Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 217 (2023) 1127–1136

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Industry 4.0 and Smart Manufacturing

Variables influence analysis of gas leak testing using belief propagation over factor graphs

Joana Martins[a,*], Diogo Costa[b], Eugénio M. Rocha[c,d]

[a]Department of Physics, University of Aveiro, 3810-193, Portugal
[b]Department of Mechanical Engineering, University of Aveiro, 3810-193, Portugal
[c]Department of Mathematics, University of Aveiro, 3810-193, Portugal
[d]Center for Research and Development in Mathematics and Applications (CIDMA), Aveiro, 3810-193, Portugal

Abstract

Leak testing provides nondestructive quality measurements and is a vital stage in the manufacturing of components that require leak-tight properties. Yet, leak tests are notorious for their sensitivity towards external and environmental factors, hampering accuracy and reliability of test results, leading to a more costly and less efficient production cycle. The classical approach to this issue is through the use of mathematical or physical models of leakage behaviour, which are of difficult creation and low generalisation. Yet, resulting calibrations made to compensate for small deviations in testing conditions seldom consider each test-jig's unique set-up, and may still lead to unsatisfactory results. Alternatively, data-driven methods, such as the novel approach presented in this work, allow for failure analysis based solely on historical data. To achieve that, we employ a set of Variable Influence Analysis (VIA) models based on the application of the Belief Propagation algorithm to factor graphs where different events are related through correlation relationships. What differentiates our approach from other data-driven methods, such as the ones based on machine learning or deep learning methods, is the interpretability of the results and more efficient and swift implementation. We then apply VIA models to a real-world use case at Bosch Thermotechnology, centred around a differential pressure leak tester where testing frequently resulted in false rejections. Our approach is able to formally determine in a data-driven manner that, unlike initial suspicion, environmental factors show negligible impact on false rejections, and issues likely stem from equipment fault.

Peer-review under responsibility of the scientific committee of the 4th International Conference on Industry 4.0 and Smart Manufacturing

Keywords: Belief propagation; factor graphs; gas leak testing; industry 4.0.

1. Introduction

Information systems are key components of any organisation, providing a basis for many of management decisions, capable of serving not only within the enterprise, but also to outside users and clients. As such, the good performance

* Corresponding author
E-mail address: joanadirce@ua.pt

1877-0509
Peer-review under responsibility of the scientific committee of the 4th International Conference on Industry 4.0 and Smart Manufacturing
10.1016/j.procs.2022.12.311

of information systems has become one of the major focus areas of senior managers [1]. Recently, to reduce wasting both time and resources, as well as to increase overall efficiency, senior managers from various organisations gave particular attention to the continuous improvement of the quality of components and organisational systems. To enhance productivity, increase user satisfaction, and agility of components, it is necessary to consider quality aspects during all different stages of information management, from designing databases to framework execution and data entries [1]. This is of particular importance to pivotal stages of the production cycle, which can further exacerbate the occurrence and effect of bottlenecks. One such example is leak testing procedures, which represent a mandatory step in the production of parts or components that require leak-tight behaviour, as is the case of containers for pressurised fluids (e.g., boilers or tanks). Leak testing enables identification and subsequent quantification of "leaks", defined as an excessive material flow above specified limits, caused by factors such as open flow paths, pinholes, broken or tampered seals or from material porosity [2].

Yet, in spite of the prominence of these processes, advances in the area seldom occur and the uncertainty and reliability of measured values is often disregarded [3]. This stems from the high sensitivity demonstrated by these methods towards small variations in environmental conditions such as temperature or ambient humidity, or other external factors such as a high dependence on operator skill [4]. Combined, these factors pose a major hindrance has they further difficult the correct machine calibration, which in turn will cause an increase in the number of falsely rejected tests. This brings additional costs to manufacturers as failed parts are repeatedly re-tested enhancing the bottlenecking effect. The classical approach to the uncertainty problem has been via the high-effort creation of mathematical or physical models, tailored to describe specific applications [2, 4, 5], yet due to the specialist nature of these solutions, generalisation is poor and small alterations to external elements or humane factors might invalidate their usage. Alternatively, a few examples of leak testing equipment are capable of automated calibration to compensate for small variations in atmospheric conditions [6]. Still, these methods, including the ones provided by manufacturers, do not consider each test-jig's unique set-up, which can result in unsatisfactory factory calibrations.

Within the scope of Industry 4.0, data-driven methods emerge as a valid alternative doing away with the need for complex models and relying instead on copious quantities of quality historical data, enriched through expert knowledge [7]. This allows for the determination of normal and abnormal working conditions, after which we can apply statistics based algorithms and techniques for future event prediction and failure mode analysis [8]. Yet, these methodologies introduce new non-trivial challenges particularly in real-world use cases, as industrial data is frequently of sub-optimum quality. Furthermore, cornerstones of data-driven methods, such as machine learning or deep learning, are capable of bringing great value to organisations through accurate and reliably predictions, however, deducing from these models behaviour actionable information for organisational improvements is much more difficult, typically acting as "black-boxes" and providing very little in the way of true interpretability [9], specially concerning the causes for the (physical) problem issues besides additionally explaining the model results.

In this paper we introduce a set of novel Variable Influence Analysis (VIA) models that make use of a Belief Propagation (BP) algorithm aimed at solving the long-standing issue of extracting actionable information and creating data-driven decisions from manufacturing data. These algorithms are able to compute the probability of a certain event being related to the cause of an issue. For that, we make use of factor graphs where all possible causes are related among them and with the issue, according to measures of correlation. As a proof of concept, this novel approach was applied to a real-world use case at Bosch Thermotechnology facilities centred around the Differential Pressure Leak detection method, one of the most widespread techniques for leak testing, for its low cost, simplicity and sensitivity [5]. At the shop floor, a large number of testing instances results in false part rejections, leading to substantial bottlenecking of production lines. Previous work in the scope of this use case saw the correct classification of several test failure types [10], however causal relationships for the existence of false rejections are still undetermined, being yet attributed to the influence of an unknown factor, initially believed to be related to environmental conditions such as pressure, temperature, or ambient humidity. Via the application of this method, we are able to formally determine, in a purely data-driven manner, that environmental factors show low impact on false rejections, with issues in this particular use case being, most likely, organisational in nature or due to equipment fault.

2. Belief Propagation Algorithm for Identification of Variable Influence

2.1. Data Representation via Factor Graphs

There are many complex systems that can be described by a factor graph representation, the only requirement is for them to be translated by a global function that factors into a product of simpler "local" functions, each of them dependent on a subset of variables [11]. In this context, a factor graph (as the one represented in figure 1), besides containing N variable nodes (represented by circles), it also contains F 'function nodes' (represented by squares).

The concept of 'function nodes' allows the dependencies between variables to be captured by connecting each set of interacting variables, $\partial a \equiv i_1, \ldots, i_{k_a}$, along edges (i, a), to a same function node a that represents the local function $\psi_a(\underline{x}_{\partial a})$, dependent on the values $\underline{x}_{\partial a} \equiv x_{i_1}, \ldots, x_{i_{k_a}}$ [11]. If a function node a is connected only to one variable node i, the function $\psi_i(x_i)$ is called an 'evidence' [12]. In the same way, each variable node i has a set of neighbouring function nodes, $\partial i \equiv a_1^i, \ldots, a_{k_i}^i$.

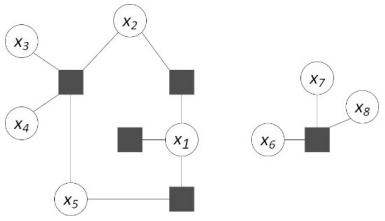

Fig. 1. Abstract example of a factor graph. The degree of a function node (the number of edges incident to it) is always superior to zero. The variable nodes are represented by circles and the function nodes represented by squares [11].

The factor graph is useful in the way that easily allows the application of the belief propagation (BP) algorithm, as a computational inference algorithm that allows the marginals probabilities (single nodes probability distributions) to be computed much more efficiently than with a naive algorithm [11].

2.2. Belief Propagation Algorithm

The Belief Propagation (BP) algorithm is based on 'message passing' through nodes and it was originally proposed by Judea Pearl in 1982 [13], who formulated it as an exact inference algorithm on tree-like graphs [11, 14]. However, on graphs with loops the situation is much less clear [15]. Some researchers have empirically demonstrated good performance of the BP algorithm on graphs with loops when iterating the messages until convergence, although in some cases the algorithm still gives poor results or fails to converge. The intuition behind its relative success lies in the fact that a graph with loops may act 'locally' like a tree, where far apart variables become almost uncorrelated [11]. In this context, the BP algorithms straddles the line between analytic and numerical techniques by iteratively solving a set of self-consistent equations satisfied by the variables of interest [11]. Such a BP approach generally belongs to the NP-hard complexity class.

On a factor graph the BP algorithm acts through two different types of messages: messages sent by variable nodes to its nearest neighbour function nodes $m_{i\to a}$, and messages received by variable nodes from its nearest neighbour function nodes, $\hat{m}_{a\to i}$, always following the existing edges. These messages take values in the space of probability distributions over the single variable space I. Therefore $m_{i\to a} = \{m_{i\to a}(x_i) : x_i \in I\}$, with $m_{i\to a}(x_i) \geq 0$ and $\sum_{x_i} m_{i\to a}(x_i) = 1$. The same happens with the $\hat{m}_{a\to i}$ messages [11]. The BP messages are then computed for each iterative step t until its convergence based on previous conditions, as described in equations 1 and 2 (where the symbol $\cong$ denotes 'equality up to a normalisation' and the symbol '\' the set subtraction) [11, 16].

$$\hat{m}_{a\to i}^{(t)}(x_i) \cong \begin{cases} \sum_{\underline{x}_{\partial a\setminus i}} \psi_a(\underline{x}_{\partial a}) \prod_{j\in\partial a\setminus i} m_{j\to a}^{(t)}\left(x_j\right), if\ |\partial a\setminus i| > 0 \\ \psi_a(x_i), if\ |\partial a\setminus i| = 0 \end{cases} \tag{1}$$

$$m_{i\to a}^{(t+1)}(x_i) \cong \prod_{b\in\partial i\setminus a} \hat{m}_{b\to i}^{(t)}(x_i) \tag{2}$$

The message $\hat{m}_{a\to i}$ results from the sum of all possible states of the neighbouring variable nodes of function node a, except for the variable node i. This sum includes the product of all messages that the function node a receives from those neighbours. However, if i is the only neighbouring variable node of function node a, there are no messages to consider and $\hat{m}_{a\to i}$ is only dependent on the action of the function node a. Meanwhile, the message $m_{i\to a}$ results from the product of all the messages sent from the neighbouring function nodes of variable node i, except for the function node a. Note that we assume that all variable nodes are connected, at least, to one function node.

It is expected that for $t \to \infty$ the messages had already converged to a fixed point. In this situation the message $m^{(\infty)}_{i\to a}(x_i)$ will give the marginal distribution of x_i on a modified graph that does not include the function node a. Analogously, the message $\hat{m}^{(\infty)}_{a\to i}(x_i)$ will coincide with the marginal distribution of x_i in a graph where all function nodes in ∂i, except a, have been erased. After the convergence at an iteration step t_{max}, it can be estimated the marginal distribution $p(x_i)$ of a variable i using all its incoming messages $\hat{m}^{(t_{max})}_{a\to i} = \hat{m}^{*}_{a\to i}$, as in equation 3 (where '*' denotes the BP fixed point):

$$p(x_i) \approx b(x_i) \cong \prod_{a\in\partial i} \hat{m}^{*}_{a\to i}(x_i) \tag{3}$$

Note that a function node with only one neighbouring variable i acts fixing the value of that variable according to the evidence $\psi_i(x_i)$. In this work, the BP algorithm was implemented in Python for $t_{max} = 100$ and the messages were considered to have converged when the difference between the messages $\hat{m}_{a\to i}$ of consecutive iterations was less than 0.0001.

2.3. Variable Influence Analysis (VIA) Models

In this work we introduce a novel modelling procedure based on the formulation of the root cause problem in terms of running the BP algorithm on a tree graph, such as the one in figure 2. For that we follow the idea that the probability of X_r being the cause of an issue X_N can be exactly computed by a BP marginal on a tree graph where all the possible causes, $X_1, ..., X_r, ..., X_{N-1}$, are correlated with X_N by a function node $a_{r,N}$.

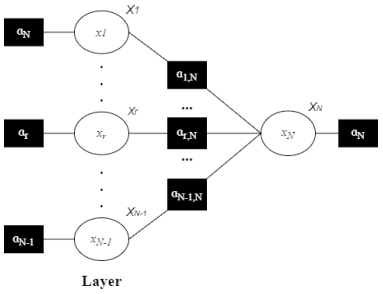

Fig. 2. Representation of the graph used to structure the root cause problem. There is an issue X_N, represented by the label variable x_N, and one layer variables for the X_r possible causes, where r is the integer $r \in [1, N-1]$. All variable nodes in the layer are connected to the label node.

In order to do that, we consider that the variable x associated with an event X can assume either the value 0 (if the event X is not verified) or the value 1 (if the event X is verified) with a certain marginal probability. In this context, we need to fix the marginal probability of the event X_N to 1, expressed by $b(x_N = 1) = 1.0$, what can be done by defining the function

$$\psi_{a_N}(x_N) = \begin{cases} 1.0 \;\; if \; x_N = 1 \\ 0.0 \;\; otherwise, \end{cases} \tag{4}$$

while all the other events, $X_1, ..., X_r, ..., X_{N-1}$, remain unfixed, and so, its corresponding variables x_r have no preferred state. Meanwhile, the functions $a_{r,N}$ connecting a variable node i_r of the layer with the label variable node, i_N, are defined as

$$\psi_{a_{r,N}}(x_r, x_N) = \begin{cases} |\rho_{r,N}| \;\; if \; x_r = x_N \\ 1 - |\rho_{r,N}| \;\; otherwise, \end{cases} \tag{5}$$

where $\rho_{r,N}$ is the correlation coefficient between the event X_r and the issue X_N.

The correlation between two events can be computed using different measures. The ones used in this work are the Pearson, Spearman rank and Kendall's tau correlation coefficients, which vary between -1 and +1, being that the closer to zero the weaker the relationship between two variables [17]. However, while the Pearson correlation measures a linear relationship between two variables, the Kendall and Spearman correlations measure the monotonic relationship, that is, how likely it is for two variables to move in the same direction, but not necessarily at a constant rate. The Kendall correlation is often preferred over Spearman's for being more robust while the Pearson correlation has the limitation of having more requirements, such as the need for both variables to be normally distributed and continuous and for the outliers to be handled ahead of time, as they can greatly affect the computed correlation coefficient [17].

The described model may yield interesting results but the root cause of an issue it's not always directly determined. The most probable event of the layer might actually be a manifestation of another event that happened earlier in time and that is the actual root cause that we wish to determine. Because of this, a modification to the initial model would be to exclude the issue X_N from the graph after the initial run and to update the label variable after every run of the BP algorithm, so that it will correspond to the event X_r that was previously determine as the most probable. Before running the BP algorithm on the new graph the label probability is fixed, either as

$$\psi_{a_r}(x_r) = \begin{cases} 1.0 \;\; if \; x_r = 1 \\ 0.0 \;\; otherwise, \end{cases} \tag{6}$$

if the model parameter f has value "True", or as $\psi_{a_r}(x_r) = b^{T-1}(x_r)$ if the model parameter f has value "False", being that b^{T-1} represents the estimate of the marginal probability previous to the current run T of the BP algorithm. The described root cause model will be referred to as $\mathcal{M}_3$, but there are still some other variations that we can apply to it concerning the events and number of nodes that are represented in the graph after each run.

In one of its variations (version $\mathcal{M}_{(3,0)}$) the number of nodes is preserved and all the events are presented in all the runs. In version $\mathcal{M}_{(3,1)}$, the event represented by the label variable cannot be part of the layer, and so, after the first run the graph only has $N-1$ variable nodes. The version $\mathcal{M}_{(3,2)}$ goes beyond this and also excludes from the graph events that where previously attributed to label variables, so that graphs of consecutive runs differ in size by 1 variable node. This last approach ensures that the model is not trapped into determining the same most probable events at different runs.

An alternative to model $\mathcal{M}_3$ would be to simply run the BP messages on a graph with more than one layer, as many as the maximum number of runs, T_{max}, of model $\mathcal{M}_3$. Such an approach will be referred to as $\mathcal{M}_{(0,0)}$, which for $T_{max} = 3$ makes use of the graph presented in figure 3. Note that now is necessary not only to compute the correlation coefficient between the issue and the other events but also the correlation coefficient between all the events, so we can define the functions connecting different layers.

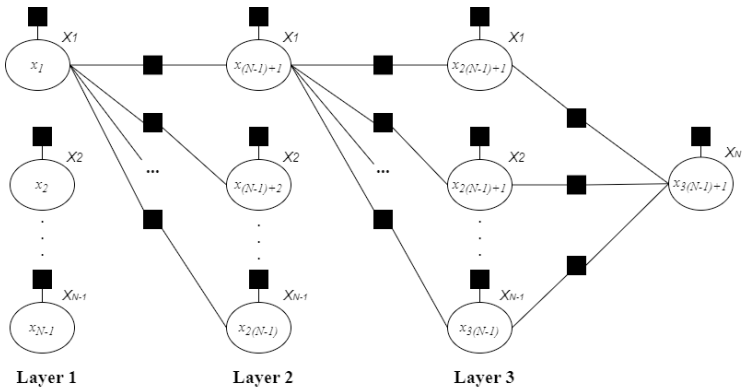

Fig. 3. Adaptation of the graph of figure 2 in order to run model $\mathcal{M}_{(0,0)}$ for 3 different layers.

Using the same graph structure it is possible to run a model $\mathcal{M}_1$ where, similarly to what is done in model $\mathcal{M}_3$, the results of different runs of the BP algorithm are incorporated into the graph. After each run of $\mathcal{M}_1$, the marginal

probabilities of some variable nodes are fixed following certain rules. In version $\mathcal{M}_{(1,0)}$, after a run T the variable which presented the higher marginal probability in layer $T_{max} + 1 - T$ is fixed either to 1 or to $b^{T-1}(x = 1)$, according to the model parameter f (as described before), while the remaining variables x_r of that layer are fixed to 0 by defining the function

$$\psi_{a_r}(x_r) = \begin{cases} 1.0 \;\; if \;\; x_r = 0 \\ 0.0 \;\; otherwise. \end{cases} \tag{7}$$

Additionally, in version $\mathcal{M}_{(1,1)}$ the most probable event determined from layer $T_{max} + 1 - T$ at iteration T is fixed not only on that layer but also on layer $T_{max} - T$, so that it is not chosen again as the most probable event. In version $\mathcal{M}_{(1,2)}$, once an event is determined as the most probable on a layer it is fixed on all the other graph layers of number smaller than $T_{max} + 1 - T$. However, once all the nodes of a layer are fixed, we see no need to run the BP messages on layers of bigger number, as they will have no influence on layers of smaller number.

In this context, in model $\mathcal{M}_2$, after each iteration T are excluded all the nodes of layers equal or superior to $T_{max} + 1 - T$, with the exception of the variable node that represents the most probable event on layer $T_{max} + 1 - T$. Such variable will be the new label and it is fixed according to the model parameter f. The versions $\mathcal{M}_{(2,0)}$, $\mathcal{M}_{(2,1)}$ and $\mathcal{M}_{(2,2)}$ of $\mathcal{M}_2$ follow the same logic of $\mathcal{M}_{(1,0)}$, $\mathcal{M}_{(1,1)}$ and $\mathcal{M}_{(1,2)}$, being that now, instead of fixing the most probable event of layer $T_{max} + 1 - T$ on layers of smaller number, the respective variable nodes are actually removed from those smaller number layers.

3. Application to Gas Leak Testing

The leak testing rig from which data was extracted, and schematically depicted in Fig. 4, is centred around the ATEQ D520 differential pressure flow meter, which can determine leakage flow rates through the pressure drop across a calibrated flow tube. Pressure drop across a calibrated flow tube with a fluid in laminar flow is proportional to flow [6], as determined by

$$\Delta P = \frac{8\mu LQ}{\pi R^4} \tag{8}$$

where ΔP is the pressure drop across the calibrated flow tube; μ is the fluid viscosity; L is the length of calibrated flow tube; Q is the flow; and R is the radius of the calibrated flow tube. Each test cycle is comprised by four phases, including: coupling between test parts and test jig; pressurisation of the test part until compensation flow is obtained; wait time for flow stabilisation; and final testing instance where the differential pressure is measure across the calibrated flow tube, thus detecting air flow induced by leaks on the testing part. After each cycle, the measuring device returns the total leak rate, pressure within test part, and an error code in case of a failed test. Furthermore, a set of external sensors was later retrofitted into the test rig (as seen in Fig. 4). For each external sensor, one measurement is taken at each phase of the test cycle, except for T5, P1 and H1, that record environmental conditions and thus only one measurement is taken. Moreover, the ATEQ D520 is capable of automatic compensation of atmospheric conditions by performing a test cycle on a master part. At the start of each shift (at a maximum of three shifts per day) a master part is connected to a calibrated leak device with a well known leak flow. The master is tested twice, once per each of two calibrated leak devices named “OK Leak” and “NOK Leak”, each having well-defined theoretical leak rate values of 5.9 cc/min and 8.7 cc/min, respectively. The equipment is locked from normal operation until an initial master part calibration is successfully achieved for each calibrated leak device. Generated sensor data points are equal for both master and regular production parts and can be summarized as: leak rate (flow rate of leak in test part); pressure (pressure value within the test part); environmental values of pressure, humidity, and temperature; and temperatures within the testing circuit at each test stage. Lastly, two further features are later computed, Leak Delta OK and Leak Delta NOK, corresponding to the differential between the obtained leak rate test results and the maximum or minimum theoretically obtainable values for valid or invalid tests, respectively.

The ATEQ device is connected to a Programmable Logic Controller (PLC) which is also responsible for managing external sensors and re-routing data to the enterprise’s Manufacturing Execution System (Nexeed-MES) and predictive platform’s Data Receiver API. This logic is depicted in Fig. 5, where the contextualisation of the current use-case within the developed smart manufacturing platform is shown, together with the general data flow.

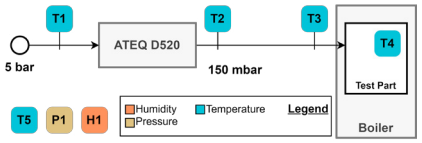

Fig. 4. Simplified schematic of the test-jig assembly.

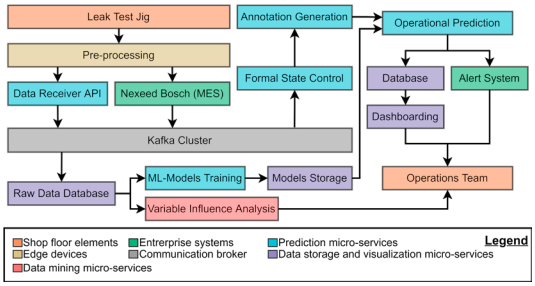

Fig. 5. Overview of smart manufacturing platform in which the developed solution will be integrated. In blue, developed or enhanced micro-services for this project.

We first introduced this use-case in [10], emerging as a multiclass classification problem. Seven relevant classes were identified closely related to different failure types, and thus striving to provide some interpretability to the ensemble of machine learning models, Fig. 6a. Due to the concise nature of most classes, this was in fact successful, yet the final model lacked true interpretability, particularly in the distinction of false versus true rejections of tests. As such, from the aforementioned seven data classes, only two are relevant for the current exemplary application: T-NOK and F-NOK. The former represents a True-NOK instance, i.e., any subsequent test will also result in a part fail; the latter signifies a False-NOK instance, where the test part fails to pass, yet upon immediate re-test it is considered non-faulty, and therefore a false part rejection. Reasons for rejection misclassification are unclear from an organisational point of view. Initially, it was believed that the issue stemmed from an unknown factor's influence, namely due to environmental conditions. However, leak testing procedures are highly dependent on other external factors, such as operator skill or even test jig set-up and characteristics, for instance seal jig restrictions or tube lengths and diameters, which can deeply influence and result in unsatisfactory results using default factory calibrations.

The initially provided data set contained leak test records spanning across a five-month time-period, representing a total of 49.649 separate testing instances, of which 40.046 represent normal behaviour (class OK), 416 represent T-NOK, and 1.100 instances are of F-NOK (false rejections). Therefore, relevant instances for this application total 41.562, Fig. 6b, where 96.35% of all data accounts for typical behaviour (class OK), and the unbalanced nature of the problem is clear from the ratio of roughly 96:1 and 36:1 between classes OK and T-NOK and OK and F-NOK, respectively.

Fig. 6. Representation of: (a) initial seven classes that constituted the data set. Red line represents the maximum leak rate value hard-coded into the differential pressure sensor as the maximum allowable leak rate. Green line is the minimum acceptable test pressurisation that must be achieved for a successful test, computed through the lower inner fence of pressure values of OK class; (b) close up of only the three relevant classes to the current problem. Note the overlap between false and true rejections. Parts classified as F-NOK will be considered up to standard in subsequent re-testing.

3.1. Results of the Variable Influence Analysis

For the Gas Leak Testing problem we defined the set of models $\mathcal{M}$ of Section 2.3 with parameters (ρ, f) and indexed by $\Omega_{(\rho,f)} = \{(0,0), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2), (3,0), (3,1), (3,2)\}$.

These models $\mathcal{M}_k$, with $k \in \Omega_{(\rho,f)}$, can all be characterised by the parameters variations: $\rho \in \{\text{Spearman}, \text{Kendall}\}$, for the metric that is used for weighting the variables influence; and $f \in \{True, False\}$, for the way the value of certain variables is fixed. Furthermore, each variable x at a layer $L \in \{1, 2, 3\}$ has an associated probability $p_{k,L}(x)$ of being the most influential variable measured by a specific model k.

In general, process experts may choose a particular model $k \in \Omega_{(\rho,f)}$ to proceed with the variable influence analysis (VIA). However, to achieve a robust result, additionally in what follows we propose the notion of variable influence score for each variable x.

Let $\mathcal{S}$ be the set of variables, L the set of layers, and $\Omega_{(\rho,f)}$ the set of models. For each variable $x \in \mathcal{S}$ and model $k \in \Omega_{(\rho,f)}$, we define

$$\zeta_k(x) = \sum_{s \in L} \gamma_L \, p_{k,s}(x), \quad \text{with} \quad \sum_{s \in L} \gamma_s = 1 \text{ and } \gamma_s > 0.$$

The coefficients chosen in Gas Leak Testing were $\gamma_1 = 0.40, \gamma_2 = 0.35, \gamma_3 = 0.25$, for 3 different layers. This accounts to the fact that the first layers evaluated describe a more direct influence than the last layers evaluated. So, $\zeta(x)$ is a weighted average of the influence probability of the variable along layers. Figure 7 shows the basic statistics of a choice of parameters.

Fig. 7. Example of basic statistics of $\zeta_k(x)$ for OK-FNOK and models $\mathcal{M}_k$ with $k \in \Omega_{(\rho,f)}$, for ρ=Kendall's tau correlation and f=False: (a) The minimum, average and maximum of $\zeta_k(x)$ for each x and all k; (b) The minimum, average and maximum of $\zeta_k(x)$ for each k and all x. As seen in Table 1, the relevant variables are Leak Rate (V1), Leak Delta OK (V66), Leak Delta NOK (V67), and Pressure (V19).

For VIA, we are interested in finding the number of times a variable attains the maximum probability among all variables, so we define

$$\delta_k(x) = \begin{cases} 1\ , \text{ if } \zeta_k(x) = \max_{y \in S}\ \zeta_k(y) \\ 0\ , \text{ otherwise} \end{cases} \quad \text{and} \quad \varphi_{\rho,f}(x) = \sum_{k \in \Omega_{(\rho,f)}} \delta_k(x),$$

where $\varphi_{\rho,f}(x)$ is the denominated **variable influence score** of the variable x under the model $\mathcal{M}_k$ with $k \in \Omega_{(\rho,f)}$.

Table 1 shows the variable influence scores obtained for the problem described in Section 3, using two distinct approaches with differing baseline events, represented as OK-FNOK and OK-TNOK. In the former, FNOK, or false rejections, serve as the event; in the latter the same logic is applied to TNOK as the event. In either case, the data sets only contained instances of the base event and of normal behaviour (OK class).

	OK-FNOK					OK-TNOK				
Variable	**S-F**	**S-T**	**K-F**	**K-T**	**Average**	**S-F**	**S-T**	**K-F**	**K-T**	**Average**
Leak Rate	6	8	4	6	6.00	6	8	7	9	7.50
Leak Delta OK	4	6	2	4	4.00	4	6	5	7	5.50
Leak Delta NOK	3	3	5	5	4.00	3	3	3	3	3.00
Pressure	2	1	2	1	1.50	2	1	2	1	1.50
G_1 (Humidity)	1	1	1	1	1.00	1	1	1	1	1.00
G_0 (Temperature)	0	0	0	0	0.00	0	0	0	0	0.00

Table 1. Values of variable influence scores for the Gas Leak Testing where less relevant variables are grouped in G_1 and G_0. The G_1 contains all the variables involving ambient humidity measurements, whereas G_0 contains all the variables related to temperature measurements. In the table header, a-b means the following model parameters assignment $\rho = a$ and $f = b$; i.e., S- means ρ=Spearman's rank correlation, K- means ρ=Kendall's tau correlation, -F means f=False, and -T means f=True.

Results show that Leak Rate has the highest variable influence score implying the highest average probability of influence over the event, for either one of the followed approaches. This is naturally expected, as it is the foremost physical quantity being measured, falling in line with the graphical interpretation seen in Fig. 6b. Moreover, differential variables, Leak Delta OK and Leak Delta NOK, closely follow the total Leak Rate value's influence, yet this also hints to badly calibrated master parts, as leak delta signifies not only the differential between obtained results and theoretical limits, but this also leads to further implications of master parts being further away from expected calibration values. In fact, this is attested by the increased influence of Leak Delta NOK in the event of false rejections comparably to true rejections, pointing to the specific miscalibration of the NOK Leak calibration device. The case of true rejections and a higher relationship with Leak Delta OK is more sensible as the increasing differential between obtained leak value and the calibrated maximum permissible leak rate leading to a true rejection of parts is expectable.

Remaining variables show lesser influence, although, interestingly, yielded the same results across each approach, implying that environmental factors exert a reduced influence over this particular problem, as temperature seemingly has no influence over the event and humidity has a distinctively low probability of influence. Note that, temperature and humidity, G_0 and G_1, respectively, correspond to the aggregation of all similar types of variables, and hence the low scores are even more poignant, corresponding to the summation of influence of a larger group of features. Pressurisation within test part also shows low influence in final influence, which in the context of the problem is viewed positively as the equipment succeeding in establishing adequate testing conditions, i.e., high leak rate values are not to be caused by leaks within the testing circuit itself (as in pipage, for instance) or in connecting seals.

4. Conclusions and Future Work

In this paper we reintroduce a use case at Bosch Thermotechnology facilities where a differential leak testing procedure introduced a bottlenecking effect with extensive repercussions during the production of water boilers, due to the commonality of false part rejections that would later require re-testing. Past work focused on the application of strong machine learning models for the classification of false rejections, however, due to the black-box nature of these methodologies, it is difficult to infer actionable organisational measures. As such, we introduced an alternative, purely data-driven, which is the Variable Influence Analysis technique for the determination of probability of causal event relationships. For that we applied multiple versions of VIA models based on the Belief Propagation algorithm to factor graphs relating different events through correlation relationships, after which we aggregated results into a more readable format where it was determined the average variable influence score.

These results enabled the logical deduction of possible procedural improvement measures for Bosch Thermotechnology. Initially, it was thought that environmental factors played a large role in the skewing of tests, however results showed that, excepting humidity, their effect was negligible. This is a sensible conclusion as humidity is notoriously difficult to fully control. Alternatively, so is the low influence of temperature, as these values show insignificant variation between tests. More importantly, the results seem to point towards an issue occurring during the calibration of master parts, which itself can be derived from: improper handling, storage, or uncalibrated master parts; or, equipment fault, e.g., ATEQ D520 self-calibration system not properly set-up for the implemented testing jig. This work allowed for determination of interpretable information, opening the possibility for Bosch Thermotechnology to initiate an in-

vestigative project into these key areas for the continuous improvement of their processes, and thus providing true actionable value.

Still open to future research is the applicability of this methodology to enhance data pipelines centred around machine learning or artificial neural network models, studying possible increases in predictive performance through the application of VIA models. Namely in providing added explainability to models, or as a feature selection mechanism. A complementary research path would be the evaluation of the solution's applicability to embedded data-science devices, either by solely relegating it the feature selection stage, or even to fully transpose the entire data-processing pipeline towards edge-devices.

Acknowledgements. The present study was developed in the scope of the Project Augmented Humanity (PAH) [POCI-01-0247-FEDER-046103], financed by Portugal 2020, under the Competitiveness and Internationalization Operational Program, the Lisbon Regional Operational Program, and by the European Regional Development Fund. The second author has a PhD grant supported by PAH. The third author was partially supported by the Center for Research and Development in Mathematics and Applications (CIDMA), through the Portuguese Foundation for Science and Technology, reference UIDB/04106/2020.

Data availability statement. The data sets used to obtain the results are confidential information of Bosch company manufacturing system, so they are not publicly available.

References

[1] M. J. Ershadi, R. Aiasi, S. Kazemi, Root cause analysis in quality problem solving of research information systems: a case study, International Journal of Productivity and Quality Management 24 (2) (2018) 284–299. doi:10.1504/IJPQM.2018.091797.

[2] A. Garcia, J. L. Ferrando, A. Arbelaiz, X. Oregui, A. Bilbao, Z. Etxegoien, Soft computing analysis of pressure decay leak test detection, in: 15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020), Springer International Publishing, Cham, 2021, pp. 299–308.

[3] C. C. Daniels, N. G. Garafolo, Effect of system variables on the uncertainty of the mass point leak rate methodology using first-order regression, Nondestructive Testing and Evaluation 29 (1) (2014) 14–28. doi:10.1080/10589759.2013.823610.

[4] H. Guntur, M. Cai, K. Kawashima, T. Kagawa, Analysis of temperature effect on differential pressure method for air leak detection, in: SICE 2004 Annual Conference, Vol. 1, 2004, pp. 159–162 vol. 1.

[5] Y. Shi, X. Tong, M. Cai, Temperature effect compensation for fast differential pressure decay testing, Measurement Science and Technology 25 (2014). doi:10.1088/0957-0233/25/6/065003.

[6] ATEQ, Ateq d520 user manual - version 1.32 (2009).
URL https://atequsa.com/wp-content/uploads/2014/07/PREMIER-D-USER-MANUAL-ENGLISH.pdf

[7] Y. Cui, S. Kara, K. C. Chan, Manufacturing big data ecosystem: A systematic literature review, Robotics and Computer-Integrated Manufacturing 62 (2020) 101861. doi:https://doi.org/10.1016/j.rcim.2019.101861.

[8] A. Bousdekis, K. Lepenioti, D. Apostolou, G. Mentzas, A review of data-driven decision-making methods for industry 4.0 maintenance applications, Electronics 10 (7) (2021). doi:10.3390/electronics10070828.

[9] P. Linardatos, V. Papastefanopoulos, S. Kotsiantis, Explainable ai: A review of machine learning interpretability methods, Entropy 23 (1) (2021). doi:10.3390/e23010018.

[10] D. Costa, E. M. Rocha, P. Ramalho, Minimizing false-rejection rates in gas leak testing using an ensemble multiclass classifier for unbalanced data, 2022 (*submitted*).

[11] M. Mezard, A. Montanari, Information, Physics, and Computation, Oxford University Press, Inc., USA, 2009.

[12] J. S. Yedidia, W. T. Freeman, Y. Weiss, Understanding Belief Propagation and Its Generalizations, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003, p. 239–269.

[13] J. Pearl, Reverend bayes on inference engines: A distributed hierarchical approach, in: Proceedings of the Second AAAI Conference on Artificial Intelligence, AAAI'82, AAAI Press, 1982, p. 133–136.

[14] A. Kirkley, G. T. Cantwell, M. E. J. Newman, Belief propagation for networks with loops, Science Advances 7 (17) (2021) eabf1211. doi:10.1126/sciadv.abf1211.

[15] A. Lipowski, A. L. Ferreira, D. Lipowska, K. Gontarek, Phase transitions in ising models on directed networks, Phys. Rev. E 92 (2015) 052811.

[16] M. Welling, Y. W. Teh, Linear response algorithms for approximate inference in graphical models, Neural Computation 16 (1) (2004) 197–221. doi:10.1162/08997660460734056.

[17] Data science stats review: Pearson's, kendall's, and spearman's correlation for feature selection, https://www.tessellationtech.io/data-science-stats-review/, accessed: 2022-06-29.