analyse the model globally or locally, i.e. allowing contrasting explanations (e.g. to compare the explanations obtained for specific datasets or even to a single data point). This is not possible to achieve with locals models like LIME [25]. However, it is also necessary to highlight that the other implementations to obtain the Shapley Values from SHAP are remarkably slower (e.g. the Deep Explainer method to obtain explanations for an LSTM Deep Learning model takes, at least, several hours to obtain the approximation of the Shapley Values).

If we continue the analysis of our proposal in terms of temporal cost, the process of building a model requires to build different models, and therefore it can take much more time than building one unique model. This problem is mitigated by using GPU accelerated version of the state-of-the-art libraries: In our case, building the CatBoost model using the CUDA acceleration is 4x faster than building using the CPU, taking only 5 minutes to train a model with several hundred thousand labelled instances with 154 features.

Finally, we have focused the benefits of using the Shapley Values to tune the model in terms of feature engineering, but it can also be useful to tune the model itself, i.e. the parameter settings. For instance, instead of using the typical Grid-Search approach to determine the optimal depth of the trees, we could use the Shapley Values to determine if the patterns learnt are better.

6 Conclusions and Future Work

The problem of building an autonomous NTL detection system for a utility company has, as it is well-known in the literature, challenges in terms of robustness due to data problems. This, and the difficulty of being able to involve the stakeholder when it used this autonomous system when the predictive model uses black-box algorithms have been two of the most challenging problems that we have had in the development of an NTL system for an international utility company from Spain. In this work, we propose our method to mitigate these two problems, a process we refer to build a model. This method is easy to implement, easy to interpret and use by the stakeholders. Moreover, it can be easily implemented in many different industrial projects where there is a human specialist, for instance, in healthcare. In Section 4, we evidenced that this approach that shares concepts with classical methods like feature engineering or feature selection but is improved with the inclusion of explainability provides better predictive models.

According to our experience, using artificial intelligence techniques in industrial processes can produce uncertainty, since the stakeholders do not properly understand how the predictive models predict, especially when a black-box algorithm is used. This work is a first step in empowering the stakeholders and increase the confidence between the company and the use of autonomous predictive algorithms.

Future work would focus on two aspects. In the short term, our effort will focus on improving this system-stakeholder interaction based on the stakeholder's feedback. In the long term, we will explore if the system can robustly assist the Stakeholder by suggesting the modifications needed to achieve more robust models or directly if the process can be automatised with an expert system.

Acknowledgements

This work has been supported by MINECO and FEDER funds under grant TIN2017-86727-C2-1-R, and a collaboration with Naturgy.

References

[1] B. Coma-Puig and J. Carmona, "Bridging the gap between energy consumption and distribution through non-technical loss detection," *Energies*, vol. 12, no. 9, 2019. [Online]. Available: http://www.mdpi.com/1996-1073/12/9/1748

[2] C. Drummond and N. Japkowicz, "Warning: statistical benchmarking is addictive. kicking the habit in machine learning," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 22, no. 1, pp. 67–80, 2010.

[3] B. Coma-Puig and J. Carmona, "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]. Available: http://doi.acm.org/10.1145/3167132.3167384

[4] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.

[5] 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. PP, no. 99, pp. 1–1, 2018.

[6] 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, 2009.

[7] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and F. Nagi, "Improving svm-based nontechnical loss detection in power utility using the fuzzy inference system," *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 1284–1285, April 2011.

[8] 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*, vol. 4, no. 6, p. 17, 2013.

[9] L. A. M. Pereira, L. C. S. Afonso, J. P. Papa, Z. A. Vale, C. C. O. 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)*, April 2013, pp. 1–6.

[10] 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)*, Dec 2014, pp. 1–6.

[11] C. C. O. Ramos, A. N. de Sousa, J. P. Papa, and A. X. Falcao, "A new approach for nontechnical losses detection based on optimum-path forest," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 181–189, Feb 2011.

[12] C. C. O. Ramos, D. Rodrigues, A. N. de Souza, and J. P. Papa, "On the study of commercial losses in brazil: A binary black hole algorithm for theft characterization," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 676–683, March 2018.

[13] V. Badrinath Krishna, G. A. Weaver, and W. H. Sanders, "Pca-based method for detecting integrity attacks on advanced metering infrastructure," in *Quantitative Evaluation of Systems*, J. Campos and B. R. Haverkort, Eds. Cham: Springer International Publishing, 2015, pp. 70–85.

[14] E. W. S. Angelos, O. R. Saavedra, O. A. C. Cortés, and A. N. de Souza, "Detection and

identification of abnormalities in customer consumptions in power distribution systems," *IEEE Transactions on Power Delivery*, vol. 26, no. 4, pp. 2436–2442, Oct 2011.

[15] J. E. Cabral, J. O. Pinto, E. M. Martins, and A. M. Pinto, "Fraud detection in high voltage electricity consumers using data mining," in *2008 IEEE/PES Transmission and Distribution Conference and Exposition*. IEEE, 2008, pp. 1–5.

[16] J. V. Spirić, M. B. Dočić, and S. S. Stanković, "Fraud detection in registered electricity time series," *International Journal of Electrical Power & Energy Systems*, vol. 71, pp. 42–50, 2015.

[17] Y. Liu and S. Hu, "Cyberthreat analysis and detection for energy theft in social networking of smart homes," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 148–158, 2015.

[18] S.-J. Chen, T.-S. Zhan, C.-H. Huang, J.-L. Chen, and C.-H. Lin, "Nontechnical loss and outage detection using fractional-order self-synchronization error-based fuzzy petri nets in micro-distribution systems," *IEEE Transactions on smart grid*, vol. 6, no. 1, pp. 411–420, 2015.

[19] P. Kadurek, J. Blom, J. 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.

[20] 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, pp. 760–775, 2017/01. [Online]. Available: https://doi.org/10.2991/ijcis.2017.10.1.51

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

[22] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, 2017.

[23] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[24] F. Chollet, "On the measure of intelligence," 2019.

[25] M. T. Ribeiro, S. Singh, and C. Guestrin, ""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*, 2016, pp. 1135–1144.

[26] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[27] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[28] B. Coma-Puig and J. Carmona, "Regression and explainability for improving non-technical losses detection in energy consumption," to be submitted.

[29] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[30] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," *arXiv preprint arXiv:1802.03888*, 2018.

[31] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.