Fraud Detection Using Event Logs with LSTM and Gradient Boosting

Emiliano Acevedo
Electrical Engineering Institute
Universidad de la República
Montevideo, Uruguay
eacevedo@fing.edu.uy

Pablo Massaferro
Electrical Engineering Institute
Universidad de la República
Montevideo, Uruguay
pmassaferro@fing.edu.uy

Alicia Fernández
Electrical Engineering Institute
Universidad de la República
Montevideo, Uruguay
alicia@fing.edu.uy

Alexander Martins
Energy recover Department
UTE
Montevideo, Uruguay
amartins@ute.com.uy

Gonzalo Caudullo
Energy recover Department
UTE
Montevideo, Uruguay
gcaudullo@ute.com.uy

***Abstract*—Automatic non-technical power loss detection methods have advanced significantly as data volume has increased with smart meter installation. Recently, academic works have mainly focused on the impact of the high resolution of the energy consumption time series, leaving aside the integration of event logs within machine learning solutions. Due to the variety of alarms and depending on electrical installation health, millions of alarm events can be generated requiring an automatic analysis of them. In this work, we propose a method that considers the sequential nature of alarm log information using a recurrent neural network and evaluate two strategies for including this information within an existing state-of-the-art NTL classifier. The experiments are reported in actual smart meter data provided by the Uruguayan utility, showing that it is possible to double the precision for on-field applicable operating thresholds.**

I. Introduction

Automatic fraud detection and prevention have been very active research topics in recent years, both at academic and business levels [1], [2]. In particular, in the detection of irregularities with the consumption of electrical energy, numerous proposals have been made, from combinatorial optimization approaches and the use of AMI network topologies to complex deep learning systems that combine numerous data sources [3], [4], [5], [6], [7]. However, very little has been written about the use of alarms events generated by smart meters. This may be due to the assumption that an alarm of tampering or leakage of current in a device necessarily configures fraud. The devices used for telemetering by energy distribution companies, known as Smart Meters (SMs), have a series of protections designed to limit or detect their manipulation. Contact sensors within the protection systems are included to detect the opening of the terminal board cover or the meter casing itself. In addition, they include current measurement in all power phases (single-phase or three-phase), which allows measuring the differential current. Setting a threshold for said leakage current can detect problems in the installation or even some types of fraud. Since the beginning of the massive installation of SMs in Uruguay by UTE in 2019, a huge volume of alarm data began to be generated. In particular, the differential current recording generated millions of events in a short time.

In practice, there can be many reasons why an alarm is generated, and this can lead to false positives, generating unnecessary on-site inspections and cost overruns for the utility. Until now, we have not found academic work that studies the data patterns of the alarms generated by smart meters. In [8], the occurrence of alarms is included within the NTL detection system. The consumption measurement is taken on a daily basis, each with a binary vector of alarms. The authors use the encoding of the IEC 870-5-102 standard, where an 8-bit code defines the quality of each measurement. Two characteristics are extracted from these alarms at different time intervals, the number of occurrences and the time since the last occurrence. These features are inputs to train an XGB (extreme gradient boosting) ensemble. Decision tree ensembles have recently been widely used algorithms to address the NTL detection problem, either trained directly on the [9], [10] data or combined with deep learning algorithms to extract features [5], [11]. For example, [9] combines the use of XGB, LigthGB and CatBoost for simulated fraud detection based on CER [12]. Buzau et al. [8] work with a database of 57k clients with real data and integrate different sources of information to train classification models. Although [8] uses information about some alarms generated by SMs, it does not consider this data's

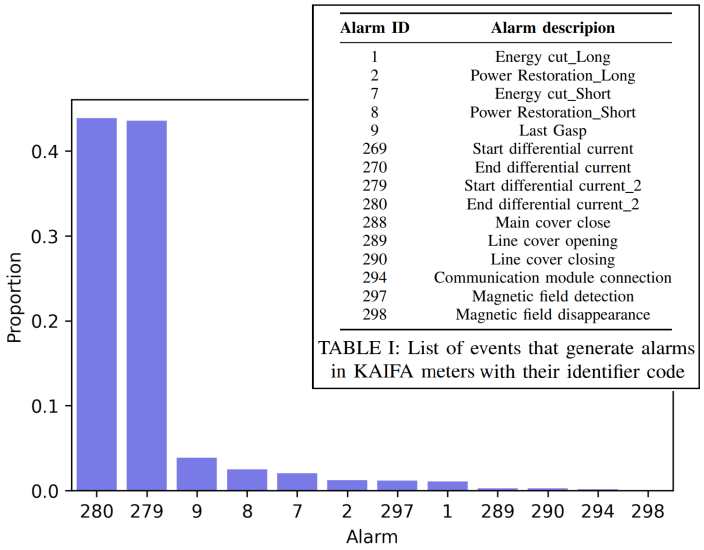
sequential nature. On the other hand, LSTM-based deep learning architectures have also been used to address the NTL problem. In [13], they propose using a two-input architecture. In the first input, an LSTM network is fed with weekly data vectors containing the daily consumption, the number of null measurements, and the missing data for that week. The second input consists of normalized categorical variables that feed an MLP.

The main contributions of this work are: (i) The proposal of two ways of extracting relevant characteristics from the alarm logs and evaluating their discrimination capacity. (ii) Analysis of performance variation with the amount of historical information. (iii) Two strategies for including the features related to alarm records in an existing deep learning NTL detection system. (iv) Experiments in real smart meter data provided by a utility.

II. Smart Meter alarm log

In addition to performing remote measurements of active and reactive energy, smart meters are equipped with sensors that allow the detection of tampering. As it was previously mentioned, these equipment includes sensors capable of detecting the opening and closing at the connection terminal cover (Terminal cover) and a sensor for opening and closing of the main cover (Top cover). Unlike other previous systems, the SM measures energy in all power phases by detecting differential currents. The differential current (current reverse) may be due to problems within the customer's facilities (eg, current leakage to earth in electrical appliances) or a bypass on the meter's terminal board. The types of alarms vary depending on the SM model used. The data for this work was acquired within the framework of a collaborative effort between our department and UTE, the Uruguayan State power generation and distribution company. The company currently has nearly 700,000 meters installed throughout the country, mostly KAIFA model MA110PU . For more information about the meter, visit the site http://kaifametering.com/. Table I presents the complete list of alarms used in this work with their identifier code.

A differential current event begins when the current difference between phases module exceeds a threshold. In the case of UTE, the equipment was configured to generate an alarm when the differential current module exceeds 15% of the phase current from a total consumption of at least 500mA. Figure 1 shows the normalized histogram of alarms. It is seen that the differential current is the event with the highest number of occurrences, while the disappearance of the magnetic field is the least frequent. Most detected events correspond to the occurrence of a phenomenon and its restoration, for example, differential current start and end or the opening and closing of a cover. Unlike the fifteen-minute period energy consumption measurements, alarm data does not have a defined cadence, as they are events generated in response to a certain condition. These events can be generated by scheduled operating activities of the company itself, measurement errors (bad contact of switches), problems in the customer's electrical installations (leaks in appliances) or improper manipulation of the meters by the customers or third parties.



Alarm ID	Alarm descripion
1	Energy cut_Long
2	Power Restoration_Long
7	Energy cut_Short
8	Power Restoration_Short
9	Last Gasp
269	Start differential current
270	End differential current
279	Start differential current_2
280	End differential current_2
288	Main cover close
289	Line cover opening
290	Line cover closing
294	Communication module connection
297	Magnetic field detection
298	Magnetic field disappearance

TABLE I: List of events that generate alarms in KAIFA meters with their identifier code

Fig. 1: Event distribution according to alarm type.

III. Relevant Feature Extraction

The occurrence of alarms holds relevant information regarding the NTL problem. Figure 2 shows the distribution of the number of differential current alarms according to the class. In order for the distributions to be comparable, they were normalized so that the area under the histogram integrates to 1. It is shown that the more alarms the client has, the greater the probability of being a fraud. This fact also occurs with the other alarms. As a result, the number of occurrences of each alarm will be used as an input characteristic. Unlike in [8] where the number of days with events are counted, in this work we will make the total count of events.

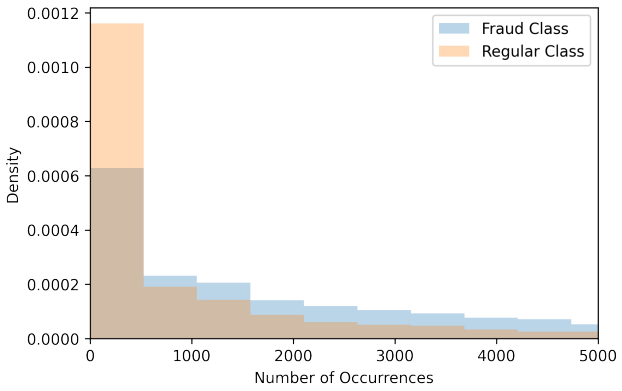


Fig. 2: Distribution of differential current events by class

Most of the detected events correspond to the occurrence of a phenomenon and its restoration as can be seen in Table I,

for example, start and end of differential current or opening and closing of a protection cover. These pairs of events have almost total correlation, and the differences that may exist are due to the problem of missing data. Since these events encode activation intervals, we define a new feature that represents the total activation time for each alarm. In order to calculate this feature, it is necessary to define a time window and accumulate the total activation time. In other words, given a time window, we integrate the number of minutes from the beginning of the alarm pair to its respective end.

Given the occurrence and the total activation time of the alarms, two approaches are proposed in this work to include them into a ML algorithm:

A. Classic Approach - SM_Logs

The *SM_Logs* consists of a vector that includes the accumulated number of event occurrences for all the events in the Table I and the total activation time for the events of differential current, magnetic field and connection cover opening. These features are computed using the latest year (50 weeks) of reported data.

B. Time Sequence Based Approach - SM_SeqLog

The *SM_SeqLog* are computed from a two-step process. First, using a weekly time window we calculate the accumulated number of event occurrence for all the events in the Table I and the total activation time for the events of differential current, magnetic field and connection cover opening. This can be seen as the calculation of *SM_Logs* by week. Unlike the previous approach, we obtain a variable sequence length vector for each feature. Secondly, we fed a two-layer 32-unit LSTM model preceded by a BatchNormalization layer with the features computed in the first step. This model generates a feature vector of lenght 32 that we call *SM_SeqLog* (See Figure 3). This approach allows to characterize the variable length alarm sequences of each client in fixed length vectors. Figure 3 corresponds to a schematic diagram of the feature extraction process. As mentioned, if the weekly vectors of the number of occurrences by events and the total activation time are accumulated, a vector is obtained that we call *SM_Logs*, while if we use the weekly vectors as inputs of a two-layer 32-unit LSTM model, we obtain the *SM_SeqLog*.

IV. Experiments

In this section we compare the proposed approaches and assesses the effect of adding these features to a detection classifier: DAICE (see section IV-E) [14]. In addition, the effect of varying the number of weeks on calculating features in the *SM_SeqLog* approach is analyzed.

A. Database

We use labeled data of 20,695 service points (SP). This database includes information on all the inspections carried out by UTE at customer facilities with SM between January 2019

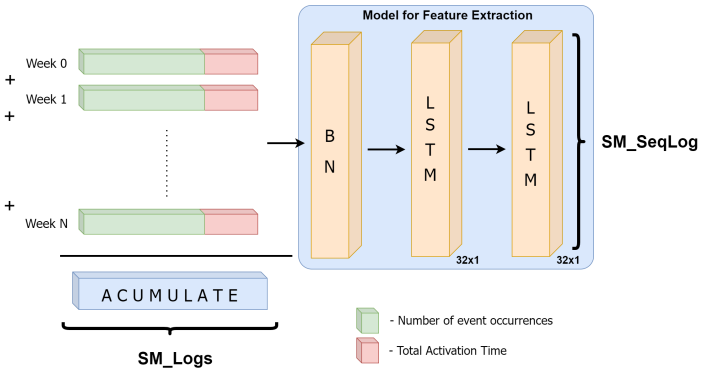


Fig. 3: Feature Extraction Diagram

and March 2022. The irregularity rate is 8% and 94% of SPs have the registration of at least one alarm event. The start date of the alarms log sequence is considered from the first occurrence of an event. The data length available for each client is variable since the SMs have been installed at different dates. We noted that in our dataset, only a few clients have more than 100 weeks of alarm logs. The database also has the active energy consumption history and a set of additional information previously used by the DAICE system. For example, Contracted Power represents the maximum power contracted by the client; (latitude, longitude) the the geographical location of the meter; Late Payment the accumulated days of delay of bills payment; and Fraud History the number of previous irregularities detected, among others [7].

B. Performance Metrics

The choice of performance metrics when dealing with unbalanced class problems is not obvious. In NTL detection problems it is common to use AUC_PR and AUC_ROC as metrics [7], [13], [14]. The AUC_PR represents the area under the precision recall curve while the AUC_ROC the area under the true positive rate (TPR) and false positive rate (FPR) curve.

We also report metrics such as Recall, Precision and F-measure at a given operation point. This is noted as $P@\beta$ which mean the precision obtained when performing inspections by labeling $\beta\%$ of the samples as positive. This is analogous for recall metric.

C. SM_Logs versus SM_SeqLog

Taking the earliest 50 weeks of SM alarm logs, the features *SM_Logs* and *SM_SeqLog* are computed. With them, we train an XGB algorithm to compare the performance of each approach. Using the weekly time sequences, the two-layer LSTM model to extract the *SM_SeqLog* features was trained with 100 epochs. A neuron was added to the output in training to perform the classification. The results obtained on a test basis of three thousand PS are reported in Table II. Superior performance is achieved when using the *SM_SeqLog* approach, obtaining an increase of five percentage points in the AUC_PR metric and the precision

at 10% of the base compared to *SM_Logs* approach. Given the advantages of the *SM_SeqLog* approach, it was chosen as the base algorithm for the inclusion of alarms in the DAICE fraud detection pipeline.

Model	P@10	R@10	$F_{Measure}$	AUC_PR	AUC_ROC
SM_Logs	0,24	0,36	0,29	0,23	0,68
SM_SeqLog	**0,29**	**0,45**	**0,35**	**0,28**	**0,73**

TABLE II: Performance of a XGB classifier with the proposed features: *SM_Logs* or *SM_SeqLog*.

D. Effect of sequence length

Before including *SM_SeqLog* features within the DAICE classification model we first analyzed the impact of alarm sequence length in this approach. Figure 4 shows the precision-recall (PR) curves obtained using *SM_SeqLog* with different numbers of weeks. Specifically, the training set uses from 10 to 100 weeks evenly spaced every 10 weeks. This Figure shows an increase in the area under the PR curve when the number of weeks increases. However, it is seen that after about 50 weeks the improvement is marginal. Table III shows the performances obtained by training with different lengths of sequences. It is observed that using 50 weeks compared to 10 increases the PR_AUC metric by eleven percentage points and the precision at 10% of the base by twelve percentage points. The results shown in Table III reinforces that after 50 weeks the performance improvement is marginal. Even though PR_AUC continues to rise with the number of weeks, the rest of the metrics do not significantly increase except for $P@1$. Particularly, it can be noted that $P@1$ actually increases with the number of weeks.

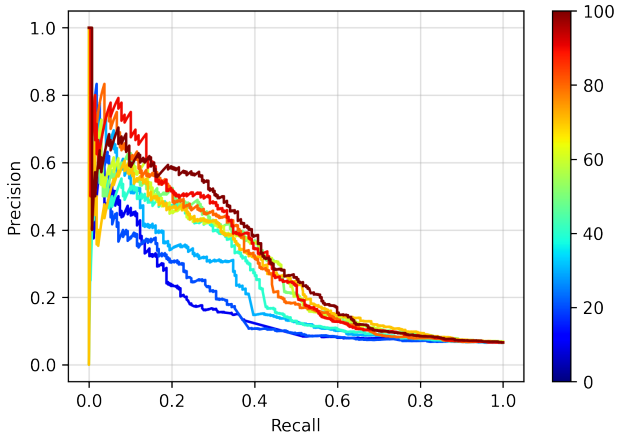


Fig. 4: Performance of the XGB classifier versus the number of weeks considered in the calculation of *SM_SeqLog*

E. DAICE system

The DAICE system refers to an XGB model that uses as input the consumption of the last 36 months and a set of ten additional characteristics of the SP, which are: latitude, longitude, days since the last inspection, number of previous irregularities, maximum power contracted, number of current reads made, the status of agreement at the time of inspection, payment delay and days since the start and renewal of service contract. The operation of the DAICE system and the contribution of the additional characteristics can be seen in greater detail in [14] where Table II shows that this model obtains the best performance in terms of AUC_PR and AUC_ROC.

Model	P@1	R@1	P@10	R@10	PR_AUC	ROC_AUC
10 weeks	0.47	0.07	0.17	0.27	0.17	0.62
20 weeks	0.45	0.07	0.20	0.31	0.18	0.62
30 weeks	0.57	0.09	0.24	0.36	0.22	0.66
40 weeks	0.53	0.08	0.26	0.40	0.24	0.68
50 weeks	0.62	0.09	0.29	0.45	0.28	0.73
60 weeks	0.57	0.09	**0.30**	**0.46**	0.28	0.72
70 weeks	0.57	0.09	0.28	0.43	0.27	**0.74**
80 weeks	0.65	**0.10**	0.28	0.43	0.29	0.70
90 weeks	**0.68**	**0.10**	0.29	0.44	**0.31**	0.71
100 weeks	0.65	**0.10**	0.29	0.45	**0.31**	0.73

TABLE III: Performance of the XGB classifier versus the number of weeks considered in the calculation of *SM_SeqLog*

F. Integration of SM_SeqLog features to DAICE system

We propose and compare two architectures capable of integrating the *SM_SeqLog* features extracted from the alarms with other relevant information. The performance of these architectures is compared to the results obtained by the detection system without the use of alarm data (DAICE). The proposed architectures consist of two different classification strategies, first an MLP (DeepDAICE_SL) and then an XGB (ExtendedDAICE). In both cases, the coded information of the alarms was used, along with the rest of the information used by DAICE. Figure 5 shows a flowchart of the whole process for the case of ExtendedDAICE since the clasifier is an XGB.

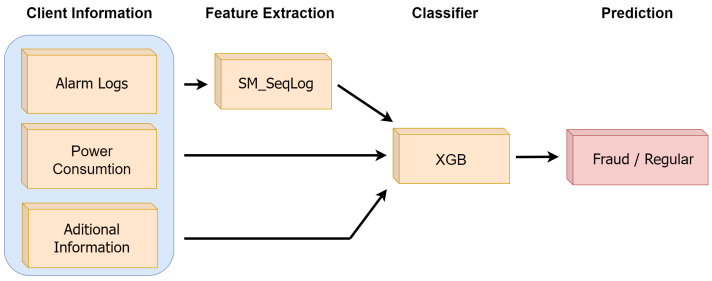


Fig. 5: Flowchart of ExtendedDAICE whole process of Fraud Detection.

Figure 6 shows the precision-recall curves for the two approaches and their comparison with the performance of the DAICE system without including alarm data. Clearly the performance of the system increases significantly when this information is included.

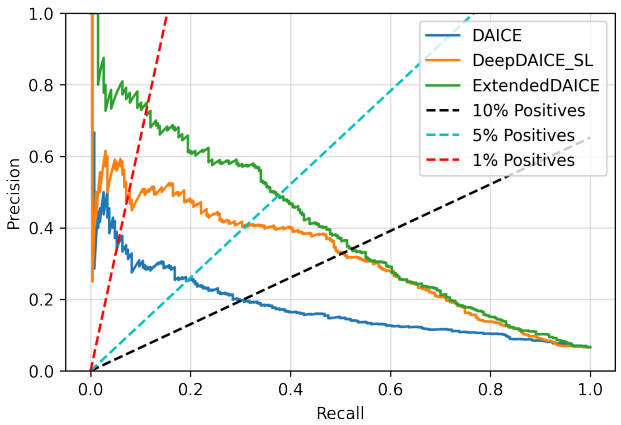


Fig. 6: Precision - Recall curve when integrating alarm information in the DAICE system

Table IV presents the performance metrics of the classification algorithms, the global performance metrics show better results when using ExtendedDAICE (see AUC_PR and AUC_ROC).

Model	P@10	R@10	$F_{Measure}$	AUC_PR	AUC_ROC
DAICE	0,20	0,30	0,24	0,18	0,72
DeepDAICE_SL	0,33	0,51	0,40	0,32	0,81
ExtendedDAICE	**0,34**	**0,52**	**0,41**	**0,40**	**0,83**

TABLE IV: Performance comparison of two architectures for integrating the alarm information in the DAICE system

Table V, presents precision and recall at different operating points. It is observed a substantial increase in precision with the use of ExtendedDaice both when inspecting 1% and 5% of the base, almost doubling the performance of the DAICE system.

Model	P@1	R@1	P@5	R@5
DAICE	0,35	0,05	0,25	0,20
DeepDAICE_SL	0.48	0,07	0,41	0,31
ExtendedDAICE	**0,75**	**0,11**	**0,49**	**0,37**

TABLE V: Performance comparison of two architectures integrating the alarm information in the DAICE system at different operating points

V. Conclusions

This paper shows that performing feature engineering based on the information provided by the smart meter alarm logs contributes to improving the performance of anomaly detection algorithms. The article proposes a set of novel features derived from alarm events, considering the temporal dependency. In addition, we proved that the inclusion of these features achieves a substantial improvement over state-of-the-art algorithms that base their detection on information derived from consumption measurements and do not consider information from alarm logs. In particular, a considerable improvement in precision has been seen when operating at low operating points, doubling the performance compared to the DAICE system. The improvement in detecting true positives without the cost of higher false positives is especially noteworthy, which also translates into a significant increase in AUC_PR and AUC_ROC.

Acknowledgment

We would like to thank UTE for funding the project as well as providing the datasets and sharing their expertise on the problem.

References

[1] K. G. Al-Hashedi and P. Magalingam, "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019," *Computer Science Review*, vol. 40, p. 100402, 2021.

[2] F. de Souza Savian, J. C. M. Siluk, T. B. Garlet, F. M. do Nascimento, J. R. Pinheiro, and Z. Vale, "Non-technical losses: A systematic contemporary article review," *Renewable and Sustainable Energy Reviews*, vol. 147, p. 111205, 2021.

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

[4] M. Hasan, R. N. Toma, A.-A. Nahid, M. Islam, J.-M. Kim, *et al.*, "Electricity theft detection in smart grid systems: a cnn-lstm based approach," *Energies*, vol. 12, no. 17, p. 3310, 2019.

[5] A. Aldegheishem, M. Anwar, N. Javaid, N. Alrajeh, M. Shafiq, and H. Ahmed, "Towards sustainable energy efficiency with intelligent electricity theft detection in smart grids emphasising enhanced neural networks," *IEEE Access*, vol. 9, pp. 25036–25061, 2021.

[6] W. Hu, Y. Yang, J. Wang, X. Huang, and Z. Cheng, "Understanding electricity-theft behavior via multi-source data," in *Proceedings of The Web Conference 2020*, pp. 2264–2274, 2020.

[7] P. Massaferro, J. M. Di Martino, and A. Fernández, "Fraud detection on power grids while transitioning to smart meters by leveraging multi-resolution consumption data," *IEEE Transactions on Smart Grids (accepted)*, 2022.

[8] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Detection of non-technical losses using smart meter data and supervised learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2661–2670, 2018.

[9] R. Punmiya and S. Choe, "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2326–2329, 2019.

[10] S. K. Gunturi and D. Sarkar, "Ensemble machine learning models for the detection of energy theft," *Electric Power Systems Research*, vol. 192, p. 106904, 2021.

[11] S. Li, Y. Han, X. Yao, S. Yingchen, J. Wang, and Q. Zhao, "Electricity theft detection in power grids with deep learning and random forests," *Journal of Electrical and Computer Engineering*, vol. 2019, 2019.

[12] Commission for Energy Regulation (CER), "CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010 [dataset].," 2012. 1st Edition. Irish Social Science Data Archive. SN:0012-00. https://www.ucd.ie/issda/data/commissionforenergyregulationcer/.

[13] M.-M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Hybrid deep neural networks for detection of non-technical losses in electricity smart meters," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1254–1263, 2019.

[14] P. Massaferro, J. M. Di Martino, and A. Fernández, "Ntl detection: Overview of classic and dnn-based approaches on a labeled dataset of 311k customers," in *2021 IEEE NA Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2021.