Understanding Overlap in Automatic Root Cause Analysis in Manufacturing Using Causal Inference

EDUARDO E. OLIVEIRA, VERA L. MIGUÉIS, AND JOSÉ L. BORGES
INESC TEC, Faculdade de Engenharia da Universidade do Porto—Rua Dr. Roberto Frias Edificio I, 4200-465 Porto, Portugal
Corresponding author: Eduardo E. Oliveira (eduardo.l.oliveira@inesctec.pt)
This work was supported by the National Funds through the Portuguese Funding Agency FCT—Fundação Ciência e Tecnologia within Project SFRH/BD/138228/2018.

ABSTRACT
Overlap has been identified in previous works as a significant obstacle to automated diagnosis using data mining algorithms, since it makes it impossible to discern how each machine influences product quality. Several solutions that handle overlap have been proposed, but the final result is a list of potential overlapped root causes. The goal of this paper is to develop a solution resilient to overlap that can determine the true root cause from a list of possible root causes, when possible, and determine the conditions in which it is possible to identify the root causes. This allows for a better understanding of overlap, and enables the development of a fully automatic root cause analysis for manufacturing. To do so, we propose an automatic root cause analysis approach that uses causal inference and do calculus to determine the true root cause. The proposed approach was validated on simulated and real case-study data, and allowed for an estimation of the effect of a product passing through a certain machine while disregarding the effect of overlap, in certain conditions. The results were on par with the state-of-the-art solutions capable of handling overlap. The contributions of this paper are a graphical definition of overlap, the identification of the conditions in which it is possible to overcome the effect of overlap, and a solution that can present a single true root cause when such conditions are met.

INDEX TERMS
Causal inference, fault diagnosis, manufacturing, root cause analysis.

I. INTRODUCTION
Manufacturing is highly competitive [8] and managing manufacturing operations can be very complex. This complexity is increasing [10] as new measures are adopted that lead to a data-intensive environment [1], [22]. Manufacturing companies should solve their operational problems efficiently and permanently in order to remain competitive. Root Cause Analysis (RCA) enables companies to do so by allowing them to focus on the origin of problems rather than on their symptoms [18]. The goal of RCA is to determine the causal mechanism behind a change from a desirable state to an undesirable one in order to ultimately keep a problem from recurring [14]. Robust RCA solutions are required [3] since diagnosing problems is very important for safe and efficient operations [25]. However, RCA can be very complicated, requiring extensive system and execution analysis [23]. As such, it is not trivial to perform RCA in manufacturing.

To increase the efficacy and the efficiency of RCA, several studies developed solutions that take advantage of the increasing volume of data generated in manufacturing environments [8], [22]. These solutions, named Automatic Root Cause Analysis (ARCA), use data mining and machine learning algorithms to automatically search for patterns in data that allow analysts to detect the root causes of problems more efficiently. Considering the framework of Industry 3.5, an intermediate stage between Industry 3.0 and to-be Industry 4.0 [7], ARCA solutions aim to improve digital decisions through data analysis.

The root cause detection problem can be arranged in three different levels: i) as a location of the root cause; ii) as the physical characteristics of the root cause; and iii) as the human/organizational characteristics of the root cause. The first level allows us to determine where the root cause is,
i.e., its location within the process. Examples of studies that deal with this type of data are [19] and [9]. The second level focuses on which physical occurrences are the root cause (e.g., extraordinary increases in pressure, high voltages). Examples of studies that focus on this kind of data are [10] and [20]. The third level focuses on the human and organizational characteristics of the root cause (e.g., equipment maintenance) to identify what triggered the physical occurrences. Both [4] and [24] are examples of studies focusing on this level of detection. In this paper, we focus on the first level, i.e., the location type data.

This paper focuses on the use of location type data. Location type data, despite being the least detailed level of data, is still worth analyzing. Some factories do not have the necessary infrastructure to collect the data required for the development of ARCA solutions focused on the physical nature of root causes. However, most factories have data on how products move through the manufacturing process. This information can be used to determine the location of root causes. When the manufacturing process is particularly complex (e.g., in semiconductor manufacturing), even production movement data can become complex to analyze, which justifies the use of data mining and machine learning techniques in the development of diagnosis solutions.

A manufacturing process is a sequence of steps that products go through, beginning as raw materials or parts and ending as a finished product. In each step, the materials or parts are processed in machines. When trying to locate the root cause of a problem in a manufacturing step, the focus is on determining the step-machine pair (meaning that a product was processed in a certain machine in a certain step) that is originating the problem, i.e., the root cause.

In [11], a phenomenon named overlap was identified, that is a serious challenge for the development of ARCA solution using location type data. An overlap between two tuples happens when there is a very high probability (close to 1) of a product going through a tuple given that it has also gone through another tuple. The occurrence of overlap makes it very difficult, or even impossible, to distinguish the influence of two different tuples on the quality of a product, as the patterns of each tuple “overlap” each other (hence the name).

Overlap can occur because of stabilization in the manufacturing process. An example of this is a process where, as soon as certain products finish processing in a step in a particular machine, those products always have the same machine available in the next step, which became available at that moment. This situation is beneficial in terms of the productivity and efficiency of the manufacturing process, but it becomes detrimental when analyzing the data resulting from such a process. This conflict between what is beneficial for production (that should be the priority) and what is beneficial for diagnosis through data analysis is noteworthy, as it means that overlap is an issue that will occur frequently whenever we analyze location type data for performing diagnosis.

Overlap is critical when there is only location data available but the data still has high dimensionality. In this scenario, traditional solutions (e.g., Pareto charts, fish-bone diagrams) are unable to deal with the high dimensionality of the data, and as such, ARCA solutions are required to efficiently obtain the location of root causes. Regarding ARCA solutions, identifying the presence of overlap provides insight to the analyst by warning about the presence of a phenomenon that degrades the analysis through data mining and machine learning algorithms, preventing them from reaching wrong conclusions and promoting the use of more robust solutions to locate true root causes.

Previous works on overlap in ARCA in manufacturing were able to provide a list of most likely root causes and identify groups of overlapped machines. However, they were still not able to determine the true root cause from the group of most likely root causes. As such, there is a research gap of not being able to identify the true root cause from a group of most likely root causes, in situations where overlap is present.

Therefore, the main aim of this paper is to develop a solution that can identify the true root cause amid a group of overlapped possible root causes. If the presence of overlap makes it impossible, it should be able to identify the conditions on which it is possible to detect the true root cause.

To tackle this issue, we propose a methodology based on causal inference and do calculus, and develop an approach that can determine the true root cause from a group of possible root causes. These two methods enable us to estimate, in certain conditions, the effect of a product passing through a certain location while disregarding the effect of overlaps. This new approach led to the development of a causal inference model that allowed us to study the effect of overlap in more detail. This, in turn, allowed us to determine the conditions when it is possible to determine the root cause tuple of a given problem and to determine the true root cause when such conditions are met. Other examples of the use of causal inference for RCA are [5] and [13].

The contributions of this paper are: i) a graphical definition and causal model of overlap; ii) the identification of the conditions required to determine the true root cause when overlap is present; iii) an ARCA solution based on causal inference that can identify the true root cause from a list of possible overlapped root causes.

The remainder of the paper is structured as follows. In Section II, we present the background theory behind this study. First, we define the problem and briefly discuss the previous works in ARCA in manufacturing. Then, we briefly present concepts of causal inference. In Section III, we describe the methodology used, explaining the causal model developed and the probabilities computed, as well as establishing the theoretical limits to the detection of root causes among overlapped tuples. In Sections IV-A and IV-B, we explain and show (respectively) the results of the experimental procedures taken to validate the proposed methodology. In Section V, we examine the results. To conclude, we present a summary of the findings and discuss future research directions.
II. BACKGROUND THEORY

A. OVERLAP IN AUTOMATIC ROOT CAUSE ANALYSIS

Overlap can occur when analyzing data to locate the root cause of a problem in a manufacturing process. A manufacturing process is defined by a sequence of steps that products go through. In each of these steps, products are transformed or assembled in a certain machine. A machine may be able to perform more than one step. A step-machine pair is called a tuple. After all steps are completed, the quality of the product is evaluated. Figure 1 illustrates such a manufacturing process, where the products $P_i \ (i = 1, 2, 3)$ are flowing through the steps of the process. In this illustrative example of a manufacturing process with three steps, $P_1$ was already processed and is being monitored, $P_2$ is being processed in Machine 3, and $P_3$ is waiting to be processed before Step B. The root cause of the problem is in Step A, Machine 1.

![Figure 1. Illustration of a process that generates data that can be used to find the location of a root cause. The manufacturing process is composed of manufacturing steps, in which a product can be processed in several machines. At the end of the process, the product quality is monitored.](image)

Table 1 represents the data generated by the example in Figure 1. In each step (column), the machine used is stored. The “Problem” column stores whether the product was problematic or not.

| Product | Step A | Step B | Step C | Problem |
|---------|--------|--------|--------|---------|
| 1       | Machine 1 | Machine 3 | Machine 6 | 1       |
| 2       | Machine 2 | Machine 4 | Machine 7 | 0       |
| 3       | Machine 2 | Machine 5 | Machine 6 | 0       |
| 4       | Machine 1 | Machine 3 | Machine 7 | 1       |
| ...     | ...     | ...     | ...     | ...     |

The quality of a product can be determined as problematic (high criticality) or normal (low criticality). If the number of problematic products increases above a defined threshold in a certain period, that means there is a problem in the manufacturing process. In that case, RCA is useful to locate the root cause, which is represented by a tuple. As described in [11], overlap happens when all the products that go through one tuple also go through another tuple, making it impossible to distinguish the influence of these tuples on the quality of the final product. If one of the tuples is the root cause, it cannot be found statistically, as it is “hidden” by being overlapped by the other tuple. Such is the case in Figure 1 and Table 1, where Step A-Machine 1 is overlapped with Step B-Machine 3. When overlap occurs, the best we can achieve is to select the tuples with the highest influence on quality and provide these to the analysts so that they can investigate the issue further and distinguish each tuple’s influence manually. It should be noted that overlap does not occur only in contiguous steps, and as such the whole manufacturing process has to be analyzed.

In general, ARCA solutions use classification algorithms to associate factors with problems and then extract the root causes from the knowledge structures (e.g., decision trees, rules, regression equations). However, the use of classification algorithms does not guarantee that the root cause is present in the knowledge structures. In other words, these algorithms may end up excluding certain tuples that have a high influence on quality. This occurs because the classifiers may select a given tuple to represent those that are correlated, excluding the others, which may “hide” the real root cause from the analyst. Thus, although classifiers may seem adequate to analyze a large amount of data, when trying to locate a root cause, they are not adequate. This is problematic in general, but it is especially critical in situations where overlap is present. For example, when analyzing data with the same structure as Table 1, a classifier will be biased towards factors with more levels and will choose Step B-Machine 3 as the root cause. In these cases, the best possibility is to identify all the overlapped tuples and let the analyst investigate each step-machine individually.

From the previous literature, it is relevant to mention the studies that addressed and proposed ways of tackling the correlation between the factors that may be the root causes of a problem. Although these studies did not mention overlap explicitly, they are an indication that the issue of overlap exists but was not explicitly identified. [26] proposed the use of clustering to group highly correlated factors and selected a representative factor from the resulting group to use during the analysis. [12] presented a two-phase approach to identify faulty factors in semiconductor manufacturing. In the first phase, Principal Component Analysis was used for feature selection, which helps to separate normal products and faulty ones. In the second phase, classifiers were used to determine the factors responsible for the faults. [6] mentioned the difficulty in concluding whether a certain factor is the root cause due to the presence of multiple faults, and highlighted that the joint effect of multiple faults can be very different from the effect of individual faults. To address this, the authors combined data analysis with cause-and-effect information. In [21], the information on the combination of faults was considered using Bayesian networks and the most relevant factors were selected using Partial Least Squares with Variable Importance in Projection (PLS-VIP) so that the rules obtained contained only the required information, discarding correlated factors.

In a previous work (submitted for publication), we proposed a mathematical definition of overlap based on the concept of Positive Mutual Information (PMI), first used...
in [2]. PMI represents the amount of information about a factor that is possible to know from another factor but taking into consideration only the positive associations. Expression (1) represents the mathematical formulation of PMI, and Expression (2) represents how to compute a measure of overlap based on PMI.

\[
\text{PMI}(x, y) = \frac{2}{H(x) + H(y)} \left( p(x, y) \log \frac{p(x, y)}{p(x)p(y)} + p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \right),
\]

(1)

where \( p(x, y) \) represents positive values ("went through machine" and "is a problematic product", respectively), and \( x \) and \( y \) represent negative values ("did not go through machine" and "is not a problematic product", respectively).

\[
\text{Overlap}_{PMI} = \sum_{i} \sum_{j} C_{i,j} \frac{N_{VI}}{N_{VII}},
\]

(2)

where:

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

and \( N_{VI} \) corresponds to the number of valid interactions between factors. \( C_{i,j} \) is 1 when the PMI between the factors \( i \) and \( j \) is greater than the threshold \( \text{Th} \), and 0 otherwise. The numerator includes the total number of interactions between \( i \) and \( j \) above the threshold, where \( i \neq j \).

The rationale behind Expression (2) is that we compare tuples and check if the PMI of their interaction is above a certain threshold. Each interaction of tuples that surpasses a threshold \( \text{Th} \) is considered to be overlapping. To evaluate the amount of overlap in a dataset (Overlap\(_{PMI}\)), we count the number of interactions that are overlapped and divide the sum by the number of valid interactions, i.e., the total number of tuples that occur at least once in the same product (otherwise, the PMI expression would give an undefined result).

We also proposed an algorithm based on the above-mentioned mathematical definition to determine the list of possible root causes of a problem, while also representing overlapped tuples. This helps reduce the number of possibilities the analyst needs to explore, making the RCA process more efficient. However, it would be relevant for a solution to automatically identify the true root cause from this list of possible root cause locations. The aim of this paper is to study such a possibility and to propose a solution that is able to automate as much as possible the identification of true root cause locations by using causal inference.

B. CAUSAL INFERENCE

A brief review of causal inference concepts is presented in this section. For a more comprehensive read, please refer to [16]. Causal inference aims to identify cause-effect relationships among factors or events. As the aim of this paper is to find the root causes of problems, it makes sense to delve into how to adequately relate causes to effects (in this case, the effects are the problems). To do this, we focus on a general theory of causation based on the Structural Causal Model (SCM), which subsumes and unifies other approaches to causation and provides a coherent mathematical foundation for the analysis of causes and counterfactuals [17].

As the name indicates, the SCM requires a structure that makes it possible to identify the relations that remain invariant, which are the ones we are trying to study. Such a structure is provided by causal graphs/models, such as the example in Figure 2. The nodes represent variables and the directed edges represent the possibility of stable causal relationships, which indicate a physical relationship between variables. In this example, \( X \) is directly influenced by its parent \( U \), and in turn \( X \) directly influences its child \( W \). Also relevant are the missing edges: in this example, the lack of an edge between \( X \) and \( Y \) indicates an assumption that \( X \) does not influence \( Y \) directly. To relate this example with RCA, we could consider \( X \) and \( W \) as tuples that a product can go through, \( Y \) as the quality of the final product, and \( U \) as representing all unknown factors that could affect both \( X \) and \( Y \) but that we cannot measure. We can also translate a causal model into a functional causal model, where each relation is represented by a function, e.g., Expressions (3). These expressions detail exactly how one variable influences the other (while the edges only indicate that there is an influence).

FIGURE 2. Example of a causal graph/model. Each node defines a variable, and each directed edge represents a causal relation between two variables.

When considering the example in Figure 2, the directed edges represent the possibility of a stable causal relationship, which indicates a physical relationship between the variables. In this case, \( X \) is directly influenced by its parent \( U \), and in turn \( X \) directly influences its child \( Z \). Also relevant are the missing edges: in this example, the lack of an edge between \( X \) and \( Y \) indicates our assumption that \( X \) does not influence \( Y \) directly. We can also translate these causal models into functional causal models, where each relation is represented by a function, like the expressions below:

\[
\begin{align*}
U &= f_1(\epsilon_1) \\
X &= f_2(U, \epsilon_2) \\
W &= f_3(X, \epsilon_3) \\
Y &= f_4(U, W, \epsilon_4)
\end{align*}
\]

(3)
Regarding Expressions (3), \( f_i (f_1, f_2, f_3, \text{ and } f_4) \) can be any function. These functions represent the stable mechanisms through which a variable affects another, e.g., if \( U \) occurs, so does \( X \), or for each unit increase in \( U \), \( X \) increases by 0.5. In this example, we can see that the arguments of each function correspond to the parents of each variable, and a \( \epsilon_i \), which represents noise or uncertainty. These functions are necessary to answer counterfactual queries, while causal graphs are required by both interventional (see Section II-B1 for more details) and counterfactual queries (relating to “what would have happened if” questions).

1) INTERVENTIONAL QUERIES

An interventional query is a type of causal query where we compute how changing or intervening in a variable affects another variable. These queries assume the form of Expression (4).

\[
P(Y = y | \text{do}(X = x_i))
\]  

Expression (4) means the probability of \( Y = y \) knowing that we perform an intervention “forcing” \( X \) to assume the value of \( x_i \) (indicated by the \( \text{do}(\cdot) \) operator). This can also be interpreted as the causal effect of \( X \) on \( Y \). This probability is different from a regular conditional probability \( P(Y = y | X = x_i) \) because the latter assumes that no changes to the causal structure are made, while the intervention indicates that the influence of the parent variables is disconnected because the value of \( X \) is not influenced by its parents, but is “forced” upon the variable by the intervention. The new graph resulting from the intervention is depicted in Figure 3.

![Figure 3. Example of the causal graph in Figure 2 after the intervention \( \text{do}(X = x_i) \). The edge between variable \( U \) on \( X \) has disappeared, as \( X \)’s value is now defined by the intervention.](image)

As can be seen in Figure 3, \( X \) and \( U \) are no longer connected by an edge, and instead, the constant \( x_i \) is imposed on variable \( X \).

There are two possible ways to compute the effect of an intervention using observational data that are relevant for this work: i) adjusting for direct causes, and ii) the Back-Door (BD) criterion. The basic idea in both is that, if we control a set of variables that d-separates \( X \) and \( Y \), it is possible to compute the effect of the intervention using preintervention probabilities.

The adjustment for direct causes is based on the following: if we control all the direct causes (or parents) of the possible cause \( X \), we eliminate all spurious correlations that could influence \( X \). Ergo, if \( pa \) is the set of direct causes of \( X \), and \( Y \) is not equal to \( X \) or any \( pa_i \), the effect of the intervention \( \text{do}(X = x_i) \) on \( Y \) is given by:

\[
P(Y | \text{do}(X = x_i)) = \sum_{pa_i} P(Y | X = x_i, pa_i) P(pa_i),
\]

where \( P(Y | X = x_i, pa_i) \) and \( P(pa_i) \) represent preintervention probabilities [16].

The adjustment for direct causes works well in the scenarios where we know and can measure all parents of \( X \). However, this is not always possible. For example, in Figure 2, we may not know or be able to measure the variable \( U \). In such a case, it would not be possible to compute the effect of the intervention by adjusting for the direct causes. However, the parent variables of \( X \) are not the only set of variables that eliminate all spurious correlations that could influence \( X \). If we are able to find such a set \( Z \) that: i) no node in \( Z \) is a descendant of \( X \), and ii) \( Z \) blocks all the paths between \( X \) and \( Y \) that contain an arrow into \( X \), \( Z \) can replace the direct causes. This is called the BD criterion [16]. Furthermore, if a set \( Z \) satisfies the BD criterion relative to \( (X, Y) \), the causal effect of \( X \) on \( Y \) is identifiable and given by:

\[
P(Y | \text{do}(X = x_i)) = \sum_{z} P(Y | X = x_i, z) P(z),
\]

where \( P(Y | X = x_i, z) \) and \( P(z) \) represent preintervention probabilities [16]. Therefore, we can “replace” the control on the direct causes (that we may not know) with a control on a set \( Z \), from which all variables are observed, and that block all spurious correlations.

III. MATERIALS AND METHODS

Previous approaches to deal with overlap resulted in a list of the tuples that were the most likely root causes, in which overlapped tuples were represented as having the same influence on quality. The methodology proposed in this section uses the results of previous studies to identify and select relevant and overlapped tuples and then tries to identify the true root cause from among them. This requires a methodology that is able to estimate how a product passing through a specific tuple affects quality while disregarding the effect of overlap. As such, we propose the use of causal inference and \( do \), calculus, as these allow us to estimate these probabilities, conceptualize overlap, and understand when and how we can detect the true root cause from a set of overlapped tuples.

From the previous studies, the one with better results used the PMI approach. As such, this is the approach we chose to select the tuples \( X_i \) before trying to find the true root cause from among them.

The motivation behind the use of causal inference to analyze overlap comes from the idea that, if we can “disconnect” a certain tuple from the influence of overlap, we should be
able to see which of the tuples is the real root cause, because we would be able to disregard the influence of overlap. This evokes the notion of intervention (as seen in Section II-B1). The causal effect of a tuple $X_i$ on the label $Y$ is given by Expression (7). Considering binary variables (1 if the product passes through $X_i$ or the label $Y$ is problematic, and 0 if the product does not pass or the product is normal), this expression represents the probability of a product being problematic if we had “forced” the product to go through that tuple.

$$P(Y = 1|\text{do}(X_i = 1))$$ (7)

To compute this probability, we require a causal graph. The causal graph we propose is defined in Figure 4. An overlap $U$ (unmeasured) makes the product go through all the tuples $X_i$ (from 1 to $N$) it affects or through none of them. Each tuple may influence the label $Y$, where a 1 signifies a problematic product.

Figure 4 has associated the functional causal model below:

$$U = \{0, 1\}$$
$$X_1 = U$$
$$\ldots$$
$$X_i = U$$
$$\ldots$$
$$X_N = U$$
$$Y = X_i^*$$

where $X_i^*$ is the root cause tuple, which is unknown, and finding it is the objective of RCA. This functional causal model simply states that $U$ randomly assumes the value of 0 or 1, and, without the interference of noise, all $X_i$ have the same value as $U$. The label $Y$ is active when the root cause node $X_i^*$ is active. As we do not know which of the nodes is the root cause, the objective is to estimate the causal effect using interventional queries.

It should be noted that for each group or cluster of overlapped tuples, one would have a different variable $U_i$, representing a local “synchronicity pattern” in the manufacturing process. In RCA, we are mainly focused on the cluster containing the label because it is the one that can lead us to the root cause of a problem identified in the label. Any tuple overlapped with the label will be overlapped with the other tuples that are also overlapped with the label.

To exemplify with a specific case, let us consider the example in Figure 1. In this example, the overlap exists between three elements: the two tuples, “Step A-Machine 1” and “Step B-Machine 3”, and the label “Problem $= 1$”. This translates to the causal model in Figure 5.

The corresponding functional causal model would be:

$$\text{Overlap} = \{\text{False, True}\}$$

“Step A-Machine 1” $= \text{Overlap}$
"Step B-Machine 3" = Overlap

"Problem = I" = Tuple*

(9)

where Tuple* is the true root cause. In this specific example, as we know that "Step A Machine I" is the true root cause, the expression becomes "Problem = I" = "Step A-Machine I".

It is clear from the model in Figure 4 that, when there is no interference with the influence of overlap in the tuples, it is impossible to distinguish between the tuples. Despite this seemingly trivial conclusion, the model still allows us to obtain knowledge that is useful to tackle the presence of overlap in ARCA. This model allows us to identify the minimum amount of interference required to compute the causal effect of a tuple $X_i$ on the label $Y$. This interference can be caused by random events that disrupt the local synchronicity in production, i.e., overlap, which leads to a few products not being affected by overlap. Knowing this can be before it is no longer possible to compute the causal effect. As such, the limit to how "pure" an overlap can be is the total number of overlapped instances is required (if a percentage of the possible, it should still present a group of the most likely root cause, the denominator $N$ of overlap in ARCA. This model allows us to identify the one of the variables is 1 while the others are 0. As such, it is possible to say that the limit is absolute and depends on the number of overlapped tuples.

As $U$ is unmeasured, it is not possible to use the adjustment for direct causes. Therefore, it is necessary to use the BD criterion. When examining Figure 4, it is possible to see that, to block any interference through the BD, we need to control all the other tuples. Therefore, we can compute this query using Expression (11), where $z$ are all possible combinations of $N - 1$ binary variables representing all the tuples except $X_i$.

$$P(Y|do(X_i)) = \sum_z P(Y|X_i, z)P(z)$$

(11)

As an example, to compute the causal effect of tuple $X_1$ for a model with three tuples, the expression would become:

$$P(Y|do(X_1)) = \sum_{x_2} \sum_{x_3} P(Y|X_1, x_2, x_3)P(x_2, x_3)$$

$$= P(Y|X_1, X_2, X_3)P(X_2, X_3) + P(Y|X_1, \overline{X_2}, X_3)P(\overline{X_2}, X_3) + P(Y|X_1, X_2, \overline{X_3})P(X_2, \overline{X_3}) + P(Y|X_1, \overline{X_2}, \overline{X_3})P(\overline{X_2}, \overline{X_3})$$

(12)

However, as explained in [15], the notion of causation can have several types, namely the Probability of Necessity (PN - how necessary it is for a product to pass through a tuple for the product to become problematic), the Probability of Sufficiency (PS - if it is sufficient for a product to pass through a tuple to become problematic), and the Probability of Necessity and Sufficiency (PNS - a mixture of the previous two probabilities). Although these were defined using counterfactuals, we adapted them to follow a similar reasoning, but for interventional queries, like in the following expressions:

$$PS = P(Y|do(X_i))$$

(13)

$$PN = P(\overline{Y}|do(\overline{X}_i))$$

(14)

$$PNS = P(Y, X_i)PS + P(\overline{Y}, \overline{X}_i)PN$$

(15)

These probabilities may be computed after the most relevant tuples have been identified using PMI to compute the overlap of the tuples with the label. This makes the process more efficient and avoids computing interventional queries on tuples of the same step. These queries are problematic because the division by zero makes them become not defined.

When the overlap is "pure" (i.e., there is no difference among tuples), it is impossible to ascertain the causal effects using interventional queries. A hypothesis could have been to use counterfactual queries, which work with "what if" scenarios. However, we are prevented from doing so because we do not know the full functional causal model of overlap. When $U$ is active, all the $X_i$ tuples are active, but we do not know which of the tuples activate the label $Y$. In other words, we do not know $X^*_i$, defined at the beginning of Section III, because that is precisely what RCA tries to identify. As such, we are limited to use the PN, PS, and PNS formulations using interventional queries.
IV. EVALUATION AND RESULTS

A. EVALUATION SETUP

To evaluate the behavior of the model and to test the performance of the methodology proposed in the previous section, we developed a mockup dataset generator. Using this generator, we can simulate an overlapped cluster where it is possible to control: i) the number of products and the number of tuples; ii) which tuple is the root cause; iii) the noise/interference that affects the influence of the root cause; and iv) the noise that can lead to misclassification. Figure 6 illustrates how both noises, $\epsilon_u$ and $\epsilon_l$, were added to the model in Figure 4 to test the behavior and sensitivity of the model in different situations.

$\epsilon_u$ represents the probability of a product in an overlapped tuple not assuming the same value as the root cause tuple, when $U = 1$. When $U = 0$, there is a $\epsilon_u\%$ of products in that tuple that are not equal to 1, and vice-versa. This noise parameter is included to test the limits to which a root cause can be found, defined in Section III. $\epsilon_u$ represents the interference in overlap, which is caused by random events that disrupt the local synchronicity in production. In the mockup datasets generated, all tuples have the same $\epsilon_u$.

$\epsilon_l$ represents the percentage of products that passed through the true root cause tuple $X^*_1$ but were not faulty. This noise parameter was included to check how resilient the model is to adverse conditions that can impact performance (e.g., imperfections in the labeling process and non-systematic or spurious problems). In the mockup datasets generated, only the true root cause is affected by $\epsilon_l$. The true root cause varies depending on the dataset (it is not always $X_2$).

For the mockup datasets generated, Table 2 summarizes the values of the parameters used for the experiments. For each combination of parameters, 25 datasets were generated, and the PN, PS, PNS, and PMI values were computed. As there are 25 possible combinations of noise repeated 25 times each, a total of 625 datasets were generated.

**TABLE 2. Values of the parameters in the mockup datasets generated.**

| Parameter | Values |
|-----------|--------|
| No. of Tuples | 3 |
| No. of Products | 10,000 |
| $\epsilon_u$ | $[1E^{-05}, 1E^{-04}, 1E^{-03}, 1E^{-02}, 1E^{-01}]$ |
| $\epsilon_l$ | $[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]$ |

A small example of a mockup dataset is represented in Table 3. This dataset has 5 products that can pass through one of the three tuples, represented by columns $X_1$, $X_2$, and $X_3$. These tuples are under the influence of overlap $U$, which makes a product pass through all products or none of them. However, noise or interference may lead to one of the tuples momentarily not being under the influence of overlap, for example in product 3 and tuple $X_1$.

**TABLE 3. Example of a small mockup dataset.**

| Product | U | $X_1$ | $X_2$ | $X_3$ | Y |
|---------|---|------|------|------|---|
| 1       | 1 | 0    | 0    | 0    | 0 |
| 2       | 2 | 1    | 1    | 1    | 1 |
| 3       | 3 | 1    | 0    | 1    | 1 |
| 4       | 4 | 0    | 0    | 0    | 0 |
| 5       | 5 | 1    | 1    | 0    | 1 |

As the mockup datasets have access to the value of $U$, that value was used to verify if the values of the causal effect obtained through the adjustment of direct causes and through the BD criterion were the same. This was verified and validated. The PN, PS, PNS, and PMI values are presented as a benchmark. Also, each mockup dataset represents a dataset where the relevant features have already been selected using an approach existing in the literature, for example, PMI.

In addition to the mockup datasets generated, the proposed methodology was also tested on several sets of stochastically simulated data and real data. These datasets are based on a real case-study in a semiconductor manufacturing setting, where the problem to be identified is overkill - situations where the Automatic Optic Inspection (AOI) generates too many false detections of defects due to exterior changes in the product that do not necessarily make the product defective. Table 4 presents the metadata of the simulated and real case-study datasets, which are different from the mockup datasets.

These four datasets consisted of three simulated datasets and one real one. In the simulated datasets, the root causes were clearly identified, but not in the real dataset. Moments in the dataset that contained systematic problems were automatically identified using an Exponentially Weighted Moving Average (EWMA) algorithm. A total of 13 moments (divided as [4,5,4]) were identified in the three simulated datasets and 16 moments were identified in the real case-study dataset. PMI was used in each of the moments identified to select the
tuples and keep the ones that had the highest overlap with the label. Only the tuples that shared the maximum overlap with the label were considered for causal inference analysis. This was a major difference in relation to the mockup datasets.

These last experiments were done not only to test the proposed methodology but also to identify how “pure” the overlap is in real-world datasets.

**B. EVALUATION RESULTS**

1) **MOCKUP RESULTS**

To test the sensitivity of the proposed model and approach to noise, a total of 625 datasets were generated and the PN, PS, and PNS queries were tested in them. A PMI method explored in a previous work was also tested as a benchmark. The interventional queries and PMI values were computed for all three tuples, and then these values were checked to see if the true root cause node was the one with the single highest value. If this happened, the root cause was correctly detected.

The results are presented in Tables 5 and 6, where it is possible to see the variation in $\epsilon_u$ and $\epsilon_l$, respectively. The values within each cell represent the percentage of datasets where the root cause was correctly detected when using one of the interventional queries or the PMI method.

**TABLE 5. Results of the interventional queries proposed and of the PMI method, when varying the values of $\epsilon_l$.**

| $\epsilon_l$ | PS | PN | PNS | PMI |
|-----------|----|----|-----|-----|
| 0, 1      | 78%| 77%| 77% | 77% |
| 0, 2      | 76%| 72%| 75% | 78% |
| 0, 3      | 75%| 74%| 74% | 74% |
| 0, 4      | 74%| 72%| 74% | 75% |
| 0, 5      | 72%| 72%| 72% | 72% |

From analyzing Table 5, we can say that the noise/interference $\epsilon_u$ had a major impact on the performance of all the methods. When the interference was 1.00E-03 or above, the influence of overlap no longer constrained the methods and they were able to detect all the root causes. As $\epsilon_u$ went below that level, overlap became an issue, because the interference threshold (mentioned in Section III) was not surpassed in many datasets, preventing the causal effect from being computed.

It is also worth mentioning that when the threshold was surpassed and the interventional queries were able to compute the causal effect, the PMI approach was also able to detect the root cause. This seems to indicate that both methods are valid for the detection of root causes as long as the interference threshold is surpassed.

**TABLE 6. Results of the interventional queries proposed and of the PMI method, when varying the values of $\epsilon_l$.**

| $\epsilon_l$ | PS | PN | PNS | PMI |
|-----------|----|----|-----|-----|
| 0, 1      | 78%| 77%| 77% | 77% |
| 0, 2      | 76%| 72%| 75% | 78% |
| 0, 3      | 75%| 74%| 74% | 74% |
| 0, 4      | 74%| 72%| 74% | 75% |
| 0, 5      | 72%| 72%| 72% | 72% |

From the results presented in Table 6, it is possible to see that $\epsilon_l$ had little to no influence on the root cause detection performance of all methods. While there was a slight decline in performance as the noise was increased, this could be anticipated, as when the noise surpasses certain levels, the label starts losing its meaning. In fact, the surprise lies in how little the effect was when only 50% of the products that went through the root cause tuple were in fact problematic. This indicates the resilience of the models tested to noisy datasets and errors in labeling.

Although not expressed in the tables presented, it is important to mention a certain behavior of the results on a finer level of analysis. It can be concluded from what was said above that $\epsilon_u$ interferes with the identifiability of the causal effect. Even though $\epsilon_l$ does not seem to interfere with this ability to identify the causal effect, the truth is that it affects the magnitude of such a causal effect. The greater $\epsilon_l$ is, the lower the value of the causal effects computed in terms of both the true root cause and of the other tuples. For example, for the datasets with $\epsilon_u = 1.00E - 01$, the average causal effect of the true root cause of all datasets is 0.90 when $\epsilon_l = 0.1$ and 0.50 when $\epsilon_l = 0.5$. For the non-root cause tuples, the average is 0.45 and 0.25, respectively. This means that the difference between the interventional probabilities of root cause tuples and the others is 0.45 when $\epsilon_l = 0.1$ and 0.25 when $\epsilon_l = 0.5$. As such, the difference between the estimated effect on quality of passing through the root cause tuple and the other tuples is reduced.

This behavior, which in this case occurs in a controlled environment with mockup datasets, was already observed in our previous works, where we already speculated that the need to lower the PMI threshold had to do with noise (defined as randomness in the labeling of the products). We believe that the behavior mentioned in this paragraph is evidence that this interpretation was indeed correct and that randomness in labeling affects the magnitude of the causal effect that one is able to compute. This randomness in labeling may come either from human/machine errors or from other small random problems, aside from the main problem caused by the root cause.

In terms of comparing the new approach to the benchmark (PMI), it is possible to see that the PS had the best performance in two levels of $\epsilon_l$ (0.1 and 0.3), while the PMI had
the best performance in another two (0.2 and 0.4), with a draw between both approaches (0.5), meaning that there were no significant differences in performance between the two approaches.

2) SIMULATION & REAL CASE-STUDY RESULTS

When analyzing the datasets based on the real case-study, only two outcomes resulted from each of the moments identified: i) a group of possible root causes was identified by the PMI filter, but the interventional queries were not able to distinguish the individual influence of each tuple; ii) the PMI filter was able to distinguish a single root cause node. In the simulation datasets, all root causes were correctly detected.

Table 7 presents the results on these datasets, based on the outcomes described above. Each dataset was divided into the problematic moments identified, which were then analyzed (there were no moments without problems). The three simulation datasets had a total of 13 moments identified, while the real case-study dataset had a total of 16 moments identified.

| Dataset       | I) Multiple RC Detected | II) Single RC Detected |
|---------------|-------------------------|------------------------|
| Simulation    | 38%                     | 62%                    |
| Real Case-Study| 50%                    | 30%                    |

The results obtained in the experiments with the mockup datasets confirm that when the interference threshold is surpassed, PMI is enough to identify the root cause. The interventional queries reached the same results as the PMI. However, even without the discriminatory power added by the interventional queries, more than half of the moments had a single root cause detected. Even in the moments where multiple root causes were detected, the number of tuples to consider was still greatly reduced.

Regarding the effect of noise (as mentioned in the previous section), and although this is not presented in Table 7, some moments in the real case-study dataset had a PNS causal effect value of only 0.166, revealing that these moments were very noisy. It is open to discussion if we should even accept such low values of causal effect as indicative of the presence of root causes.

V. DISCUSSION

This section summarizes the results of the different experiments and discusses how these can be combined to form coherent conclusions.

In the mockup experiments, the proposed approach achieved a performance comparable to that of the PMI approach proposed in a previous study, both when varying $\epsilon_u$ and $\epsilon_l$. The results also indicate that $\epsilon_u$ affected the identifiability of the true root cause, as hypothesized. The lesser the interference, the more overlap prevents the discovery of the true root cause, as the data generated does not provide enough information to distinguish the effect of the different tuples on the quality label. Meanwhile $\epsilon_l$ affected the magnitude of the causal effect, or how low the threshold needs to be in order to capture the true root cause. As the effect of random events on the quality of products increases, the relative strength of the causal effect of the passage by the root cause tuple becomes diluted, which requires the methods to be tuned to detect smaller differences between the effects of passing through the different tuples.

The results of the experiments with the stochastic simulation and real case-study data showed that when the conditions allowed the true root cause to be identified, the PMI approach had the same performance as the one proposed in this paper. As the PMI approach represents the best results from previous works, we can say that the proposed approach based on causal inference achieves a similar performance than the best of the previous methods. This similarity in performance between both methods indicates that overlap is a phenomenon that, when present in a “pure” form, makes impossible the discovery of the true root cause, independently of the methodology used. However, the approach based on causal inference has the advantage of providing a perception of such limitation, and even allowing for the definition of the overlap limits.

To summarize the results of all experiments, it is possible to conclude that the approach proposed in this paper and the PMI approach proposed in a previous study perform similarly and are able to identify true root causes as long as the conditions identified in Section III are met. It was also possible to find further evidence that supports previous hypotheses on the effects of noise on the performance of ARCA solutions.

VI. CONCLUSION AND FUTURE RESEARCH

In this paper, we presented a different perspective on the issue of overlap in Automatic Root Cause Analysis (ARCA). We attempted to “untangle” the effect of overlap and ultimately allow analysts to clearly identify the root cause from a group of overlapped variables. The proposed approach was validated in real datasets from a case-study in semiconductor manufacturing and stochastically simulated datasets based on the case-study, which showed further evidence of the conclusions reached in the mockup experiments: the PMI approach is able to detect the root cause by itself once the theoretical limit is surpassed and noise does decrease the magnitude of the causal effect. The contributions of this paper are: i) a causal model of overlap, which further improves our understanding of this issue, ii) a clear definition of the conditions necessary to locate root causes in the presence of overlap, and iii) an ARCA solution robust to overlap based on causal inference.

In terms of shortcomings, the lack of root cause definition in the real case-study data hinders the validation in real data. Future work could improve upon this by having access to real data with the root causes clearly identified. It would also be interesting to consider more complex structures of functional causal models, e.g., instead of having a single root cause, having multiple root causes joined by either “or”/“and” operators. Another topic for further research would be the application of the proposed approach to other...
manufacturing sectors, besides the semiconductor manufacturing data explored in this paper. In addition to the manufacturing sector, this method can be applied to other areas that generate data similar to the location type of data, such as logistics sector, to diagnose problems in different levels of the supply chain.

REFERENCES

[1] O. Abdelrahman and P. Keikhosrokiani, “Assembly line anomaly detection and root cause analysis using machine learning,” IEEE Access, vol. 8, pp. 189661–189672, 2020.

[2] A. Brun, S. Castagnos, and A. Boyer, “A positively directed mutual information measure for collaborative filtering,” in Proc. 2nd Int. Conf. Inf. Syst. Econ. Intell. Nancy, France: Nancy Univ., 2009, pp. 943–958.

[3] Z. Cai, W. Li, W. Zhu, L. Liu, and B. Yang, “A real-time trace-level root-cause diagnosis system in Alibaba datacenters,” IEEE Access, vol. 7, pp. 142692–142702, 2019.

[4] P. Chemwevo, L. Pintelon, L. Jongers, and P. Mchiri, “I-RCAM: Intellig-ent expert system for root cause analysis in maintenance decision making,” in Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM), Jun. 2016, pp. 1–7.

[5] X. Chen, J. Wang, and J. Zhou, “Process monitoring based on multivari-ate causality analysis and probability inference,” IEEE Access, vol. 6, pp. 6360–6369, 2018.

[6] L. H. Chiang, B. Jiang, X. Zhu, D. Huang, and R. D. Braatz, “Diagnosis of multiple and unknown faults using the causal map and multivariate statistics,” J. Process Control, vol. 28, pp. 27–39, Apr. 2015.

[7] C.-F. Chien, T. Hong, and H. Guo, “A conceptual framework for ‘industry 3.5’ to empower intelligent manufacturing and case studies,” Proc. Manuf., vol. 11, pp. 2009–2017, Jan. 2017.

[8] A. K. Choudhary, J. A. Harding, and M. K. Tiwari, “Data mining in manufacturing: A review based on the kind of knowledge,” J. Intell. Manuf., vol. 20, no. 5, pp. 501–521, 2009.

[9] M. Donauer, P. Peças, and A. Azevedo, “Identifying nonconformity root causes using applied knowledge discovery,” Robot. Comput. Integr. Manuf., vol. 36, pp. 84–92, Feb. 2015.

[10] J. Dong, R. Sun, K. Peng, Z. Shi, and L. Ma, “Quality monitoring and root cause diagnosis for industrial processes based on Lasso-SAE-CCA,” IEEE Access, vol. 7, pp. 90230–90242, 2019.

[11] E. Oliveira, V. L. Miguéis, and J. L. Brites, “On the influence of overlap in automatic root cause analysis in manufacturing,” Int. J. Prod. Res., pp. 1–17, Oct. 2021.

[12] S.-K.-S. Fan, S.-C. Lin, and P.-F. Tsai, “Wafer fault detection and key step identification for semiconductor manufacturing using principal component analysis, AdaBoost and decision tree,” J. Ind. Prod. Eng., vol. 33, no. 3, pp. 151–168, Apr. 2016.

[13] Y.-K. Gu, M. Zhang, and X.-Q. Zhou, “Fault diagnosis of gearbox based on improved DUGC with combination weighting method,” IEEE Access, vol. 7, pp. 92955–92967, 2019.

[14] P.-L. Ong, Y.-H. Choo, and A. K. Muda, “A manufacturing failure root cause analysis in imbalance data set using PCA weighted association rule mining,” Jurnal Teknologi, vol. 77, no. 18, pp. 103–111, Nov. 2015.

[15] J. Pearl, “Probabilities of causation: Three counterfactual interpretations and their identification,” Synthese, vol. 121, no. 1, pp. 93–149, 1999.

[16] J. Pearl, Causality: Models, Reasoning Inference, 2nd ed. Cambridge, MA, USA: Cambridge Univ. Press, 2009.

[17] J. Pearl, “Causal inference in statistics: An overview,” Statist. Surv., vol. 3, pp. 96–146, Oct. 2009.

[18] N. Radziwill. (2019).

[19] H. Sim, D. Choi, and C. O. Kim, “A data mining approach to the causal analysis of product faults in multi-stage PCB manufacturing,” Int. J. Precis. Eng. Manuf., vol. 15, no. 8, pp. 1563–1573, Aug. 2014.

[20] S. Sinha, P. Franciosa, and D. Ceglar, “Building a scalable and inter-pretable Bayesian deep learning framework for quality control of free form surfaces,” IEEE Access, vol. 9, pp. 50188–50208, 2021.

[21] H. J. Steinhaus, A. Karlsson, G. Mathiason, and T. Helldin, “Root-cause localization using restricted Boltzmann machines,” in Proc. 19th Int. Conf. Inf. Fusion (FUSION), 2016, pp. 248–255.

[22] Z.-H. Sun, R. Liu, and X. Ming, “A fault diagnosis and maintenance decision system for production line based on human-machine multi-information fusion,” in Proc. Artif. Intell. Cloud Comput. Conf. (AICCC), New York, NY, USA: Association for Computing Machinery, 2018, pp. 151–156.

[23] J. Wang and K. Chen, “Selection of root-cause process variables based on qualitative trends in historical data samples,” IEEE Access, vol. 7, pp. 138637–138644, 2019.

[24] M. Zanon, G. A. Susto, and S. McLoone, “Root cause analysis by a combined sparse classification and Monte Carlo approach,” IFAC Proc. Volumes, vol. 47, no. 3, pp. 1947–1952, 2014.

EDUARDO E. OLIVEIRA received the M.Sc. degree in industrial engineering and management from the Faculty of Engineering of the University of Porto, in 2015, where he is currently pursuing the Ph.D. degree. He is also a Researcher at INESC TEC. He has also an Invited Auxiliary Professor in information systems with the Department of Industrial Engineering and Management, Faculty of Engineering of the University of Porto. His research interests include data mining and diagnosis in manufacturing.

VERA L. MIGUEIS received the Ph.D. degree in industrial engineering and management from the Faculty of Engineering of the University of Porto, Portugal. She is currently an Assistant Professor with the Department of Industrial Engineering and Management, Faculty of Engineering of the University of Porto. She is also a Researcher at INESC TEC. She has published papers in several international journals, namely focusing on the retail, education, and manufacturing sectors. Her research interests include data analytics and quantitative methods to support decision making processes. She has been mainly working on analytical customer relationship management and data mining.

JOSÉ L. BORGES received the degree in mechanical engineering and the M.Sc. degree in electronic engineering and computers from the Faculty of Engineering of the University of Porto, and the Ph.D. degree in computer science from the University College of London, U.K. He is currently an Associate Professor with the Department of Industrial Engineering and Management, Faculty of Engineering of the University of Porto and a Researcher at INESC TEC. He teaches courses in statistics, data mining, information systems, human–computer interaction, and data visualization. He has published more than 30 papers in international journals, ISI proceedings and book chapters. He is a Supervisor or a Co-Supervisor of four completed Ph.D. theses and of three ongoing theses. He is also a Supervisor of more than 45 master’s theses.