

# A Large-scale Replication of Smart Grids Power Consumption Anomaly Detection

Bruno Rossi<sup>1,2</sup> 

<sup>1</sup>*Faculty of Informatics, Masaryk University, Brno, Czech Republic*

<sup>2</sup>*Institute of Computer Science, Masaryk University, Brno, Czech Republic*

**Keywords:** Smart Grids, Smart Meters, Anomaly Detection, Power Consumption, Replication Study.

**Abstract:** Anomaly detection plays a significant role in the area of Smart Grids: many algorithms were devised and applied, from intrusion detection to power consumption anomalies identification. In this paper, we focus on detecting anomalies from smart meters power consumption data traces. The goal of this paper is to replicate to a much larger dataset a previously proposed approach by Chou and Telaga (2014) based on ARIMA models. In particular, we investigate different model training approaches and the distribution of anomalies, putting forward several lessons learned. We found the method applicable also to the larger dataset. Fine-tuning the parameters showed that adopting an accumulating window strategy did not bring benefits in terms of RMSE. While a  $2\sigma$  rule seemed too strict for anomaly identification for the dataset.

## 1 INTRODUCTION


An anomaly (also known as an outlier, deviation, data irregularity, among other synonyms) has been defined as any data point significantly different from the remaining datapoints (Aggarwal, 2015). The importance of identifying such elements is given by the fact that we can discover unexpected behaviours in the process that generated the data under analysis (e.g., a series of readings from sensor data to represent a failure in the sensor itself or a network trace that signals an intrusion in the system). This is the main reason why anomaly detection has gained more and more importance in recent years in many fields, like network intrusion detection, law enforcement to detect criminal activities, detection of anomalies in IoT devices communication / behaviour, and many more (Cramer et al., 2018; Caithness and Wallom, 2018).

In the context of Smart Grids (SG), varieties of data analysis approaches have been applied to disparate research problems, such as power consumption forecasting, demand response optimization, non-intrusive appliances load monitoring, false data injection attacks (Rossi and Chren, 2019). In this paper, we focus in the area of power consumption anomaly detection. Power consumption data traces are typically collected from smart metering devices that signal the levels of energy usages within (smart) homes (Chren et al., 2016). In this area, anomaly detection is a very

relevant activity, as the identification of anomalies can give indications about potentially malicious actions (Jung et al., 2019), such as data injection attacks or data theft. For this reason, many techniques have been adopted (Zhang et al., 2011; Chou and Telaga, 2014; Saad and Sisworahardjo, 2017; Sial et al., 2018; Buzau et al., 2018; Liu and Nielsen, 2016; Rossi et al., 2016; García et al., 2018).

The main goal of this paper is to provide a replication of a previous approach (Chou and Telaga, 2014) for the detection of anomalies in power consumption data. Running replications of previous empirical studies is a popular way to increase the confidence in the results of previous studies (Gómez et al., 2010). However, common in replication is the modification of some conditions of the original study (e.g., study subjects) (Juristo and Vegas, 2011), to look at the impact of variations of the context to the final results. Our independent external replication is focused on running the same approach (with some small inevitable variations discussed in the threats to validity sections), but on a different (larger) dataset than in the original paper. By running the replication, we can summarize then lessons learned based both on the results of the replication and on running the experimentation on a different dataset. Furthermore, we can discuss some of the main algorithms proposed over time and their peculiarities. We have the following contributions:

- The application of a previous method of power consumption anomaly identification (Chou and

<sup>a</sup>  <https://orcid.org/0000-0002-8659-1520>

Telaga, 2014) to a large dataset of power consumption traces (more than 9M events from the Smart\* dataset (Barker et al., 2012) vs. 171K events processed in the original paper), looking at the impact of a different training process and different thresholding;

- A series of lessons learned derived by the application of the approach. Looking at aspects such as automation, streaming data in the context of current algorithms;

The paper is structured as follows. Section 2 discusses background about power consumption anomaly detection, providing some of the main algorithms that were proposed over the time for anomaly detection in the area. In Section 3, we report about the results of re-implementation of the algorithms in (Chou and Telaga, 2014) for power consumption anomaly detection applied to a larger dataset. In Section 4, we provide the lessons learned both from the literature review and the experimental results. Section 5 concludes the paper with also some future works.

## 2 POWER CONSUMPTION ANOMALY DETECTION ALGORITHMS

To show how anomaly detection is a non-trivial activity, some authors introduce a categorization of different types of anomalies: point anomalies, context anomalies, and collective anomalies (Chandola et al., 2009; Aggarwal, 2015). The usual understanding about anomalies might be about single point anomalies, that is a single instance that could be considered anomalous from the rest of the data (e.g., by some frequency threshold). Context anomalies take into account other factors, so the same data point, for example, might or might not be an anomaly depending on seasonality of time series. Collective anomalies move away from these concepts and consider a data point as an anomaly only if taking place within specific patterns identified over time. This categorization gives the clear idea that, in some cases, the identification of anomalies by looking at threshold intervals, might not be enough.

There are different ways in which an anomaly can be detected. Some probabilistic models might look into data distributions, defining properties for anomalies, other models might look at data proximities (such as k-nearest neighbour, looking at distances of k-data points in space), others might look into the time series properties (Aggarwal, 2015).

In the areas of power consumption anomaly detection, the goal is to detect anomalous data traces that could represent a relevant domain event, like an attempt to steal energy, to tamper with smart meters, or simply a device failure. In this area, time series are typically used to represent power traces changing over the time with the option to use multiple time series (e.g., power traces and weather data) to identify data anomalies.

Discussing some of the representative models in time (Table 1), there is a significant variation of the models applied. In general terms, some models look at some statistical properties, with temporal aspects that are taken into account by the majority of the models. At the base level, we can distinguish between several models for anomaly detection: *linear* (e.g., regression-based), *proximity* (e.g., k-nearest neighbour), *statistical* (e.g., two sigma rule), *density-based* (e.g., clustering). However, a clear-cut distinction might be complex, as multiple approaches might be combined in ensemble models (e.g., using some time series forecasting method with some extreme values anomaly detection approach).

There are further distinctions we can make about algorithms for power consumption anomaly detection. On one side, each algorithm might be different in the level of granularity of the results, e.g., one technique might be just binary (*anomaly / normal*), while other techniques might have some level of abnormality, that allows *ranking* of the anomalies (Aggarwal, 2015). Another distinction is about whether the technique is *supervised* or *unsupervised* (or in-between, semi-supervised). Supervised anomaly detection relies on the availability of labelled instances of anomalies, while such information is not available to unsupervised methods.

Initial models, such as the one in (Zhang et al., 2011), were using estimation of regression models for power consumption and temperature, marking as anomalies all data points in which estimated values and real values were deviating by a pre-defined threshold. A similar approach was used in (Chou and Telaga, 2014)—the approach replicated in this paper. In this case, however, models built were based on AutoRegressive Integrated Moving Average (ARIMA) on the single power consumption time series. Similar approach based on Periodic Auto Regression with exogenous variables (PARX) was proposed in (Ardakanian et al., 2014), and then replicated / improved in (Liu and Nielsen, 2016). Other approaches took into account clustering of power consumption data by temporal properties (Saad and Sisworahardjo, 2017), or even data anomaly detection heuristics based on the domain, like in (Sial et al., 2018), proposing grouping

Table 1: Some Representative Algorithms Applied for Power Consumption Anomaly Detection (S=supervised, U=unsupervised, Bi=binary Output, Rk=ranking Output).

Algorithm	Type	U	S	Bi	Rk	Authors
Long Short Term Memory (LSTM) Network - clustering user profiles and looking at errors from LSTM regression	Proximity + linear models	✓	—	✓	—	(Fenza et al., 2019)
K-NN + SVM + Linear Regression + XGBoost	Proximity	—	✓	✓	—	(Buzau et al., 2018)
Heuristic based on clustering + k-Nearest Neighbour (k-NN)	Extreme Values + Proximity	✓	—	—	✓	(Sial et al., 2018)
Contextual clustering based anomaly score	Proximity	✓	—	—	✓	(Saad and Siswora-hardjo, 2017)
Periodic Auto Regression with exogenous variables (PARX) + Gaussian statistical distribution.	Statistical model	—	✓	✓	—	(Liu and Nielsen, 2016; Ardakanian et al., 2014)
Two-Sigma Rule applied to time series after Neural Network ARIMA for power forecasting	Statistical model	—	✓	✓	—	(Chou and Telaga, 2014)
Regression, Entropy, Clustering	Linear models, information theoretic, proximity	✓	—	✓	—	(Zhang et al., 2011)

of data traces depending on time and type of the day, and then looking at the distance (e.g., by kNN) of similarly grouped smart meters. (Buzau et al., 2018) use several machine learning approaches (k-NN, SVM, Logistic Regression, XGBoost) for the identification of anomalies from users’ power consumption profiles. Recent models push in the direction of the importance of the stochastic nature of the underlying processes. For these reasons, detecting concept drifts—changes in the behaviour of users over time—is an important feature of such models, like in (Fenza et al., 2019).

### 3 EXPERIMENTAL RESULTS

To showcase the challenges in power consumption anomaly detection, we replicated a previous method proposed by (Chou and Telaga, 2014) with differences discussed in section 4.1 Threats to Validity. This method is particularly interesting, as it represents an instance of a time series based method that is typical in the area of power consumption anomaly detection. We are particularly interested in looking at a much larger dataset than in the original paper, and issues that derive from the application to a different context. In running the replication, we have the following research questions:

- RQ1. Given the importance of historical patterns considered by newer models, what is the impact of an *accumulating window* training approach compared to *sliding windows* from the original paper?
- RQ2. Given the training/test periods and the  $\sigma$ -rule level parameters, are these acceptable parameters

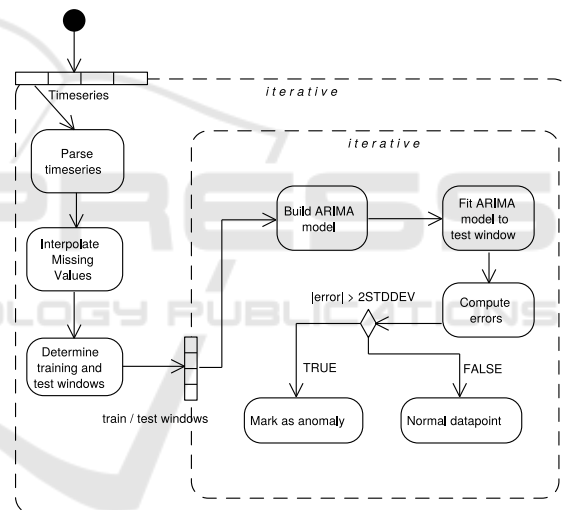


Figure 1: Anomaly Detection Process.

for the new dataset?

Following (Chou and Telaga, 2014) approach, the process of anomaly detection is shown in Fig. 1. The input is a set of time series of power consumption data for which we compute anomalies. The first step is on parsing the data and replacing missing values by linear interpolation. Training and test windows are determined next. As in the original paper, the ARIMA model is trained on a sliding window of 4 weeks and then tested on the subsequent week/day (Fig. 3). The window is then moved and the process is repeated for each of the defined windows to compute the error from the forecast and the predicted ones. Anomalies are defined following the 2- $\sigma$  rule, as any value greater than 2 Standard Deviations from the average

of the errors in the prediction from the ARIMA model (so-called error residuals). Final results are summarized in terms of plots showing anomalies on top of the original time series. Fig. 2 shows the final result for one time series as implemented in our replication, in which the identified anomalies parts of the time series are marked in red (parts in which ARIMA residuals errors are greater than  $2\text{-}\sigma$  from the average).

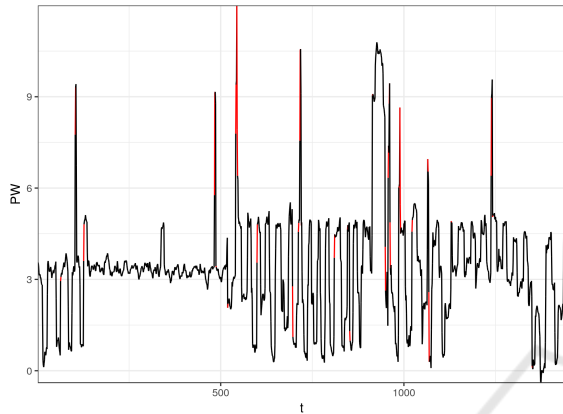


Figure 2: Example of Anomalies Identified (in Red) Plotted on Top of a Daily Testing Data Window.

In (Chou and Telaga, 2014) paper, authors performed an experiment with real smart meters, collecting data for a 17-weeks period on a minute granularity (we can estimate overall 171 360 datapoints,  $60m * 24h * 7d * 17w$ ). In this experiment, we use the Smart\* dataset providing energy traces for series of smart homes of 114 apartments (Barker et al., 2012). We considered data from each apartment taken from 2016 (the dataset comprises also years 2014, 2015). Each apartment dataset represents power consumption at a granularity of 1 minute, and each apartment contains traces for around 500K events, with some variation. In this work, we consider each apartment as a time series. Overall, for the whole analysis, we analyzed the first 80K events of each of the apartments, overall more than 9,120,000 events<sup>2</sup>.

**Method.** We replicated the approach of (Chou and Telaga, 2014) and analyzed the results of the application of the method to 114 apartments of the Smart\* public dataset for year 2016. Based on *RQ1*, *RQ2*, we aim at evaluating two aspects:

- The evaluation of three different training and test strategies: a sliding window of 4 weeks (SL4W, Fig. 3) - the same applied in the paper (Chou and Telaga, 2014), a sliding window of 1 week (SL1W), and an aggregated window training and

<sup>2</sup>since in the method we used a sliding / accumulating window for testing the models, a small ending part of events was not included, to keep the test window of the same size

testing (AGGRW, Fig. 4). While the sliding window has a limited history about previous periods for training, the accumulating window keeps more information about previous periods of time series to build the ARIMA models;

- The distribution of anomalies on the Smart\* public dataset after the application of the method, that is how many anomalies are detected for each of the apartments;

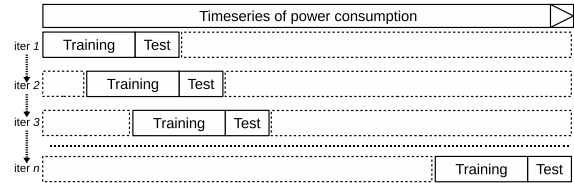


Figure 3: Sliding Window Training and Testing of the Model (as Applied Originally in (Chou and Telaga, 2014)).

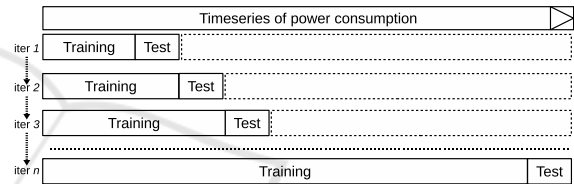


Figure 4: Accumulating Window Training and Testing of the Model.

**Different Training-testing Strategies.** We run the method based on ARIMA and residuals threshold anomalies on all the 114 apartments for the year 2016 in the Smart\* dataset, applying the different training and testing strategies. To evaluate the differences, we calculated the testing Root Mean Square Error (RMSE), that is how well the ARIMA model was approximating the real data. RMSE is calculated as:  $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$ , where  $\hat{y}$  are predicted values,  $y_i$  observed values, and  $n$  is the sample size. RMSE is an indication of how well the ARIMA model fits the data. The lower the value the better, though absolute values cannot be compared to different contexts, as they are dependent on the measure used for building the ARIMA model (power consumption in this case kWh).

We can see in Fig. 5 the comparison of the distribution of RMSE for the three different strategies. The effort of taking more historical data in the training process by using the AGGRW strategy does not bring benefits, on the contrary, the best strategy is SL4W, while differences are lower between a one-week training vs. four-weeks training as in the original paper. Furthermore, by automating such large number of time series predictions with ARIMA models, we lose the possibility to fine-tuning predictions

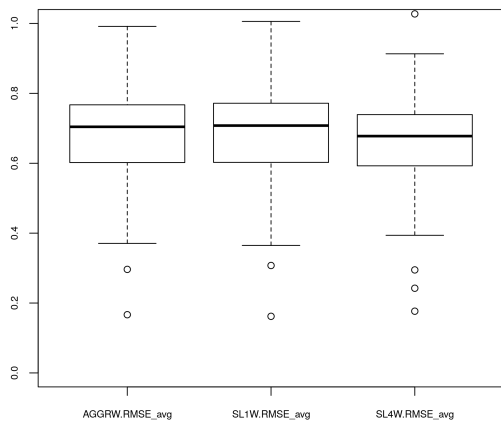


Figure 5: RMSE of SL4W, SL1W, AGGRW Training Test Methods on the 114 Apartments.

on a time series-by-time series. For example, residuals should follow a non-normal distribution of the errors in fitting the ARIMA model.

To look if the differences are statistically significant, we run Wilcoxon Signed-Rank Tests, paired tests to evaluate the mean ranks differences between the different strategies. For Wilcoxon Signed-Rank Test, we calculate effect size as  $r = Z/\sqrt{N}$ , where  $N = \# \text{ cases} * 2$ , to consider non-independent paired samples, using Cohen’s definition to discriminate between small (0.0 – 0.3), medium (0.3 – 0.6), and large effects (> 0.6). The difference is statistically significant for SL1W vs. SL4W ( $p$ -value < .00001 –  $p \geq 0.05$ , two-tailed, medium effect size ( $r = 0.50$ )), SL1W vs. AGGRW ( $p$ -value 0.00064,  $p \geq 0.05$ , two-tailed, small effect size ( $r = 0.22$ )), and SL4W vs. AGGRW ( $p$ -value < .00001,  $p \geq 0.05$ , two-tailed, medium effect size ( $r = 0.49$ )).

**Findings.** Considering the Smart\* dataset, there are no tangible benefits in terms of fitting ARIMA models with an accumulating window strategy. A sliding window strategy brings statistically equivalent results.

**Performance of the Three Training Strategies.** We compared the time performance of the three testing strategies (Fig. 6). Overall, the AGGRW strategy is the most expensive in terms of time required (mean 25min. per iteration), compared to SL4W (mean 12min.), vs. SL1W (mean 5min.). Furthermore, the time required for the accumulating window strategy grows linearly with the dataset size for each training-testing iteration, making it unbearable for larger datasets. As specified in the previous analysis, considering the specific Smart\* dataset, such strategy is not worth the application in terms of final results.

**Findings.** The AGGRW testing-training strategy becomes unbearable in case of large datasets as it grows linearly at each iteration with the size of the datasets. If historical information in training does not give benefits (as in the current datasets), sliding windows might be a better choice. In the case of the Smart\* dataset, a 1-week training with a 1-day testing gives the best in terms of results.

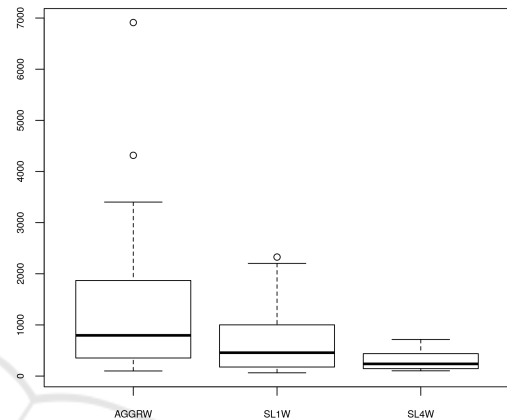


Figure 6: Distribution of Running Time (Seconds) of Training-Testing Strategies SL4W, SL1W, AGGRW.

**Distribution of Anomalies: 2σ vs 3σ Rules.** We looked also at the difference between running a 2σ rule (as in the original paper) and 3σ rules for filtering anomalies from the ARIMA models residuals (Fig. 7). Running 2σ rules, as in the original paper brings to 5%-8% anomalies identified for each apartment time series, while 3σ rule bring 2%-4% of the total items identified as anomalies. We believe the latter to be a more realistic scenario given the Smart\* dataset. It is, however, responsibility of domain experts to evaluate events tagged by the algorithms to see if false positive or false negative, or at least to focus on specific patterns of events in time like concatenating anomalies. There is not a large difference in the detection by using the three different testing strategies.

**Findings.** Using the original 2σ rule used in (Chou and Telaga, 2014) in the context of the Smart\* datasets brings anomalies in the range of 5%-8% of each subset, which might be excessive for domain-experts to be evaluated. A 3σ rule might be more appropriate, taking the anomalies to 2%-4%.



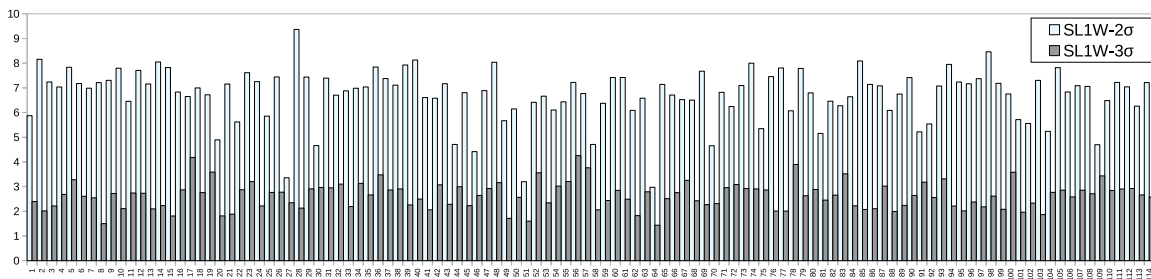


Figure 7: Percentage of Anomalies Detected by Applying SL1W Training-Test Strategies with  $2\sigma$  and  $3\sigma$  Rules.

## 4 LESSONS LEARNED FROM THE REPLICATION

By means of the independent external replication on the Smart\* public power consumption data traces, we can derive several indications based on the application of the replicated anomaly detection approach.

**The Selection of the Model is Context-based.** As in other domains, the selection of the best model is context-based. However, since power consumption traces translate into time-series, consideration of data temporal contextual properties are quite relevant in this area. The generation process is stochastic in nature, so models have to take into account temporal, as well as contextual information. The approach by (Chou and Telaga, 2014) that we replicated, is a representative of such time series based models—based on a single time series, while more advanced models take into account several time series, like power and weather data, e.g., (Liu and Nielsen, 2016).

**Assumptions are Relevant in the Application of a Method.** As usual in Data Science, all models are based on specific assumptions. Replications follow the same set of assumptions as in the replicated paper. As an example, (Chou and Telaga, 2014) model is based on the assumption that errors from ARIMA models are normally distributed to apply sigma rules, like the fact that 95% of the values lie within two standard deviations to the mean. What happens, however, if the dataset has a skewed distribution? Applied a method with wrong parameters / assumption can be ineffective in the identification of anomalies.

**Concept Drift.** Most of the models assume no change of underlying process over time, however, considering concept drifts like changes in owners of an apartment (Fenza et al., 2019) might be necessary for the accuracy of anomaly detection. Recent models seem to go into this direction, by considering a more dynamic process model, rather than static processes.

**Ground Truth & Anomaly Injection.** Gathering datasets in which anomalies have been tagged is diffi-

cult, most of the datasets do not come with an indication of anomalies. If supervised techniques have to be applied, can be good practice to “inject” some known anomalies as series of datapoints that can be detected by supervised learning. Domain knowledge is very important in this area. For example, the method of (Chou and Telaga, 2014) that we replicated, in the original paper was fine-tuned to consider in some cases the duration of events, given knowledge of some device that was generating false positives.

**Effort to Re-implement the Algorithms.** Many techniques might be time-consuming to re-implement and validate based on the information provided in research papers. There is need of replications and platforms that can allow the comparability of the results, such as the NILMTK platform (Batra et al., 2014).

**Fine-tuning and Automation.** When we extend the analysis to larger datasets (as in our case, around 57M events vs. the original estimated 171K events in (Chou and Telaga, 2014)), we lose the possibility of fine-tuning several aspects. In our case, ARIMA was fitted with the *autoarima* function, as any manual optimization would be unfeasible due to the large number of running iterations (114 apartments per several training / test sets). Q-Q plots can be used to look at the normality of residual errors of the models, but become unfeasible to evaluate when building tens of models for fitting data of different training-test iterations. Likely, models that are easier to fine-tune will be preferred in case of larger datasets.

**Online Learning.** Related to automation, early models were tested on smaller datasets. With the emergence of Big Data, online learning becomes a more effective domain of application, to test if models can scale to streaming datasets (Lipčák et al., 2019; Drakontaidis et al., 2018). Online learning can also be “simulated” with public datasets by generating new itemsets with similar properties as the underlying data. Generators should take into account not only data properties to generate realistic streams of data, but also some delays in generation / receiving of data

that is specific of windowed data streams (Liu et al., 2016).

#### 4.1 Threats to Validity

Replicating previous research based on the information provided in articles is a challenging task as many parameters are not reported, as well as some of the underlying software versions used. We can report the following threats to validity in our replication.

We tried to use the same programming languages mentioned in the original article (Chou and Telaga, 2014), that is *R* (R Core Team, 2018) with the *forecast* library for time series analysis. Where not specified, we used the standard parameters provided by the methods (e.g., *nnetar*). Our environment used *R 3.4.4*, *forecast 8.9*, *0.10-47*, under *Ubuntu 18.04.3 LTS*: even with the same implementation, some differences might be due to changes in defaults of the libraries. The analysis was run on an Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz, plus the support for virtual machines on CERIT-SC Center infrastructure.

Compared to (Chou and Telaga, 2014), we also skipped one step in the process, that is the usage of k-means clustering on time-series. In the original paper, the clustering step was done after interpolation to decide whether to perform the training on a daily or weekly basis. To be consistent with the original paper, we use the same 4-weeks windows training period that we compare with alternative strategies. Also, in the original paper there is a comparison between ARIMA and Neural Network Auto Regressive (NNAR) models, we only focused on ARIMA as it was not our goal to compare different time series fitting methods. Another difference is on the way anomalies were calculated, in the original dataset it was observed that there was a printing activity usually taking less than five minutes that was often tagged as anomaly—for this reason authors decided to mark as anomaly any activity over 2 Standard Deviations and at least 5 min. of duration. This was an *ad-hoc* rule based on the specific domain, so we did not consider the additional timing constraint, rather we only compared different  $\sigma$  rules.

The time series approach examined is univariate, but multiple time series could be exploited for anomaly detection, as having support of weather information time series could be useful to detect anomalies. To be consistent with the original paper, we also used a single time series, even though weather data is available in the Smart\* dataset. The other issue is the availability of ground truth, as without insider knowledge about the datasets, can be difficult to understand which elements could be really posi-

tive cases of anomalies. For this reason, we could not calculate performance of the models in terms of false positives – false negatives. One approach to overcome such issue could be based on injecting anomalies and looking at the performance of the models, but we kept this as future work.

## 5 CONCLUSIONS

The area of power consumption anomaly detection has adopted many approaches for the identification of anomalous patterns for issues such as energy theft prevention. The goal of this paper was to run an independent external replication of a previous approach by (Chou and Telaga, 2014) on a larger dataset (~9M datapoints compared to the original ~170K).

Overall, we evaluated different strategies of training and testing (*RQ1*): sliding and accumulating windows, to keep more into account history of time series in learning the models, and the impact of different threshold options on the newer dataset (*RQ2*). First of all, we found that the method of (Chou and Telaga, 2014) was applicable without issues to a much larger dataset that in the original paper. The original sliding window training strategy is performing equivalently to an accumulating window strategy in terms of RMSE. A training of one week and one day testing had the best results in terms of RMSE for the Smart\* dataset considered. Using a  $2\sigma$  rule reports too many data traces as anomalous—likely as in the original paper this threshold was combined with another *ad-hoc* temporal rule. Furthermore, the real presence of anomalies is difficult to evaluate as in both the original paper and the current replication there is no ground truth about anomalies. Some anomaly-injection activities could be performed to evaluate the quality of the anomaly identification technique.

Based on the results, we discussed several lessons learned for current and future power anomaly detection algorithms, like the impact of automation on larger datasets, the impact of concept drifts and availability of ground truth for the evaluation of models' performance. With so many approaches proposed over the time for power consumption anomaly detection, and the many datasets available, replications can be a useful instrument to understand how an approach applies to different contexts.

## ACKNOWLEDGEMENTS

The work was supported from European Regional Development Fund Project *CERIT Scientific Cloud* (No.

CZ.02.1.01/0.0/0.0/16\_013/0001802). Access to the CERIT-SC computing and storage facilities provided by the CERIT-SC Center, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CERIT Scientific Cloud LM2015085), is greatly appreciated.

## REFERENCES

- Aggarwal, C. C. (2015). Outlier analysis. In *Data mining*, pages 237–263. Springer.
- Ardakanian, O., Koochakzadeh, N., Singh, R. P., Golab, L., and Keshav, S. (2014). Computing electricity consumption profiles from household smart meter data. In *EDBT/ICDT Workshops*, volume 14, pages 140–147.
- Barker, S., Mishra, A., Irwin, D., Cecchet, E., Shenoy, P., Albrecht, J., et al. (2012). Smart\*: An open data set and tools for enabling research in sustainable homes. *SustKDD, August*, 111(112):108.
- Batra, N., Kelly, J., Parson, O., Dutta, H., Knottenbelt, W., Rogers, A., Singh, A., and Srivastava, M. (2014). Nilmtk: an open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th international conference on Future energy systems*, pages 265–276. ACM.
- Buzau, M. M., Tejedor-Aguilera, J., Cruz-Romero, P., and Gómez-Expósito, A. (2018). Detection of non-technical losses using smart meter data and supervised learning. *IEEE Transactions on Smart Grid*, 10(3):2661–2670.
- Caithness, N. and Wallom, D. (2018). Anomaly detection for industrial big data. In *Proceedings of the 7th International Conference on Data Science, Technology and Applications, DATA 2018*, page 285–293, Setubal, PRT. SCITEPRESS - Science and Technology Publications, Lda.
- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.
- Chou, J.-S. and Telaga, A. S. (2014). Real-time detection of anomalous power consumption. *Renewable and Sustainable Energy Reviews*, 33:400–411.
- Chren, S., Rossi, B., and Pitner, T. (2016). Smart grids deployments within eu projects: The role of smart meters. In *2016 Smart cities symposium Prague (SCSP)*, pages 1–5. IEEE.
- Cramer, I., Govindarajan, P., Martin, M., Savinov, A., Shekhawat, A., Staerk, A., and Thirugnana, A. (2018). Detecting anomalies in device event data in the iot. In *IoT BDS*, pages 52–62.
- Drakontaidis, S., Stanchi, M., Glazer, G., Hussey, J., Leger, A. S., and Matthews, S. J. (2018). Towards energy-proportional anomaly detection in the smart grid. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–7.
- Fenza, G., Gallo, M., and Loia, V. (2019). Drift-aware methodology for anomaly detection in smart grid. *IEEE Access*, 7:9645–9657.
- García, J., Zamora, E., and Sossa, H. (2018). Supervised and unsupervised neural networks: Experimental study for anomaly detection in electrical consumption. In *Mexican International Conference on Artificial Intelligence*, pages 98–109. Springer.
- Gómez, O. S., Juristo, N., and Vegas, S. (2010). Replications types in experimental disciplines. In *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*, page 3. ACM.
- Jung, O., Smith, P., Magin, J., and Reuter, L. (2019). Anomaly detection in smart grids based on software defined networks. In *Proceedings of the 8th International Conference on Smart Cities and Green ICT Systems - Volume 1: SMARTGREENS*, pages 157–164. INSTICC, SciTePress.
- Juristo, N. and Vegas, S. (2011). The role of non-exact replications in software engineering experiments. *Empirical Software Engineering*, 16(3):295–324.
- Lipčák, P., Macak, M., and Rossi, B. (2019). Big data platform for smart grids power consumption anomaly detection. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 771–780.
- Liu, X., Iftikhar, N., Nielsen, P. S., and Heller, A. (2016). Online anomaly energy consumption detection using lambda architecture. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 193–209. Springer.
- Liu, X. and Nielsen, P. S. (2016). Regression-based online anomaly detection for smart grid data. *arXiv preprint arXiv:1606.05781*.
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Rossi, B. and Chren, S. (2019). Smart grids data analysis: A systematic mapping study. *IEEE Transactions on Industrial Informatics*.
- Rossi, B., Chren, S., Buhnova, B., and Pitner, T. (2016). Anomaly detection in smart grid data: An experience report. In *2016 IEEE international conference on systems, man, and cybernetics (smc)*, pages 2313–2318. IEEE.
- Saad, A. and Sisworahardjo, N. (2017). Data analytics-based anomaly detection in smart distribution network. In *2017 International Conference on High Voltage Engineering and Power Systems (ICHVEPS)*, pages 1–5. IEEE.
- Sial, A., Singh, A., Mahanti, A., and Gong, M. (2018). Heuristics-based detection of abnormal energy consumption. In *International Conference on Smart Grid Inspired Future Technologies*, pages 21–31. Springer.
- Zhang, Y., Chen, W., and Black, J. (2011). Anomaly detection in premise energy consumption data. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–8. IEEE.