applied sciences

MDPI

Article

Overlap in Automatic Root Cause Analysis in Manufacturing: An Information Theory-Based Approach

Eduardo e Oliveira [1], Vera L. Miguéis [2,*] and José L. Borges [2]

[1] Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial (INEGI), Associate Laboratory for Energy, Transports and Aerospace (LAETA), Campus da FEUP, R. Dr. Roberto Frias 400, 4200-465 Porto, Portugal
[2] Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência (INESC TEC), Faculdade de Engenharia da Universidade do Porto, Campus da FEUP, R. Dr. Roberto Frias 400, 4200-465 Porto, Portugal
* Correspondence: vera.migueis@fe.up.pt; Tel.: +351-22-508-1400

Abstract: Automatic Root Cause Analysis solutions aid analysts in finding problems' root causes by using automatic data analysis. When trying to locate the root cause of a problem in a manufacturing process, an issue-denominated overlap can occur. Overlap can impede automated diagnosis using algorithms, as the data make it impossible to discern the influence of each machine on the quality of products. This paper proposes a new measure of overlap based on an information theory concept called Positive Mutual Information. This new measure allows for a more detailed analysis. A new approach is developed for automatically finding the root causes of problems when overlap occurs. A visualization that depicts overlapped locations is also proposed to ease practitioners' analysis. The proposed solution is validated in simulated and real case-study data. Compared to previous solutions, the proposed approach improves the capacity to pinpoint a problem's root causes.

Keywords: Root Cause Analysis; manufacturing; fault diagnosis; data mining; information theory

Citation: e Oliveira, E.; Miguéis, V.L.; Borges, J.L. Overlap in Automatic Root Cause Analysis in Manufacturing: An Information Theory-Based Approach. *Appl. Sci.* **2023**, *13*, 3416. https://doi.org/10.3390/app13063416

Academic Editor: Dimitris Mourtzis

Received: 10 Feburary 2023
Revised: 2 March 2023
Accepted: 6 March 2023
Published: 8 March 2023

1. Introduction

Manufacturing is highly competitive [1], and its operations are very complex [2]. Quality is essential for companies aiming to improve customer satisfaction and competitiveness [3]. Maintaining quality and improving operations is essential to remain competitive [4]. One process required for the maintenance of and improvement in operations is Root Cause Analysis (RCA) [5]. This process's goal is to find the true origin (or root cause) of a problem. It is an essential process, as it allows companies to find a problem's root cause, enabling manufacturers to learn the underlying issue and improve the manufacturing process [6,7]. Finding the root causes and solving the problems at the source allows stopping the problem from persisting, which means that locating and eliminating the source of the problem are critical [8,9]. RCA emphasizes the complexity of manufacturing operations, as it is not trivial to distinguish the root causes from the symptoms, and it requires extensive system and execution analysis [10,11].

The above-mentioned characteristics of the RCA problem led to several studies proposing solutions to improve the efficiency and efficacy of RCA. Recently, some studies have used the increase in the volume of data generated in manufacturing environments [2,12–14], to develop solutions called Automatic Root Cause Analyses (ARCA), which automatize at least part of the RCA process. However, even the development of these solutions is not trivial, as it requires ensuring that they are a good fit for the data's characteristics. It is essential to take into consideration data characteristics in order to develop solutions with satisfactory performance. ARCA solutions aim at improving decisions through the analysis of data, as conceptualized under the framework of Industry 3.5, an intermediate stage before achieving Industry 4.0 [15–17]. Some studies, such as [18,19], propose a combination

between traditional RCA methods and automated solutions. More recently, efforts have been made to develop automated solutions for root cause analysis that incorporate notions of causality instead of relying only on correlations. References [20–22] are examples of such works.

As described in [23], root causes can be understood at three different levels, depending on the types of data available: (i) the root cause's location, (ii) the physical characteristics of the root cause, and (iii) the human/organizational aspects of the root cause. The first level portrays the location of the root cause in the manufacturing process. For example, [24,25] are studies that use this type of data. The second level focuses on what physical attributes are the root cause (e.g., a sudden increase in voltages or high temperature). Examples of studies using this type of data are [26–28]. The third and final level centers on the human and organizational aspects of the root cause (e.g., equipment maintenance), so as to identify what triggered the physical problem. References [29,30] are examples of studies exploring this final level of root causes. In this paper, we use the first level of data and try to find the location of the root cause. We choose this type of data as it is the most available one, and solutions based on it can be advantageous to more factories.

Analyzing location-type data is still relevant, despite being just the first level of data. To detect the physical aspects of ARCA, factories require proper infrastructure, which some factories do not have. However, data on how the product flows through the manufacturing process are usually readily available, which enables the determination of the root causes' locations. In situations where the manufacturing process is particularly complicated (e.g., in semiconductor manufacturing [31]), even production flow data can become difficult to analyze, which justifies the use of Data Mining (DM) and Machine Learning (ML) techniques in the development of solutions that aid in diagnosis.

An issue of locating the root causes is overlap, as presented in [32]. A manufacturing process can be seen as a progression of steps products go through, starting as raw materials or parts and becoming a finished product in the end. Overlap is a phenomenon that happens when, in two separate manufacturing steps, all products that go through a certain machine in one of the steps are all processed in another given machine in a later step. With the data generated by this phenomenon, it is very hard to distinguish the influence each machine has on the quality of the final product, especially if it is analyzed through classification algorithms.

Overlap may happen due to stabilization in the manufacturing process. For example, if we have a process where products always flow from the same two machines in contiguous steps, as these have their times synchronized, as soon as the machine in the earlier step is finished, it is always the same machine in the later step that becomes available . Despite the example, this does not mean that overlap can only occur between machines that operate in contiguous steps. Attaining this sort of balance in the manufacturing process can be positive in terms of productivity and efficiency, but it becomes an issue during the analysis of the data generated from such a process. This contention between what is positive from a production perspective (that should be prioritised) and what is advantageous for diagnosis based on data analysis is relevant, as this signifies that overlap will occur regularly when performing diagnosis on data of the location type.

It is important to mathematically measure overlap in order to correctly gauge its effect on the development of ARCA solutions. However, that measure needs to consider two aspects: (i) the direction of the association between factors (manufacturing process variables), and (ii) it needs to be able to recognize the effect of overlap between individual machines (further explanations can be found in Section 3). These aspects are not considered in [32].

The goal of this paper is to develop an ARCA solution that is able to locate the root cause of a problem in a manufacturing process, and that is robust to overlap. To do so, we propose an overlap measure that considers the aspects mentioned in the above paragraph. The measure is based on information theory, in particular, the use of a variant of mutual information named Positive Mutual Information (PMI), proposed in [33]. Building on

top of this new measure, we introduce a new method that is used to handle RCA and identify the most probable root causes and is robust to overlap. A visualization tool is also proposed, making the task of identifying overlap and the root causes easier for practitioners. We validate the proposed approach in simulated data and real data from a case study in semiconductor manufacturing, which is a highly competitive sector [34].

The rest of the paper has the following structure. In Section 2, an overview of previous works relevant to this study is presented. First, a brief discussion about previous works of ARCA in manufacturing is presented. This is followed by a definition of the problem of overlap in Section 3. In Section 4, the proposed measure of overlap is described, in addition to the proposed method for RCA based on this measure. Sections 5.1 and 5.2 explain and show (respectively) the results of the experimental procedures used to validate the proposed methods. Section 6 discusses and summarises the results obtained in the previous section and discusses future research directions. Finally, we summarise our findings and present the main conclusions.

2. Previous Works

A manufacturing process consists of the processing of products and materials through several steps (detailed in Section 3). This paper's goal is to locate root causes in a manufacturing process, that is, to determine which machines were the root causes of a problem in the process. In [35], the authors addressed this issue, calling it the "root cause machineset identification problem". This study uses a three-phase method to locate the root cause: (i) it processes the dataset of a moment with a problem; (ii) it generates various candidate machinesets (groups of machines); and (iii) it applies association rule mining to the dataset. The rules obtained are analyzed based on a novel measure of interest that considers both confidence and the continuity between the defective products for a candidate machineset. The method extracts the location of the root cause from the antecedent of the association rule, with the consequent side representing whether the product is normal or defective.

Another relevant work on the same topic is [24]. This study proposes a solution that is able to identify quality drifts and the root causes continuously and automatically. A two-phase algorithm is proposed, which first clusters common defects and then determines the root causes. The Squeezer algorithm is used for clustering. To locate the root causes in the second phase, each group of machines is examined to verify if it can be identified as a root cause or not, based on the sequence of defects. Ref. [25] proposes a visualization technique based on the Herfindahl–Hirschman Index (HHI) to identify patterns in the concentration of faults in the machines. This technique helps practitioners find root causes in a quick and transparent way.

There are also some studies in the literature that mention that a correlation between factors when trying to identify root causes can become an issue, and strategies are proposed to tackle it. Ref. [36] uses clustering to group highly-correlated factors, and selects an archetypal factor from them to use during modeling and analysis. Ref. [37] presents a two-step technique to identify parameters with faulty values in semiconductor manufacturing. In the first step, Principal Component Analysis is used for feature engineering, making the distinction between normal products and faulty ones clearer by increasing the separation between both classes. In the second step, classifiers are used to identify the factors leading to the appearance of faults. Ref. [38] mentions the complexity in determining whether a specific factor is the root cause when multiple faults are present and explains that the compound effect of multiple faults on factors can be very distinct from the effect of the individual faults. The authors couple data analysis with cause-and-effect information in order to address this issue. In [39], the combination of faults and its resulting information are analysed using Bayesian networks. Partial Least Squares with Variable Importance in Projection (PLS-VIP) is used to select the most relevant factors, which ensures that the rules obtained contain only the necessary information.

Most papers that develop solutions based on location data use classifiers, analyzing the knowledge structure (e.g., decision trees, rules) to determine the root causes. We argue that

solutions based on classifiers can be impaired in terms of performance by the presence of overlap. Although the method proposed by [24] does not use classifiers, it does use product queues that can nevertheless be affected by overlap. Ref. [25], despite also proposing a method not using classifiers, is based on visualization and concentration measures, that are not able to identify overlap and its detrimental effect. Ref. [40] has a broad scope and aims at identifying pitfalls of applying data science to manufacturing problems. It alludes to some pitfalls in determining important factors but does not mention overlap. Ref. [41] also focuses on feature selection in the context of RCA, but again does not present any mention of overlap. The issue of overlap was first identified in [32], and the authors proposed measuring the overlap using the strength of association between factors. However, this measure does not consider how the association is directed. This method and measure were further extended in [20] by using the concept of causality. A literature review about the topic of ARCA in general can be consulted in [23].

Given the above-mentioned background, this paper contributes to the literature by presenting a novel measure of overlap rooted in information theory, which considers the aspect of the direction of the association. This measure is resilient to overlap, a phenomenon that can be detrimental to the performance of solutions focused on the use of classification algorithms for analyzing location data with the aim of identifying root causes. This paper also proposes an ARCA solution based on the novel measure that is robust to overlap.

3. Problem Definition

Overlap is a phenomenon specific to the problem of locating the root causes of problems in a manufacturing process. As products go through a sequence of manufacturing steps in a manufacturing process, in each of those steps, they are processed in a certain machine. A step–machine combination means that a product was processed in a certain machine in a certain step. This combination is also called a tuple. Figure 1 depicts such a manufacturing process, where a product goes through several steps, and is processed in a machine in each step. The squares with P_i illustrate products as they flow through a manufacturing process. In this depiction, Product 1 is in line to be processed before Step B, Product 2 is still being assembled/transformed in Machine M_3, and Product 3 was already processed and is currently being monitored. The problem's root cause is located in Step A, Machine M_1.

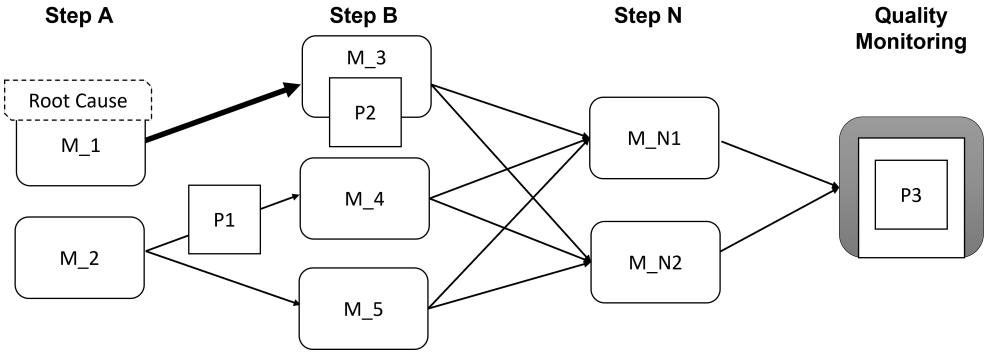


Figure 1. A depiction of an example of a process that produces data that can be used to locate a root cause [32].

At the end of the process, the product has its quality monitored, and it is defined as normal or problematic. If the number of products with problems increases sharply in a short period, it indicates that the process has a problem needing to be tackled, which requires RCA to determine its origin.

The manufacturing process described above generates data similar to those shown in Table 1. Each row corresponds to a product, each column corresponds to a step, and each cell indicates the machine where that row's product was processed for that column's step. The "Problem" column represents the final quality of the product with respect to whether it

had a problem or not. We have chosen to use four instances in this table, as this provides enough variety of examples, but not enough complexity to prevent the comprehension of the conceptual example.

Table 1. Data generated by the example depicted in Figure 1, in tabular form.

Product	Step A	Step B	Step N	Problem
1	**M_1**	**M_3**	M_N1	**1**
2	M_2	M_4	M_N2	0
3	M_2	M_5	M_N1	0
4	**M_1**	**M_3**	M_N2	**1**

The objective when trying to locate a root cause in this type of data is to determine the step–machine tuple that represents the location of the origin of the problem. However, this can become extremely hard if overlap is present.

Overlap can be understood as a synchronization within the manufacturing problem that makes all products that pass through a given machine in a certain step also pass through another specific machine in a step further ahead in the process. In addition, all the products that pass through this later step have passed through the same machine in that previous step. This synchronization makes it extremely hard to differentiate how each of these machines influences the quality of the product. Note that the entire trace of step–machine tuples has to be analyzed, as the overlap may also occur between tuples of steps that are not contiguous.

Figure 1 and Table 1 depict an example of overlap: the problem's origin is located in Step A—Machine M_1; however, all products that go through the root cause tuple also go through Step B—Machine M_3. In such a scenario, it is not possible to distinguish which of the tuples was the origin of the problem. Overlap is particularly problematic when trying to use classifiers to automatically extract root causes, as previous solutions in the literature have proposed. This arises due to the knowledge structures (e.g., decision trees, rules) being generated through the selection of the most representative factors, regularly discarding factors highly correlated with the representative ones, due to their supposed redundancy. The criterion used to identify redundant factors can lead to hiding factors that are the true root cause, giving more relevance to the representative factors. For example, when generating a Decision Tree (DT), using the information-gain criterion for splitting promotes the use of factors with more levels, although the root cause may be a factor with a smaller number of levels. In the example above, a DT based on information gain would select Step B–Machine M_3 as a root cause, discarding the true root cause (Step A–Machine M_1) and providing a wrong diagnosis.

Overlap is significant when we only have location data available but the data are high-dimensional (both in number of products and number of steps/machines). In this context, traditional approaches (e.g., Ishikawa diagrams, Failure Mode, and Effects Analysis) are incapable of efficiently dealing with the high dimensionality of the data, and as such, ARCA solutions are necessary for efficiently obtaining the root causes' locations. There are also works that try to expand these traditional solutions in order to improve them, such as [42]. In what concerns ARCA solutions, determining the presence of overlap aids analysts by signaling them about an issue that has repercussions in the analysis through DM and ML algorithms, and prevents them from reaching wrong conclusions, which enables the use of more resilient solutions to locate the true root cause.

Given the description of the problem, it is necessary to consider two aspects when measuring overlap. First, one needs to take into consideration whether an association between factors is positive or negative. This means that we should only consider associations generated by a product that goes through a certain machine in a step and *always* goes through a certain machine in another step (positive association), and not when a product goes through a certain machine in a step and it *never* goes through a certain machine in another step (negative association). Overlap is problematic only in the case of positive asso-

ciation, as it is in that scenario that it becomes impossible to distinguish the tuples. In the case of negative association, the tuple that the product goes through can be immediately determined as a root cause (in the situation where the product is problematic and only those tuples are considered as possible root causes). The second aspect is that the measure needs to focus on comparing tuples (step-machine pairs) and not simply steps.

4. Proposed Methodology

4.1. Proposed Measure

A measure of overlap is necessary to ensure that we can correctly measure its presence and its effects on the performance of ARCA solutions. As mentioned before, overlap is present when the values of two columns in a dataset such as the example in Table 1 are synchronized (in the example, "Machine M_1" with "Machine M_3"). As overlap has a pernicious effect on the analysis using DM and ML algorithms, it is better to consider other perspectives to measure overlap. As such, and to include resilience to noise, we measure overlap from a probabilistic and information theory perspective. Noise is understood as random elements that can affect the final quality of the product. Noise can affect the label, as a small percentage of normal products are labeled as faulty and vice-versa.

Considering overlap from a probabilistic perspective, an overlap between two tuples represents the very high probability of a product going through a certain machine given that we know that it went through another machine in another step. Formally, a high overlap between two tuples occurs when

$$P(Y = y|X = x) \geq Th, \tag{1}$$

That is, overlap occurs when the probability of going through machine y in step Y, given that a product goes through machine x in step X, is above a certain threshold Th (we propose 0.9 to be the default, as it considers cases of very high overlap while leaving some margin for errors caused by noise).

From an information theory perspective, if we consider each tuple as a random variable, it is possible to say that knowing the value of one of the variables provides a high amount of information about the other variable. This information "shared" by both variables is quantified in [43] as mutual information or

$$I(X;Y) = \sum_{x}\sum_{y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}, \tag{2}$$

It varies between 0 and 1, where 1 means the highest association between variables. Alternatively, mutual information can be defined in terms of entropy,

$$I(X;Y) \equiv H(X) - H(X|Y) = H(Y) - H(Y|X), \tag{3}$$

where $H(X)$, $H(Y)$ represent the entropy of X and Y (two random variables), respectively. Mutual information has been widely used in fault detection applications (see [44] as an example). This measure aligns with the intuition developed for overlap, except that it considers both positive and negative associations. In order to improve the measure, we propose the use of Positive Mutual Information (PMI), defined in [33], which takes into consideration only the positive (normalized) mutual information between the two binary variables:

$$PMI(x,y) = \frac{2}{H(x) + H(y)} \left(p(x_p, y_p) log \frac{p(x_p, y_p)}{p(x_p)p(y_p)} + p(x_n, y_n) log \frac{p(x_n, y_n)}{p(x_n)p(y_n)} \right), \tag{4}$$

where x_p and y_p represent positive values (in this specific case "went through machine" and "problematic product", respectively), and x_n and y_n represent negative values ("did not go through machine" and "normal product", respectively). Therefore, in the case of

overlap between two tuples, PMI only takes into consideration the cases where the product goes through either both tuples or neither of them. PMI aligns better with the intuition developed for overlap than mutual information. To compute the effect of overlap on a dataset such as the one in Table 1, first one-hot encoding of the factors representing steps is performed in order to obtain binary variables representing the tuples, and then the PMI between all the new tuples is computed. We then count the number of interactions that have a PMI above a certain threshold and divide by the number of interactions with a valid value of PMI. The number of valid interactions is used, because the PMI on factors that have no positive association is undefined, and considering the total number of interactions would "dilute" the presence of overlap, diminishing the perception of the issue.

Expression (5) measures the amount of overlap on a dataset, based on the concept of PMI. $C_{i,j}$ is one when the PMI between the factors i and j is greater than the threshold Th and zero otherwise. The numerator counts all interactions between i and j, where $i \neq j$, above the threshold.

$$Overlap_{PMI} = \frac{\sum_i^I \sum_j^J C_{i,j}}{N_{VI}} \tag{5}$$

where

$$C_{i,j} = \begin{cases} 1, & \text{if } PMI(i,j) \geq Th \\ 0, & \text{if } PMI(i,j) < Th \end{cases}$$

and $i \neq j$. N_{VI} stands for the number of valid interactions between factors.

Considering the example in Table 1, we would perform one-hot encoding, which would yield Table 2. Each column represents a tuple, identified by the letter of the step followed by the code of the machine. The values of the label *Problem* are included, as they are relevant to identify which tuples are more closely associated with a problematic product.

Table 2. Example from Table 1 transformed with one-hot encoding before computing the PMI values.

A1	A2	B3	B4	B5	NN1	NN2	Problem 0	Problem 1
1	0	1	0	0	1	0	0	1
0	1	0	1	0	0	1	1	0
0	1	0	0	1	1	0	1	0
1	0	1	0	0	0	1	0	1

With this table, it is possible to compute the PMI values between all the tuples using Expression (4), which can be arranged in a matrix, as represented in Table 3. Cells with a "-" represent undefined values of PMI.

Table 3. Matrix with the PMI values of the interactions between tuples in the example from Tables 1 and 2.

	A1	A2	B3	B4	B5	NN1	NN2	Problem 0	Problem 1
A1	1	-	**1**	-	-	0	0	-	**1**
A2	-	1	-	0.505	0.505	0	0	1	-
B3	**1**	-	1	-	-	0	0	-	**1**
B4	-	0.505	-	1	-	-	0.505	0.505	-
B5	-	0.505	-	-	1	0.505	-	0.505	-
NN1	0	0	0	-	0.505	1	-	0	0
NN2	0	0	0	0.505	-	-	1	0	0
Problem 0	-	1	-	0.505	0.505	0	0	1	-
Problem 1	**1**	-	**1**	-	-	0	0	-	1

Notice that, in this example and in other situations, the PMI between tuples of the same step does not exist, which is expected because those tuples are mutually exclusive. Finally, it is possible to compute the $Overlap_{PMI}$ of the whole dataset using Expression (5). With the default threshold value of 0.9 and for the example discussed, the numerator of the expression is 8. The denominator, i.e., the number of valid interactions, is 40. This yields a value of $Overlap_{PMI} = 0.2$.

4.2. Proposed Method

Given the measure defined above, we propose a new factor ranking algorithm to evaluate the overlap between tuples and detect the ones that are most likely to be root causes. Mutual information has already been used for factor ranking in previous studies (e.g., [45–47]).

The proposed factor ranking algorithm is integrated into a method illustrated in Figure 2. As the manufacturing process is operating, data are extracted from it and are analyzed by a problematic-moment-identification algorithm, which tries to identify parts of data relating to periods where there were problems in the manufacturing process. It then selects those parts of the data, which are analyzed by a factor-ranking algorithm that outputs a list of the most likely root causes.

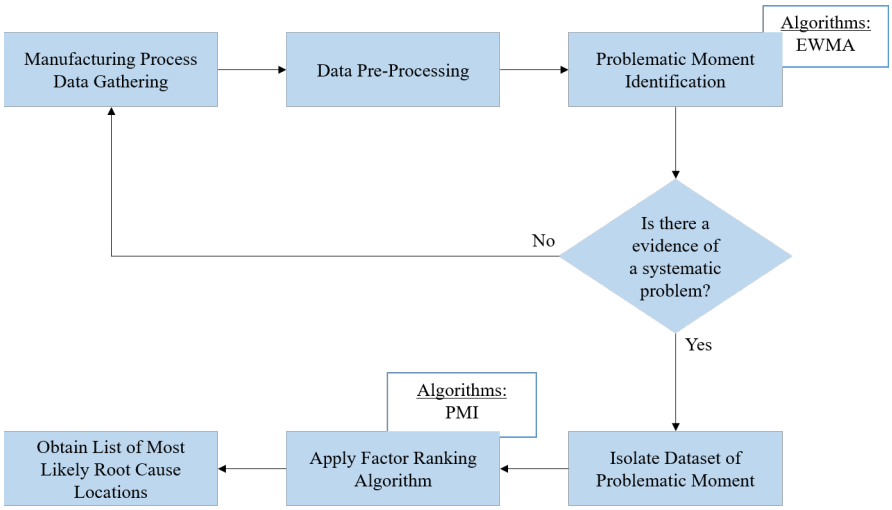


Figure 2. Illustration of the proposed ARCA method.

The algorithm selected for the problematic moment identification stage was Exponentially Weighted Moving Averages (EWMA) [48]. In the proposed solution, EWMA is used to control the proportion of problematic products in a lot. This algorithm was selected after a thorough study analysing the performance of three control charts (EWMA, Cumulative Sum (CUSUM) control charts, and Shewhart charts). The variance to be used in EWMA was estimated using the proportion of problematic products per lot, modeled using the beta distribution. We estimated the α and β parameters and then used them to compute the variance. The choice of using the beta distribution was based on its flexibility, which enables it to fit into the proportion of problematic products that appeared in the real case-study.

In what concerns the novel factor-ranking algorithm proposed, the intuition is straightforward: the goal is to find the tuples that are overlapped with the tuple of the factor representing the final quality (e.g., the label) of a problematic product. In addition, this procedure makes it possible to determine the tuples that are overlapped among themselves and the label, finding clusters of machines that are related to problematic products. After the PMI is computed for all the interactions, the tuples presenting a PMI value in relation to a "Label High" tuple greater than the threshold are kept. These values are then

sorted in decreasing order and presented as a list to the analyst. Algorithm 1 summarises the novel factor ranking algorithm.

Algorithm 1: PMI algorithm

```
Input: Dataset of problematic moment; Threshold
Output: Ordered list of most likely root causes
begin
    Output ← [∅];
    To-use-dataset ← OneHotEncoding(Input);
    Compute PMI Matrix of To-use-dataset;
    Output ← PMI Matrix's column related to problem w/ tuples' names;
    Remove from Output values below Threshold;
    Sort Output;
end
```

Considering the example in Table 3, the only tuples that overlap (i.e., have a PMI value above the threshold) with the problematic product tuple ("Problem 1") are the tuples "A1" and "B3" (excluding the self overlap of "Problem 1"). Of these two tuples, we can see that they also overlap each other, and that there exist no other overlap clusters among other tuples. Therefore, the tuples "A1" and "B3" would be presented to the analyst as the most likely root causes.

To ease the detection of root causes by practitioners, and in addition to the list, we propose a visualization that depicts tuples as nodes and overlap between them as edges between the nodes. We opted to show directed edges to represent the order in which the steps occurred. The nodes are colored according to their steps, e.g., one color is used for Step04 and another for Step06. Figure 3 is an example of the visualization on a small part of the stochastic simulation experiment (different from the previous example). The visualization tool displays all the nodes and overlaps as a grid of nodes, but these can be moved to ease the perception of clusters. Specifically, we focus on the cluster that contains the node "Label High", which indicates a problematic product. We can see a cluster connected with the tuple Label High, which is composed of three Step–Machine tuples. The arrows indicate the order in which the steps occur (e.g., in this example, Step04_Eq occurs before Step06_Eq and Step20_Eq). Each node is colored according to its step.

In the example provided by Figure 3, the true root cause is "Step04_Eq Equipment14". We can see that it is correctly identified (it appears in the network associated with "Label High"), but also that it is overlapped with two other tuples. These three tuples would compose the list given by the algorithm. This exemplifies a situation where overlap is present and makes it impossible to determine which of the three tuples identified is the root cause. However, we have greatly reduced the number of possibilities in this experiment, from 91 possible tuples (22 steps with four possible machines each, plus 1 step with three possible machines) to 3, easing the effort required by the analyst to detect to root cause and making the process more efficient.

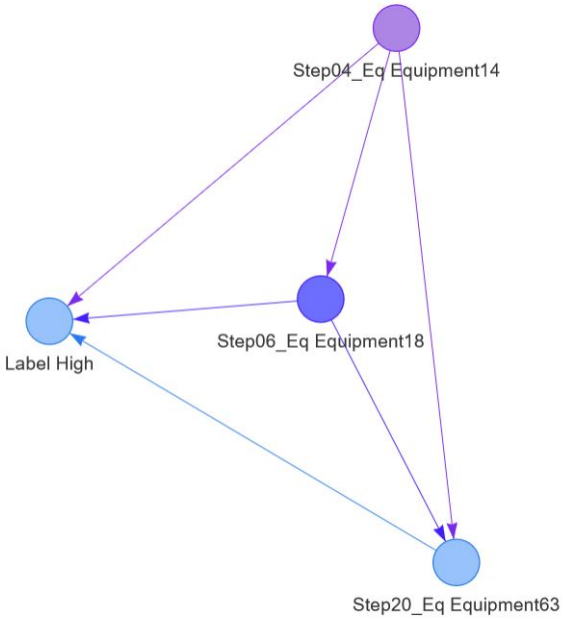


Figure 3. Detail of an example of visualization of PMI overlap among different tuples in simulated data.

5. Evaluation

5.1. Evaluation Setup

To validate the performance of the proposed method, experiments using several datasets were conducted. These datasets are the same as the ones used in [32] to ease comparison. All the datasets have a structure similar to the conceptual examples given in Section 3 and 4. However, we do not have the liberty to divulge or depict the structure of the manufacturing process used in the experiments due to a non-disclosure agreement. The datasets can be divided into three experiments:

- **Mockup** data are generated with the aim of representing simple situations where a part of the factors are overlapped and where it is possible to control if the root cause is one of the overlapping factors. A mockup dataset generator was developed, allowing the control of (i) the number of products and steps in the manufacturing process, (ii) the percentage of factors overlapping with each other, and (iii) if the root cause is an overlapping factor or not. Each dataset represents a problematic moment, and its analysis does not require problematic moment identification;
- **Stochastic Simulation** data are generated using a stochastic simulator, which aims to emulate the data from a case study in semiconductor manufacturing. The aim of using these datasets is to use data as similar to a real case as possible, but where we are still able to control the locations of root causes. Contrary to the mockup datasets, it is not possible to preemptively establish overlap as a parameter. The overlap surges spontaneously through the emulation of a manufacturing process. In these experiments, the proposed method is validated and compared to another factor-ranking algorithm, as well as a classifier. A total of three datasets were used. These simulations' datasets emulate different conditions regarding the presence of noise. Dataset 1 represents a medium level of noise, while dataset 2 has low noise. Finally, the third dataset has a high noise level. The noise is included in the dataset through the labeling process. As products are labeled as problematic or not, a certain amount of products are mislabeled (either they were problematic without passing through the root cause, or were normal despite having passed through a root cause). The higher the noise, the greater the amount of mislabeled products.

- **Real Case-Study** data were provided by a semiconductor manufacturing company. The objective when testing these data is two-fold: (i) to illustrate that overlap is indeed a real problem that occurs when analyzing data from a manufacturing process, and (ii) to validate the proposed algorithm with real data. The actual root causes were not identified by the company used as a case study, and as such, validation of the RCA algorithms is carried out by comparing their results and verifying if there is a convergence in the proposed lists of most likely root causes.

The proposed method (denominated PMI) is compared with the Co-Ocurrences (CO) algorithm (a factor ranking algorithm proposed in [32]) and a Decision Tree (DT) classifier. The CO algorithm has no parameters to be tuned, while the DT classifier has its parameters (minimum number of instances per split and a confidence factor) tuned in each dataset using grid search and 5-fold cross-validation. The idea behind the CO algorithm is to compare the proportion of products with problems that went through a certain step–machine tuple with the proportion of problematic products that did not go through that tuple. If a tuple has a greater proportion of problematic products that went through it than normal ones, it is likely a root cause. The Co-Occurrences (CO) algorithm is shown in Algorithm 2.

Algorithm 2: Co-Ocurrences algorithm

```
Input: Dataset of problematic moment
Output: Ordered list of most likely root causes
begin
    Output ← [∅];
    To-use-dataset ← OneHotEncoding(Input);
    for Each Step-Machine Tuple do
        Construct Co-Occurrences table;
        Compute importance;
        if importance > 0.5 then
            Output ← [Output, (Step-Machine Tuple, importance)];
        end
    end
    Sort Output;
end
```

The input is the dataset of problematic moments identified in the first stage of the proposed method. The output is a list with two columns, one that identifies the tuple and another that identifies its importance in terms of co-occurrence with the label. The output is initialized as an empty list. The dataset is encoded using one-hot encoding to group the information in the dataset by the step–machine combination, instead of just factors. Then, for each step-machine combination, a table such as Table 4 is constructed. Each line represents the counts of products that have gone through a specific tuple ("Yes"), or not ("No"), and each column represents whether the products had problems (NP/YP) or not (NN/YN). For each step–machine combination tuple, a table is constructed.

Table 4. Table that serves as the basis for comparing the proportion of problems in a step–machine tuple [32].

Step-Machine Tuple	Product Quality—Problematic	Product Quality—Normal
Passed through this tuple?—No	NP	NN
Passed through this tuple?—Yes	YP	YN

The importance of each combination is calculated using Expression (6):

$$\text{importance} = \frac{\frac{YP - YN}{YP + YN} - \frac{NP - NN}{NP + NN}}{2} \tag{6}$$

If the importance of the tuple is above 0.5, the Output list stores the tuple and its importance. After all tuples are evaluated, the list is sorted in descending order of importance.

We consider that this combination of simulated data (from the two initial experiments) and real data (from the final experiment), with growing complexity, can provide a robust validation. In the final experiment, using real-case data, it was possible to identify 16 moments.

5.2. Evaluation Results

5.2.1. Mockup

This section focuses on evaluating the effect of overlap on the performance of both factor ranking algorithms and the classifier used. A total of 250 mockup datasets were generated, and the results are presented in Figures 4 and 5. The RC detection is the percentage of datasets where the root cause was correctly identified by the algorithm. The percentage of overlapped columns represents the percentage of tuples that overlapped with each other. The capacity for root-cause detection of the different factor-ranking algorithms (CO and PMI) and the classifier algorithm (DT) is evaluated in two different scenarios, i.e., one where the root cause is one of the overlapping factors, and another where it is not.

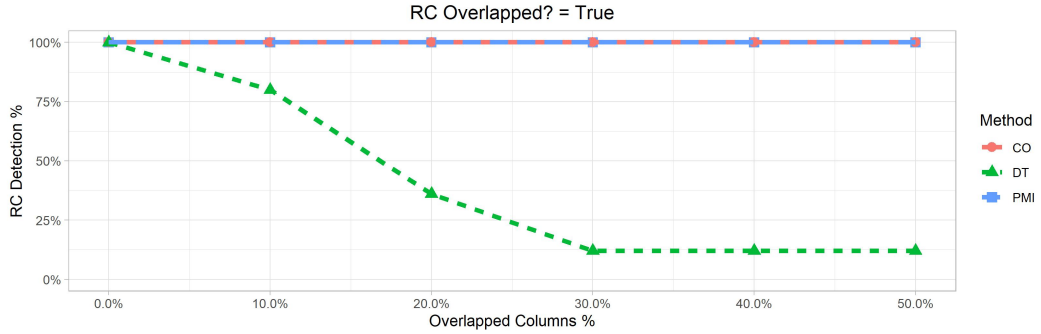


Figure 4. Results of the Mockup experiments when the root cause is in one of the overlapping factors.

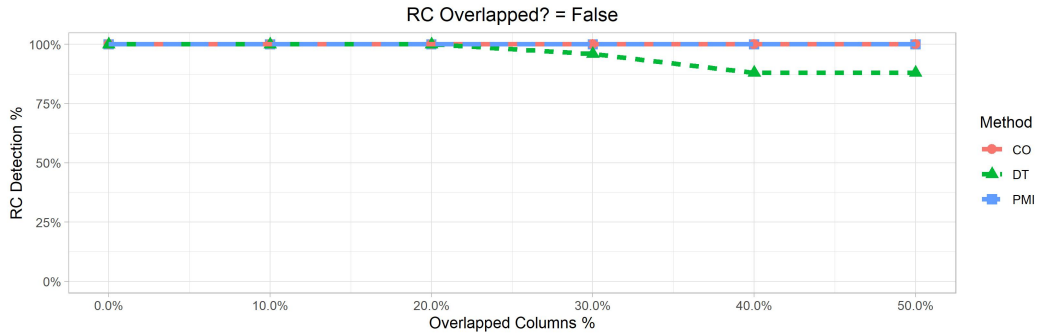


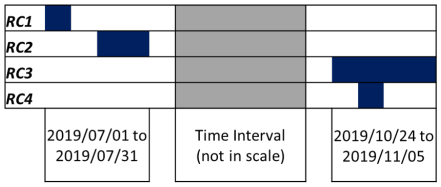
Figure 5. Results of the Mockup experiments when the root cause is not in one of the overlapped factors.

It is possible to see that, both when the root cause is in one of the overlapped factors and when it is not, the PMI algorithm is able to identify a small group of factors that contain the root cause in all datasets. This is in line with the best performance of the CO algorithm, which achieved the same result (the graphs for CO and PMI overlap each other in the figure).

Regarding the results of the DT classifier, its performance declines in the presence of overlap, especially when the root cause is in one of the overlapping factors.

5.2.2. Stochastic Simulation

This section focuses on evaluating the performance of the proposed solution as a whole. The root causes included in the simulated data are depicted in Figure 6. Each dataset corresponds to a period of four months. On the left side of the figure, we can see the temporal distribution of the different root causes. The first two root causes are located at the beginning of the simulated period, while the last two are at the end. This concentration of root causes was (i) to mirror the imbalance between normal products (the vast majority) and problematic products (a small minority) that exist in real data, and (ii) to simulate the simultaneous occurrence of root causes, which may also occur in real datasets. The right side of the figure explains in more detail each root cause and the reason why it was used. RC1 is a failure in part of equipment or a machine. RC2 is a standard malfunction in a single machine when performing a single function (or step). RC3 is a simultaneous failure that occurs at the same time as RC4. This last root cause is a failure that happens when a machine is malfunctioning in all its functions/steps.



Root Cause	Reason	Step	Equipment
RC1	Sub-Equipment in single Step	Step01	Equip03.2
RC2	Equipment in single Step	Step04	Equip14
RC3	Two RC at same time	Step20	Equip60
RC4	One faulty Equipment performing several Steps	All	Equip63

Figure 6. Root causes defined in the simulated datasets [32].

Table 5 presents the results of the RCA algorithms used, divided by dataset, and each moment where it is possible to detect a problem. Each column represents the root causes detected, and in brackets, the position in the ranking of variables. The code of the root causes is explained in Figure 6. Ties among positions are signaled with "*". False positives in the DT column are signaled with an FP. Each time a root cause was correctly detected, it is presented. In the case of CO and PMI, the ranking of the factor is also presented. In the ranking, ties between factors may occur, which means that the root cause was not the only factor identified as having the same importance, meaning that the root cause was not isolated. As such, we signal the ties with "*". For the DT classifier, we indicate the false positives/incorrect detection, in addition to the root causes. These false positives are presented as "FP". The reasons for doing so is (i) due to the classifier not providing a ranking and (ii) to have an improved understanding of the false detections that occurred.

Table 5. Table with the results of the RCA algorithms (FP—False Positive).

Dataset	Moment	$Overlap_{PMI}$	CO	PMI	DT
1	*1*	4.20%	RC2 (1st); RC1 (12th)	**RC2 (1st); RC1 (4th)**	RC2
1	*2*	0.05%	**RC2 (1st)**	**RC2 (1st)**	RC2; FP; FP
1	*3*	16.11%	**RC3 (1st) ***	**RC3 (1st) ***	FP
1	*4*	0.16%	**RC3 (1st); RC4 (2nd) ***	RC3 (1st)	RC3; FP;FP
2	*1*	14.24%	RC2 (6th) *	**RC2 (2nd) ***	FP
2	*2*	0.10%	**RC2 (1st)**	**RC2 (1st)**	RC2; FP
2	*3*	0.00%	**RC3 (1st)**	**RC3 (1st)**	RC3; FP; FP
2	*4*	0.67%	**RC3 (1st)**	**RC3 (1st)**	**RC3**
2	*5*	0.22%	RC3 (6th)	**RC3 (1st)**	FP; RC3
3	*1*	3.60%	RC2 (1st)	**RC2 (1st); RC1 (10th)**	RC2
3	*2*	0.00%	**RC2 (1st)**	**RC2 (1st)**	RC2; FP
3	*3*	18.38%	**RC3 (1st) ***	**RC3 (1st) ***	FP
3	*4*	0.61%	**RC3 (1st) ***	**RC3 (1st) ***	RC3, RC4; FP

When considering the performance of the algorithms, it is possible to see that the PMI algorithm has, overall, a better performance than the CO algorithm. Examples of this improved performance are dataset 1, moment 1; dataset 2, moments 1 and 5. In these examples, the PMI algorithm is able to detect what is detected by the CO algorithm but does so while placing the true root causes in better positions.

5.2.3. Real Case Study

The experiments with data from a case study have the objective of further validating the use of the proposed algorithm in real data. As mentioned in Section 5.1, due to the lack of information on the real root causes, a more robust validation is attempted by comparing the results of each algorithm and verifying whether there exists a convergence in the results, which would strongly indicate a root cause. The results of this experiment are expressed in Table 6. Only the steps of the step–machine tuples are shown to facilitate reading and due to a confidentiality agreement with the company used for the case study.

Table 6. Table summarizing the problematic moments identified in the case study dataset, as well as the common root causes among the different algorithms proposed.

Moment	Begin	End	Overlap$_{PMI}$	CO	PMI	DT
1	60	240	0.96%	COATFUSE	-	COATFUSE
2	312	360	4.74%	PLATINGRDL	PLATINGRDL	PLATINGRDL
3	420	516	1.25%	EXPOSEFUSE	EXPOSEFUSE	EXPOSEFUSE
4	516	660	1.67%	EXPOSEFUSE	EXPOSEFUSE	EXPOSEFUSE
5	672	720	15.88%	CUREFUSE	CUREFUSE	CUREFUSE
6	876	960	1.89%	COATRDL	COATRDL	COATRDL
7	948	1656	0.67%	EXPOSERDL	-	EXPOSERDL
8	1656	1752	3.12%	-	AOI1DL1	AOI1DL2
9	1860	1908	9.16%	-	-	-
10	2484	2592	1.50%	-	PLATINGRDL	PLATINGRDL
11	2580	3096	0.79%	-	AOI1DL2	AOI1DL2
12	3132	3240	1.45%	-	-	CLEANTOPWLB
13	3228	3432	0.74%	AOI1DL2	AOI1DL2	AOI1DL2
14	3456	3492	9.87%	-	AOI1DL2	-
15	3480	3888	0.66%	-	-	-
16	3888	3984	1.54%	-	EXPOSERDL	-

The threshold used for PMI is fixed at 0.3, as in dataset number 3 of the simulation experiments. Even with such a low threshold, there are many moments when the PMI algorithm is not able to detect any root cause (four in total). However, such is the case with the CO algorithm as well.

When considering the moments identified with detection, it is possible to see that there is indeed a convergence towards a common root cause, even in moments 14 and 16, where, of the three listed, only the PMI algorithm is able to detect a root cause.

6. Discussion

In this section, we discuss and summarise the results of the different experiments and how these combine to form coherent conclusions.

In the Mockup experiments, the proposed PMI method was able to achieve a performance equal to the best solutions based on the previous overlap measure (CO) and a much better performance than the DT classifier. Through a comparative analysis of Figures 4 and 5, we can see that an increment in the amount of overlap leads to degradation in the performance of the classifier. This is evidence in favor of the argument presented in Section 2.

In the experiments with the Stochastic simulation data, PMI achieved better performance than the other algorithms, as it put the true root causes in the same or higher rankings than the CO method. When applying the different algorithms to the real case study, the proposed method was found to be able to identify most root causes in common with other methods.

An additional note on the experiments with the Stochastic simulation data is that, to enable the detection of tuples that overlapped with the Label, there was a need to lower the threshold (of Expression 5) to 0.8 for the first two datasets and 0.3 for the last dataset. As these datasets have greater levels of noise, this seems to indicate that the threshold needs to be adapted to the datasets at hand and that the noisier the dataset is, the lower the threshold needs to be to detect tuples overlapping with the label. The default threshold value of 0.9 was chosen so as to include instances with very high overlap but with some leeway for noise. The fact that we needed to lower the threshold in response to an increase in noise indicates that there is a reduction in the signal-to-noise ratio in terms of evidence of the presence of the root cause in the data. While overlap may be less evident due to the presence of noise, so is the root cause signal.

From the results of all the experiments, we can clearly reach the conclusion that the proposed PMI algorithm has better performance than both the classifier solution and the CO algorithm.

A relevant shortcoming of this work is that it was not possible to establish the root causes in the case study data, which hinders the validation in real data. This work would be improved if it was possible to access real data with the root causes identified. Another avenue for future work could be analyzing the effect of noise on the performance of the factor-ranking algorithms, as it is not clear what is causing the differences in the performance of the algorithms (although it is clear that they are resilient to overlap). Finally, it would be interesting to see if it is possible to find a method that is able to untangle the statistical impossibility of distinguishing between overlapped factors.

7. Conclusions

This work presents a new measure of overlap. Overlap is a synchronization in the manufacturing process that makes all products that pass through a given machine in a certain step also pass through another specific machine in a step further ahead in the process. With data generated in the presence of this phenomenon, it becomes impossible to discern the influence of the machines that processed all these products on the products' quality .

We propose a novel measure of overlap that uses information theory concepts such as Positive Mutual Information (PMI). This measure considers two critical aspects, namely whether the associations are positive or negative, and whether it is appropriate to detect overlap among step-machine tuples. This measure is the basis of a factor-ranking algorithm that is used to detect root causes, in an ARCA solution.

To validate this new method, three experiments were conducted: (i) using mockup data, (ii) using simulated data that emulates a case study, (iii) using real data from the case study itself. It was possible to conclude that the proposed algorithm achieved better performance with simulated data, competing with the benchmark algorithms in the other two experiments.

This paper contributes to the literature by presenting a robust measure of overlap, which allows for a better understanding and analysis of this characteristic of the problem. In addition, a new factor-ranking algorithm is presented with positive results. A visualization tool was also developed that eases the analysis by practitioners, by depicting the relevant overlaps between tuples in a manipulable graph.

This work allows researchers and practitioners to have an improved comprehension of a new concept, which can lead to the development of improved ARCA solutions, making the management of manufacturing operations faster and reducing the associated workload.

Author Contributions: Conceptualization, E.e.O.; methodology, E.e.O.; software, E.e.O.; validation, E.e.O., V.L.M. and J.L.B.; formal analysis, E.e.O.; investigation, E.e.O.; data curation, E.e.O.; writing—original draft preparation, E.e.O.; writing—review and editing, E.e.O. and V.L.M.; visualization, E.e.O. and J.L.B.; supervision, V.L.M. and J.L.B.; project administration, E.e.O., V.L.M. and J.L.B.; funding acquisition, E.e.O., V.L.M. and J.L.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Funds through the Portuguese funding agency, FCT—Fundação Ciência e Tecnologia within the project SFRH/BD/138228/2018.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Mockup and Stochastic Simulation data are available from the corresponding author upon reasonable request. Real Case-Study data are not available due to commercial restrictions.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Sun, J.; Xi, L.; Pan, E.; Du, S.; Xia, T. Design for diagnosability of multistation manufacturing systems based on sensor allocation optimization. *Comput. Ind.* **2009**, *60*, 501–509. [CrossRef]
2. Deng, Y.; Huang, D.; Du, S.; Li, G.; Zhao, C.; Lv, J. A double-layer attention based adversarial network for partial transfer learning in machinery fault diagnosis. *Comput. Ind.* **2021**, *127*, 103399. [CrossRef]
3. Shiau, Y.R.; Wang, S.Y. Key improvement decision analysis mechanism based on overall loss of a production system. *J. Ind. Prod. Eng.* **2021**, *38*, 66–73. [CrossRef]
4. Jabrouni, H.; Kamsu-Foguem, B.; Geneste, L.; Vaysse, C. Analysis reuse exploiting taxonomical information and belief assignment in industrial problem solving. *Comput. Ind.* **2013**, *64*, 1035–1044. [CrossRef]
5. Du, S.; Lv, J.; Xi, L. A robust approach for root causes identification in machining processes using hybrid learning algorithm and engineering knowledge. *J. Intell. Manuf.* **2012**, *23*, 1833–1847. [CrossRef]
6. Tarakci, H. Two types of learning effects on maintenance activities. *Int. J. Prod. Res.* **2016**, *54*, 1721–1734. [CrossRef]
7. Sahoo, S. Big data analytics in manufacturing: a bibliometric analysis of research in the field of business management. *Int. J. Prod. Res.* **2021**, 1–29. [CrossRef]
8. Wee, Y.Y.; Cheah, W.P.; Tan, S.C.; Wee, K. A method for root cause analysis with a Bayesian belief network and fuzzy cognitive map. *Expert Syst. Appl.* **2015**, *42*, 468–487. [CrossRef]
9. Tan, C.M.; Chen, H.H.; Wu, J.P.; Sangwan, V.; Tsai, K.Y.; Huang, W.C. Root Cause Analysis of a Printed Circuit Board (PCB) Failure in a Public Transport Communication System. *Appl. Sci.* **2022**, *12*, 640. [CrossRef]
10. Steinhauer, H.J.; Karlsson, A.; Mathiason, G.; Helldin, T. Root-cause localization using Restricted Boltzmann Machines. In Proceedings of the 2016 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, 5–8 July 2016; pp. 248–255.
11. Agarwal, K.; Shivpuri, R. Knowledge discovery in steel bar rolling mills using scheduling data and automated inspection. *J. Intell. Manuf.* **2014**, *25*, 1289–1299. [CrossRef]
12. Yan, R.; Chen, X.; Wang, P.; Onchis, D.M. Deep learning for fault diagnosis and prognosis in manufacturing systems. *Comput. Ind.* **2019**, *110*, 1–2. [CrossRef]
13. Lechevalier, D.; Narayanan, A.; Rachuri, S.; Foufou, S. A methodology for the semi-automatic generation of analytical models in manufacturing. *Comput. Ind.* **2018**, *95*, 54–67. [CrossRef]
14. Papacharalampopoulos, A.; Giannoulis, C.; Stavropoulos, P.; Mourtzis, D. A digital twin for automated root-cause search of production alarms based on KPIs aggregated from IoT. *Appl. Sci.* **2020**, *10*, 2377. [CrossRef]
15. Chien, C.F.; yen Hong, T.; Guo, H.Z. A Conceptual Framework for "Industry 3.5" to Empower Intelligent Manufacturing and Case Studies. *Procedia Manuf.* **2017**, *11*, 2009–2017. [CrossRef]
16. Chien, C.F.; Lin, Y.S.; Lin, S.K. Deep reinforcement learning for selecting demand forecast models to empower Industry 3.5 and an empirical study for a semiconductor component distributor. *Int. J. Prod. Res.* **2020**, *58*, 2784–2804. [CrossRef]
17. Ku, C.C.; Chien, C.F.; Ma, K.T. Digital transformation to empower smart production for Industry 3.5 and an empirical study for textile dyeing. *Comput. Ind. Eng.* **2020**, *142*, 106297. [CrossRef]
18. Bennacer, L.; Amirat, Y.; Chibani, A.; Mellouk, A.; Ciavaglia, L. Self-Diagnosis Technique for Virtual Private Networks Combining Bayesian Networks and Case-Based Reasoning. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 354–366. [CrossRef]

19. Xu, Z.; Dang, Y. Automated digital cause-and-effect diagrams to assist causal analysis in problem-solving: A data-driven approach. *Int. J. Prod. Res.* **2020**, *58*, 5359–5379. [CrossRef]
20. e Oliveira, E.; Miguéis, V.L.; Borges, J.L. Understanding Overlap in Automatic Root Cause Analysis in Manufacturing Using Causal Inference. *IEEE Access* **2022**, *10*, 191–201. [CrossRef]
21. Razouk, H.; Kern, R. Improving the Consistency of the Failure Mode Effect Analysis (FMEA) Documents in Semiconductor Manufacturing. *Appl. Sci.* **2022**, *12*, 1840. [CrossRef]
22. Zhu, Y.J.; Guo, W.; Liu, H.C. Knowledge Representation and Reasoning with an Extended Dynamic Uncertain Causality Graph under the Pythagorean Uncertain Linguistic Environment. *Appl. Sci.* **2022**, *12*, 4670. [CrossRef]
23. e Oliveira, E.; Miguéis, V.L.; Borges, J.L. Automatic root cause analysis in manufacturing: An overview & conceptualization. *J. Intell. Manuf.* **2022**. [CrossRef]
24. Rokach, L.; Hutter, D. Automatic discovery of the root causes for quality drift in high dimensionality manufacturing processes. *J. Intell. Manuf.* **2012**, *23*, 1915–1930. [CrossRef]
25. Donauer, M.; Peças, P.; Azevedo, A. Identifying nonconformity root causes using applied knowledge discovery. *Robot. -Comput.-Integr. Manuf.* **2015**, *36*, 84–92. [CrossRef]
26. Chen, Z.; Liu, Y.; Valera-Medina, A.; Robinson, F.; Packianather, M. Multi-faceted modelling for strip breakage in cold rolling using machine learning. *Int. J. Prod. Res.* **2020**, *59*, 1–14. [CrossRef]
27. Saez, M.A.; Maturana, F.P.; Barton, K.; Tilbury, D.M. Context-sensitive modeling and analysis of cyber-physical manufacturing systems for anomaly detection and diagnosis. *IEEE Trans. Autom. Sci. Eng.* **2019**, *17*, 29–40. [CrossRef]
28. Sun, Y.; Qin, W.; Zhuang, Z.; Xu, H. An adaptive fault detection and root-cause analysis scheme for complex industrial processes using moving window KPCA and information geometric causal inference. *J. Intell. Manuf.* **2021**, *32*, 2007–2021. [CrossRef]
29. Chemweno, P.; Pintelon, L.; Jongers, L.; Muchiri, P. i-RCAM: Intelligent expert system for root cause analysis in maintenance decision making. In Proceedings of the 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), Ottawa, ON, Canada, 20–22 June 2016; pp. 1–7. [CrossRef]
30. Sun, Z.H.; Liu, R.; Ming, X. A Fault Diagnosis and Maintenance Decision System for Production Line Based on Human-Machine Multi- Information Fusion. In Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference, Tokyo, Japan, 21–23 December 2018; Association for Computing Machinery: New York, NY, USA, 2018; AICCC' 18; pp. 151–156. [CrossRef]
31. Lima, A.; Borodin, V.; Dauzère-Pérès, S.; Vialletelle, P. A sampling-based approach for managing lot release in time constraint tunnels in semiconductor manufacturing. *Int. J. Prod. Res.* **2021**, *59*, 860–884. [CrossRef]
32. e Oliveira, E.; Miguéis, V.L.; Borges, J.L. On the influence of overlap in automatic root cause analysis in manufacturing. *Int. J. Prod. Res.* **2022**, *60*, 6491–6507. [CrossRef]
33. Brun, A.; Castagnos, S.; Boyer, A. A positively directed mutual information measure for collaborative filtering. In Proceedings of the 2nd International Conference on Information Systems and Economic Intelligence—SIIE 2009, Hammamet, Tunisia, 12–14 February 2009; pp. 943–958.
34. Hsu, C.Y.; Lin, S.C.; Chien, C.F. A back-propagation neural network with a distributed lag model for semiconductor vendor-managed inventory. *J. Ind. Prod. Eng.* **2015**, *32*, 149–161. [CrossRef]
35. Chen, W.C.; Tseng, S.S.; Wang, C.Y. A novel manufacturing defect detection method using association rule mining techniques. *Expert Syst. Appl.* **2005**, *29*, 807–815. [CrossRef]
36. Zanon, M.; Susto, G.A.; McLoone, S. Root Cause Analysis by a Combined Sparse Classification and Monte Carlo Approach. *IFAC Proc. Vol.* **2014**, *47*, 1947–1952. [CrossRef]
37. Fan, S.K.S.; Lin, S.C.; Tsai, P.F. Wafer fault detection and key step identification for semiconductor manufacturing using principal component analysis, AdaBoost and decision tree. *J. Ind. Prod. Eng.* **2016**, *33*, 151–168. [CrossRef]
38. Chiang, L.H.; Jiang, B.; Zhu, X.; Huang, D.; Braatz, R.D. Diagnosis of multiple and unknown faults using the causal map and multivariate statistics. *J. Process Control.* **2015**, *28*, 27–39. [CrossRef]
39. Sim, H.; Choi, D.; Kim, C.O. A data mining approach to the causal analysis of product faults in multi-stage PCB manufacturing. *Int. J. Precis. Eng. Manuf.* **2014**, *15*, 1563–1573. [CrossRef]
40. Lee, C.Y.; Chien, C.F. Pitfalls and protocols of data science in manufacturing practice. *J. Intell. Manuf.* **2020**, *33*, 1189–1207. [CrossRef]
41. Detzner, A.; Eigner, M. Feature selection methods for root-cause analysis among top-level product attributes. *Qual. Reliab. Eng. Int.* **2021**, *37*, 335–351. [CrossRef]
42. Gu, Y.K.; Zhang, J.; Shen, Y.J.; Fan, C.J. Fault tree analysis method based on probabilistic model checking and discrete time Markov Chain. *J. Ind. Prod. Eng.* **2019**, *36*, 146–153. [CrossRef]
43. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
44. Lv, F.; Yu, S.; Wen, C.; Principe, J.C. Interpretable fault detection using projections of mutual information matrix. *J. Frankl. Inst.* **2021**, *358*, 4028–4057. [CrossRef]
45. Bennasar, M.; Hicks, Y.; Setchi, R. Feature selection using Joint Mutual Information Maximisation. *Expert Syst. Appl.* **2015**, *42*, 8520–8532. [CrossRef]
46. Hoque, N.; Bhattacharyya, D.; Kalita, J. MIFS-ND: A mutual information-based feature selection method. *Expert Syst. Appl.* **2014**, *41*, 6371–6385. [CrossRef]

47. Bi, N.; Tan, J.; Lai, J.H.; Suen, C.Y. High-dimensional supervised feature selection via optimized kernel mutual information. *Expert Syst. Appl.* **2018**, *108*, 81–95. [CrossRef]
48. Roberts, S.W. Control Chart Tests Based on Geometric Moving Averages. *Technometrics* **1959**, *1*, 239–250. [CrossRef]