ANOMALY DETECTION IN SMART METERS

Thanasis Vafeiadis[1], Anastasios Alexiadis[1], Valia Dimaridou[1], Stelios Krinidis[1,2], Kostas Kitsikoudis[1], Lambros Makris[1], Danijel Davidović [3], Dimosthenis Ioannidis[1], Dimitrios Tzovaras[1]

[1] – Information Technologies Institute (ITI), CERTH Thessaloniki 57001, Greece
{thanvaf, talex, valia, krinidis, kzkitsik, lmak, djoannid, dimitrios.tzovaras}@iti.gr

[2] – Pragma-IoT Solutions IKE, Perikleous 153 str. Drosia distr., 57001, Thesaloniki, Greece
krinidis@pragma-iot.com

[3] – Elektro Ljubljana d.d., Slovenska cesta 56, 1000 Ljubljana
danijel.davidovic@elektro-ljubljana.si

Abstract – In this work, a successful project regarding the detection of abnormal events in the smart meters of the network of a large Slovenian power distribution company is presented, via conducting a large anomaly detection study on the data from the entire smart sensors (meters) network, namely the recorded events of the smart meters. By analysing the data from smart meters and utilizing statistical, machine and deep learning techniques and especially Autoencoders the proposed system is able to detect anomalies on each smart meter as well as on the distribution network, fast and accurate.

Keywords: Electricity, anomaly detection, smart meters, machine learning, big data, data analytics.

1 INTRODUCTION

In recent years, smart metering is at the heart of energy transition and sharing from energy suppliers Electricity Distribution Operators (DSOs) to customers and local authorities. In general, smart meters collect a vast amount of every day data on customer level electricity consumption. Anomaly detection raises the major problem, namely the detection of abnormal events or unusual consumption behaviours.

Electricity companies face a lot of anomalies in the energy distribution grid. Some of them have been eliminated by remote electricity measuring (metering), but still a lot of anomalies/ problems remain (e.g. broken smart meters, electricity theft, etc.). On one hand, remote sensing can provide a huge amount of data about the grid, but, on the other, human intervention is still needed to read, analyse, understand and detect anomalies. Thousands of records should be manually checked for identifying potential anomalies. The majority of the electricity companies and providers that are equipped with similar sensors, do not even process these data. Thus, there is a high need for an automatic process.

The main goal is to process and analyse data from smart meter in real-time. Smart metes are the next generation of energy meters that can record electricity consumption and power quality remotely. Generally, the technology around smart meters delivers significant benefits to consumers as well as a more sustainable energy management and an increased security of energy supply [1, 2]. To this end, a variety of algorithms has been developed and tested capturing different type of errors. The first algorithm identifies opponent events performed without work orders. For example, if a smart sensor produces events, where one is the opposite of the previous one (e.g. open cover and close cover), and there is no respective work order at this sensor, then an error is recorded. Another algorithm identifies successive events. If specific events are performed in a specific sequence within a specific time period (e.g. open cover, no power, power, close cover), then this is marked as an error. Another type of error is the identification of erroneous smart meters. When a smart meter produces a large number of events for a period of time, then an error is also identified. Furthermore, if a specific error is produced by multiple smart sensors at the same time, then we may have an error at the distribution network.

There is great amount of work in scientific literature regarding anomaly detection in the smart energy sector. In [3], the authors propose a prediction-based detection system which is a supervised machine learning method that includes a training and a detection process. This anomaly detection system is built on the lambda architecture [4] type with real time capability of big data. The authors in [5] compared clustering, regression and entropy methods for detecting anomalous electricity consumption. The utilization of normal distribution for anomaly classification is an effective statistical method and is widely used [3, 6, 7, 8]. However, statistical methods have a more supportive role in anomaly detection and have to be used with other methodologies and techniques, such as clustering or regression. The authors in [7] used clustering along with a statistical method to find outliers in electricity consumption time series from smart homes. In [9], the authors detect electricity intensity anomalies based on seasonal patterns.

The remainder of this paper is organized as follows. In Section II, the different data sources and types of IoT sensors are discussed. Data analytics tools, developed and tailored to the needs and characteristics of this project, are described in Section III, while in Section IV the visualization and statistics of the proposed solutions are presented. Finally, we draw our conclusions with a brief outlook on future work in Section V.

2 DATA SOURCES - ERROR TYPES - IOT SENSORS

2.1 Data sources description

Data is comprised by events produced by smart meters. The whole dataset consists from a whole year recording (~40 millions events), which are produced from 141.433 smart meters. Each smart meter has the ability to produce 160 distinct events which are of various types (Section 2.2), and can be categorized into three main different groups: *Error*, *Quality*, and *Control*. These events have different importance, which is categorized into three different importance levels: *Low*, *Medium* and *High*.

2.2 Error types

The major errors, that can be detected analysing these events, are *opposite errors* occurred without any work order, *successive errors, network quality errors* and *smart meter errors*. Some examples of opposite events are: *Power down L1* opposite to (o.t.) *Power restored L1*, *Terminal cover removed* o.t. *Terminal cover closed*, *Under voltage on L1* o.t. *No under voltage on L1 anymore*, *Power down* o.t. *Power up*, *Power down* o.t. *Power up after long power down*, etc. Overall, there are 32 different combinations of opposite cases/ errors. Some examples of successive cases/ errors are: (1) *Power down* → (2) *Terminal cover removed (more than 1 minute delay between*

(1) and (2)) → (3) *Terminal cover closed* → (4) *Power up* or *Power up after long power down,* (1) *Terminal cover removed* → (2) *Power down L1* or *L2* or *L3 (more than 1 minute delay between (1) and (2))* → (3) *Power restored L1* or *L2* or *L3,* (1) *Terminal cover removed* → (2) *Missing voltage L1* or *L2* or *L3(more than 1 minute delay between (1) and (2))* → (3) *Power restored L1* or *L2* or *L3* etc. Overall, there are 472 different combinations of successive cases/ errors. The sequences of successive cases vary from two to four events per case. Furthermore, network quality errors occur when many smart meters produce the same error, which can be translated that these events are not errors from smart meters but errors from the distribution network. These types of errors are very important for the DSOs, since they can check the quality and status of the distribution network at real-time. Finally, smart meter errors appear when an abnormal number (usually very large) of events is occurred in a smart meter.

2.3 IoT smart meters

Smart meter is the fundamental component of an advanced metering system. Smart meters, not only provide us data remotely, but also access and capabilities (e.g. remote connection/ disconnection of the power supply, etc.) that requires too much time with the traditional ways, that is site visits. One type of data that are collected from the smart sensors are events. Almost all distribution and electricity companies do not exploit this kind of data, but only electricity data are utilized. However, these types of events could help the company to maintain the metering infrastructure in an efficient way and to explore the status of the sensors as well as the distribution network. Events give us information about missing voltage by phase, power down, working of breaker, strong DC field, open cover of meter, local communication establishment and many other events. Usefulness of events can improve maintenance of smart meters in near future, while fraud, broken smart meters, and other erroneous situation can be detected at real-time. The proposed methodology for anomaly detection based on the recorder events of smart meters is suitable for every smart meter that provide similar type of events.

3 ANOMALY DETECTION ALGORITHMS

3.1 Successive Events/ Errors

This kind of errors can be detected utilizing a windowed greedy algorithm, which search along the sequence of the events produced by a sensor. The algorithm looks in the overall sequence of the events for specific patterns. There are 472 patterns identified so far. The algorithm can be described by the following equation:

$$\sum_{m=0}^{N-1} |p_m - e_{i+m}| \quad \forall i \tag{1}$$

where e_i is the i^{th} event, p_m is the m^{th} component of the pattern, *N* is the number of elements in the pattern, while |.| is the Boolean process that matches two identical events. When (1) is equal to zero, then a perfect match exists, and a successive error is identified. This process is repeated for all events produced by all sensors.

3.2 Opponent Events/ Errors

These errors occurred when a sensor produces two successive events that are opponent, e.g. cover open & close, power down and up, etc. There are 32 different combinations of opponent errors. These kinds of errors are identified through a greedy search at the event sequence, where each pair of events is checked:

$$e_i \neq e_{i+1} \quad \forall i \tag{2}$$

where e_i is the i^{th} event, *and* ≠ is the Boolean process that matches two opponent events. When (2) is triggered, then an opponent error exists. This process is repeated for all events produced by all sensors for a specific period.

3.3 Smart Meter Errors

In this case, an algorithm has been developed searching for smart maters with erroneous behaviour, e.g. a smart meter that produces more than 500 events in specific time period. Thus, the events produced by a sensor for a specific time slot are encountered:

$$f(tp) = \sum_{i=1}^{tp} h(e_i) \tag{3}$$

where e_i is the i^{th} event, *tp* is the desired time period, and *h(.)* is the enumeration function. When *f(tp)* is greater than a predefined threshold, then a smart meter error is identified. This process is repeated for all smart meters for the desired time period. The threshold is fully parameterizable and can be modified from time to time, as well as from sensor to sensor.

3.4 Network Quality Errors

In this case, the algorithm is looking for network quality errors. This kind of errors can be identified, when multiple smart meters produce the same error at the same time. For example, if multiple sensors produce a "power L1 problem", at the same time, then this means that it is not a problem of the sensors, but a problem of the network. Thus, the number of sensors that produce similar events are encountered:

$$g(j) = \sum_{k=1}^{SM} \left(\sum_{j=1}^{tp} h(e_j^k) \right) \quad (4)$$

where e_j^k is the i^{th} event produced by the k^{th} smart meter, *tp* is the desired time period, *SM* is the total number of smart meters, *h(.)* is the enumeration function, and *g(j)* is the total number of sensors produced the specific e_j^k event. When *g(j)* is greater than a threshold, then the network is assumed that it has a problem. The threshold is fully parameterizable and can be modified according to the network.

3.5 Proposed Error Sequences

As described in Section 2.2, a list of successive cases that frequently occur in smart meters was created using visual (manual) observation of each meters' errors. However, such a process is time consuming and error prone, since there is no guarantee that all the possible successive cases will be taken into account. To this end, a statistical methodology based on frequencies of occurrence of events is proposed, that processes all the event types that occurred in each smart meter, and tries to identify some new, unknown and not expected successive cases. The suggested approach is applied for every smart meter, and it is based on (1).
The suggested approach initially identifies all possible events sequences of length two, three and four, such as the given length from error types. The time difference between the first and the last event in the sequence is also calculated. All successive cases, known and unknown (or recommended) that occur for the tested each smart meter are merged, so that to avoid duplicates. Subsequently, a list with the error series, its frequency, and the mean and median time interval between the first and the last event is provided. Finally, the suggestions for all the smart meters are merged, so that a final list with recommendations to be constructed. For each type, and each suggestion, we present the total number of times that the proposed sequence occurred on all sensors, the number of different sensors it appeared and a mean and median interval between the first and the last error. Table 1 shows three examples of sequences of events containing two, three and four events that were detected by the current methodology.

Table 1. Proposed error sequences.

Event 1	Event 2	Event 3	Event 4	Mean minutes	Median minutes	Number of Sensors	Number of appearances
Phase leakage	Power down	-	-	18.94	0	8819	9103
Invalid start-up sequence	Maximum demand exceeded	Maximum demand OK	-	29.26	16	208	340
Power down	Remote communication ok	Terminal cover closed	Adjust time/date	21.67	3.0	210	210

3.6 Autoencoders

Autoencoders are used in order to help with maintenance by differentiating the sensors that needed servicing. The purpose of this pattern analysis is to seek for event patterns that indicate that a sensor might go wrong, and therefore better organize its maintenance. The proposed approach is as follows: for each observation the sensor id and event id are listed while some new features are created. The new features are: (a) the time difference from the previous value of the same sensor, (b) the time difference from start of day, (c) the time difference from last measurement of the same sensor (d) and the time difference from the last work order of the same sensor. The previous event id from the last measurement of the same sensor, is also added.
The training dataset consists of 5.024.241 event observations, out of which the 22.668 are work orders. A deep autoencoder is used and trained for approximately 3 days and 300 epochs to MSE loss 0.00079. To evaluate the performance of the model, the work orders that were due to service are coloured in red. In Figure 1 the results of the application of the autoencoder are depicted. Most red points were gathered in specific areas of the presented

clusters, which means that the model succeeds to identify the sensors that will need servicing in short time period. Figure 1 also visualizes the sensors that had been recently service in blue colour, mainly gathered in the third cluster. Those, also validate the results of the autoencoder due to the fact that they are mapped far away from red points (sensors that need service).

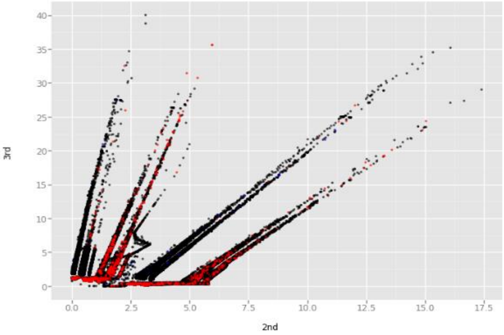
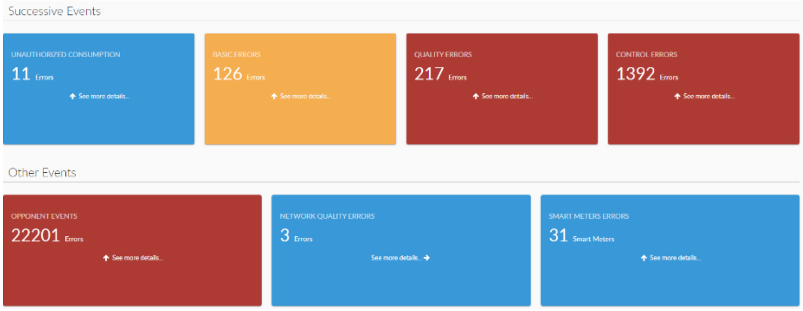


Figure 1: Latent variables of the autoencoder output layer.

4 EXPERIMENTAL RESULTS – VISUALIZATIONS

This section focuses on real-time data analysis, so that companies that use smart meters, can easily monitor the current behaviour of their devices (smart meters). To this end, a visualization platform has been developed, so as to aggregate the errors from sensors' network and to show the results of the analysis that was proposed in Section III. Figure 2, shows the main dashboard of the proposed platform, where the end-user can observe the general state of all sensors in a nutshell, while at the same time being able to navigate through different dates. For the selected time period, the backend pulls all the error data from the database, and performs the beforementioned analysis, in order to visualize the successive events summary of all the installed sensors, split into 4 main categories. Furthermore, summary of other type of events, such as opposite and similar events are depicted.



Successive Events

UNAUTHORIZED CONSUMPTION
11 Errors
See more details...

BASIC ERRORS
126 Errors
See more details...

QUALITY ERRORS
217 Errors
See more details...

CONTROL ERRORS
1392 Errors
See more details...

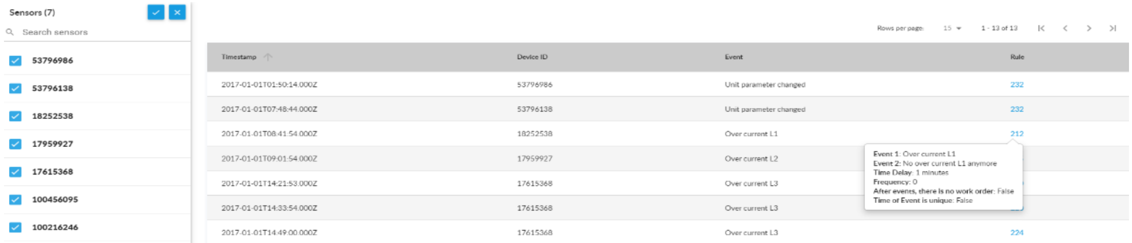
Other Events

OPPONENT EVENTS
22201 Errors
See more details...

NETWORK QUALITY ERRORS
3 Errors
See more details...

SMART METERS ERRORS
31 Smart Meters
See more details...

Figure 2: Main dashboard of the proposed platform. Anomalies in a nutshell.



Sensors (7)

- ☑ 53796986
- ☑ 53796138
- ☑ 18252538
- ☑ 17959927
- ☑ 17615368
- ☑ 100456095
- ☑ 100216246

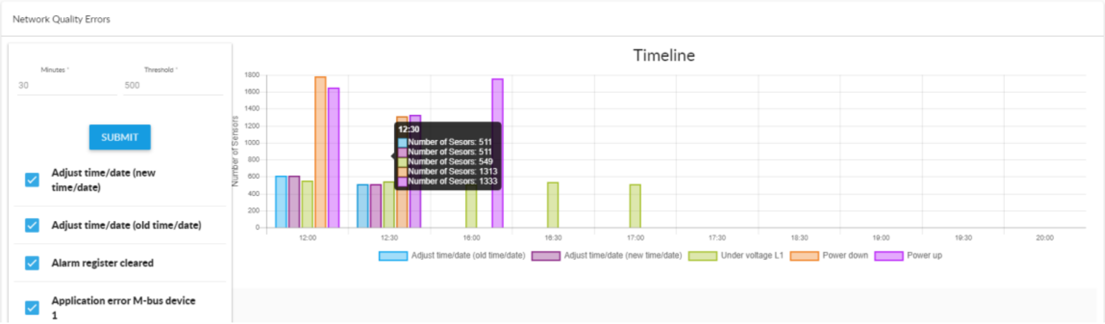
Timestamp	Device ID	Event	Rule
2017-01-01T01:50:14.000Z	53796986	Unit parameter changed	232
2017-01-01T07:48:44.000Z	53796138	Unit parameter changed	232
2017-01-01T08:41:54.000Z	18252538	Over current L1	212
2017-01-01T09:01:54.000Z	17959927	Over current L2	[illegible]
2017-01-01T14:21:53.000Z	17615368	Over current L3	[illegible]
2017-01-01T14:33:54.000Z	17615368	Over current L3	[illegible]
2017-01-01T14:49:00.000Z	17615368	Over current L3	224

Event 1: Over current L1
Event 2: No over current L1 anymore
Time Delay: 1 minutes
Frequency: 0
After events, there is no work order: False
Time of Event is unique: False

Figure 3: Unauthorized consumption dashboard.

While the main dashboard can provide an overview of the sensors' state, more details are required by specialized users. The user has the option to choose the type of events he/she is mostly interested in, and acquire more information about it. As an example, Figure 3 provides a drill-in view from unauthorized consumption event type from the successive events/ errors category. All the sensors that presented unauthorized consumption are showed, for the selected dates. For each error, the platform presents the details of the rule that was detected, as well as information about the rule type. The user has also the option to filter the sensors and isolate selected sensors from the sensor selection list (left column of the dashboard). Similar dashboards can be provided for all four categories of successive events.

Another example of data visualization is illustrated at Figure 4, where the end-user can navigate at the network quality errors detection (Section 2.4), the similar events identification. The results are provided in a histogram-like visualization for the selection period, where the number of sensors with similar events occurred at the same time are presented. The user has the option to easily adjust the time period and the minimum number of sensors the events appear.



Network Quality Errors

Minutes * 30

Threshold * 500

SUBMIT

☑ Adjust time/date (new time/date)

☑ Adjust time/date (old time/date)

☑ Alarm register cleared

☑ Application error M-bus device 1

Figure 4: Network quality errors dashboard.

5 CONCLUSIONS

The data analysis and visualization system has been applied and validated in a Slovenian power distribution company. The main contribution of this work along with the develop algorithm for anomaly detection is the real-time data analysis. The benefits gained so far are the vast amount of errors detected in the whole sensors network and the proposition for new and unknown possible errors. Furthermore, the manual process required for detecting similar events for one day is more than a month, with high costs in personnel, low accuracy and huge amount of electricity loss, especially in cases where a fraud exists. Finally, the proposed algorithms presents a high potential for the predictive maintenance activities for the smart sensor network. Future work is the extension of these anomaly detection algorithms aiming, firstly to deal with more complex errors cases.

ACKNOWLEDGMENT

This work is partially supported by the "EDI – European Data Incubator" project funded by the EU H2020 Programme under grant agreement no. 779790, and by "enCOMPASS – Collaborative Recommendations and Adaptive Control for Personalised Energy Saving"' project funded by the EU H2020 Programme, grant agreement no. 723059.

REFERENCE

[1] DEPURU S.S.S.R., WANG L. DEVABHAKTUNI V.: Smart meters for power grid: challenges, issues, advantages and status. Renewable and sustainable energy reviews, 16(6), 2011, 2736-2742.

[2] McLOUGHLIN F., DUFFY A., CONLON M.: A clustering approach to domestic electricity load profile characterisation using smart metering data, Applied Energy, 141, 2015, 190-199.

[3] LIU X., NIELSEN P.S.: Scalable prediction-based online anomaly detection for smart meters data, Information system, 77, 2018.

[4] LIU X., IFTIKHAR N., NIELSEN P.S., HELLER A.: Online anomaly energy consumption detection using lambda architecture, International Conference on Big Data Analytics and Knowledge Discovery, 2016.

[5] ZHANG Y., CHEN W., BLACK J.: Anomaly detection in premises energy consumption data, Power and Energy Society General Meeting, 2011, 1-8.

[6] CHOU J.S., TELAGA A.S.: Real-time detection of anomalous power consumption, Renewable and Sustainable Energy Reviews, 33, 2014, 400-411.

[7] JAKKULA V., COOK D.: Outlier detection in smart environment structured power datasets, International Conference on Intelligent Environments, 2010, 29-33.

[8] JANETZKO H., STOFFEL F., MITTELSTADT S., KEIM D.A.: Anomaly detection for visual analytics of power consumption data, Computers & Graphics, 38, 2014, 27-37.

[9] HERRERIAS M.J.: Seasonal anomalies in electricity intensity across Chinese regions, Applied Energy, 112, 2013, 1548-1557.