Power passports for fault tolerance: Anomaly detection in industrial CPS using electrical EFB

Odyurt, U.; Roeder, J.; Pimentel, A.D.; Gonzalez Alonso, I.; de Laat, C.

DOI
10.1109/ICPS49255.2021.9468262

Publication date
2021

Document Version
Final published version

Published in
Proceedings 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)

Power Passports for Fault Tolerance: Anomaly Detection in Industrial CPS Using Electrical EFB

Uraz Odyurt*, Julius Roeder*, Andy D. Pimentel*, Ignacio Gonzalez Alonso† and Cees de Laat*
*Informatics Institute (IvI), University of Amsterdam, Amsterdam, The Netherlands
Email: {u.odyurt, j.roeder, a.d.pimentel, delaat}@uva.nl
†ASML Netherlands B.V., Veldhoven, The Netherlands
Email: ignacio.alonso@asml.com

2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS) | 978-1-7281-6207-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPS49255.2021.9468262

***Abstract*—We present how electrical Extra-Functional Behavioural (EFB) metrics can be used in generating accurate representations of executional units for industrial Cyber-Physical Systems (CPS). We achieve this by employing our concept of *power passports* (the representation), created per metric and per *execution phase* (executional unit). We employ these representations alongside supervised classification algorithms, i.e., Decision Tree and Random Forest, in an effective data analytical pipeline. Our approach is capable of detecting anomalous operational conditions and predicting the type of anomaly, out of different known types, with significant overall accuracies, as high as 99% in certain set-ups. We consider anomalous operational conditions as non-reference conditions, resulting in loss of performance or unreliable operation of a system.**

Our experiments are designed to reflect real-world conditions as much as possible and all of our collected raw data comes from real executions, normal and anomalous, with no synthetic manipulation. Our results show that a black box approach towards systems under scrutiny for anomaly detection and classification, given its accuracy and considering the limitations applicable to low-power industrial CPS, can be the preferred one.

***Index Terms*—Automated performance anomaly detection, Industrial cyber-physical systems, Fault tolerance, Classification**

I. Introduction

The expanding complexity of modern Cyber-Physical System (CPS) platforms forces us to think about and develop solutions for the challenges resulting from this complexity. CPS platforms, whether having a distributed or a centralised architecture, are becoming more and more software-heavy, i.e., the software running the platform is playing a more prominent role in its operational specifics. As the software grows in size and complexity, also considering the interactive relation between different software components on both internode (amongst multiple platforms) and intranode (within a single platform) scales, the potential for unforeseen anomalies increases. Another source is the complexity factor imposed upon CPS by means of environmental uncertainties, making it exceptionally challenging and expensive to account for all possibilities at design time [1]. Thus, online measures of keeping the system in check to allow detection and prediction of anomalies is not a luxury, but a necessity.

The economic impact of the challenge is especially prevalent in *industrial CPS* platforms [2]. Industrial CPS come in different flavours, depending on the industrial use-case they are intended to serve. These systems have a broad spectrum in terms of size, performance, hardware architecture, interconnectivity, and other defining descriptions. When it comes to less capable industrial CPS in terms of performance, the computational burden of monitoring tools and their noise-like effect on the device develops into a limitation. We would like to avoid these noises, for they do not belong to the original behaviour and they contaminate collected data. As such, amongst different approaches, the reduction of the need to study the system under scrutiny in great detail, would be a promising approach. In this fashion, the goal is to treat such devices not as white boxes, but as black/grey boxes. The term grey box here denotes the fact that although we would like to avoid as much detail regarding the internal mechanisms of the device as possible, we still do need an overview. A black/grey box approach is also beneficial for highly capable industrial CPS, e.g., semiconductor photolithography machines, since it is extremely challenging to capture their vast complexity.

The use of external metrics are especially advantageous for less capable systems, for the monitoring and tooling involved with the collection of internal system metrics dictates the performance penalty of its own. In this context, external metrics, e.g., electrical, are the ones collected without interfering with the operation of the platform. Other systems that would benefit from an external approach are the ones being closed to inspection and probing, necessitating a black box approach.

The foundation for the following discourse in this paper is the fact that industrial CPS are *inherently repetitive*. As such, whichever perspective they are looked at from, their operation and the information related to their operation can be compartmentalised. This will pave the way to the notion of *execution phases*, explained in Section II.

Contribution: In this paper, we will show how Extra-Functional Behavioural (EFB) metrics external to the system, e.g., electrical metrics such as current and power, can be used to distinguish between normal and anomalous behaviour, with the ability to detect the type of anomaly. We will demonstrate our findings in the form of supervised anomaly classification

This paper is composed as a collaboration between the research project 14208, titled *"Interactive DSL for Composable EFB Adaptation using Bisimulation and Extrinsic Coordination (iDAPT)"*, funded by The Netherlands Organisation for Scientific Research (NWO); and the research project titled *"Time, Energy and security Analysis for Multi/Many-core heterogeneous PLAtforms (TeamPlay)"*, supported and partly funded by the European Union Horizon-2020 research and innovation programme under grant agreement No. 779882.

results with exceptional overall accuracy, as high as 99% in certain cases. As far as we are aware, external electrical metrics have not been studied for anomaly detection/classification of a CPS, in a black box approach.

II. Background

A. Extra-functional behaviour

EFB include a computing system's behaviour and as the name suggests, it is not directly derived from functional aspects of the system. Examples are, execution time, different latencies, throughput, power and energy consumption, amongst others. Metrics reflecting EFB are not only dependent on functional behaviour, but also on environmental circumstances, such as the platform itself, the input to the system and operational conditions. These being important variables for CPS, necessitate the role of EFB metrics in their monitoring and analysis.

B. Execution phases

Execution phases are basically repeated units of execution, which are especially noticeable in industrial CPS operations. Repeated tasks can be broken down to their subtasks and higher granularity can be achieved to describe repeated units of execution. We call the smallest of these units an *atomic execution phase*. There may be combined atomic phases that are also repetitive. Such repetitions involving multiple atomic phases are called *combo execution phases* in [3].

C. Signatures and passports

We have adopted and extended the concept of software passports from [3], which were built based on internal system metrics, requiring probing from within the software, e.g., CPU time. A signature is the representation of an execution phase, modelling the trend of EFB metrics collected during that phase. Our technique of choice for this modelling is regression. A passport is a reference signature, collected under normal and reference execution conditions, which is used in comparisons. Note that signatures are calculated per metric.

III. Methodology

We strive to detect and classify performance anomalies in an online fashion. As shown in Fig. 1, the high-level steps of *online monitoring*, *detection* and *identification* are covered by our analytics pipeline. Data collection is followed by its compartmentalisation and application of regression modelling next, resulting in power signatures/passports. The goodness-of-fit values are acquired by comparing a signature to a passport.

During the online monitoring and relevant data manipulations, we compartmentalise (cut) monitoring data based on execution phases, as this will result in comparable pockets of information. These pockets correspond to platform repetitions. Since the amount of monitoring data can be rather large, we use the concept of signatures in general and passports in particular to represent pockets of unknown and reference information, respectively. As passports are essentially regression functions generated over cumulative EFB metric values, they can be conveniently stored and utilised for comparison purposes. We consider goodness-of-fit tests, coefficient of determination (R^2)

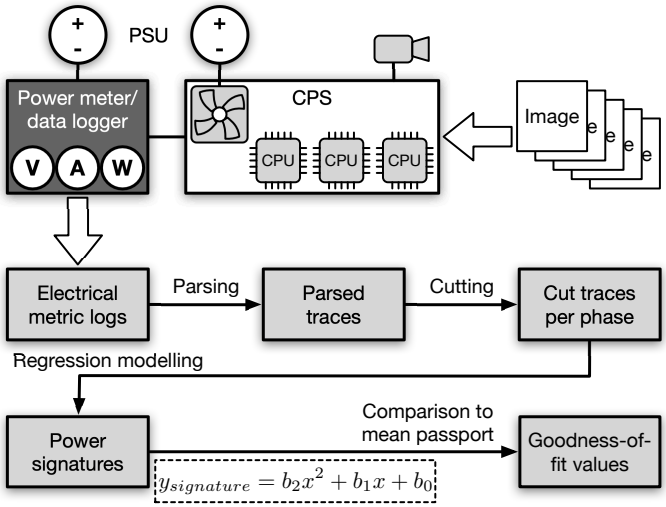


Fig. 1: Our experimental platform, electrical EFB metric collection set-up and the data processing pipeline (note the independent power supply to the fan)

and Root-Mean-Square-Deviation ($RMSD$), when comparing a sample signature to a reference passport. Goodness-of-fit test results are the basis for detection, quantifying the deviation between normal and anomalous conditions. The role of identification is to pinpoint the actual anomaly, with the ability to separate between many, even unknown ones.

The identification step involves the use of classification algorithms such as Decision Tree and Random Forest. The accuracy of these classifiers is affected based on the choice of feature set and input data set. Accordingly, there may be a need for intermediate *analysis* to adjust and further tailor classification parameters. For instance, upon recognition of an unknown anomaly, or low accuracy for a certain known one, the feature set considered can be adjusted. Depending on the actual use-case, this analysis could be an automated one.

IV. Implementation

We have chosen an image processing application, running on an embedded board, as our proof-of-concept. Such systems are deployed in the industry, from assembly lines to unmanned vehicles. Such a system also fits our goal for the lower end of industrial CPS performance spectrum, since these are low-power systems with limited computational capabilities. Furthermore, a platform running an image processing application is highly repetitive, thus mimicking an industrial CPS fairly well.

A. Proof-of-concept platform

Our proof-of-concept set-up is depicted in Fig. 1 and consists of an ODROID-XU4 computing device, implementing the ARM big.LITTLE computing architecture. We are running a stripped-down Linux distribution and the main running application is a neural network-based image analysis software, detecting if cars are present in images. The platform is capable of receiving images from either a camera, or a storage device and in our case, images are provided via a storage device.

For our electrical EFB data collection, we rely on the Otii Arc power data logger unit [4]. The data logger collects

electric potential in Volts and electric current in Amps with the sampling rates of 1 kHz and 4 kHz, respectively. Timestamps for each data collection is also recorded alongside these metrics.

B. Data collection and preparation pipeline

Our experiments do not include any synthesised data and all the data collection was performed during real executions for both regular and anomalous scenarios. Given that we have electrical potential, electrical current and time readings, we consider the three metrics, *current* in Amps, *power* in Watts and *energy* in Milliwatt-hours, for generation of signatures.

The image analysis running on the target platform has two main operations, i.e., reading images from a storage and applying a neural network detection algorithm on them. Thus, we have considered the following atomic and combo phases for current, power and energy readings, as depicted in Fig. 2:

- Image op.: An *atomic* phase for image loading,
- Neural op.: An *atomic* phase for neural network,
- Cycle op.: A *combo* phase for a full image cycle, including image loading and neural network operations.

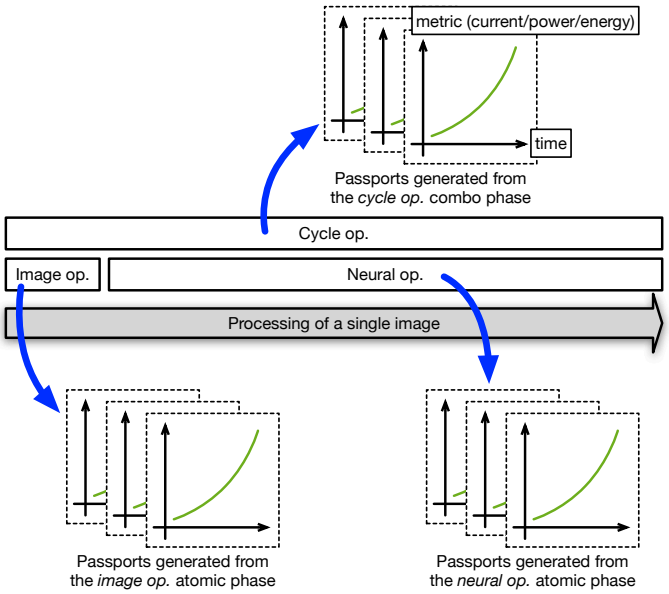


Fig. 2: Different execution phases for an image processing task, i.e., *atomic image operation*, *atomic neural operation* and *combo cycle operation*, combining the first two

C. Implementing power passports

Our tools of choice to represent unique execution phases, as given in Section II, are signatures and passports, with the latter being the signature for a reference phase. Based on the nature of the utilised EFB metrics, the resulting execution representations are what we call *power signatures* and *power passports*. Having three considered metrics means that we will generate three passports per operation. We also transform metric readings and create cumulative values, since creating a regression model over cumulative values results in a more accurate fit and a monotonically increasing regression function. We have experimented with regression functions of different orders and have reached a satisfactory fit with second degree polynomial regressions.

In addition to individual power passports, we are also generating *mean power passports* as a unified representation for many executions with different input data. Mean passports are generated per metric and per phase type. When it comes to mean power passports, their generation is not a straightforward task, as there needs to be matching timestamps. Basically, we are calculating the mean of many regression functions, which can be written as,

$$y_{mean}(x) = \frac{f(x) + g(x) + h(x) + ...}{n},$$

with x being the independent variable, time, and n the number of functions.

Either we have to do listwise deletion and remove independent variable readings which do not exist in all phases, or we have to perform data imputation. The latter is much more preferable, for there are executions that take slightly longer and we do not wish to disregard valid data collection points residing at the end of longer executions. In principle, this applies to other unmatched points as well, regardless of their location within the execution time frame. We have already generated regression models for data collections as their representations and as such, we can perform regression-based imputation by predicting the dependent variable values (metric) for missing independent variable values. This arrangement is especially convenient, since we already have generated and we use these very same regression models in comparisons and goodness-of-fit tests. There will be no extra bias other than what already exists as part of regression models. Fig. 3 depicts this process.

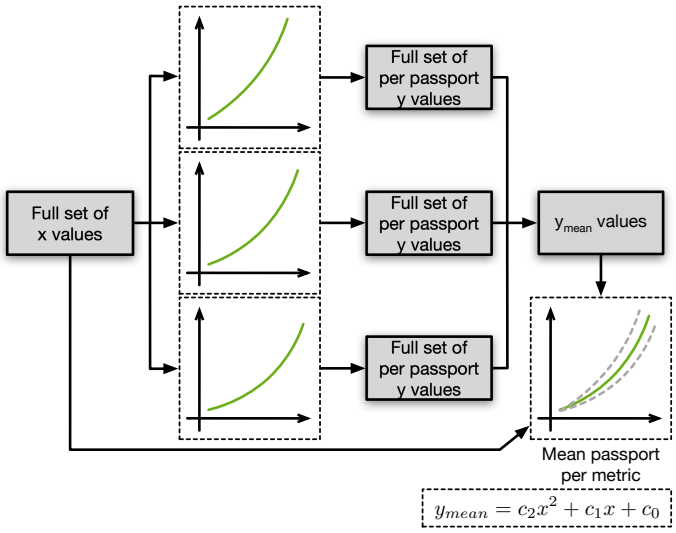


Fig. 3: Mean passport generation flow using the full set of x values (independent variable) from all available passports for the same metric and generating equal size sets of y values (dependent variable) by means of regression-based imputation

D. Comparisons

Having regression functions as representations of reference units of execution (phases) allows us to compare new sets of

sample data against these functions using goodness-of-fit tests. In our case, coefficient of determination (R^2) and Root-Mean-Square-Deviation ($RMSD$) are taken to quantify and compare the amount of deviation. These are standard statistical tests to determine how close a regression function is interpolating data points. Although we have individual and mean passports as potential reference points in comparisons, the only meaningful comparison to perform is a *one-to-mean* comparison..

One-to-mean comparison: This considers the collected sample related to a specific input against the universal mean passport. The metric and the phase should be the same. Therefore, there is no need for special considerations on the mean passport leg of the comparison, as there is only one per metric and per phase, i.e., we do not need to match input images.

Two other comparisons, one-to-one and mean-to-mean, are biased and not realistic for a production environment. Note that for normal conditions, we also perform one-to-mean comparisons, which essentially compares a normal phase signature (passport) to a normal mean passport and will result in minute deviations. This will indicate a normal case for classification.

E. Classification

Out of different employed classification algorithms, best results are coming from Decision Tree (DT) and Random Forest (RF) classifiers[1]. Here, we list the feature set involved in the classification and their brief descriptions.

- Metric: Considered metric out of current, power and energy consumption
- Execution time: Execution time for the phase (this is the one piece of information, turning the view into grey box, instead of a black box one, since we have the boundaries of phases in time)
- Coefficient 2: Coefficient for the second degree term, x^2, of the regression function
- Coefficient 1: Coefficient for the first degree term, x, of the regression function
- Intercept: Intercept value of the regression function
- One-to-mean R^2: Goodness-of-fit value for sample points from one image against the reference mean passport
- One-to-mean $D_i(R^2)$: Absolute difference between the one-to-mean R^2 value and the R^2 value of the reference mean passport
- One-to-mean $RMSD$: Goodness-of-fit value for sample points from one image against the reference mean passport
- One-to-mean $D_i(RMSD)$: Absolute difference between the one-to-mean $RMSD$ value and the $RMSD$ value of the reference mean passport
- Label: Normal, Anomaly 1, Anomaly 2, etc. (NoFan and UnderVolt in this paper)

We split the data into 70% training and 30% test data. Our data analysis pipeline has been written in Python 3.7 and for our regression and classification needs, we rely on Scikit-learn 0.23.1 machine learning library [5].

[1]The hyperparameters used with Scikit-learn library for each classifier besides the default values are as follows: $criterion = entropy$ for DT; $n\_estimators = 100$ for RF.

V. Experimental set-up

The input data for the image analysis application are provided on a storage device. We have considered two different sets of images, first one being proper images with meaningful scenery. This batch includes images with and without a car depicted in them. The second batch includes images that do not depict any particular shape and have purely randomised pixels, introducing variation in the input. In this fashion, we could evaluate if the composition of an image is a factor for our workflow. Each batch is used in two different executions, one involving a single round of image analysis and the second, involving ten rounds of image analysis, meaning the same batch is processed ten times, sequentially. We have chosen the number of rounds arbitrarily and with the aim to have a long enough execution, reflecting the effects of anomalies. Each batch includes 30 images, making the workload for ten rounds of processing as 300 images. We have also executed every combination of conditions twice, by assigning the application to either a big, or a little core on the platform. The list of performed data collections are as follows:

- Case 1: 1 execution round, regular images, little core
- Case 2: 10 execution rounds, regular images, little core
- Case 3: 1 execution round, regular images, big core
- Case 4: 10 execution rounds, regular images, big core
- Case 5: 1 execution round, randomised images, little core
- Case 6: 10 execution rounds, randomised images, little core
- Case 7: 1 execution round, randomised images, big core
- Case 8: 10 execution rounds, randomised images, big core

The structuring of our workloads for the aforementioned data collection cases, fits the principle of repetitive task execution for industrial CPS rather well. Keep in mind that although these tasks are repetitive, but the underlying non-determinism is still present, as the behaviour of the system has subtle variations per execution, even with the same exact input. Now that different cases are defined, we have considered two different anomalies, affecting the performance and the reliability of the system.

a) Malfunction of the cooling system: For this anomaly, henceforth called *NoFan*, we have disabled the cooling fan of the platform's CPU block. Keep in mind that in our set-up, the fan has a separate supply of power (PSU) to begin with and will not directly affect electrical EFB metric readings of the platform, as depicted in Fig. 1.

b) Unstable power delivery: For this scenario, henceforth called *UnderVolt*, we have reduced the voltage supply to a level below the required amount for the platform, but still keeping the device functional. This was a reduction from 5.0 Volts to 4.7 Volts. We have also made sure that the voltage supply is at a sufficient level and it will not result in glitches.

We have considered the exact same cases as the normal circumstances, with different batches of images, different numbers of processing rounds and different cores. Accordingly, our experiments resulted in eight cases for each scenario, representing Normal, NoFan and UnderVolt situations. It must be mentioned that having equal number of cases and basically equal number of data fields for all scenarios is advantageous as it will result in

a balanced data set for classifier algorithms. Having a balanced data set will eliminate the need for imbalance countering techniques, e.g., undersampling and oversampling.

VI. Results

Our results mostly focus on the prediction accuracy of classification algorithms as it is our ultimate goal to be able to detect and classify anomalies. We have tried Decision Tree (DT) and Random Forest (RF) classifiers with different subsets of our data set and we observe that the choice of phase and metric as sources of data has a considerable effect on the prediction accuracy of the classification. We have tried classifications using data from all three metrics at the same time, as well as every metric individually. For all of these classification trials, we have considered either atomic image op., atomic neural op., combo cycle op., or the combination of the data from both atomic image and atomic neural operations. Table I presents these choices alongside their resulting overall prediction accuracies.

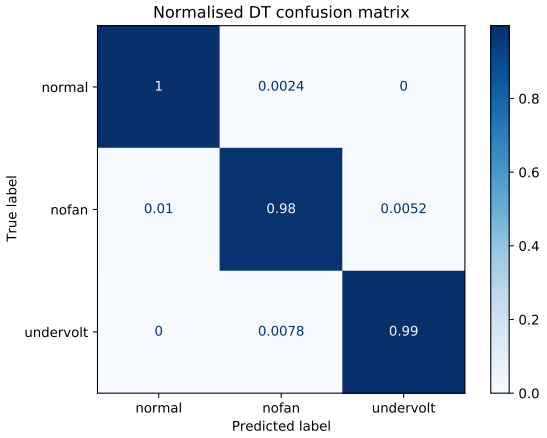
TABLE I: Overall prediction accuracy of Decision Tree (DT) and Random Forest (RF) classifiers based on data sets generated from different combinations of metrics and phases

Metric	Phase	DT acc. (%)	RF acc. (%)
current	image op.	89.83	92.80
current	neural op.	98.89	98.81
current	cycle op.	99.15	99.23
current	image op. + neural op.	94.23	96.06
power	image op.	81.79	83.23
power	neural op.	96.86	96.61
power	cycle op.	97.03	97.54
power	image op. + neural op.	89.70	91.48
energy	image op.	74.25	77.13
energy	neural op.	96.69	97.37
energy	cycle op.	97.29	97.71
energy	image op. + neural op.	86.99	88.47
all three	image op.	83.05	84.89
all three	neural op.	97.85	98.22
all three	cycle op.	97.96	98.50
all three	image op. + neural op.	89.86	91.52

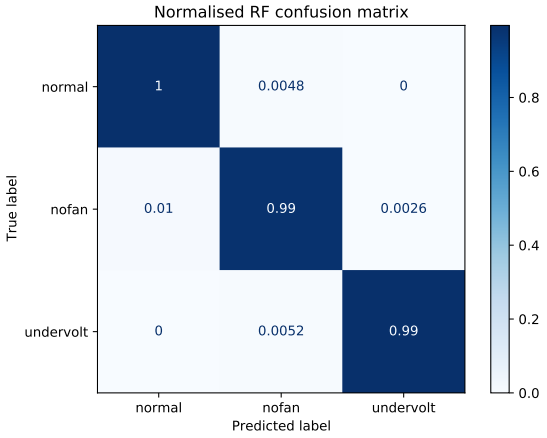
The overall prediction accuracy of DT and RF classifiers, while using the data generated from *combo cycle op.* phases together with the electrical current metric, are 99.15% and 99.23% respectively, which are the highest accuracies out of all combinations. The confusion matrices are depicted in Fig. 4. High prediction accuracy is observable across the board as a result of having a balanced data set.

A. Analysis and discussion

Contrary to the common intuition, combining different phases and passports generated from these phases in the data set does not improve the accuracy beyond what we have achieved with data sets based on a single phase. The same also applies to combinations of metric data. It is possible to investigate the suitability of phase data for these classifiers by looking at the DT graph visualisation. The more contained the graph, the easier it is for the classification algorithm to distinguish between labels. In this case, the DT graph visualisation for the



(a) DT classifier



(b) RF classifier

Fig. 4: Normalised confusion matrices for (a) Decision Tree (DT) classifier and (b) Random Forest (RF) classifier, considering the anomaly categories, Normal, NoFan and UnderVolt as labels, with the data set put together for the electrical current metric and for the combo cycle op. phase

combo cycle operation with electric current as the metric of choice, has a rather contained complexity.

Looking back at the definition of our phases, this is an expected effect. Considered anomalies in this paper are of the type affecting the computational capability of the system, whereas an atomic image operation mostly reflects the I/O activity while loading an image. However, we must take into account that for other types of anomalies, it might be the case that the atomic image op. is a more definitive source of data, so we cannot flat out disregard such sources of data.

One might argue that anomaly specific monitoring is also capable of detection, e.g., monitoring voltage supply can reveal instabilities. Such an approach is just that, single anomaly specific. Such detection assumes prior knowledge of possible anomaly types and every anomaly will require its own detection. However, our methodology can detect and differentiate between multiple anomalies by virtue of looking into the system behaviour, making it capable of handling anomalies with unknown causes as well.

B. Explainable output

One of the characteristics of our methodology is the fact that it provides analytically comprehensible output. Accordingly, we can backtrack our steps leading to a specific result and check the data relevant to each step. The advantages are twofold here. While such a capability can help with optimising the workflow itself, e.g., by choosing better metrics and phases as the sources of data, it could also provide information for better future designs. For instance, if a specific batch of inputs to the system results in unexpected classification results, individual passports can reveal which atomic phase of the execution is misbehaving.

VII. Related work

The research community has been aware of the importance of anomaly detection and classification, with efforts in this regard being well-documented in surveys by Chandola et al. [6] and Ibidunmoye et al. [7]. The challenging nature of decision support, leading to actuations and better designs for industrial systems, is also attested by the grand challenges presented by Fowler [8]. Most of the body of work given in [7] focus on performance anomalies in distributed systems [9], cloud environments [10] and web applications [11], whereas our methodology is specifically tailored towards industrial CPS. We are of the opinion that our approach helps in simplifying the task for repetitive systems by considering execution phases, regression-based representations and more importantly, black box views of industrial CPS.

Electrical metrics, especially electrical power analysis, have a profound role in cybersecurity research. The famous and now classic paper by Kocher et al. [12] is a great example. While such publications consider electrical metrics as a source of side-channel information and an offensive, secret-revealing metric, we consider them as reflections of functional behaviour.

To the best of our knowledge, the use of electrical metrics for anomaly detection/classification of CPS, while adopting a black box approach, has not been studied. The only aspect distancing us from a purely black box approach is the knowledge of phase durations. Change Point Detection (CPD) [13] can automate phase detection using the collected external metric data.

The survey by Chandola et al. [6] includes classification algorithms from other works. These algorithms are applicable to repetitive data such as ours. We specifically have focused on DT [14] and RF [15] algorithms, as these were the most promising in terms of prediction accuracy. As provided in Table I, we have improved the initially observed accuracy by choosing the right sources of data.

VIII. Conclusion and future work

We have shown that the concept of software passports based on system-level EFB metrics, along with the accompanying tooling, can be successfully extended to EFB metrics external to the system, e.g., voltage, current, power and energy consumption metrics. We have specifically considered electrical metrics, current, power and energy consumption in our experiments, data collections and *power passport* calculations. Power signatures/passports can be successfully used to detect anomalies and contribute to fault tolerance.

Given this methodology's previous application with a large-scale and arguably much more complex industrial CPS platform, a semiconductor photolithography machine, both ends of the industrial CPS spectrum have been covered, from large platforms to small ones. The approach has been valid across-the-board and passports are shown to be rather flexible constructs. We have shown that a detailed white box view of industrial CPS is not a necessity and one can still generate definitive signatures/passports based on a grey/black box view. What we needed was just the start and end timestamps for different operations, denoting the boundaries of execution phases.

Our classification pipeline is capable of labelling cases with gradually changing feature values, as the move from the normal condition to anomalous, e.g., NoFan, is a gradual process.

Future work: There are quite a few interesting extensions conceivable for this paper. Considering that our methodology is an explainable one, using supervised and rather comprehensible machine learning algorithms at its final step, it would be interesting to explore the application of more advanced algorithms, such as Convolutional Neural Networks. Speaking of automation, CPD as a method of automatic phase detection is a natural extension to this work. We would like to examine the performance and the accuracy of neural networks by applying them directly to the collected and parsed data at different steps.

References

[1] P. Derler, E. A. Lee, and A. Sangiovanni Vincentelli, "Modeling cyber-physical systems," *Proceedings of the IEEE*, 2012.

[2] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *CIRP Annals*, 2016.

[3] U. Odyurt, H. Meyer, A. D. Pimentel, E. Paradas, and I. G. Alonso, "Software passports for automated performance anomaly detection of cyber-physical systems," in *Embedded Computer Systems: Architectures, Modeling, and Simulation*, 2019.

[4] Qoitech AB, "Otii by qoitech," 2019. [Online]. Available: https://www.qoitech.com/otii/

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, 2011.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, 2009.

[7] O. Ibidunmoye, F. Hernández-Rodriguez, and E. Elmroth, "Performance anomaly detection and bottleneck identification," *ACM Comput. Surv.*, 2015.

[8] J. W. Fowler and O. Rose, "Grand challenges in modeling and simulation of complex manufacturing systems," *SIMULATION*, 2004.

[9] D. Gunter, B. L. Tierney, A. Brown, M. Swany, J. Bresnahan, and J. M. Schopf, "Log summarization and anomaly detection for troubleshooting distributed systems," in *2007 8th IEEE/ACM International Conference on Grid Computing*, 2007.

[10] S. Fu, "Performance metric selection for autonomic anomaly detection on cloud computing systems," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011.

[11] L. Cherkasova, K. Ozonat, , J. Symons, and E. Smirni, "Anomaly? application change? or workload change? towards automated detection of application performance anomaly and change," in *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*, 2008.

[12] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, 1999.

[13] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and Information Systems*, 2017.

[14] L. Rokach and O. Maimon, *Decision Trees*, 2005.

[15] L. Breiman, "Random forests," *Mach. Learn.*, 2001.