

Learning Causal Graphs in Manufacturing Domains using Structural Equation Models

Maximilian Kertel
Technology Development Battery Cell
BMW Group
 Munich, Germany
 maximilian.kertel@bmw.de 

Stefan Harmeling
Department of Computer Science
TU Dortmund University
 Dortmund, Germany
 stefan.harmeling@tu-dortmund.de

Markus Pauly
Department of Statistics
TU Dortmund University
 Dortmund, Germany
Research Center Trustworthy
Data Science and Security
 UA Ruhr, Germany
 pauly@statistik.tu-dortmund.de 

Abstract—Many production processes are characterized by numerous and complex cause-and-effect relationships. Since they are only partially known they pose a challenge to effective process control. In this work we present how Structural Equation Models can be used for deriving cause-and-effect relationships from the combination of prior knowledge and process data in the manufacturing domain. Compared to existing applications, we do not assume linear relationships leading to more informative results.

Index Terms—Causal Discovery, Bayesian Networks, Industry 4.0

I. INTRODUCTION

To be published in the Proceedings of IEEE AI4I 2022.

Complex manufacturing processes as, e.g. for battery cells show high scrap rates and thus high production costs and large environmental footprints. One of the driving factors is the missing knowledge on the interdependencies between the process parameters, intermediate product properties and the quality characteristics [1]. Together we call this the cause-and-effect relationships (CERs). CERs can be visualized as a network with the process and product characteristics as nodes and the CERs as directed edges [1], [2]. It is the goal of our paper to unify expert knowledge and process data to derive such a network, which allows the visual identification of

- root-causes of erroneous products,
- relevant parameters for process control during successive production steps and
- important characteristics to predict the quality of the final product.

In complex manufacturing domains, CERs form a linked mesh of hundreds of involved factors [1]. Typically, CERs are derived by running Designs of Experiments (DOEs). However, DOEs can be time-demanding and the production line has to be stopped in the meantime leading to prohibitively high costs. Moreover, if there are many potential CERs, the number of experiments can become infeasible.

At the same time, the Internet of Things (IoT) allows data processing and storage along the whole production line, leading to a vast amount of accessible information. It is thus desirable

to derive the CERs from the existing observational (or non-experimental) data. For this purpose, Bayesian Networks can be used to unify expert knowledge and data. From these, CERs can be derived under the assumption of causal sufficiency [3]. This approach is called *causal discovery* or *structure learning*. The most common example in the manufacturing domain [4]–[6], is the PC algorithm [3]. This algorithm relies on the assumption of faithfulness and on efficient statistical tests for conditional independence. In principle the PC algorithm can be applied with any test for conditional independence. However, existing nonparametric tests do not scale well [7], [8]. Most of the applications of the PC algorithm either discretize the measurements, or researchers approximate the joint distribution of the variables by a multivariate normal distribution. For discrete data and normally distributed data fast tests for conditional independence exist. However, the former leads to a loss of information, while the latter requires a linear dependency between the variables to be exact. In case of manufacturing data this is most likely a misspecification [9]. Simulation studies show, that the performance of the PC algorithm can be poor in case of non-linearity [10]. This questions the application of the PC algorithm for large or high-dimensional manufacturing data.

In recent years, Structural Equation Models (SEM), which can incorporate arbitrary functional relationships, were increasingly proposed to derive Causal Bayesian Networks. They replace the assumption on faithfulness by a functional form of the conditional distributions (see Equation (1)). While the PC algorithm returns a set of graphs, methods based on SEMs often derive a single graph. To the best of our knowledge, we are the first to apply SEMs to derive such graphical models in the manufacturing domain.

The paper is structured as follows. In Section II we present potential prior knowledge and available data in manufacturing domains. We continue in Section III by reviewing Bayesian Networks and SEMs and explain Causal Additive Models (CAM). In Section IV we present an extension of CAM, called TCAM, which efficiently incorporates prior knowledge. We apply our method in Section V to process data of the assembly of battery modules at BMW. We conclude in Section VI.

II. DATA AND CHALLENGES IN COMPLEX MANUFACTURING DOMAINS

In this section we describe the data sources and propose a preprocessing of the data. Then, we explain the broad prior knowledge in manufacturing domains. Finally, we mention common challenges with production data.

A. Data Sources along the Production Line

The assembly of products consists of production lines, which again contain several stations, which are passed in a fixed order and where process steps are carried out. During those process steps the piece is transformed or it is combined with other parts in order to achieve a predefined outcome. All involved parts are assigned to unique identifiers. Data of different types is collected along the production process:

- Process data: the stations take measurements of the involved parts (e.g. thickness of the piece) and the parameters of the machine (e.g. weight of applied glue).
- End-of-Line (EoL) tests take additional quality measurements of the intermediate or final products.
- Station information: at some production steps the pieces are spread out to identical stations, such that parts can be processed in parallel and every piece is assigned to one of the stations.
- Bill of Material (BoM): the BoM contains the information which pieces were merged together and on which position they have been worked in.
- Supplier data: suppliers transmit data on provided goods.

The preprocessing of the data, which is depicted in Figure 1, consists of the following steps:

- 1) Collect the data for every intermediate product.
- 2) Iteratively merge the data of all subcomponents of a final product.

Measurements of identical subcomponents, which are placed in the same position, can be found in the same column. Eventually, the final tabular data set contains all measurements that can be associated with a final product.

B. Prior Knowledge

As the stations are passed in a fixed order, we know that CERs across different stations can only act forward in time. Additionally, in many manufacturing organizations, tools as the Failure Mode and Effect Analysis (FMEA) [11] are implemented to extract expert knowledge on CERs in the production process and to provide the information in a structured form.

C. Challenges of Data Analysis in Manufacturing

Often, similar information is recorded multiple times along the production line, leading to multicollinearity [4]. Also, sensors might deliver non-informative data by recording implausible values. Industrial data is also reported to be drifting over time. However, even in shorter time intervals, data of a series production contains thousands of observations. This distinguishes the manufacturing domain from other applications of causal discovery as medicine, genetics or the social sciences.

III. STRUCTURE LEARNING OF GRAPHICAL MODELS

A. Some Preliminaries on Graphical Models

Let $G = (\mathbf{V}, \mathbf{E})$ be a directed acyclic graph (DAG) [12, Chapter 6] with nodes $\mathbf{V} = (V_1, \dots, V_p)$ and edges \mathbf{E} . The node V_i is called a parent of V_j if the edge $V_i \rightarrow V_j$ is in \mathbf{E} . We denote the set of all parents of V_j as $pa(V_j)$. A tuple of nodes $(V_{j_1}, \dots, V_{j_\ell})$, such that V_{j_k} is a parent of $V_{j_{k+1}}$ for all $k = 1, \dots, (\ell - 1)$, is called a *directed path*. Nodes that can be reached from X_j through a directed path are called the *descendants* of X_j .

In the following we denote random vectors with bold letters as \mathbf{Z} and random variables as Z . Let $\mathbf{X} = (X_1, \dots, X_p)$ be a random vector representing the data generating process. For a graph G with nodes X_1, \dots, X_p , we call (\mathbf{X}, G) a Bayesian network if the local Markov property holds, i.e.

$$X_i \perp X_j | pa(X_i)$$

for any X_j that is not a descendant of X_i in G . Here, $X \perp Y | \mathbf{Z}$ denotes the conditional independence of X and Y given \mathbf{Z} . In that case, we can deduce additional conditional independencies for \mathbf{X} from the graph G using the concept of *d-separation* [12]. For a Bayesian Network (\mathbf{X}, G) , it then holds that $X_i \perp X_j | \mathbf{S}$ if X_i and X_j are d-separated by \mathbf{S} in G . On the other hand, if there is a graph G , such that $X_i \perp X_j | \mathbf{S}$ implies that X_i and X_j are d-separated given \mathbf{S} in G , then \mathbf{X} is called *faithful with respect to G* . As multiple graphs can contain the same d-separations, this graph G is in general not unique.

To promote the intuition, assume that \mathbf{X} has a joint density f . Then $X_i \perp X_j | \mathbf{S}$ can be characterized by

$$f(x_i | X_j = x_j, \mathbf{S} = \mathbf{s}) = f(x_i | \mathbf{S} = \mathbf{s}),$$

where $f(x_i | \mathbf{Z} = \mathbf{z})$ denotes the conditional density function of X_i given $\mathbf{Z} = \mathbf{z}$. Thus, if we already know \mathbf{S} , then X_j does not provide additional information on X_i . Assume that we are interested which variable in $\{X_j, \mathbf{X}_{\mathbf{S}}\}$ causes the variable X_i to be out of the specification limits. Then we know, that the root causes can be found within \mathbf{S} .

B. Graph Learning with Structural Equation Models

While the PC algorithm is the classic approach for deriving a Causal Bayesian Network, recent research focused on identifying it using acyclic SEMs [10], [13]–[15]. They assume that there exists a permutation $\Pi^0(1, \dots, p) = (\pi^0(1), \dots, \pi^0(p))$ and functions $\{f_\ell, \ell = 1, \dots, p\}$, such that

$$X_\ell = f_\ell(X_{\ell_1}, \dots, X_{\ell_v}, \varepsilon_\ell), \ell = 1, \dots, p, \quad (1)$$

where $\pi^0(\ell_k) < \pi^0(\ell)$ for all $k = 1, \dots, v$ and $\varepsilon_1, \dots, \varepsilon_p$ are i.i.d. noise terms. As the estimation of f_ℓ in Equation (1) is difficult in high dimensions, one typically restricts the function class and the distribution of the noise terms. In this work, we assume that the functions follow the additive form

$$f_\ell(X_{\ell_1}, \dots, X_{\ell_v}, \varepsilon_\ell) = c_\ell + \sum_{k: \pi^0(k) < \pi^0(\ell)} f_{k,\ell}(X_k) + \varepsilon_\ell, \quad (2)$$

where $\varepsilon_\ell \sim \mathcal{N}(0, \sigma_\ell)$ and $c_\ell \in \mathbb{R}$. To ensure the uniqueness of the $f_{k,\ell}$ and without loss of generality, we set $\mathbf{E}(X_\ell) = 0$

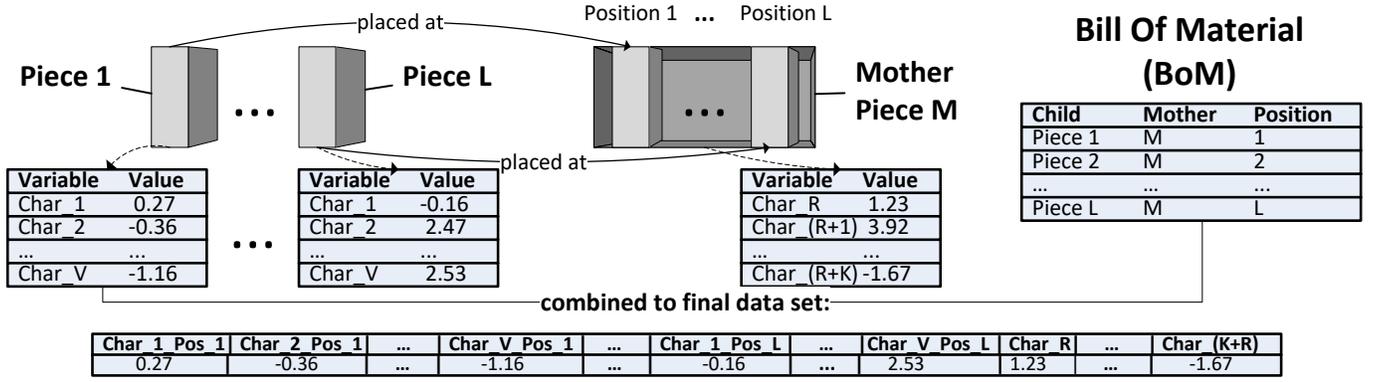


Fig. 1. Visualization of the data preparation described in Section II. The same measurements are collected for piece 1 to piece L . Then they are placed in their mother piece with identifier M . Finally, the resulting data set consists of all measurements of M and those from piece 1 to piece L , where the positioning of the measurements of the child pieces within the data frame depends on their placement according to the BoM. This step is carried out repeatedly, if M itself is positioned in another mother piece.

and $\mathbf{E}(f_{k,\ell}(X_k)) = 0$, for all $\ell = 1, \dots, p, \pi^0(k) < \pi^0(\ell)$. From Equations (1) and (2) we derive that

$$X_\ell \perp X_k | (X_{v_1}, \dots, X_{v_j}),$$

with $\pi^0(k) < \pi^0(\ell), \pi^0(v_1) < \pi^0(\ell), \dots, \pi^0(v_j) < \pi^0(\ell)$ if and only if $f_{k,\ell} = 0$. Let G^0 be the graph on X_1, \dots, X_p , that contains the edge $X_i \rightarrow X_j$ if and only if $f_{i,j} \neq 0$ for $\pi^0(i) < \pi^0(j)$. Then (\mathbf{X}, G^0) is a Bayesian network, as it is fulfilling the Markov property.

If we assume that the functions $f_{k,\ell}$ in Equation (2) are non-linear and smooth, then [15] show that G^0 is identifiable from observational data. This is in contrast to the PC algorithm, which typically returns a class of graphs. Note that we do not presume that the distribution is faithful to some DAG, which is a central assumption of the PC algorithm. We emphasize that for the PC algorithm non-linearity is an obstacle as efficient conditional independence testing is just feasible for multivariate normal data. In contrast, we can utilize the non-linearity for identifying SEMs to receive more informative results (under the assumption of Equation (2)).

An example of a learning algorithm for SEMs is the Causal Additive Model (CAM, [10]). We will focus on CAM due to its applicability to high-dimensional data, its ability to capture non-linearity and due to the theoretical justification that G^0 can be identified, if the functions on the right-hand side of Equation 2 are nonlinear and smooth. [10] propose to find G^0 with the following steps:

- 1) Find the underlying node ordering Π^0 of X_1, \dots, X_p .
- 2) Identify the influential functions $f_{k,\ell}$ with feature selection methods.

To make things more precise, consider N observations $(x_{i1}, \dots, x_{ip}), i = 1, \dots, N$ from \mathbf{X} and call the data matrix $\mathbf{D} \in \mathbb{R}^{N \times p}$.

1) *Finding the node ordering:* [10] show that if

- the functions $f_{k,\ell}$ are smooth and non-linear and can be approximated well and

- the derivatives of $f_{k,\ell}$ and the fourth moments of $f_{k,\ell}(X_k)$ and X_k are bounded.

then the following estimator for Π^0 is consistent as $N \rightarrow \infty$:

$$\hat{\Pi} = \underset{\Pi}{\operatorname{argmin}} \sum_{\ell=1}^p \|x_\ell - \sum_{\pi(k) < \pi(\ell)} \hat{f}_{k,\ell}(x_k)\|_{2,N}^2 \quad (3)$$

Here, we define $\|x_k\|_{2,N}^2 := \frac{1}{N} \sum_{k=1}^N x_{k\ell}^2$ and $\hat{f}_{k,\ell}$ is found by running an additive model regression [16] of X_ℓ on $\{X_k : \pi(k) < \pi(\ell)\}$.

For large p , [10] propose a greedy method to find $\hat{\Pi}$. Let G be a DAG on \mathbf{X} with edges $E(G)$. For simplicity we denote the edge $X_k \rightarrow X_\ell$ by (k, ℓ) . A score for G is defined by

$$S(G) = \sum_{\ell=1}^p \|x_\ell - \sum_{(k,\ell) \in E(G)} \hat{f}_{k,\ell}(x_k)\|_{2,N}^2.$$

The functions $\hat{f}_{k,\ell}$ are estimated by running an additive model regression of X_ℓ on its parents in G . Intuitively, $S(G)$ indicates how much variation of \mathbf{D} is captured by G . The edges that can be added to G without causing cycles are denoted by

$$A(G) := \{(i, j) \in \{1, \dots, p\} \times \{1, \dots, p\} : (\mathbf{X}, E(G) \cup \{(i, j)\}) \text{ is DAG}\}.$$

Starting with the empty graph G_0 , [10] iteratively add the edge $(k^0, \ell^0) = \operatorname{argmax}_{(k', \ell') \in A(G_t)} M_t(k', \ell')$, where

$$M_t(k', \ell') = \|x_\ell - \sum_{(k,\ell) \in E(G_t)} \hat{f}_{k,\ell}(x_k)\|_{2,N}^2 - \|x_\ell - \sum_{(k,\ell) \in E(G_t) \cup \{(k', \ell')\}} \tilde{f}_{k,\ell}(x_k)\|_{2,N}^2. \quad (4)$$

The functions $\hat{f}_{k,\ell}$ are found by regressing X_ℓ on its parents in G_t , while $\tilde{f}_{k,\ell}$ are found by regressing X_ℓ on its parents in $G' = (\mathbf{X}, E(G_t) \cup \{(k', \ell')\})$. Thus, the edge (k^0, ℓ^0) maximally reduces the unexplained variance. We set $G_{t+1} = (\mathbf{X}, E(G_t) \cup \{(k^0, \ell^0)\})$ and continue until we obtain

a complete DAG, which implies the node ordering.

This greedy method is still computationally intense for large p . Thus, [10] propose to take advantage of sparse structures, where p is large but the number of edges in the graph is assumed to be small: to this end they start by a preliminary neighborhood selection (PNS) step. Here, initially for every $\ell \in \{1, \dots, p\}$ a superset of neighbors of X_ℓ in G^0 is identified. In the subsequent node ordering step, one only considers the superset of the neighbors, when greedily adding new edges. This reduces the computation time of the algorithm significantly, if the sizes of the supersets are considerably smaller than p .

2) *Identifying edges*: After the node ordering is set, we need to identify the influential characteristics for every X_ℓ among those X_k for which $\hat{\pi}(k) < \hat{\pi}(\ell)$. The idea is to detect those $f_{k,\ell}$ which are not 0, using feature selection methods [16], [17]. For those k , a change in X_k has an effect on X_ℓ . For a comparison of CAM and the PC algorithm based on simulated data sets with known ground truth, see [10].

IV. METHODOLOGY

The goal of this section is to derive a method that combines the current results on structure learning of SEMs with the features of the manufacturing domain in Section II.

A. Recap of Common Prior Knowledge

Compared to other applications of causal discovery, it is typical for the manufacturing domain, that there exists prior knowledge, see Section II. In particular, there is a partial and transitive ordering of the variables implied by the stations' ordering. Additionally, we include expertise on the absence of edges. Both facets shall improve the algorithm's runtime.

B. Adaptions to CAM

The data generating process behind manufacturing data sets often leads to a low number of conditional independencies in \mathbf{X} , when compared to p . Thus, the Causal Bayesian Network of \mathbf{X} is not sparse. This poses a challenge to many structural learning algorithms in higher dimensions. We show in this subsection how prior knowledge on the node ordering and the existence of edges can be incorporated so that structure learning remains feasible. To formalize our prior knowledge, let $t: \{1, \dots, p\} \rightarrow \{1, \dots, T\}$, so that $t(k) < t(\ell)$ means that there can only be edges from X_k to X_ℓ but not vice versa. Further, let F be a boolean matrix, where $F_{k,\ell} = \text{True}$ if the edge from X_k to X_ℓ is known to be absent.

1) *Preliminary Neighborhood Selection*: For every measurement X_ℓ , we determine a set of possible parents among those k , where $F_{k,\ell} = \text{False}$ and $t(k) \leq t(\ell)$. Denote that set for index ℓ by P_ℓ .

2) *Node Ordering*: We start by adding all potential edges that go across stations and add them to the initial graph G_0 , as those can not cause any cycle. The score of G_0 hence is

$$S(G_0) = \sum_{\ell=1}^p \sum_{k \in P_\ell, t(k) < t(\ell)} \|x_\ell - \hat{f}_{k,\ell}(x_k)\|_{2,N}^2. \quad (5)$$

We continue by determining the node ordering as in Section III-B. Note that we only need to determine the node ordering for indices k, ℓ so that $t(k) = t(\ell)$. The initial inclusion of across-station-edges saves update steps of M (Equation 4). This makes the algorithm feasible even for non-sparse high-dimensional settings, if the number of tiers T or the number of edges known to be absent is sufficiently large.

3) *Pruning*: The pruning step is identical to CAM.

In the manufacturing industry, the prior knowledge on $t(k) < t(\ell)$ is often given by the temporal nature of the production process. We therefore call our adaption *TCAM* (*Temporal Causal Additive Models*). It is sketched in Algorithm 1.

Algorithm 1: TCAM Algorithm

Input: D, F, t as in Section IV-B
Result: DAG G
// Preliminary Neighborhood Selection (PNS)
SupersetNeighbors = list();
for $\ell = 1, \dots, p$ **do**
 $I = \{k \text{ s.t. } t(k) \leq t(\ell) \ \& \ F(k, \ell) = \text{False}\}$;
 SupersetNeighbors[ℓ] = PNS(X_ℓ, \mathbf{X}_I);
 // Other edges are now forbidden
 for $k = 1, \dots, p$ **do**
 if $k \notin \text{SupersetNeighbors}[\ell]$ **then**
 $F(k, \ell) = \text{True}$;
 end
 end
end
// Add across-tier edges
Set G as empty graph on X_1, \dots, X_p ;
for $k, \ell = 1, \dots, p$ **do**
 if $k \in \text{SupersetNeighbors}[\ell] \ \& \ t(k) < t(\ell)$ **then**
 Append (k, ℓ) to edges of G ;
 end
end
 $M(k, \ell) = \text{right-hand side of (5)}$;
for $k, \ell = 1, \dots, p$ **do**
 if $((t(k) > t(\ell)) \mid (F(k, \ell) = \text{True}))$ **then**
 $M(k, \ell) = -\infty$;
 end
end
// Add within-tier edges
while $\max(M) > -\infty$ **do**
 Find $(k_0, \ell_0) = \text{argmax}_{(k,\ell) \in A(G)} M(k, \ell)$;
 Append (k_0, ℓ_0) to edges of G ;
 $M(k_0, \ell_0) = -\infty$;
 Update $M(\cdot, \ell_0)$;
 Set $M(k, \ell) = -\infty$ for all
 $\{1, \dots, p\} \times \{1, \dots, p\} \ni (k, \ell) \notin A(G)$;
 end
// Pruning like CAM (details omitted)
return G

V. APPLICATION

The energy storage of electric vehicles is called a battery pack which is composed of battery modules, which in turn contain a fixed number of battery cells. A battery module connects the battery cells in series or parallel and it protects those cells against shock, vibration and heat. Thus, the battery module is a key component for the safety of battery-electric vehicles. We apply TCAM to data collected at the assembly at BMW. The data set under investigation contains 7254 battery modules with 738 variables each.

A. Data Preparation

As the missing values rate is low (around 2.4%) we apply naive mean imputation instead of more sophisticated method as [18]–[20]. We then continue by removing features that have only one distinct value and hence provide no information. As this data set also shows multicollinearity, we apply an expert-based approach. We asked experts to identify clusters of variables containing similar information and to define representatives for them. For a purely data-driven approach in manufacturing, see [5]. Those steps reduced the number of characteristics from 738 to 491. Finally, we standardize the data so the variables’ empirical mean and standard deviation is 0 and 1 respectively.

Beyond the temporal ordering of the stations, it is reasonable that the production measurements of identical intermediate products as depicted in Figure 1 are independent. Thus, it is possible to restrict the potential edges that have to be considered. Additionally, we assume that some of the recorded measurements as the facility temperature and the selection of the stations are not affected by other measurements. We can mark those values as *root nodes*, meaning that they have no incoming edges. This further restricts the number and orientation of possible edges.

B. Choice of Software and Hyperparameters

1) *Preliminary Neighborhood Selection*: For our application of TCAM, we find supersets of the neighbors by applying the LASSO. For $\ell \in \{1, \dots, p\}$, we run a regression of X_ℓ on those components of \mathbf{X} , which are possible parents according to our prior knowledge. Going forward we mark those variables as potential parents of X_ℓ , where the corresponding regression coefficient is above 10^{-2} . The penalty parameter λ is chosen via cross-validation. Let λ_{min} be the penalty parameter that minimizes the mean squared cross-validation error. Then we choose the maximal λ such that the mean cross-validation error is within one standard deviation of the minimum λ_{min} .

2) *Node Ordering and Pruning*: For the node ordering we employ the package `mgcv` by [16]. Let us call the graph after node ordering G_{NO} . In the pruning step we run a sparse additive regressions of X_ℓ on its parents in G_{NO} for $\ell = 1, \dots, p$. This step returns p-values for the parents of X_ℓ in G_{NO} . We follow [10] and set the regressands as parents of X_ℓ in the final graph, whose p-values are below the threshold of 10^{-3} .

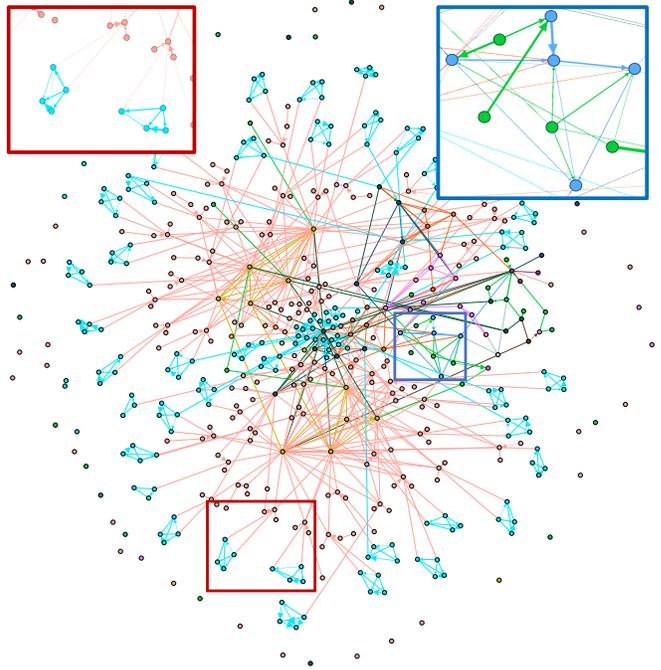


Fig. 2. Resulting graph of TCAM where the nodes correspond to characteristics of the product and edges correspond to detected CERs. The node coloring is according to the station, where the variable was measured. Edge colors are according to respective source node’s color. The blue box highlights the detected relationship between the choice of the stations (green nodes) and the product quality (blue nodes). The red box depicts the similarities between structures of identical subcomponents.

C. Results

The resulting graph is depicted in Figure 2 and contains 491 nodes and 859 edges. We observe that there are a few nodes that have a large number of neighbors. In general this poses a difficulty for most structure learning algorithms and CAM did not finish in reasonable time. For details on the runtime for a low-dimensional special case, see Section V-D. With TCAM and the inclusion of prior knowledge we were able to overcome those obstacles.

Further, substructures of identical parts show similar patterns. The red box in Figure 2, highlights patterns consisting of two linked clusters, where one cluster consists of four nodes, while the other one consists of three nodes. Together with process experts we could further verify that many CERs detected by TCAM are plausible.

This application is confidential, but we would still like to share one of the insights. TCAM discovered a CER between one station that processed the part and the part’s quality. Experts derived that the maintenance of that station was overdue and the CER can be used to find better maintenance intervals. This is one example how graphical models can contribute to an effective and proactive process control.

D. Evaluation against Expert Knowledge

For the characteristics of one of the subcomponents, we derived an expert-based graph, which is depicted in Figure 3. Here, the blue CERs potentially exist, while green CERs surely

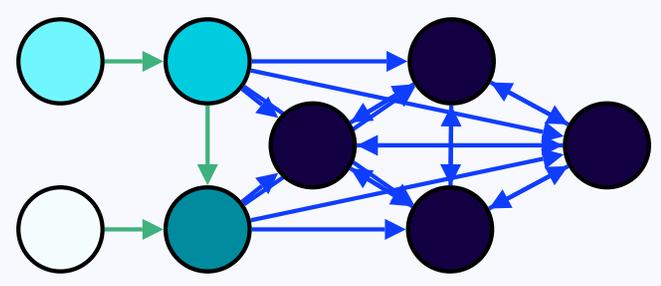


Fig. 3. Expert-based graph on measurements for subcomponents. The green edges are known to exist, while the blue edges potentially exist. Edges beyond the ones depicted are known to be absent. The darker the node, the later the corresponding variable is measured in the production process.

	aSHD	sd(aSHD)	#edges	sd(#edges)	time (s)
CAM	3.496	1.442	9.464	0.948	1.342
TCAM	1.120	0.343	8.084	0.778	1.000
TPC	1.108	0.866	7.463	1.623	0.013

TABLE I

THE AVERAGE ASHD ($\overline{\text{ASHD}}$), THE STANDARD DEVIATION OF THE ASHD ($\text{SD}(\text{ASHD})$), THE AVERAGE NUMBER OF EDGES ($\overline{\text{\#EDGES}}$) AND THE STANDARD DEVIATION OF THE NUMBER OF EDGES ($\text{SD}(\text{\#EDGES})$) FOR ALL THREE METHODS OF SECTION V-D AND FOR 500 REPLICATIONS.

exist. Other CERs can be ruled out. We compare the estimated graphs and runtimes of TCAM, CAM and a variant of the PC algorithm called TPC [21], which allows the inclusion of temporal background knowledge. The significance level is set to 0.01. We run 500 experiments, where we randomly draw 500 subcomponents, while each of them appears in at most one of the runs. We define an adapted Structural Hamming Distance (aSHD) [22] between an estimated graph G_{est} and the one in Figure 3 by the sum over the number of green edges that are not in G_{est} and the number of edges G_{est} that do not appear in Figure 3. The results are depicted in Table I. TPC and TCAM perform better than CAM, which shows the advantage of the inclusion of prior knowledge. Additionally, even in this low-dimensional setting the average runtime for TCAM is smaller than for CAM. Further, we observe that the aSHD of TCAM and TPC is on average quite similar. However, the standard deviation of the aSHD and the standard deviation of the number of edges is smaller for TCAM. This indicates that TCAM delivers more stable and informative results in the manufacturing domain. The original PC algorithm performed worse than TPC and is omitted.

VI. CONCLUSION

We have presented a method to derive the graphical representation of CERs of manufacturing processes based on SEMs. While existing approaches for causal discovery in the manufacturing domain assumed linear relationships between the process characteristics, we applied CAM to find arbitrary additive functional relationships in data. We showed how existing prior domain knowledge can be included and improves the computational burden of CAM. A case study on manufacturing data reveals that the learned graph detects unknown root-causes, delivers more informative results and

paves the way to an efficient and proactive process control.

REFERENCES

- [1] T. Kornas, R. Daub, M. Z. Karamat, S. Thiede, and C. Herrmann, "Data- and expert-driven analysis of cause-effect relationships in the production of lithium-ion batteries," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 380–385.
- [2] T. Wuest, C. Irgens, and K.-D. Thoben, "An approach to monitoring quality in manufacturing using supervised machine learning on product state data," *Journal of Intelligent Manufacturing*, vol. 25, no. 5, pp. 1167–1180, 2014.
- [3] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, prediction, and search*. MIT press, 2000.
- [4] M. Vuković and S. Thalmann, "Causal discovery in manufacturing: A structured literature review," *Journal of Manufacturing and Materials Processing*, vol. 6, no. 1, p. 10, 2022.
- [5] K. Marazopoulou, R. Ghosh, P. Lade, and D. Jensen, "Causal discovery for manufacturing domains," 2016. [Online]. Available: <https://arxiv.org/abs/1605.04056>
- [6] J. Li and J. Shi, "Knowledge discovery from observational data for process control using causal bayesian networks," *IIE transactions*, vol. 39, no. 6, pp. 681–690, 2007.
- [7] K. Zhang, J. Peters, D. Janzing, and B. Schölkopf, "Kernel-based conditional independence test and application in causal discovery," in *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, ser. UAI'11. Arlington, Virginia, USA: AUAI Press, 2011, p. 804–813.
- [8] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf, "Kernel measures of conditional dependence," *Advances in neural information processing systems*, vol. 20, 2007.
- [9] P. Lade, R. Ghosh, and S. Srinivasan, "Manufacturing analytics and industrial internet of things," *IEEE Intelligent Systems*, vol. 32, no. 3, pp. 74–79, 2017.
- [10] P. Bühlmann, J. Peters, and J. Ernest, "CAM: Causal additive models, high-dimensional order search and penalized regression," *The Annals of Statistics*, vol. 42, no. 6, pp. 2526 – 2556, 2014. [Online]. Available: <https://doi.org/10.1214/14-AOS1260>
- [11] D. H. Stamatis, *Failure mode and effect analysis: FMEA from theory to execution*. Quality Press, 2003.
- [12] J. Peters, D. Janzing, and B. Schölkopf, *Elements of Causal Inference: Foundations and Learning Algorithms*. Cambridge, MA, USA: MIT Press, 2017.
- [13] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, "Dags with no tears: Continuous optimization for structure learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [14] S. Shimizu, T. Inazumi, Y. Sogawa, A. Hyvärinen, Y. Kawahara, T. Washio, P. O. Hoyer, and K. Bollen, "Directingam: A direct method for learning a linear non-gaussian structural equation model," *The Journal of Machine Learning Research*, vol. 12, pp. 1225–1248, 2011.
- [15] J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf, "Causal discovery with continuous additive noise models," *The Journal of Machine Learning Research*, vol. 15, no. 1, p. 2009–2053, 01 2014.
- [16] S. N. Wood, *Generalized additive models: an introduction with R*. Chapman and Hall/CRC, 2006, vol. 2.
- [17] J. Huang, J. L. Horowitz, and F. Wei, "Variable selection in nonparametric additive models," *Annals of statistics*, vol. 38, no. 4, p. 2282, 2010.
- [18] S. van Buuren, *Flexible Imputation of Missing Data*. Chapman and Hall/CRC, 2012, vol. 2.
- [19] B. Ramosaj, J. Tulowitzki, and M. Pauly, "On the relation between prediction and imputation accuracy under missing covariates," *Entropy*, vol. 24, no. 3, p. 386, 2022.
- [20] M. Kertel and M. Pauly, "Estimating gaussian copulas with missing data," 2022. [Online]. Available: <https://arxiv.org/abs/2201.05565>
- [21] R. M. Andrews, R. Foraita, V. Didelez, and J. Witte, "A practical guide to causal discovery with cohort data," 2021. [Online]. Available: <https://arxiv.org/abs/2108.13395>
- [22] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing bayesian network structure learning algorithm," *Machine learning*, vol. 65, no. 1, pp. 31–78, 2006.