A Data Management and Anomaly Detection Solution for the Entertainment Industry

(Discussion Paper)

Michele Berno[1], Marco Canil[1], Nicola Chiarello[1], Luca Piazzon[1], Fabio Berti[2], Francesca Ferrari[2], Alessandro Zaupa[2], Nicola Ferro[1], Michele Rossi[1] and Gian Antonio Susto[1]

[1]*Department of Information Engineering, University of Padua, Italy*
[2]*Antonio Zamperla S.p.A., Italy*

Abstract

We present a smart monitoring system that combines a data management architecture with an unsupervised anomaly detection technique, targeting the automated equipment in the entertainment industry. Anomaly detection uses state-of-the-art univariate and multivariate algorithms, as well as recently proposed techniques in the field of explainable artificial intelligence, to achieve enhanced monitoring capabilities and optimize service operations. The monitoring system is here presented and tested on a real-world case study, i.e., an amusement park ride.

Keywords

anomaly detection, predictive maintenance, data management, entertainment industry

1. Introduction

Antonio Zamperla S.p.A.[1] designs and develops entertainment rides, continuously striving for their rides to become safer, greener and more efficient. Smart monitoring systems are expected to enrich the entertainment industry with a number of key features. For example, faults could be predicted in advance, or data analysis during the testing of the ride's prototypes could give deeper insights onto various design choices. Furthermore, maintenance operations are nowadays performed manually, following ride's manuals or government directives that are usually scheduled on a periodic basis, regardless of the actual conditions of the machines. This implies that, oftentimes, maintenance is performed without a real need, entailing a waste of time, human resources and material. This practice, although being cautious and robust, is quite inefficient. Smart monitoring systems would allow a change of paradigm from the current conservative approach to a greener and more efficient one. Lastly, automated supervision techniques would enforce the safeness of the rides by detecting subtle anomalies, which would be hardly identified by a human supervisor.

In the context of advanced monitoring of complex systems, two main Machine Learning

SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy
gianantonio.susto@unipd.it (G. A. Susto)

[1]https://www.zamperla.com/

(ML)-based technologies have emerged in recent years: unsupervised *anomaly detection* (AD) [1], that aims at providing enhanced diagnostic capabilities, and *predictive maintenance* [2], with the purpose of predicting failures/degradation to enable the early intervention of maintenance operators.

In this work, we present a data management architecture that combines state-of-the-art univariate and multivariate approaches for AD to reach enhanced monitoring capabilities and optimize service operation. We use a database to store the streaming data acquired by sensors, feed them to the AD algorithms, store back the features extracted by the AD algorithms, relate them to alerts and maintenance events, and support an overall Web application.

The paper is organized as follows: Section 2 briefly summarizes related work; Section 3 presents the considered use case; Section 4 describes the Entity-Relationship schema of the datastore; Section 5 introduces the proposed approach for AD; Section 6 discusses the current AD prototype and the related Web application. Finally, concluding remarks and future works are reported in Section 7.

2. Related Work

Unsupervised AD tools adopt (i) multivariate approaches based on tabular data and (ii) univariate approaches working with time-series [3]: (i) multivariate approaches [4] have the advantage of capturing multivariate anomalous behaviour that typically goes undetected by classic chart-based monitoring tools, but, when applied to time-series data, they entail the use of feature extraction procedures that are typically time-consuming for the developers and may lead to loss of information; (ii) univariate approaches typically work by predicting residuals, i.e., comparing measured and forecast time-series data, and raising an alarm as their difference exceeds a threshold. While Deep learning techniques are available for (i) and (ii), they typically need to be adapted to cope with discrete production data, where time-series are usually split into batches representing the machine cycles [5]. Another relevant issue in the monitoring field is the so-called *concept drift*, which means that the statistical proprieties of the target variables change over time in an unforeseen way [6], posing the additional challenge of tracking or estimating it.

3. Use Case

The case study is represented by a coaster (Zamperla's *DangleZ*) whose seats can freely move during the ride, independently of the underlying chassis, see Figure 1. In this case study, the coaster track is 133 meters long.

The data acquisition process provides, by means of an on-board *PLC*, a set of $q = 55$ different time series acquired with irregular sampling rate. Each time series represents a signal which can be analyzed to monitor the behavior of the machine: 9 signals are generated by the equipment sensors that report the transit of the coaster over different locations of the rail; 15 signals describe voltages, currents, frequencies and other physical quantities denoting the consumption and the movement of the machine; 5 signals are acquired by a weather station that detects the condition of the surrounding environment; the remaining signals are needed to check the

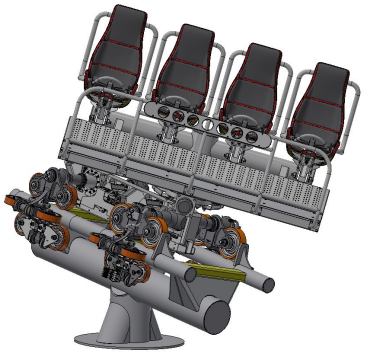

Figure 1: Case study: schema of Zamperla's *DangleZ* ride.

correct working conditions and the security of the machine (for instance, the state of the safety button).

Data are divided into *sessions* (or tests) and each session is composed of a number of *cycles*. Each cycle collects the data generated during one run of the coaster, from the moment it leaves the station to its return to the station. Each session contains data coming from an homogeneous set of measurements, typically measurements in the same day or under the same wheater conditions.

4. Conceptual Design

Figure 2 shows the Entity-Relationship schema of our database, containing four principal sets of entities:

- *parks and users* (green): park names with a reference email and Profile information of the users allows to access and insert acquisition data (Park Managers) or analysis data (Analysts);
- *ride related information* (red): each ride has a specific model and is equipped with a set of acquisition devices. All the maintenance events are stored together with their category (ordinary/extraordinary maintenance or hardware upgrade) and their type (e.g., components lubrication, replacement of gaskets). Events data are used in the AD system to identify the different feasible working conditions of the machines;
- *acquisition data* (yellow): data are organized into sessions, each of which represents a set of ride cycles under the same underlying conditions, i.e., the acquisition data related to a full morning or afternoon. These conditions are stored in the form of metatag values related to the session (e.g., weather condition, load, notes). As explained above, the acquisition is organized into cycles, each one representing a full run of the coaster. During the acquisition, each device gathers data for a set of variables, each identified by a unique TAG name. Synonymy relations between tags are stored to deal with old tag names in legacy acquisitions. Data are stored in packets, each one containing a vector of values (data_array) for a TAG coming from a device during a specific cycle;

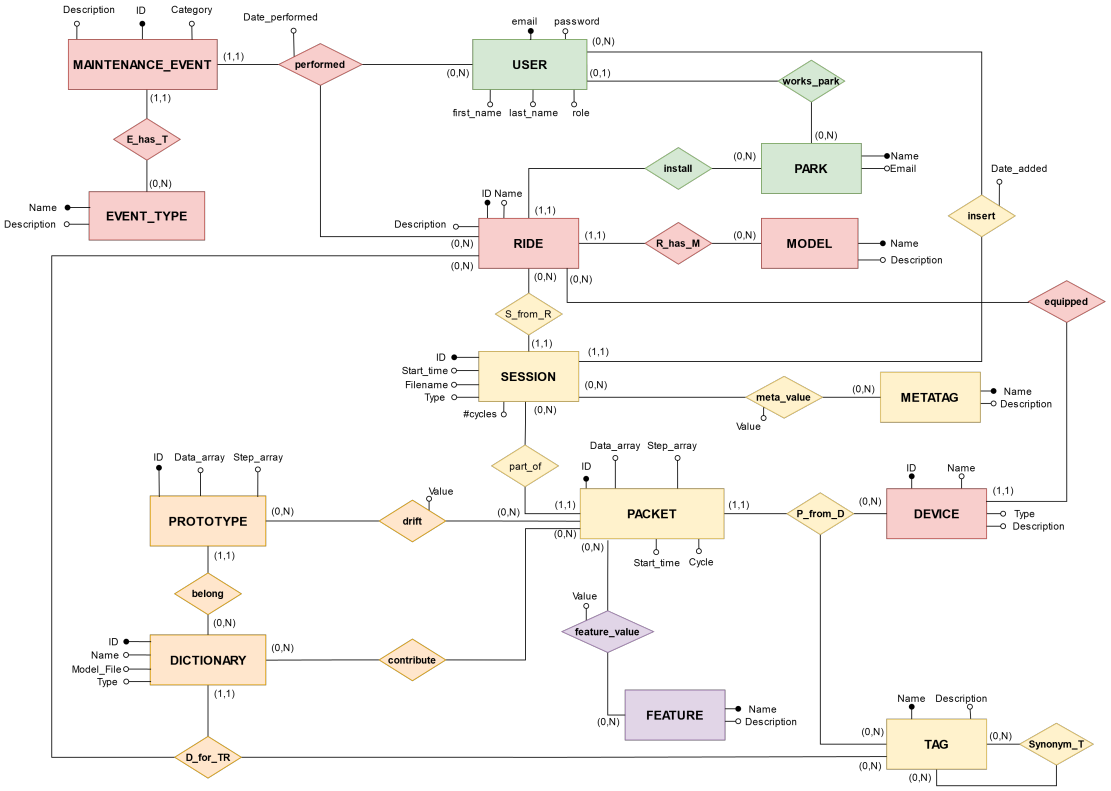

Figure 2: Entity-Relationship schema of the database.

- *analysis related data*: dictionaries (the obtained prototypes) for the univariate AD and the corresponding drift value of each packet are stored (orange entities). Features for the multivariate AD and their values for each packet are stored (purple entities).

5. Anomaly Detection

The proposed approach integrates univariate and multi-variate methods, as illustrated in Figure 3:

- at a given machine cycle, a set of raw signals $\mathcal{X} = \{X_1, \ldots, X_q\}$ coming from the equipment sensors (and additional context information) are fed to a *feature extraction* block that computes features $x_1, \ldots, x_p$, where it typically holds $p \neq q$. While signals X_j are usually time-series, features x_i are scalars. Here, we consider the case where each feature x_i is computed from a single raw signal X_j and we denote the set of features computed from X_j by $\mathcal{S}_{X_j}$;
- such features are used within a multivariate AD block having two objectives: (i) obtaining an *anomaly score* $s(\cdot)$, a quantitative indicator that summarizes the degree of "outlierness" of the machine cycle $\mathcal{X}$ under exam; (ii) using an XAI approach to obtain a *feature*

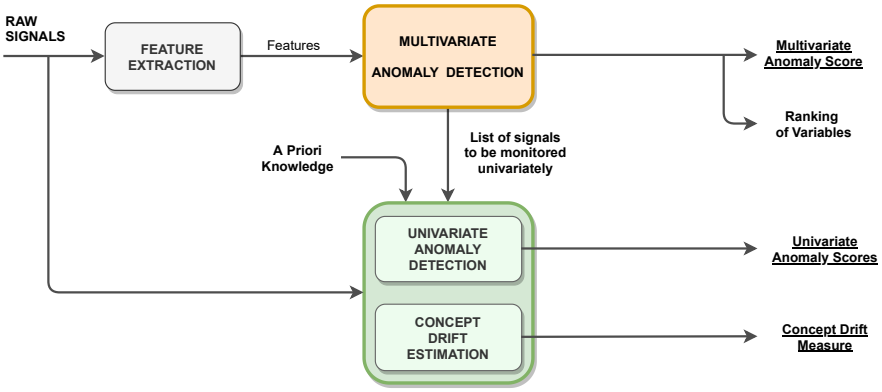

Figure 3: Functional schema of the proposed Anomaly Detection approach.

importance score $f_i(\mathcal{X}), \forall\, i \in \{1, \ldots, p\}$. $f_i(\mathcal{X})$ is a quantitative index that summarizes the impact of feature x_i in identifying machine cycle $\mathcal{X}$ as anomalous.

- raw signals that deserve to be monitored separately with a time-series based (univariate) approach are detected based on (i) expert knowledge and on (ii) the feature importance $f_i(\mathcal{X})$ coming from the AD module. The rationale is that some anomaly types are better identified by analyzing a specific time-series. To be reliably detected, the respective signal X_i should be independently assessed, thus avoiding the loss of information descending from a multi-variate feature extraction procedure. To identify such signals, all data points $\mathcal{X}$ are considered. Raw signals to be processed by a univariate approach are identified by applying the following conditions:

  1. a cycle $\mathcal{X}$ is tagged as "anomalous" if it holds

  $$s(\mathcal{X}) > \tau, \tag{1}$$

  where τ is a pre-defined threshold on the anomaly score (under the assumption that "high" values of the anomaly score indicate a high degree of outlierness). For the problem at hand, we used $\tau = 0.55$ w.r.t. the anomaly score generated by an Isolation Forest;

  2. if, for an anomalous cycle, $\mathcal{X}_a$, we detect that the features obtained from a single time-series X_i are more important than all the others in explaining the anomaly, then the corresponding time-series X_i is sent to the univariate monitoring block. Formally X_i is selected

  $$\begin{aligned} \text{if there exists} \quad & f_j(\mathcal{X}) > \delta f_k(\mathcal{X}), \\ \text{where} \quad & j, k \in \{1, \ldots, p\}, \\ \text{and} \quad & x_j \in \mathcal{S}_{X_i}, x_k \notin \mathcal{S}_{X_i}, \end{aligned} \tag{2}$$

  where $\delta > 1$ is a pre-defined quantity (for our results, we have set $\delta = 2$);

  3. if, besides the two previous steps, some expert knowledge indicates that some raw signals are particularly relevant by themselves, then, they should be monitored univariately as well.

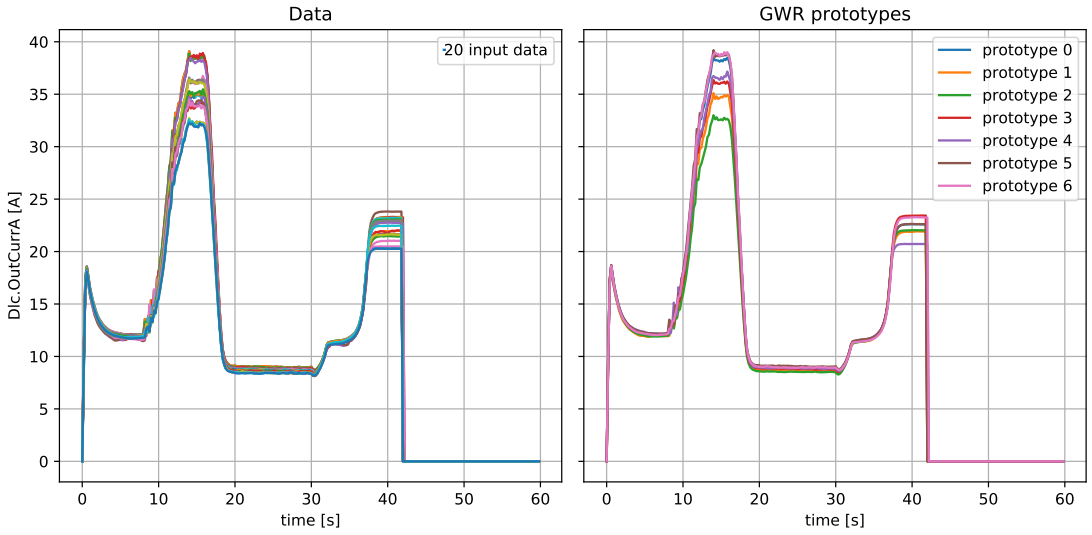

Figure 4: The left part of the figure shows the values of an engine's current consumption in a particular session. These values have been used to create the reference dictionary depicted in the right part of the figure, using GWR.

The user is finally provided with complementary information coming from the two module types, with anomaly scores, system monitoring indications, ranking of variables and concept drift estimations, allowing for a guided Root Cause Analysis.

6. Prototype

We remark that features are not only related to physical signals acquired from the machine, but also to parameters that characterize its working condition and the surrounding environment, like the machine load (number of customers carried), the lubrication, the ambient temperature and the presence of rain. To exemplify, Fig. 4 shows a particular electric current consumption during a machine cycle: this signal was processed by extracting features related to the value of the first current peak, the rising time (to rise to 90% of the peak's amplitude), the maximum current value, a standard deviation in its neighborhood, and the value of the last peak before the signal drops to zero. Once all the features were computed, a correlation analysis was performed to remove redundancy, obtaining vectors composed of $p = 23$ independent features ready to be inputted into the Isolation Forest. For the *concept drift* estimation, we focused on one specific signal, namely, the current consumption of a particular engine, since it highly correlates with the lubrication that is performed periodically on the machine. We used, as a reference, measurements collected in a day when the gears were lubricated and during which the machine was tested in all its possible load configurations (i.e., recalling that, as shown in Fig. 1, four seats are available, the possible configurations correspond to 1, 2, 3 or 4 occupied seats). The left part of Fig. 4 illustrates these data. Some of these configurations generate traces that are clearly identifiable also by visual inspection, as a higher number of occupied seats implies more

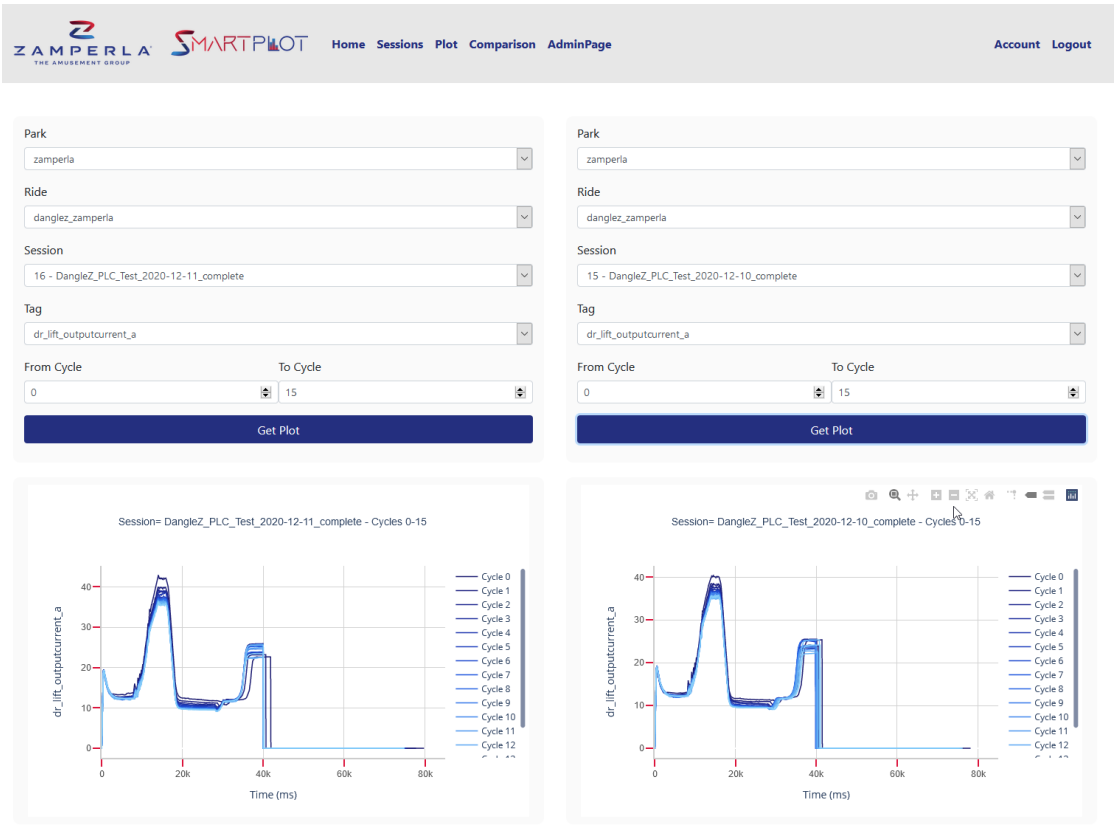

Figure 5: Example of the Web application functionalities.

weight and, therefore, a higher current consumption. However, the proposed GWR approach does this automatically, extracting and then storing the typical patterns (the prototypes) into the dictionary depicted in the right part of Fig. 4.

Data acquired from the sensors are temporarily stored into the file system where there is a folder for each Ride containing metadata about the Ride and a set of csv files corresponding to each Session for data Ride. These files are parsed by using Python scripts and stored in a relational database, namely PostgreSQL[2]. Then, a data access layer, written in Python, provides API to query the data to both the anomaly detection algorithms and to the Web application. The server-side of the Web application is implemented by using the Django REST framework[3]. On the client-side, we rely on AJAX calls implemented by using jQuery[4] and the PLOTLY[5] libraries for delivering interactive plots.

Figure 5 shows an example screenshot of the Web application we have developed. It allows designers to explore, visualize, and analyze the acquired data, according to the approaches

[2]https://www.postgresql.org/
[3]https://www.django-rest-framework.org/
[4]https://jquery.com/
[5]https://plotly.com/

described above. In particular, Figure 5 shows the side-by-side comparison of two different sessions of acquisitions and the related analysis.

7. Conclusions and Future Work

In this work, an AD system to monitor amusement rides has been presented. The system combines a database management system and a multi-faceted AD engine performing univariate and multi-variate analyses, exposed through a Web application. The approach is unsupervised, making it appealing for scenarios where tagged information is unavailable or unreliable. Also, features are ranked according to their importance in explaining an alarm, using an explainable artificial intelligence method.

Future work may concern different areas: *improvements* of the current framework by (i) using other norms (e.g., $|\cdot|_\infty$ or DTW [7]) to account for different types of anomalies, and (ii) adopting semi-supervised learning techniques, including tagged data from either manual labeling or new sensors; *extensions* of the system beyond AD, using it in the ride design phase to maximize its efficiency and the service life of wear components.

Acknowledgments

This work has been supported by the Regione Veneto POR FESR 2014-2020. Asse 1. Azione 1.1.4 (*Bando per il sostegno a progetti sviluppati da Aggregazioni di imprese*) initiative for the project "Trasformazione digitale innovativa nell'industria dell'entertainment" and by MIUR (Italian Minister for Education) under the initiative "Departments of Excellence" (Law 232/2016).

References

[1] L. Stojanovic, M. Dinic, N. Stojanovic, A. Stojadinovic, Big-data-driven anomaly detection in industry (4.0): An approach and a case study, in: 2016 IEEE International Conference on Big Data (Big Data), IEEE, 2016, pp. 1647–1652.

[2] G. A. Susto, et al., An adaptive machine learning decision system for flexible predictive maintenance, in: 2014 IEEE International Conference on Automation Science and Engineering (CASE), 2014, pp. 806–811.

[3] H. Wang, M. J. Bah, M. Hammad, Progress in outlier detection techniques: A survey, IEEE Access 7 (2019) 107964–108000.

[4] Y. Zhao, Z. Nasrullah, Z. Li, Pyod: A python toolbox for scalable outlier detection, Journal of Machine Learning Research 20 (2019) 1–7.

[5] M. Carletti, C. Masiero, A. Beghi, G. A. Susto, A deep learning approach for anomaly detection with industrial time series data: a refrigerators manufacturing case study, Procedia Manufacturing 38 (2019) 233–240.

[6] A. Tsymbal, The problem of concept drift: definitions and related work, Computer Science Department, Trinity College Dublin 106 (2004) 58.

[7] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE transactions on acoustics, speech, and signal processing 26 (1978) 43–49.