The Choice of AI Matters: Alternative Machine Learning Approaches for CPS Anomalies

Odyurt, U.; Sapra, D.; Pimentel, A.D.

DOI
10.1007/978-3-030-79463-7_40

Publication date
2021

Document Version
Final published version

Published in
Advances and Trends in Artificial Intelligence : From Theory to Practice

The Choice of AI Matters: Alternative Machine Learning Approaches for CPS Anomalies

Uraz Odyurt(✉), Dolly Sapra, and Andy D. Pimentel

Informatics Institute (IvI), University of Amsterdam, Amsterdam, The Netherlands
{u.odyurt,d.sapra,a.d.pimentel}@uva.nl

Abstract. We compare the pros and cons of two Artificial Intelligence (AI) solutions, addressing the anomaly detection and identification challenge in industrial Cyber-Physical Systems (CPS). We demonstrate how our current approach, *Advanced DL*, based on Convolutional Neural Networks (CNN) differs from a previous one, *Classic ML*. Though both workflows prove to result in highly accurate classification of anomalies, Classic ML is superior in this regard with 99.23% accuracy against 94.85%. This comes at a cost, as Classic ML requires total insight and expertise regarding the system under scrutiny and heavy amounts of feature engineering, while Advanced DL treats the data as a black box, minimising the effort. At the same time, we show that finding the best performing CNN model design is not trivial. We present a quantitative comparison of both workflows in terms of elapsed times for training, validation and preprocessing, alongside discussions on qualitative aspects. Such a comparison, involving analysis of workflows for the given use-case, is of independent interest. We find the choice of AI solution to be use-case dependent.

Keywords: Machine learning · Convolutional neural network · Behavioural passports · Anomaly identification · Industrial cyber-physical systems

1 Introduction

We have witnessed the emergence of solutions based on classic Machine Learning (ML) and more advanced models, i.e., Deep Learning (DL) with Convolutional Neural Networks (CNN), for a plethora of problems for quite some time now. These techniques have become an integral part of any method of choice. The industry in particular, reaps the benefits of such solutions in production systems. As ML and DL provide more than just one way to solve a given problem, it is of utmost importance to pick the right solution and to employ the right workflow. For industrial systems, the extent of resource consumption and timely operation could very well mean the difference between success and failure, depending on the relevant requirements. In other words, it is not just about the accuracy of answers to problems, but also how fast and how efficiently they can be found.

H. Fujita et al. (Eds.): IEA/AIE 2021, LNAI 12799, pp. 474–484, 2021.
https://doi.org/10.1007/978-3-030-79463-7_40

We explore the balance between these factors for a given problem, which is a simplified version of an industrial use-case. We deal with an industrial Cyber-Physical System (CPS) with embedded computing nodes, for which we actively detect and identify anomalies. Anomalies can manifest themselves as diminished performance or other harmful behaviour. Anomaly detection and identification are open challenges for industrial CPS. As such systems evolve and become more complex, mainly as a result of software complexity, not every operational corner case can be covered at design time. Considering deployment in critical applications, anomalies are often very costly to rectify and leave costly effects behind when they occur. We aim to solve this challenge in a ML-based workflow, monitoring the system behaviour and identifying anomalies online.

We will be comparing two of these ML-based solutions, namely, *Classic ML* workflow developed earlier and *Advanced DL*, developed as an alternative in this paper. The Classic ML workflow incorporates regression modelling and classic algorithms, i.e., decision tree and random forest. Our Advanced DL workflow incorporates limited data preprocessing steps and takes advantage of CNNs. The main aspect driving us towards the Advanced DL workflow is the amount of domain specific knowledge, expertise and understanding of the system that is necessary for the Classic ML workflow. Our Advanced DL approach is a truly black box one, requiring no insight into the data or the internals of the system, but at the same time, has its own shortcomings.

Contribution. We have developed an alternative approach based on advanced DL to detect and identify anomalies in industrial CPS. We perform quantitative and qualitative comparisons between this approach and a previous one, utilising classic ML. Though we are dealing with a specific use-case, our comparison addresses the characteristics of the general methodology (depicted in Fig. 1) and the use-case is a demonstrator to generate data for it. We argue that there is no absolute winner and the choice of the workflow depends on the expected classification accuracy, the ability to explain the outcome based on the input data, the amount of internal knowledge, workflow development time and preference of a white box versus a black box approach towards data.

This introduction is followed by core concepts of our workflows for industrial CPS. Section 3 details our overall methodology, including the two approaches for its realisation, while Sect. 4 elaborates the implementation of the second approach, based on CNNs. Results and comparisons are given in Sect. 5, followed by the related work and concluding remarks in Sects. 6 and 7, respectively.

2 Machine Learning for Industrial CPS

When it comes to the industrial applications of CPS, there are high-value use-cases for the deployment of ML algorithms. One such use-case is the detection and classification of anomalies. Figure 1 showcases the high-level view of a methodology to address such a challenge. Different flavours of ML, whether classic ML or DL algorithms, are good fits when dealing with large amounts of data.

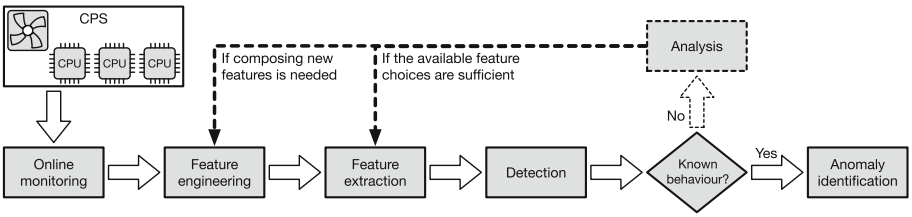


Fig. 1. Our reference analytics-based pipeline, provisioning the presence of feature engineering and anomaly identification steps, alongside anomaly detection and an optional analysis step, upon discovery of unseen anomalies.

Given that modern industrial CPS provide this large amount of monitoring data generation capability through software and hardware probes, the use of ML is not a preference, but a necessity. As such, the *analytics-based* pipeline shown in Fig. 1 is designed with data-centricity in mind [9].

Depending on the type of ML algorithm, certain amount of preprocessing is needed to transform the data into consumable forms. As it will be shown in Sect. 3, major parts of this preprocessing will be implemented rather differently, resulting in alternative characteristics and performance. Our anomaly detection is based on monitoring the system's Extra-Functional Behaviour (EFB), representing the behavioural traits of a system beyond its functional definitions and semantics. EFB is generated from different performance and operational metrics, e.g., execution time, latencies, power and energy consumption. EFB representations composed from such metrics can uniquely identify a specific system, under specific operational conditions [9].

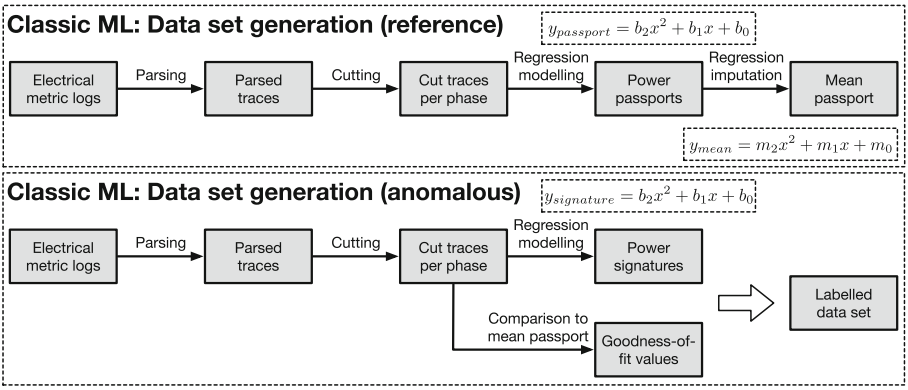
3 One Challenge, Two Approaches

We have chosen the high-level methodology given in [9,11], aiming at detection and classification of anomalies in industrial CPS, as our reference. Since its implementation is based on classical ML algorithms [11], requiring much feature engineering effort, intimate knowledge of system internals and the data itself, we have devised a competing workflow, based on deep learning with CNNs.

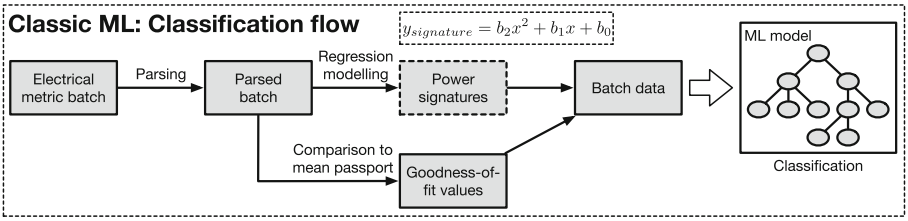
3.1 Classic ML Workflow

Figures 2a and 2b visualise data set generation and anomaly classification flows for the Classic ML workflow, respectively. Here, we interpret and realise our reference workflow with classic ML classifiers, e.g., decision tree.

The Classic ML workflow involves the concept of *execution phases*, i.e., repetitive units of execution during the operational timeline of a system. Industrial CPS in particular, reveal the presence of such repetitions, as they are purpose-built systems with limited operational variety [10]. In other words, these are repeated smaller tasks, making up the complete execution.



(a) Data set generation (preprocessing)



(b) Anomaly classification

Fig. 2. The two flows involved with the Classic ML approach, (a) data set generation flow from large amounts of trace data, leading to a training data set and (b) an anomaly identification flow, using much smaller trace batches with a previously trained classifier.

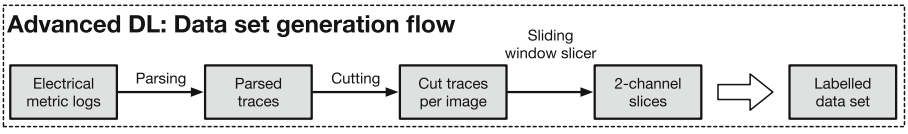
During data set generation, EFB metric logs, e.g., electrical current, are collected and parsed. What follows involves cutting of parsed traces into parts corresponding to desired executional phases, i.e., tasks from the actual operation of the CPS, reflecting its behaviour. For instance, a CPS collecting imagery and processing them could perform `load_image` and `process_image` tasks, leaving us with three possible execution phases. These are *image load atomic phase*, *image processing atomic phase* and the combination, *image combo phase*. It is yet unknown to a designer which phase is the best to choose from and cut the traces based on, at solution design time.

For a collection of recorded readings over time for any EFB, it is possible to generate a regression function, which will serve as a unified representation of the readings. We call this a behavioural signature and when generated for a reference execution, we call it a behavioural passport [10]. Accordingly, data points from the log, including timestamps and metric readings, are used to generate a regression function as a representation. Passports are generated per metric and per phase. Numerous inputs to the CPS under reference circumstances result in numerous passports. To simplify future comparisons, we generate *mean passports* out of many passports, again per metric and per phase. By collecting anomalous

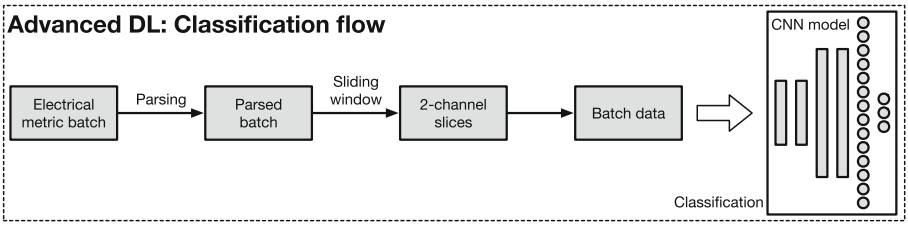
traces and generating signatures in the same manner, we are able to calculate the amount of deviation between corresponding signatures and mean passports. The final outcome is a labelled data set, which in turn can train a classifier in a supervised fashion.

3.2 Advanced DL Workflow

Our Advanced DL flows for data set generation and anomaly classification are depicted in Figs. 3a and 3b, respectively. For both flows, whether the learning leading to the labelled data set, or the classification, the amount of data preparation is minimal. This preparation includes parsing of the raw metric logs, cutting of the parsed traces per image and running a sliding window algorithm to generate two-channel slices of fixed size. These two channels include the time data (timestamps) and the metric data (metric readings). It is necessary to consider the time data as a separate channel since the metric data collection happens at high frequency, with non-determinism for system behaviour present, resulting in timestamps that do not exactly match for different experiments. This is an expected effect as industrial CPS are inherently non-deterministic. We have only considered the metric resulting in the highest accuracy for the Classic ML flow as it was seen in [11], i.e., electrical current. Note that in this approach, there is no need for an intimate understanding of the data to reveal atomic phases within the processing of an image and the trace data related to each image is considered as a whole.



(a) Data set generation (preprocessing)



(b) Anomaly classification

Fig. 3. The two flows involved with the Advanced DL approach, (a) data set generation flow from large amounts of trace data, leading to a training data set (note the reduction in number of steps and their complexity) and (b) an anomaly identification flow, using much smaller trace batches with a previously trained classifier.

The Classic ML flow already has a rather high accuracy [11]. To push the classification accuracy of our Advanced DL flow to similar levels, we have performed

a grid search for hyperparameter optimisation. The three groups of considered hyperparameters are data preparation, learning and CNN model parameters, further elaborated in Sect. 4.2.

4 Implementation

Regarding notable details of the Classic ML flow, in our experiments, polynomial regression functions of degree two provide sufficient interpolation accuracy. We transform the time-series data from traces into cumulative data for the metric part to make the regression function a monotonically increasing one. For goodness-of-fit tests, we use both coefficient of determination (R^2) and Root-Mean-Square-Deviation ($RMSD$) to compare a sample signature to a reference passport. The identification step uses Decision Tree (DT) [13] and Random Forest (RF) [2] classifiers. Considering the Classic ML implementation from [11], we focus on the Advanced DL elements, as it involves the bulk of this work.

4.1 Data Set

Our data sets are generated from the same raw electrical metric readings, collected via an external power data logger unit, Otii Arc [12], connected to an ODROID-XU4 computing device. These traces are in the form of time-series and every data point has a timestamp and a metric value. The data set for the Classic ML flow has many columns, such as execution time, regression function coefficients and intercept, goodness-of-fit test values and labels [11]. The Advanced DL data set on the other hand is rather simple, only including two separate time and metric data channels and corresponding labels.

For both workflows, we are considering three labels, i.e., Normal, NoFan and UnderVolt. The methodology can be implemented with any number of labels. Our demonstrator involves these labels corresponding to, normal circumstances for reference executions, faulty cooling fan for the system-on-chip, and unstable power supply, respectively. Both data sets are balanced as we have performed equal number of experiments for all scenarios (labels). For Advanced DL, the data is normalised at preprocessing. Training set and test set ratios to the whole data set are 80% and 20% for the Advanced DL trainings, respectively and 70% and 30% for the Classic ML trainings, respectively.

4.2 CNN Structure and Search Space

To arrive at an acceptable CNN design, we have performed a grid search for the hyperparameter variations listed in Table 1.

The most optimised model we arrived at consists of six convolutional layers with sizes 64, 64, 128, 128, 256, 256, a Fully Connected (FC) layer of size 4096, all kernel sizes 5×1, ReLU activation for each convolutional layer and the FC layer, and MaxPool layers after even convolutional layers. Our data analysis pipelines have been written in Python 3.8 and we use the Scikit-learn 0.23.2 package for

Table 1. Hyperparameters considered during the grid search and their variations

Parameter type	Parameter	Variations
Data preparation	Slice sizes	50, 100
	Slice shifts	10, 20
Learning	LR at start	0.01, 0.001, 0.0001, 0.00005
	Epochs	10, 20, 30, 40, 45, 50, 60
	Batch sizes	10, 20, 50, 100
	LR decay	present (mul. factor 0.1), absent
	Decay periods	8, 10, 20
CNN model	Conv. layers	2, 4, 6
	Conv. layer size	8, 16, 32, 64, 128
	Kernel size	3, 5
	FC layer size	512, 1024, 2048, 4096

regression and classical ML classification, as well as the PyTorch 1.6.0 package for CNN implementations. The hardware infrastructure for our experiments is a machine with a 2.20 GHz Intel® Xeon® E5-2650 v4 CPU, 64 GB of RAM and a GeForce RTX 2080 Ti graphics card, with CUDA release 10.0, v10.0.130.

5 Results: Classic ML Vs Advanced DL

Considering the hyperparameters listed in Table 1, Fig. 4 displays an overview of our grid search for paths achieving higher accuracies.

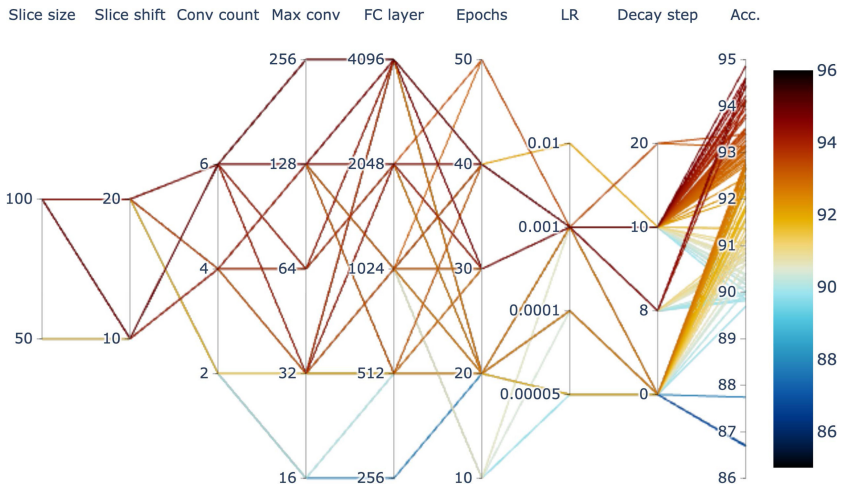


Fig. 4. The set of higher accuracies from our grid search, visualised in a parallel coordinates plot (decay step 0 denotes absence of decay)

To be able to quantitatively compare the two workflows, we consider *elapsed time* for different operations, collected with the `time.perf_counter()` call, providing a high-accuracy monotonic clock. Timing results are given in Table 2. Note that the Classic ML preprocessing is highly parallelised. Model training has to be considered for our industrial use-cases, since upon the introduction of a new anomaly, i.e., a new label, retraining will be necessary.

Table 2. Elapsed times in seconds during different stages of the two approaches

Workflow	Preprocessing	Training	Validation
Classic ML - DT (CPU)	204 648 (∼57 h)	0.02	0.001
Classic ML - RF (CPU)	204 648 (∼57 h)	0.32	0.021
Advanced DL (CPU)	2576	239 976 (∼67 h)	125
Advanced DL (GPU)	2576	18 535 (∼5 h)	25.5

It is evident from our observations during the CNN training that there is a limit to the Advanced DL workflow's achievable accuracy. This is considering the fact that minimal amount of preprocessing has been applied for this particular workflow on purpose. We also see that this achievable accuracy is an effective one, up to 94.85%, depending on hyperparameters. The accuracies for our Classic ML workflow using DT and RF classifiers are 99.15% and 99.23%, respectively. However, the high accuracy provided by the Classic ML workflow comes at a cost and arguably, a high one. The amount of analysis, design effort, experimentation and in short, feature engineering required for the Classic ML workflow is rather vast. Accordingly, there is much need for domain specific knowledge and understanding of the internals of the system under scrutiny. The workflow designer has to know beforehand, or explore, to understand which phases best reflect the overall behaviour of the system for the specific set of anomalies.

One of the capabilities missing in our Advanced DL workflow is the possibility to detect unknown behaviour, i.e., unseen anomalies. Though the CNN model itself can be retrained upon the addition of a new anomaly, the workflow does not include steps facilitating new anomaly discoveries. The reference methodology from Fig. 1 provisions this possibility, for we can use goodness-of-fit tests and detect unseen levels of deviation from a passport. Following this detection, further analysis will result in a new class of anomaly, which can be added and considered for feature engineering in future data sets. This addition of unknown anomalies is achievable in the Classic ML workflow. However, it does require the designer to go through the whole process again, as the new anomaly may or may not be easily detectable using the same phase data.

We would like to emphasise the fact that our Advanced DL workflow is a truly black box approach, requiring no insight into the data or the system internals. In this fashion, the Advanced DL flow cuts through the data processing complexities of the Classical ML flow. Though optimising hyperparameters is

a time-consuming process, it does not depend on the internals of the system and is reusable in the future for more anomalies. We just have to retrain the network. On top of that, neural network frameworks are highly optimised for GPU acceleration, requiring minimal changes to the implementation code.

Stability and maturity of frameworks, in the sense that how much code transformation is enforced from one version to the next is another aspect. In our experience, the change is rapid and substantial with deep learning frameworks, as the field is constantly changing and evolving. This could very well be a factor in a business environment striving for long-term deployment.

Last but not least, with Classic ML workflow, we are able to explain why the classification has resulted in a certain label. Models such as decision trees can be traversed and every processing block in the Classic ML flow can be backtracked to initial trace values, directly connecting the outcome to the input.

6 Related Work

Anomaly detection for industrial CPS and relevant methodologies work upon various input sources, e.g., power signals, sensor data, network traffic data and system calls. Kim et al. [7] were among the first to highlight that power consumption can be used for anomaly detection. Caviglione et al. [4] detected attacks related to covert channels using the power consumption of the running processes. Covert channels occur when malicious applications exploit different assigned permissions and are able to exchange information. Liu et al. [8] developed a strategy using power side-channel data to detect anomalous behaviour in control flow execution applied to IoT microcontrollers. Similarly, Xu et al. [14] used power channels to detect attacks on the Distribution Terminal Unit.

In the last few years, the complexity of CPS has led to elusive and indiscernible faults. Conventional anomaly detection methods are increasingly substituted with state of the art deep learning techniques [5]. Moreover, CNNs have proven to be well suited for analysis of power signals and other similar time-series data for fault detection and classification [6]. Albasir et al. [1] proposed a CNN-based approach to detect malware activity, utilising the power consumption behaviour of smartphones. Canizo et al. [3] deployed CNNs together with recurrent cells to detect anomalies in time-series data from multiple sensors.

7 Conclusion and Future Work

We have developed an alternative AI workflow to a previously devised one, to detect and classify anomalies in industrial CPS. Both workflows, the earlier Classic ML and the new Advanced DL, show high classification accuracies, 99.23% and 94.85%, respectively. While achieving the high accuracy of the Classic ML required extensive design, feature engineering effort and costly computations, the Advanced DL also required extensive optimisation effort. We have discussed

different qualitative aspects of both workflows, such as dependence on the intimate knowledge of the system and the data, stability AI frameworks, efficient GPU implementation and root-cause analysis.

In our opinion, there is no clear winner between these workflows. Critical applications and use-cases can benefit from highest accuracies and analytical capabilities provided by the Classic ML workflow, allowing the study of root-causes behind anomalies, while ease of extension with different anomalies is best served by the Advanced DL workflow. It is totally use-case dependent.

Acknowledgements. This paper is composed as a collaboration between the research project 14208, titled *"iDAPT"*, funded by The Netherlands Organisation for Scientific Research (NWO); and the research project titled *"ALOHA"*, supported and partly funded by the European Union Horizon-2020 research and innovation programme under grant agreement No. 780788.

References

1. Albasir, A., Manzano, R., Naik, K.: Deep learning based approach for classifying power signals and detecting anomalous behavior of wireless devices. In: 2019 IEEE World Congress on Services (SERVICES) (2019)
2. Breiman, L.: Random forests. Mach. Learn (2001)
3. Canizo, M., Triguero, I., Conde, A., Onieva, E.: Multi-head CNN-RNN for multi-time series anomaly detection: an industrial case study. Neurocomputing **363**, 246–260 (2019)
4. Caviglione, L., Gaggero, M., Lalande, J., Mazurczyk, W., Urbański, M.: Seeing the unseen: revealing mobile malware hidden communications via energy consumption and artificial intelligence. IEEE Trans. Inf. Forensics Secur. **11**(4), 799–810(2016)
5. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: a survey (2019)
6. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A.: Deep learning for time series classification: a review. Data Min. Knowl. Discov. **33**(4), 917–963 (2019). https://doi.org/10.1007/s10618-019-00619-1
7. Kim, H., Smith, J., Shin, K.G.: Detecting energy-greedy anomalies and mobile malware variants. In: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (2008)
8. Liu, Y., Wei, L., Zhou, Z., Zhang, K., Xu, W., Xu, Q.: On code execution tracking via power side-channel. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (2016)
9. Meyer, H., Odyurt, U., Pimentel, A.D., Paradas, E., Alonso, I.G.: An analytics-based method for performance anomaly classification in cyber-physical systems. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing (2020)
10. Odyurt, U., Meyer, H., Pimentel, A.D., Paradas, E., Alonso, I.G.: Software passports for automated performance anomaly detection of cyber-physical systems. In: Pnevmatikatos, D.N., Pelcat, M., Jung, M. (eds.) SAMOS 2019. LNCS, vol. 11733, pp. 255–268. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-27562-4_18
11. Odyurt, U., Roeder, J., Pimentel, A.D., Gonzalez Alonso, I., de Laat, C.: Power passports for fault tolerance: anomaly detection in industrial CPS using electrical EFB. In: 2021 IEEE Conference on Industrial Cyberphysical Systems (2021)

12. Qoitech AB: Otii arc - otii by qoitech (2020). https://www.qoitech.com/otii/
13. Rokach, L., Maimon, O.: Decision Trees (2005)
14. Xu, A., Jiang, Y., Cao, Y., Zhang, G., Ji, X., Xu, W.: ADDP: anomaly detection for DTU based on power consumption side-channel. In: 2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2) (2019)