



Unsupervised Anomaly Detection in Production Lines

Alexander Graß, Christian Beecks, Jose Angel Carvajal Soto

Fraunhofer Institute for Applied Information Technology FIT, Germany
{alexander.grass, christian.beecks, angel.carvajal}@fit.fraunhofer.de

Abstract. With an ongoing digital transformation towards industry 4.0 and the corresponding growth of collected sensor data based on cyber-physical systems, the need for automatic data analysis in industrial production lines has increased drastically. One relevant application scenario is the usage of intelligent approaches to anticipate upcoming failures for maintenance. In this paper, we present a novel approach for anomaly detection regarding predictive maintenance in an industrial data-intensive environment. In particular, we are focusing on historical sensor data from a real reflow oven that is used for soldering surface mount electronic components to printed circuit boards. The sensor data, which is provided within the scope of the EU-Project COMPOSITION (under grant no. 723145), comprises information about the heat and the power consumption of individual fans inside a reflow oven. The data set contains time-annotated sensor measurements in combination with additional process information over a period of more than seven years.

Keywords: Unsupervised Learning, Industry 4.0, Anomaly Detection

1 Introduction

In the last couple of years, the importance of cyber-physical systems in order to optimize industry processes, has led to a significant increase of sensorized production environments. Data collected in this context allows for new intelligent solutions to e.g. support decision processes or to enable predictive maintenance. One problem related to the latter case is the detection of anomalies in the behavior of machines without any kind of predefined ground truth. This fact is further complicated, if a reconfiguration of machine parameters is done on-the-fly, due to varying requirements of multiple items processed by the same production line. As a consequence, a change of adjustable parameters in most cases directly leads to divergent measurements, even though those observations should not be regarded as anomalies.

In the scope of the EU-Project COMPOSITION (under grant no. 723145), the task of detecting anomalies for predictive maintenance within historical sensor data from a real reflow oven was investigated. While the oven is used for soldering surface mount electronic components to printed circuit boards based on continuously changing recipes, one related problem was the unsupervised recognition

of potential misbehaviors of the oven resulting from erroneous components. The utilized data set comprises information about the heat and power consumption of individual fans. Apart from additional machine parameters like a predefined heat value for each section of the oven, it contains time-annotated sensor observations and process information recorded over a period of more than seven years.

As one solution for this problem, in the upcoming chapters we will present our approach named Generic Anomaly Detection for Production Lines, short GADPL. After a short introduction on related approaches, in the upcoming chapters we will focus on a description of the algorithm. Afterwards we outline the evaluation carried out on the previously mentioned project data, followed by a concluding discussion on the approach and future work.

2 Related Work

While the topic of anomaly detection and feature extraction is covered by a broad amount of literature, in the following we will focus on a selection of approaches that led to the here presented algorithm. Recently, the automatic description of time series, in order to understand the behavior of data or to perform subsequent operations has drawn the attention of many researchers. One idea in this regard is the exploitation of Gaussian processes [3, 5] or related structural compositions [4]. Here, a time series is analyzed using a semantically intuitive grammar consisting of a kernel alphabet. Although corresponding evaluations show impressive results, they are rather applicable to smaller or medium sized historical data, since the training of models is comparatively time consuming. In contrast, other approaches exist, which focus on the extraction of well-known statistical features, further optimized by means of an additional feature selection in a prior stage [2]. However, the selection of features is evaluated based on already provided knowledge and thus not applicable in unsupervised use-cases. A last approach discussed here, uses the idea of segmented self-similarity joins based on raw data [7]. In order to decrease the complexity, segments of a time series are compared against each other in the frequency domain. Even though this idea provides an efficient foundation for many consecutive application scenarios, it lacks the semantic expressiveness of descriptive features as it is the case for the already mentioned methods.

In the upcoming chapter we consequently try to deal with those challenges, while presenting our approach for unsupervised anomaly detection.

3 Approach

The hereafter presented description of GADPL is based on the stage-wise implementation of the algorithm. After an initial clustering of similar input parameters (3.1) and a consecutive segmentation (3.2), we will discuss the representation of individual segments (3.3) and the corresponding measurement of dissimilarity

(3.4). GADPL is also summarized in figure Algorithm 1, at the end of this chapter.

3.1 Configuration Clustering

In many companies, as well as in the case of COMPOSITION, a single production line is often used to produce multiple items according to different requirements. Those requirements are in general defined by varying machine configurations consisting of one or more adjustable parameters, which are changed 'on-the-fly' during runtime. For a detection of deviations with respect to some default behavior of a machine, this fact raises the problem of invalid comparisons between sensor measurements of dissimilar configurations. If a measurement or an interval of measurements is identified as an anomaly, it should only be considered as such, if this observation is related to the same configuration as observations representing the default behavior. In other words:

If $C_k = \{x_l := \lambda_l | 1 \leq l \leq M\}$ is a configuration with M parameters x_l of value λ_l , then for the dissimilarity δ of two measurement representations $y_{1,i}$ and $y_{2,j}$ with associated configurations C_i and C_j , it holds that:

$$\delta(y_{1,i}, y_{2,j}) \text{ is defined iff. } i == j$$

Therefore in advance to all subsequent steps, at first all sensor measurements have to be clustered according to their associated configuration.

For simplicity, in the following subsections we are only discussing the process within a single cluster, although one has to keep in mind, that each step is done for all clusters in parallel.

3.2 Segmentation

As a result of the configuration-based clustering, the data is already segmented coarsely. However, since this approach describes unsupervised anomaly detection, the idea of a further segmentation is, to create some kind of ground truth, which reflects the default behavior of a machine. In subsection 3.4 we will see, how the segmentation is utilized to implement this idea. In an initial step, a maximum segmentation length is defined, in order to specify the time horizon, after which an anomaly can be detected. Assuming a sampling rate of 5mins per sensor, the maximum length of a segment would consequently be $(60 \cdot 24)/5 = 288$ to describe the behavior on a daily basis. Although a decrease of the segment length implies a decrease of response time, it also increases the computational complexity and makes the detection more sensitive to invalid sensor measurements. In this context, it needs to be mentioned that in this stage segments are also spitted, if they are not continuous with respect to time as a result of missing values. Another fact that has to be considered is the transition time of configuration changes. While the input parameters associated with a configuration change directly, the observations might adapt more slowly and therefore blur the expressiveness of the new segment. To prevent this from happening,

the transition part of all segments, which have been created due to configuration changes, gets truncated. If segments become smaller than a predefined threshold, they can be ignored in the upcoming phases.

3.3 Feature Extraction

Having a set of segments for each configuration, the next step is to determine the characteristics of all segments. While the literature presents multiple approaches to describe the behavior of time series, we will focus on common statistical features extracted from each segment. Nonetheless, the choice of features is not fixed, which is why any feature suitable for the individual application scenario can be used. One example for rather complex features could be the result of a kernel fitting in the context of Gaussian processes, accepting a decrease in performance. Since the goal is to capture comparable characteristics of a segment, we compute different real-valued features and combine them in a vectorized representation. In the case of COMPOSITION, we used the mean to describe the average level, the variance as a measure of fluctuation and the lower and upper quartiles as a coarse distribution-binning of values. Due to the expressiveness of features being dependent from the actual data, one possible way to optimize the selection of features is the Principal Component Analysis [6]. Simply using a large number of features to best possibly cover the variety of characteristics might have a negative influence on the measurement of dissimilarity. The reason for this is the partial consideration of irrelevant features within distance computations.

Moreover, since thresholds could be regarded as a more intuitive solution compared to additionally extracted features, this replacement would lead to a significant decrease in the number of recognized anomalies. Apart from the sensitivity to outliers, the reason is a neglect of the inherent behavior of a time series. As an example consider the measurements of an acoustic sensor attached to a motor that recently is sending fluctuating measurements, yet within the predefined tolerance. Although the recorded values are still considered as valid, the fluctuation with respect to the volume could already indicate a nearly defect motor. Finally, one initially needs to evaluate appropriate thresholds for any parameter of each configuration.

3.4 Dissimilarity Measurement

For now we discussed the exploitation of inherent information, extracted from segmented time series. The final step of GADPL is to measure the level of dissimilarity for all obtained representatives. Since no ground truth is available to define the default behavior for a specific configuration, the algorithm uses an approximation based on the given data. One problem in this regard is the variability of a default behavior, consisting of more than one pattern. Therefore, a naive approach as choosing the most occurring representative, would already fail for a time series consisting of two equally appearing patterns captured by different segments, where consequently half of the data would be detected as

Algorithm 1 GADPL

Require: Time series T , Machine parameters M , Configuration transition time p , Segment length (l_{min}, l_{max}) , Number of nearest neighbors k , Dissimilarity threshold Δ_{max}

```

 $C' = \text{cluster\_configurations}(T, M)$ 
 $R' = \{R_1, \dots, R_{|C'|}\}$ 
for all configuration segments  $C_i$  in  $C'$  do
  for all segments  $s_j$  in  $C_i$  do
     $s_j = \text{truncate\_transitions}(s_j, p)$ 
    if  $|s_j| < l_{min}$  then
       $C_i = C_i \setminus s_j$ 
    else if  $|s_j| > l_{max}$  then
       $s'_j = \text{split\_segments}(s_j, l_{max})$ 
       $C_i = C_i \cup s'_j$ 
       $C_i = C_i \setminus s_j$ 
    end if
     $R_i = R_i \cup \text{extract\_features}(s'_j)$ 
  end for
end for
for all configuration representatives  $R_i$  in  $R'$  do
  for all representatives  $r_j$  in  $r_i$  do
     $NN_k = \text{query\_index}(r_j, k)$ 
    if  $\Delta(r_j, NN_k) > \Delta_{max}$  then
       $\text{emit\_anomaly}(i, j)$ 
    end if
  end for
end for

```

anomalous behavior.

As one potential solution GADPL instead uses the mean over a specified size of nearest neighbors, depicting the most similar behavior according to each segment. The idea is that even though there might multiple distinct characteristics in the data, at least a predefined number of elements represent the same behavior compared to the processed item. Otherwise, this item will even have a high average dissimilarity with respect to the most similar observations and can therefore be classified as anomaly.

Let r_i be the representative vector of the i -th segment obtained by feature extraction and let $NN_k(r_i)$ be the according set of k nearest neighbors. The dissimilarity measure Δ for r_i is defined as:

$$\Delta(r_i, NN_k(r_i)) = \frac{1}{k} \sum_{j=1}^k \delta(r_i, NN_k^j(r_i))$$

where $NN_k^j(r_i)$ corresponds to the j -th nearest neighbor and δ to a ground distance defined on \mathbb{R}^n .

Here, for the vectorized feature representations, any suitable distance function δ is applicable. In the context of COMPOSITION we decided to use the Euclidean distance for a uniform distribution of weights, applied to normalized

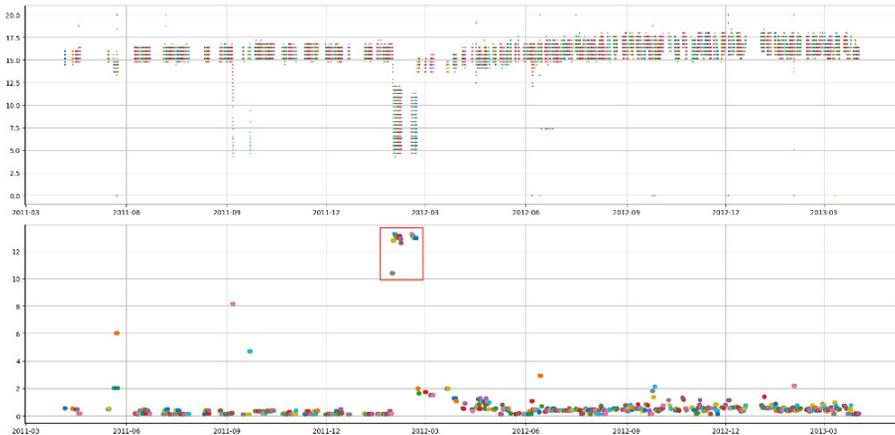


Fig. 1. Application of GADPL: The upper part shows the segmentation of time annotated power consumption data in percent. The lower part illustrates the result of the dissimilarity measurement, where the red rectangle indicates classified anomalies.

feature values. To further increase the performance of nearest neighbor queries, we exploited the R*-tree [1] as a high-dimensional index structure.

Given the dissimilarity for each individual representative together with a predefined anomaly threshold, GADPL finally emits potential candidates having an anomalous behavior.

4 Evaluation

In this section we will discuss the evaluation performed on a historical data set, provided in the scope of COMPOSITION. While in future, the algorithm should be applied to continuously streamed sensor data, the initial evaluation was performed on recorded data, captured over a period of seven years. The data consists of machine parameters (already classified by recipe names) and time-annotated sensor measurements including temperature value and power consumption, based on a sampling rate of 5 minutes. In addition, a separate maintenance log covers the dates of previous fan exchanges. However, malfunctions only occurred two times during runtime and are therefore comparatively rare. A confirmation of results due to actual defect components is consequently restricted to some extent. Since this project and the here presented approach are regarded as ongoing work, the outlined evaluation is continued likewise.

Figure 1 illustrates the application of GADPL, including segmentation (upper part) and dissimilarity measurement (lower part), for the time around one fan failure. Here, differently colored circles represent slices of the time series after segmentation, describing the percentage power consumption of a fan. Using the features mentioned in section 3.3, we intended to perceive deviating values and untypical fluctuations within the data, without being sensitive to outliers arising from single incorrect sensor measurements. Having one of the recorded fan

exchanges at the end of February 2012, the result of the algorithm clearly shows significantly higher values for the dissimilarity (red rectangle) prior to the event. Even though increased dissimilarity values at the end of May 2011 and around September 2011 can be explained by analyzing the original data, yet there were no recordings for a defect component at those times. However this does not automatically imply incorrect indications, since defect machine parts are not the only reasoning for anomalous characteristics in the data. An appropriate choice for the value of a maximal dissimilarity, defining the anomaly threshold, can therefore highly influence the accuracy.

Both cases of a defect fan behavior were clearly captured by the algorithm and emphasized by a high dissimilarity.

5 Conclusion

With GADPL we introduced a solution to the relevant topic of unsupervised anomaly detection in the context of configuration-based production lines. After a short outline on the topic and related work, we discussed the algorithm and the associated intention of our approach, before briefly showing the evaluation results based on the project data.

Since the approach is ongoing work, in the future we will primarily extend our evaluation based on streaming data. Although we described the algorithm using historical data, the procedure for streaming data is carried out analogously. Another point in the scope of future evaluations is the choice of more complex features and a related automated feature selection. Another idea to further improve the approach is a semantic segmentation of the time series. While currently a time series is segmented exploiting domain knowledge, a segmentation based on characteristics in the data might potentially increase the accuracy. This would also prevent from an unappropriated choice of the maximal segmentation length, which could result in a split of data within a potential motif.

Finally, we plan to investigate the correlation of anomalies within multivariate data. If GADPL in its current state is used for multivariate time series data, each dimension is processed independently. Combining inter-dimensional information within a single dissimilarity measure to cover anomalies would therefore be a useful functionality to further optimize the approach.

6 Acknowledgements

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 723145 - COMPOSITION. This paper reflects only the authors views and the commission is not responsible for any use that may be made of the information it contains.

References

1. Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r^* -tree: an efficient and robust access method for points and rectangles. In *Acm Sigmod Record*, volume 19, pages 322–331. Acm, 1990.
2. Maximilian Christ, Andreas W. Kempa-Liehr, and Michael Feindt. Distributed and parallel time series feature extraction for industrial big data applications. *CoRR*, abs/1610.07717, 2016.
3. David Duvenaud, James R. Lloyd, Roger Grosse, Josh B. Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search. In Sanjoy Dasgupta and David McAllester, editors, *ICML 2013: Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *JLMLR Proceedings*, pages 1166–1174. JMLR.org, June 2013.
4. Roger Grosse, Ruslan Salakhutdinov, William T. Freeman, and Joshua B. Tenenbaum. Exploiting compositionality to explore a large space of model structures. In Nando de Freitas and Kevin Murphy, editors, *Proceedings of the 28th Conference in Uncertainty in Artificial Intelligence*, Corvallis, Oregon, USA, 2012. AUAI Press.
5. James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models. *CoRR*, abs/1402.4304, April 2014.
6. Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
7. Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 1317–1322. IEEE, 2016.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

