A data-driven approach for multivariate contextualized anomaly detection: industry use case

Nenad Stojanovic
Nissatech
Serbia
Nenad.Stojanovic@nissatech.com

Marko Dinic
Nissatech
Serbia
Marko.Dinic@nissatech.com

Ljiljana Stojanovic
Fraunhofer IOSB
Germany
Ljiljana.Stojanovic@iosb.fraunhofer.de

***Abstract*—Anomaly detection is the process of discovering some anomalous behaviour in the real-time operation of a system. It is a difficult task, since in a general case (multivariate anomaly detection, an anomaly can be related to the behaviour of several parameters which are not necessarily behaving anomalously per se, but their (complex) relation is anomalous (not usual/normal). This implies the need for a very efficient modeling of the normal behavior in order to know what should be treated as anomalous/outlier/unusual. Consequently, classical model-driven approaches, due to their focusing on the selected parameters for creating models, are not able to model the behavior if the whole system. This is why data-driven approaches for anomaly detection are getting ever more important for the industry use cases where hundreds (thousands) of parameter should be taken into account. However, current approaches are usually focused on the univariate anomaly detection (or some variations of it), so without going into observing the entire space of relations (computation very difficult).**

In this paper we present a novel approach for the multivariate anomaly detection that is based on modeling and managing the streams of variations in a multidimensional space. The main advantage of this approach is the possibility to observe the relations between variations of a large set of parameters and create clusters of "normal/usual" variations. In order to ensure scaling, which is one of the most challenging requirements, the approach is based on the usage of the big data technologies for realizing data analytics tasks/calculations.

The approach is realized as a part of D2Lab (Data Diagnostics Laboratory) framework and has been applied in several industrial use cases. In this paper we present a very interesting usage for the anomaly detection in the process of functional testing of home appliances devices (in particular case refrigerators) after manufacturing/assembling process. It has been done for a bug vendor (Whirlpool), who expects huge saving in testing and improved customer satisfaction from this approach.

Keywords: anomaly detetction, big data analytics, scalability quality control process

I. INTRODUCTION *(HEADING 1)*

According to recent analyses[1], the number of smart connected devices will grow beyond 50 billion by 2020 and Internet of Things (IoT) has the potential to represent 11% of the world's economy over the same period. The manufacturing sector is expected to be one of the top adopters of IoT technologies. On the other hand, IoT has been considered as one of disruptive technologies that have the maximum potential for revolutionizing the landscape of manufacturing. While huge progress on making assets 'smarter' and production more efficient have been made during last years, the full potential of using IIoT (Industrial IoT) has not yet been exploited sufficiently. Indeed, manufacturing generates a huge amount of data, but this data is still mainly used in the various types of the analytical processing (e.g. diagnostics, predictions) to improve some KPIs of a process, although the main benefit is related to the possibility to enable a continuous process improvement. One of the most promising advantages of IIoT is the possibility to realize a very efficient real-time monitoring and react in the case of some problems/anomalies detected, which is the main focus of this paper.

Anomalies are abnormal events or patterns that do not conform to expected events or patterns [1]. Identifying anomalies is important in a broad set of disciplines; including, medical diagnosis, insurance and identity fraud, network intrusion, and programming defects. Anomalies are generally categorized into three types: point, or content anomalies; context anomalies, and collective anomalies [2]. Point anomalies occur for data points that are considered abnormal when viewed against the whole dataset. Context anomalies are data points that are considered abnormal when viewed against meta-information associated with the data points. Finally, collective anomalies are data points which are considered anomalies when viewed with other data points, against the rest of the dataset.

Anomaly detection is the process of discovering some anomalous behaviour in the real-time operation of a system. In general case, an anomaly can be related to the behaviour of several parameters (so called multivariate anomaly detection), which are not necessarily behaving anomalously per se, but their (complex) relation is anomalous. This implies the need for a very efficient calculation of the anomalies in the huge datasets, which makes the process of anomaly detection very difficult[2]. On the other hand, univariate anomaly detection is a much simpler methods but

[1] Big Data in Manufacturing: BDA and IoT Can Optimize Production Lines and the Bottom Line— but Much of the Industry Isn't There Yet, Frost & Sullivan, Big Data & Analytics, December 2016

[2] In the theory this situation is related to detecting so called unknown unknowns/anomalies, i.e. not previously known anomalies. Most of the approaches is focused on detecting known unknowns/anomalies

it is difficult to perform root cause analysis of an issue because “it is hard to see the forest for the trees”. It is easier to scale in terms of computation. Less data is needed to learn what is normal because the system looks at each metric/parameter by itself, as opposed to looking at combinations of metrics. Figure 1 illustrates the constraints of a univariate anomaly calculation. Let us consider a device to be monitored that is sensed with two parameters, presented on Figure 1 (in case of a machine these could be *Power* and *Speed*, for example).

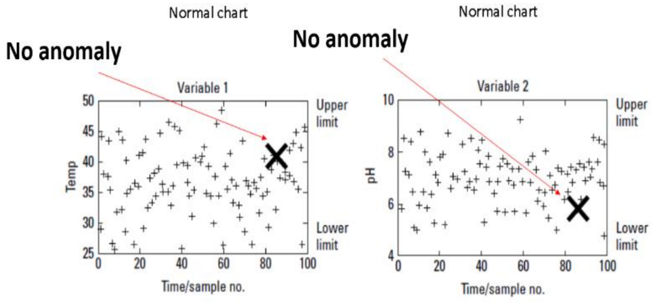

Figure 1. Device with two parameters – normal behavior of both parameters (univariate analysis)

If we are observing values of both parameters at a certain moment, individually, we could conclude that there is no problem in a process, and that the device functions correctly. But this could be misleading, because even though these values are in range of normal values, if we consider both of them at the same time, the image we have about the device can become different. This is presented on Figure 2.

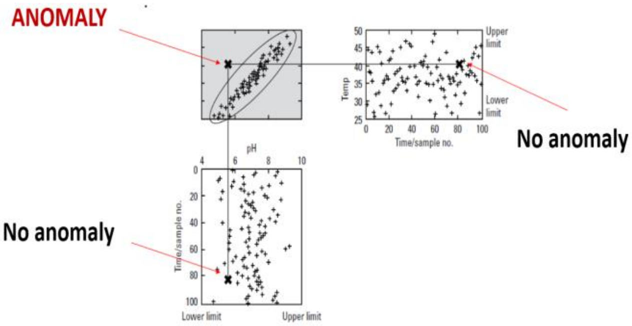

Figure 2. Multivariate analysis enables an efficient anomaly detection process

This shows that it is not enough to consider parameters in isolation (univariate anomaly detection, like to define outliers based on relational class attributes [3]), but rather there is a need to correlate them. By knowing correlations between parameters we can expect to identify anomalous behavior even in cases when values of individual parameters seem normal, but their relationship is not. However, this is one of the most challenging parts in the realization of the multivariate anomaly detection: computational complexity which requires novel and efficient big data driven methods.

Indeed, a prevalent challenge in existing works in anomaly detection for industry is their scalability to large amounts of data, which is usually a case due to a huge expansion of the usage of sensors (IoT in general), as already mentioned. In most cases the algorithms have increased their complexity to overcome more naïve methods, but in doing so have limited their application scope to offline detection. Additionally, where an algorithm may have excelled in its serial elision, it is now necessary to view the algorithm in parallel; using concepts such as divide and conquer, or MapReduce [4]. Many common anomaly detection algorithms such as k-nearest neighbour, single class support vector machines, and outlier-based cluster analysis are designed for single machines [5].

Another approach that can be used for extending univariate anomaly detection is the model-based monitoring [6], driven by the set of predefined (in design time) models, based on some expert knowledge and theories. However, this approach is usually expensive/difficult to be developed and in general case it requires a continual maintenance/evolution. Indeed, due to some changes which might happen in the environment or in the internal structure, an evolution of the model is required (caused by so called model drift). An additional constraint in the model-based anomaly detection is the infeasibility of the models to capture the dependencies between all parameters which influence the behaviour of the system. There are two challenging cases:

- when the behaviour of some parameters cannot be captured in the models created by experts, e.g. due to a large number of parameters (1000+) and no corresponding background knowledge
- the influence of contextual parameters (like air temperature, pollution) which are not part of the models

On the other hand, the industry requires ever more (scalable and affordable) approaches for analysing patterns of complex and heterogeneous data streams and looking for anomalies that can reveal something unexpected (anomalies).

In this paper we present such an (scalable and affordable) approach. The main idea is in handling contextual anomaly detection by excessive computation of the co-relations using all the data concurrently (multidimensional space), using a very efficient big data analytics infrastructure. The main advantage of this approach is the possibility to observe the relations between variations of a large set of parameters and create clusters of “normal/usual” variations. The benefits are high, especially when the context of the data cannot be broken into discrete categories, or when new records cannot easily be placed within one of the given contexts. This approach generally requires a higher computational complexity than a univariate or model-based anomaly detection, as the underlying algebra in calculating multivariate anomalies is computationally expensive.

The approach is realized as a part of D2Lab (Data Diagnostics Laboratory, d2lab.nissatech.com) framework and has been applied in several industrial use cases. In this paper we present a very interesting usage for the anomaly detection in the process of functional testing of home appliances devices (in particular case refrigerators), done after manufacturing/assembling process. The existing approach for anomaly detection is based on the univariate analysis of the measured data from the testing process, which doesn't allow to detect complex anomalies that are a huge challenge for the quality engineers. The approach has been realized for a big vendor (Whirlpool), who expects huge saving in the testing process and an improved customer satisfaction from this approach.

The paper is organized in the following way: in the second section we describe our approach for data-driven multivariate contextualized anomaly detection, whereas Section III presents the results from the use case study. In Section IV we give the details about the implementation and the deployment. In Section VI we present the most important related work. Section V contains concluding remarks.

II. D2LAB APPROACH FOR MULTIVARIATE CONTEXTUAL ANOMALY DETECTION

We use the notion of anomalies as something that is "significantly different than the past" [7] without pretending that an anomaly is a mistake, but rather an outlier. Consequently, the detection process is related to comparing the real-time status of a system with a model of its normal behaviour, whereas the creation of this model is the most challenging task. Indeed, since there are, usually, many parameters influencing a process it is difficult to design/create (by an expert) the model of the normal behavior of the process (which enables model-based anomaly detection), as discussed in the previous section.

Our approach is based on the fact that modern industrial processes, esp. those driven by IIoT, usually generate MB/GB of data. Since this data reflects the real-time behavior of the process, it represents a very valuable source for deriving (model of) normal behavior of the process, which is the crucial task in the anomaly detection. However, this task, which is in the nutshell of D2Lab approach, has two main requirements:

R1: It has to scale properly in order to enable multivariate anomaly detection, even in the cases of thousands of parameters and GB of past data

R2: It has to enable observation of the co-relations between all parameters, incl. contextual ones, altogether

In the following two subsections we provide more details about the realization of these requirements.

A. Scalability

Regarding the first requirement, our approach relies on the big data technologies. In the nutshell of the approach is a novel highly scalable clustering based method, KmedoidsUsingFAMES described in [9, 10], which represents a combination of several algorithms – K-medoids [11], FAMES [12], K-means|| (K-means parallel) [13] and uses DTW [14, 15] as a distance measure. The main advantages of clustering based anomaly detection methods are that they are a) unsupervised (which means affordable since the inclusion of domain experts is limited) and b) scalable (R1). K-medoids is a clustering algorithm which represents the basis of our solution as it is very flexible about the data it clusters and allows to use any distance measure. FAMES (Fast MEdoid Selection) represents a special technique of medoid selection which is very fast, while for the initialization we use a modification of K-means|| algorithm, which allows selection of good initial medoids, improving the quality of final clusters and reducing the number of iterations needed, at the same time. DTW is used as a distance measure, as it enables to compare sequences of different length by finding the optimal alignment of the two.

The approach is implemented in D2Lab framework, which also allows that a sequence of transformations is performed prior to clustering (e.g. filtering, windowing, padding, standardization), to create most adequate features (see Section IV). This solution is implemented as a sequence of MapReduce jobs [16], making a very scalable, distributed, parallel solution. The algorithm runs on a YARN cluster, with a number of machines which belong to the *commodity hardware* group.

B. Contextualization

The realization of this requirement is the main contribution of this paper. We emphasize that the role of contextualization is critical for the IIoT systems due to the possibility to generate a false positive anomaly when context such as the time of day, time of year, or type of location is missing. For example, hydro sensor readings in the winter may fluctuate outside the acceptable anomaly identification range, but this could be due to varying external temperatures influencing how a building manages their heating and ventilation. The problem is that there are many contextual parameters and their influence on the behavior of the system (that can be depending on thousands of parameters) should be checked in an efficient and scalable (systematic) way. This is the reason why we have developed a separate set of methods for dealing with contextualization, described in the following text.

We make a distinction between contextual and behavioral parameters [8]. Contextual parameters define the context of an instance. Behavioral parameters represent non-contextual instance characteristics. For example, if we are talking about a location-based use case, where certain parameters are measured at a certain location, coordinates of that location would make the contextual parameters, while the parameters measured at the point would be behavioral parameters. As already illustrated, it is crucial to include the contextual parameters into analysis, as an instance might be anomalous in one context and normal in another one.

Form of contextual parameters might be different than behavioral ones. For that reason a method to include them into analysis (with the rest of parameters) is needed. Through concrete industry use cases we were able to identify the following types of contextual parameters, used as the basis for realizing our approach (and ensuring the advantages):

- *grouping contextual parameters* – contextual parameters which define a group, for example – id of a location where measurements were performed
- *single value contextual parameters* – contextual parameters which have a single value during the whole period of observation (parameters measurement), for example – temperature measured once during whole period of observation
- *time series contextual parameters* – parameters which are measured for the whole time during period of observation, for example – temperature measured at a certain frequency

The first case "grouping contextual parameters" is easiest to handle, as there is a limited number of groups we can perform separate analysis for each of the groups.

We treat "time series contextual parameters" in the same way as behavioral parameters which are time series by nature.

The main challenge is the "single value contextual parameters" group, which requires special treatment to be included in our clustering algorithm, as the algorithm works with time series parameters. In order to use this data for the training of a machine learning algorithm we develop a novel technique, called context boosting. Briefly, it 1) creates time series parameter out of single value parameter by repeating the value *N* times (where *N* corresponds to the length of other time series parameters) and 2) performs *boosting* of values, to make contextual parameters equally important as behavioral ones (as in most cases number of contextual parameters is much smaller than the number of behavioral ones). One of the main advantages is the scalability of this approach since it relies on the D2Lab computation model, as mentioned previously.

The importance of contextual parameters in the analysis and the results of approach are further explained through the use case.

III. Use case

A. Introduction

In order to illustrate the approach, in this section we present the details from the industry use case, which has been performed for Whirlpool in the domain of the ZHQ (Zero Hour Quality), a quality management approach that aims in having a valid product (high quality) from the beginning of the usage (zero hour) of a product/device. In order to achieve this, there is a very strict process of functional testing after assembling process that consists of 10+ measurement stations where different parameters of the operation of the assembled devices are checked.

The quality control process is done using a traditional threshold-based process, producing so called out-of-limits OOL alarms. It consists of defining an upper and lower threshold so that when a measurement goes above the upper limit or below the lower one, an alarm is triggered. Then engineers will inspect the parameter that is out of limits and determine whether it is an anomaly or not and decide which action to take (for example, run a procedure).

Therefore, this is a "traditional" univariate (threshold-based) anomaly detection process, which suffers from all the drawbacks we mentioned in the first section. Our task was to develop a novel approach for the anomaly detection in the functional testing process, which should 1) exploit the value of past measurement data for the detection of complex anomalies and 2) include the influence of the contextual parameters, i.e. to develop a multivariate contextualized anomaly detection approach. The approach should be tested on the functional testing of refrigerators.

B. Use case settings and preliminary analysis

We briefly list the most important elements from the Whirlpool use case:

- For each refrigerator that Whirlpool produces, a functional test is performed by attaching sensors and measuring a number of parameters
- Functional tests validate that a refrigerator works as it should and represent one of basic measures of quality control
- For the dataset provided following 6 (behavioral) parameters are measured (see Figure 3 for an example):
  - Temp. Destra (Rossa), Temp. Sinistra (Nera), T EV Cap Fz., T EV Cap. Fr., Cos Fi, Potenza
- As it was previously explained, there was a need to include contextual parameters into analysis as well. Following contextual parameters were provided:
  - Time when the functional test was performed
  - Temperature of the environment in which functional test was performed
  - Location (part of the factory) where the functional test was performed
- Data is delivered as a database containing compressed content (decompression was first needed)
- The dataset was too large to be processed on a single machine (measurements for 44165 functional tests) and that is why we needed a scalable, distributed solution which is able to bare with Big data.
- A certain number of instances contained missing values (missing parameters, to be more exact) and needed special treatment

Length of time series parameters differs for a number of tests (see Figure 4 for an example of the histograms for one parameter). However, this issue can be easily resolved using DTW as a distance measure and padding as a preprocessing technique, as we designed in D2Lab. DTW allows us to compare functional tests of different length, while we use padding to get all parameters of a single functional test to the same length.

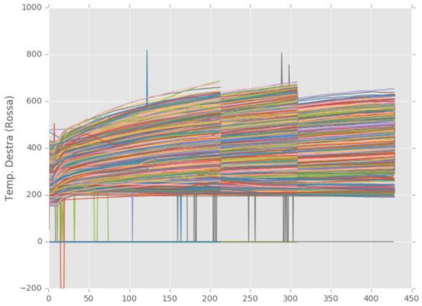

Figure 3. Temp. Destra (Rossa)

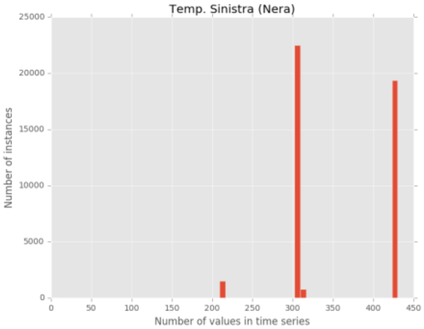

Figure 4. Length of Temp. Sinistra parameter for instances in the dataset

In order to get a better overview of the distribution of data we performed PCA (Principal Component Analysis) to be able to visualize it in 3D (cf. Figure 5). Each point on the plot represents a functional test determined by its six parameters. We can notice one large group and a certain number of points deviating from it. It means that the functional tests are performed in an unified way, but there are (not a few) cases which appear like outliers.

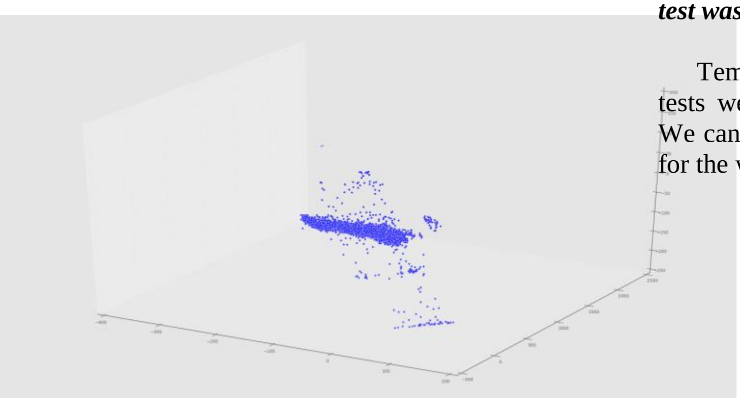

Figure 5. Scatter plot (3D) of dataset instances after performing PCA

C. Contextualized anomaly detection

The main goal of this paper is to explain the influence of contextual parameters, based on the approach described in previous section. We have examined the influence of contextual parameters before including them into further analysis. The results for each contextual parameter are the following:

- Time when the functional test was performed

Influence of time of the year when the functional test was performed is given on Figure 6. We have tried different number of groups and at the decided to split the dataset into four groups:

Light blue – Summer,
Dark blue – Spring
Red – Winter,
Yellow – Autumn

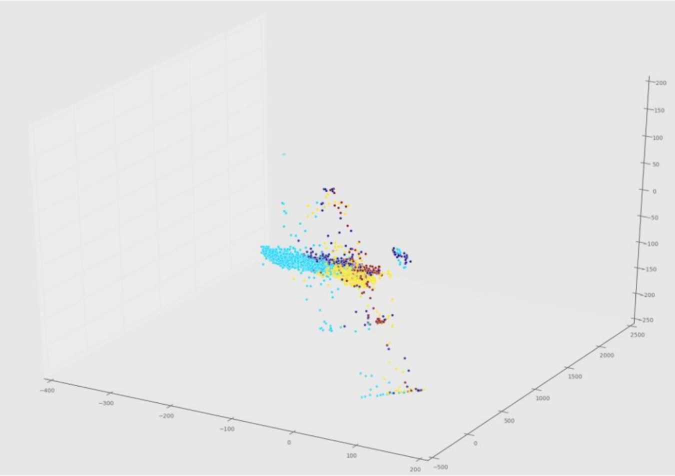

Figure 6. Influence of time when the functional tests were performed

We have concluded that time of testing has some influence on refrigerator behavior during functional tests, but only to a certain measure, as these periods are not separated clearly and there is some amount of overlapping and mixing of functional tests from different periods.

- Temperature of the environment in which functional test was performed

Temperatures of the environment in which functional tests were performed were given as single measurements. We can observe how this temperature changed during time, for the whole dataset, on Figure 7.

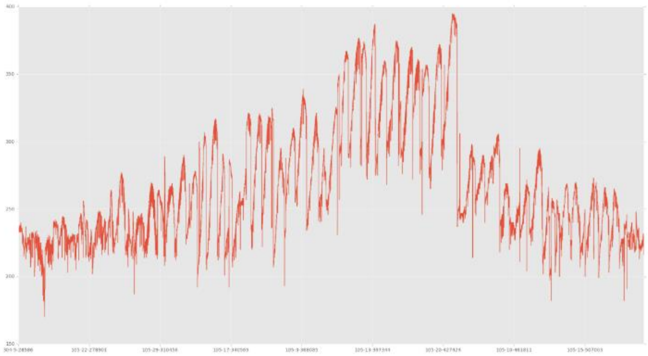

Figure 7. Environment temperature change during time

Then we observe the temperature influence on Figure 8. In this case our analysis showed that it is best to split the temperature into 5 ranges.

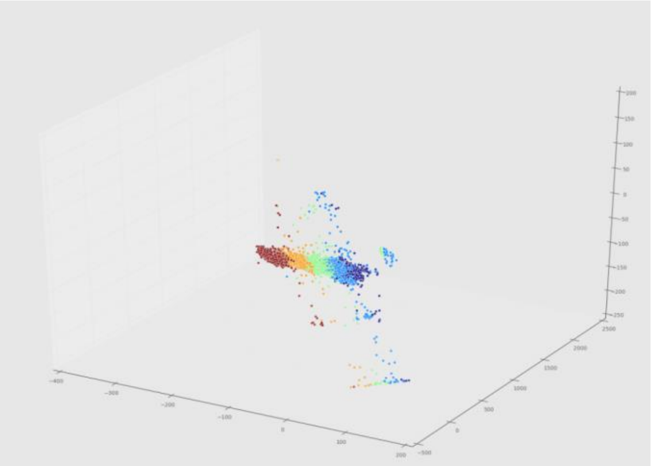

Figure 8. Environment temperature influence onto functional tests

We can see that functional tests are very well grouped if we consider the temperature at which they were performed. This indicates that this parameter may have great influence, as it will show later.

- Location (part of the factory) where the functional test was performed

It was interesting to see whether the location where the functional tests were performed has any influence on the results of the functional tests. The results are given on Figure 13. Two locations are colored in red in blue.

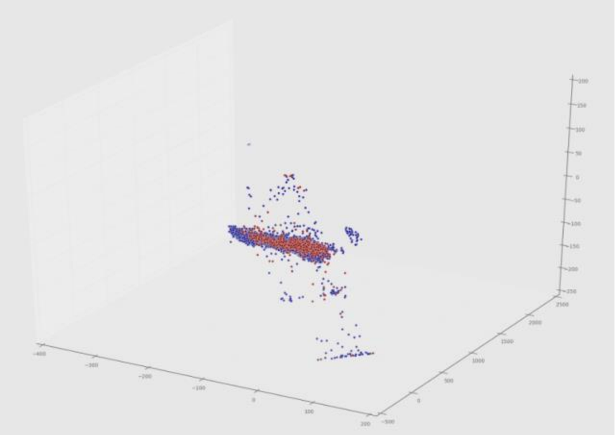

Figure 9. Location influence onto functional tests

We have concluded that the location where the functional tests were performed had no influence onto functional tests result.

D. Interpretation of results

In communication with domain experts we were able to validate our conclusion about the influence of environment temperature on results of functional tests. Domain experts stated that they were not aware of this influence and that they plan to use this information for future tests as it is of great importance. Namely, tests are performed for a refrigerator, a device whose main and only purpose is to cool, hence a lot of measured parameters are actually different temperatures measured inside the refrigerator (Temp. Destra (Rossa), Temp. Sinistra (Nera), T EV Cap. Fz. and T EV Cap. Fr.). The temperature inside the refrigerator is correlated to the temperature of the environment, hence, the temperature of the environment needs to be taken into consideration when performing functional tests. We have done another experiment to prove this. We took two functional tests:

- functional test of refrigerator performed in environment with low temperature
- functional test of refrigerator performed in environment with high temperature

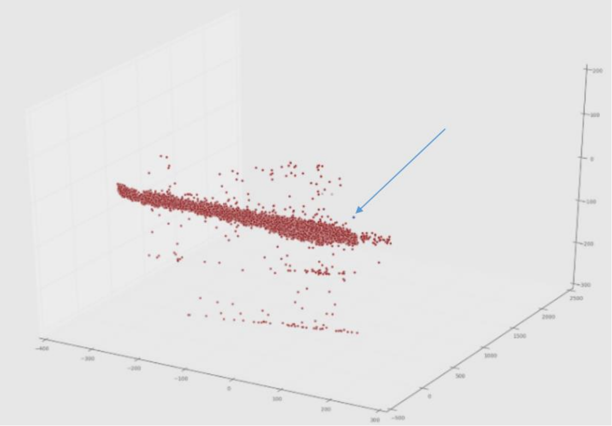

Figure 10. Testing the data-driven model: successful detection of an artificial (anomalous) instance

We took environment temperature (contextual parameter) of one refrigerator and behavioral parameters of the other refrigerator and combined them as they come from a single functional test. Then we run PCA on this artificial test like on all the others and visualized it (Figure 10). We can see that such a functional test represents an outlier, being far from the central group.

After concluding that out of the three contextual parameters environment temperature is to be included, we performed D2Lab clustering (see Section 2) to detect existing anomalies and to generate a model which can be used in real time. Clustering results are presented in Figure 11. The number of clusters is six, which is at the same time the number of medoids included in the model. Number of clusters was determined using combination of *Elbow* and *Silhouette* methods [17, 18].

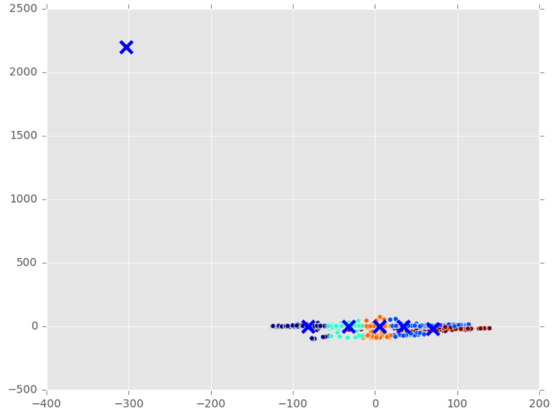

Figure 11. D2Lab clustering results

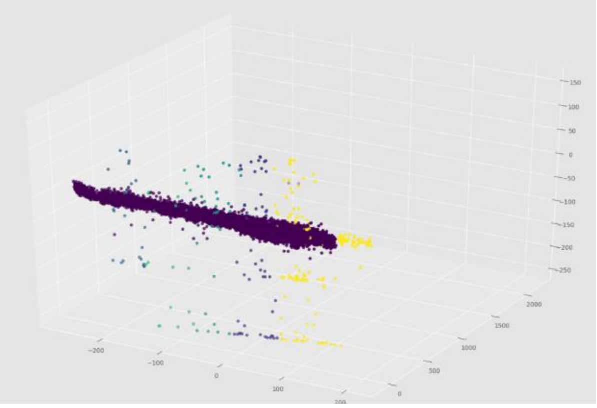

Figure 12. Detected anomalies

The smallest cluster containing only two points represents a cluster of anomalies, while other five clusters contain mostly normal functional tests with a smaller number of anomalies on the boundary of those clusters. Identified anomalies are marked on Figure 12.

IV. Implementation and Deployment

In this section we give the most important details about the implementation and deployment of the presented approach in the Whirlpool environment.

A. *Technical architecture*

Core D2Lab components modified for the purpose of specific, Whirlpool use case, are given on Figure 13.

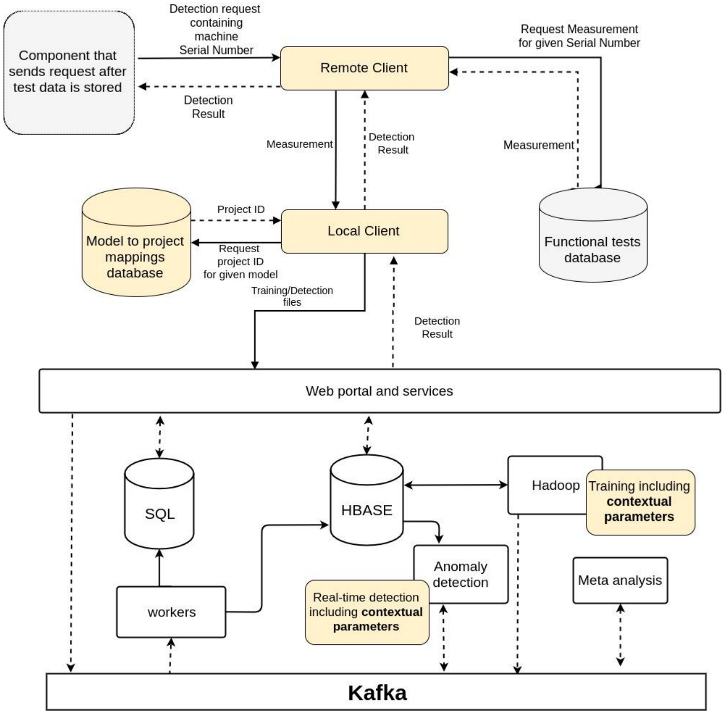
Component that sends request after test data is stored
Detection request containing machine Serial Number
Detection Result
Remote Client
Request Measurement for given Serial Number
Measurement
Detection Result
Measurement
Model to project mappings database
Project ID
Request project ID for given model
Local Client
Functional tests database
Training/Detection files
Detection Result
Web portal and services
SQL
HBASE
Hadoop
Training including **contextual parameters**
Anomaly detection
Meta analysis
workers
Real-time detection including **contextual parameters**
Kafka

Figure 13. D2Lab architecture – Whirlpool instance

D2Lab represents a highly scalable system, based on Big data technologies such as Hadoop, HBase and Kafka. In this case all components are deployed inside Whirlpool facility (private cloud) with no need for external communication. Core features of the system, specific to this use case are:

1. *Remote Client*

Represents the entry point for anomaly detection. When a functional test is finished the remote client receives a request to analyze the functional test saved in the functional test database. It reads the measurements for the database, transforms the data according to specified format and contacts the local client. After it receives answer from the local client it returns the result to the component which initiated the procedure.

2. *Local Client*

Local client has access to a database which contains mapping between different types (models) of devices. It determines the model of the machine tested by the functional test and contacts D2Lab service, passing the functional test and information about the model.

3. *Model to project mappings database*

Contains mappings for different kinds of devices (machines).

4. Training procedure

Training procedure had to be extended to support analysis which includes contextual parameters. Specific steps were needed to determine the influence of contextual parameters and to emphasize it. This steps have already been described in the previous sections.

5. Real time anomaly detection

Influence of contextual parameters also had to be included in real time anomaly detection. Similar steps need to be performed during both phases, to have an identical workflow for a functional test, regardless of whether it comes for training, or in real time, for anomaly detection.

B. Hardware requirements

In this part we specify minimal hardware requirements for D2Lab deployment. D2Lab requires a cluster of machines to exist, for Hadoop, HBase and Kafka components. Additionally, there is a need for at least another machine, central server, on which all the other components will be deployed. Hardware requirements for the central server and cluster components are as follows:

- central server: 12GB RAM, 8 cores, 500GB HDD
- cluster master 12GB RAM, 8 cores, 300GB HDD
- 3 slaves 12GB RAM, 8 cores, 400GB HDD

C. Processing pipeline

D2Lab processing pipeline is depicted on Figure 14. These are only the phases of analysis and for each phase there is a certain number of possibilities which can be used depending on the use case and concrete data. For example, in case of Whirlpool refrigerator use case data was stored in a relational database, from which it needed to be extracted using SQL queries, to be imported into D2Lab system for training. We performed exploratory analysis to find out about the influence of contextual parameters and to conclude that we will need a special learning method as the instances in dataset may differ in length. A lot of instances had missing values, which needed to be dealt with. Data was transformed using standardization method. There was no need to create windows as the data was not in a streaming form. PCA was selected as dimensionality reduction method, while Fast Fourier Transform (FFT) and statistical features were also considered. Similarly, Elbow method gave better results when trying to determine the optimal number of clusters than the Silhouette method. The whole procedure was unsupervised due to lack of labels, so we used our scalable clustering algorithm KmedoidsUsingFAMES during training. We inspected the clusters, cluster representatives and anomalies found during training. We compared clusters and anomalies to try to identify the root cause of anomalies. Along the way we produced a lot of 2D/3D plots (scatter plots, silhouette plots, box plots, histograms), and at the end produced a report called “Process Data Atlas”. Model produced during training (containing cluster representatives) was then used in real-time to detect new anomalies between functional tests, as they are performed.

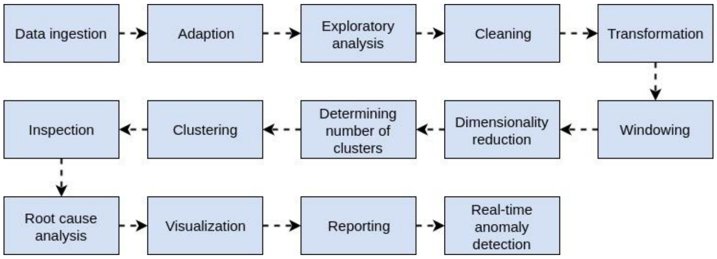

Figure 14. D2Lab processing pipeline

V. RELATED WORK

Little work has been performed in providing context-aware anomaly detection algorithms.

Srivastava and Srivastava [17], proposed an approach to bias anomaly detectors using functional and contextual constraints. Their work provides meaningful anomalies in the same way as a post-processing algorithm would, however, their approach requires an expensive dimensionality reduction step to flatten the semantically relevant data with the content data.

Mahapatra et al. [18] propose a contextual anomaly detection framework for use in text data. Their work focuses on exploiting the semantic nature and relationships of words, with case studies specifically addressing tags and topic keywords. They had some promising results, including a reduction in the number of false positives identified without using contextual information. Their approach was able to use well-defined semantic similarity algorithms specifically for identifying relationships between words.

A different approach for contextual detection is that work of AlEroud et al. [19], who apply contextual anomaly detection to uncover zero-day cyber attacks. Their work involves two distinct steps, similar to the modules described in this paper: contextual misuse module, and an anomaly detection technique. There are other minor modules, such as data pre-processing, and profile sampling. The first major component, contextual misuse, utilizes a conditional entropy-based technique to identify those records that are relevant to specific, useful, contexts. The second component, anomaly detection, uses a 1-nearest neighbour approach to identify anomalies based on some distance measure. This component is evaluated over the records individually to determine whether connections between records indicate anomalous values. This work is similar to the work presented in this paper in that the detection is composed of two distinct modules. However, the content component of their work involves calculating difficult distance measures that are not always easily definable. For example, when faced with many features that each have different data types or domains, it is difficult to calculate suitable distance metrics as finding a common method to aggregate the features is also difficult. Another drawback is that each module is normally evaluated for all new incoming values. While the authors do say that the first component aims to reduce the dimensionality required for the second component, they go on to mention that both the contextual component and anomaly detection component are calculated individually to evaluate the anomaly detection prowess of the approach.

Miller et al. [20] discuss anomaly detection in the domain of attributed graphs. Their work allows for contextual data to be included within a graph structure. One interesting result is that considering additional metadata forced the algorithm to explore parts of the graph that were previously less emphasized. A drawback of this work is that their full algorithm is difficult for use in real-time analytics. To compensate, they provide an estimation of their algorithm for use in real-time analytics, however the estimation is not explored in detail and so it is difficult to determine its usefulness in the real-time detection domain.

Other work has been done in computationally more expensive algorithms, such as support vector machines (SVMs) and neural networks. In general, these algorithms require a large amount of training time, and little testing time. In most cases this is acceptable as models can be trained in an offline manner, and then evaluated in real-time. One disadvantage to using these classification-based algorithms is that many require accurate labels for normal classes within the training data. This is difficult in scenarios such as environmental sensor networks where there is little to no labelling for each sensor value. Shilton et al. [21] propose a SVM approach to multiclass classification and anomaly detection in wireless sensor networks. Their work requires data to have known classes to be classified into, and then those data points which cannot be classified are considered anomalous. One issue that the authors present is the difficulty in setting one of the algorithm's parameters.

To reduce the effect of the computational complexity of these algorithms, Lee et al. [22] have proposed work to detect anomalies by leveraging Hadoop. Hadoop is an open-source software framework that supports applications to run on distributed machines. Their work is preliminary in nature and mostly addresses concerns and discussion related to anomaly detection in Big Data.

Another online anomaly detection algorithm has been proposed by Xie et al. [23]. Their work uses a histogram-based approach to detect anomalies within hierarchical wireless sensor networks. A drawback to their approach is their lack of consideration for multivariate data. That is, their work focuses strictly on developing histograms for the data content but not the context of the data.

VI. CONCLUSION

In this paper we presented a method for data driven quality control using highly scalable clustering based anomaly detection. We introduced the notion of contextual and behavioral parameters and extended existing methods to include contextual parameters during analysis to get more accurate results.

In this paper we presented the results from a large case study that has been done for Whirlpool in the domain of the ZHQ (Zero Hour Quality), a quality management approach that aims in having a valid product (high quality) from the beginning of the usage (zero hour) of a product/device. We tested this approach in a case related the anomaly detection in the functional testing use case, proving the value of our approach to the domain experts by finding causalities that they were not aware of. As a result of this research a model of normal (*usual*) behavior was created, which will be used in real time to detect anomalous refrigerators, to reduce scrap and improve the quality of the products.

We argue that this approach can be very useful for an efficient and affordable anomaly detection in a wide set of cases related to IIoT. The main advantage is that the approach offer a scalable data-driven development of complex models in order to enable real-time multiparametar monitoring of processes. This will enable a transformation of manufacturing systems from reactive into proactive, i.e. they will be able not only to react on a problem at hand, but rather to sense the problem (in advance) and proactively resolve the situation/anomaly (ahead of time).

REFERENCES

[1] Chandola V, Banerjee A, Kumar V, Anomaly detection**:** a survey**.** *ACM Comput Surv* 2009,**41**(3):1–58.

[2] Michael A Hayes and Miriam AM Capretz, Contextual anomaly detection framework for big sensor data, Journal of Big Data 2015 2.2, Hayes and Capretz; licensee Springer. 2015

[3] Kou Y, Lu C-T, Spatial weighted outlier detection In: Proceedings of SIAM Conference on Data Mining.. SIAM, 2006

[4] Dean J, Ghemawat S: MapReduce: Simplified data processing on large clusters. Commun ACM 2008, 51(1):107–113

[5] Rajasegarar S, Leckie C, Palaniswami M: Anomaly detection in wireless sensor networks. Wireless Commun IEEE 2008,15(4):34–40.

[6] Donald L. Simon, Aidan W. Rinehart, A Model-Based Anomaly Detection Approach for Analyzing Streaming Aircraft Engine Measurement Data, NASA/TM—2015-218454

[7] David Evans, Jose Martinez and Moritz Korte-Stapff Data Mining to Drastically Improve Spacecraft Telemetry Checking: A Scientist's Approach, SpaceOps Conferences, 16-20 May 2016, Daejeon, Korea

[8] C.C. Aggarwal, Outlier Analysis, Springer, New York, Heidelberg, Dordrecht, London, 2013.

[9] Stojanovic Nenad, Dinic Marko, Stojanovic Ljiljana. (2015). Big data process analytics for continuous process improvement in manufacturing. 1398-1407. 10.1109/BigData.2015.7363900.

[10] Stojanovic Ljiljana, Dinic Marko, Stojanovic Nenad, Stojadinovic Aleksandar. (2016). Big-data-driven anomaly detection in industry (4.0): An approach and a case study. 1647-1652. 10.1109/BigData.2016.7840777.

[11] Kaufman, L. and Rousseeuw, P.J. (1987), Clustering by means of Medoids, in Statistical Data Analysis Based on the L1 -Norm and Related Methods, edited by Y. Dodge, North- Holland, 405– 416

[12] Adriano Arantes Paterlin, Mario A. Nascimento, Caetano Traina Junior, Using Pivots to Speed-Up k-Medoids Clustering, Journal of Information and Data Management, Vol. 2, No. 2, June 2011, Pages 221–236.

[13] D. Arthur, S. Vassilvitskii, K-means++: The Advantages of Careful Seeding, SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, January 7-9, New Orleans, Louisiana, 2007

[14] S. Salvador, P. Chan, FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space, https://gi.cebitec.uni-bielefeld.de/teaching/2007summer/jclub/papers/Salvador2004.pdf

[15] C.A. Ratanamahatana, E. Keogh, Everything you know about Dynamic Time Warping is Wrong, http://wearables.cc.gatech.edu/paper_of_week/DTW_myths.pdf.

[16] J. Dean, S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, *Google, Inc*, 2004

[17] Srivastava N, Srivastava J (2010) A hybrid-logic approach towards fault detection in complex cyber-physical systems. In: Prognostics and Health Management Society, 2010 Annual Conference of The. IEEE. pp 13–24

[18] Mahapatra A, Srivastava N, Srivastava J (2012) Contextual anomaly detection in text data. Algorithms 5(4):469–489

[19] AlEroud A, Karabatis G (2012) A contextual anomaly detection approach to discover zero-day attacks. In: Cyber Security, 2012 International Conference On. IEEE. pp 40–45

[20] Miller BA, Arcolano N, Bliss NT (2013) Efficient anomaly detection in dynamic, attributed graphs: Emerging phenomena and big data. In: Intelligence and Security Informatics (ISI), 2013 IEEE International Conference On. IEEE. pp 179–184

[21] Shilton A, Rajasegarar S, Palaniswami M (2013) Combined multiclass classification and anomaly detection for large-scale wireless sensor networks. In: Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference On. IEEE. pp 491–496

[22] Lee JR, Ye S-K, Jeong H-DJ (2013) Detecting anomaly teletraffic using stochastic self-similarity based on Hadoop. In: Network-Based Information Systems (NBiS), 2013 16th International Conference On. IEEE. pp 282–287

[23] Xie M, Hu J, Tian B (2012) Histogram-based online anomaly detection in hierarchical wireless sensor networks. In: Trust, Security and Privacy in Computing and Communications, 2012 IEEE 11th International Conference On. IEEE. pp 751–759