Journal of Process Control 106 (2021) 142–154

Contents lists available at ScienceDirect

Journal of Process Control

2区

journal homepage: www.elsevier.com/locate/jprocont

A method for detecting causal relationships between industrial alarm variables using Transfer Entropy and K2 algorithm

Rute Souza de Abreu [a,*], Yuri Thomas Nunes [a], Luiz Affonso Guedes [b], Ivanovitch Silva [b]

[a] Post-Graduate Program in Electrical and Computing Engineering, Technology Center, Federal University of Rio Grande do Norte, Natal 59078-970, Brazil

[b] Post-Graduate Program in Electrical and Computing Engineering, Department of Computer Engineering and Automation, Federal University of Rio Grande do Norte, Natal 59078-970, Brazil

ARTICLE INFO

Article history:
Received 28 January 2021
Received in revised form 26 July 2021
Accepted 1 September 2021
Available online 23 September 2021

Keywords:
Bayesian networks
Causal relationships
Transfer Entropy

ABSTRACT

Advances in technology allowed the fast and easy creation and configuration of industrial alarms. The growth in the number of alarms, however, brought some problems to the alarm management systems, for example, alarm flooding. In this paper, it is propose a new method to detect the causal relationships between industrial alarm variables using Transfer Entropy theory and the structural learning of the Bayesian networks K2 Algorithm as its fundamental basis. This work argues that the detection of these relationships can help to reduce the number of alarms in the alarm management systems, avoiding the overloading of operators in case of plant failures. To validate the proposal, a case study using the well-known plant-wide simulator Tennessee Eastman Process was performed. In this study case the graph of the causal relationships obtained by the proposed method was compatible with the system functioning.

1. Introduction

In the last years, the number of industrial alarms set in plants and factories has hugely increased. That occurs because the alarm creation process was benefited by the advances of technology, allowing the creation and configuration of thousands of alarms. The rise in the alarm variables often overloads the alarm systems causing a drop in their performance [1].

When an atypical event occurs in the industry the alarm logs becomes very useful to investigate the root cause of the problem. However, analyzing the alarm interrelationship is often a complex task. Modern industries possess a highly integrated process coordinated by controllers. Thus, if an abnormal situation occurs, many process variables may demonstrate a strong drift in their usual levels. If the situation is not rapidly controlled there can be production loss, damage to the equipment ,or even human life severe harm, and environmental disasters can occur [2–6].

Aware of the challenges in the alarm management systems, such as chattering and nuisance alarms, poor alarm design and configuration, abnormality propagation through plant interconnection [7], this work proposes a method for detecting the causality between industrial alarms, aiming to reduce alarm flooding and operators overloading.

Using Transfer Entropy (TE) and the K2 Algorithm this works produces a model capable of discovering the most probable causal tree, identify the alarm root cause. The idea is that understanding how alarms relate to each other makes it possible to improve the performance of alarm management systems, reducing operators overloading and response time in case of failure occurrences.

In this proposal the TE was used for identifying an initial graph of causal relationships from one series to another. This method was chosen for being able to handle the non-linearities imposed by the behavior of process and alarm variables from the industrial scenarios. However, as the TE produces a very complex graph, this work performed a set of processing on the data to obtain a more significant causal direct acyclic graph. Finally, this graph was used as the pre-order to the K2 Algorithm producing the final version of the causal graph.This latter stage is used for discerning direct from indirect relationships.

To demonstrate how the presented method works a simulation of a real chemical process was performed, using the plant-wide simulator Tennessee Eastman Process.

As the main contributions, this work highlights:

- A data-driven method to detect causal relationships between industrial alarms.
- Insertion of temporal notion into the algorithm.
- Insertion of virtual nodes in K2 preorder, increasing parents search space.

* Corresponding author.
E-mail address: rute.s.abreu@gmail.com (R.S.d. Abreu).

https://doi.org/10.1016/j.jprocont.2021.09.001
0959-1524/

The rest of this paper is divided into more six sections. Section 2 describes the related papers with this work. Section 3 presents the theoretical foundation that is necessary to understand this work. Section 4 details the proposal. In Section 5 a case study, using industrial alarm data from the simulator Tennessee Eastman Process (TEP) [8], is performed. Finally, Section 6 discusses the results and Section 7 draws some conclusions.[1]

2. Related work

In the scientific community, there are some approaches to identify causality and root cause nodes, using either TE, Granger Causality or Bayesian Networks. This work claims that the proposed idea is innovative for joining both fields and introducing the notion of time into the K2 Algorithm, allowing it to work with time-series data.

Although this paper intends to create a method for detecting causal relationships, the research made by the authors have also permeated the themes of alarm flooding reduction, an obvious application to the detection of causal relationships, and alarm management.

Therefore, the following related work was grouped into three main categories: Detection of causal relationships and fault diagnosis, Alarm flood reduction, and Alarm management. Some of the main works in each category will be presented below.

For a better understanding, Table 1 resumes the mentioned works found in literature and provides a comparison between them and the method proposed by this work in terms of the aspects generation of a causality model, detection of root cause variables and indirect relationship analysis, where the latter regards to the analysis of multiples paths coming from one same node to another.

In the next few paragraphs some of the main articles for detecting causal relationships and for performing fault diagnosis will be described. This work focused on articles that used Transfer Entropy, Granger Causality and Bayesian Networks, since these techniques are related to those used in the proposal of this work.

For root cause diagnose in industrial process variables, a data-driven method using Granger Causality (GC) was proposed by [19]. They performed conditional GC to produce a graph in which the process variables were the nodes, and the F-value obtained from the GC test was the edge's weights. They used a maximum spanning tree algorithm to produce an acyclic graph maximized by the level of GC causality.

A causal map of industrial alarms was produced by [22] using a modified version of the Transfer Entropy, Normalized Entropy, in which the measure is computed considering the effect of a historical window of the alarm samples. For relationships between two variables that appeared directly, and indirectly they used the Direct Transfer Entropy, which considers the effect of the addition of a third variable. A causality relationship graph was provided in [18] using a modified version of TE. In the computation of $TE_{Y->X}$, it was fixed the reference for the X variable, following the displacement of the prediction horizon.

An scoring function named Family Transfer Entropy was proposed in [14] for performing Bayesian learning of causal relationships between industrial alarms. Besides, a method for finding root cause alarm using Bayesian networks was proposed in [25]

Transfer Entropy and Mutual Information were used by [20] to detect causality between multi-valued alarm series and for performing fault diagnosis. To overcome the issue of indirect relationships between the alarm variables, it is used Conditional Mutual Information.

[1] This paper presents and summarizes the techniques and processes that were developed by the first author in her Master thesis [9].

Besides, an approach using TE lasso regularization was proposed by [29] for performing fault diagnosing. Also, for fault diagnosing a Gaussian model together with TE application was used in [30]. Granger Causality test and Transfer Entropy were used in [16] to perform causal feature selection. The selected variables were then used to improve the prediction performance of two developed inferential models. They performed their method on process variables obtained through TEP simulation.

Fault detection, root cause diagnosis, and propagation pathway were made using Dynamic Bayesian Networks in [28] and Principal Analysis and Bayesian Networks in [26,27]. An investigation on combining human knowledge, machine learning, and Bayesian networks was made in [23]. The authors used an ontology for maintaining expert knowledge and showed how the use of constraint algorithms can produce promising causality networks.

A framework for support decision in root cause identification in alarm flooding was proposed by [21]. The authors proposed four steps: detecting alarm floods using a standardized norm, extracting relevant signals using tagged root cause alarm, using TE for causality analysis, and extracting root cause alarms by gathering the results with information on plant connectivity. Finally, a non-linear acyclic causal model for predicting alarms in Cyber–Physical Systems based on causality analysis and probability inference was proposed by [17].

The following articles presents different ways of providing alarm flood reduction, which a is direct application of the method that is proposed in this work.

An online algorithm for predicting alarm flooding was proposed by [11], when it is matched online alarm sequences with a pattern database for providing warnings on oncoming alarms. In [15], it was used the Smith–Waterman algorithm for clustering alarm flood sequences. The central idea in that work was that alarm floods coming from distinct faults can be grouped. Therefore being useful to identify patterns and consequently root cause alarms. A new method for flood alarm reduction was proposed in [24]. Their goal was to identify the root cause of the alarm flooding. To do so, their approach applies some steps such as chattering removal, identification of flood periods, flooding clustering sequences, among others, and finally, isolation of the root cause.

A method for identifying groups of alarms for dynamic suppression was proposed in [13]. Through the analysis of historical alarm data, the authors identified the groups of alarm floods and performed mining techniques to discover frequent alarm patterns on the flood groups.

Finally, the following papers refer to alarm management. Alarm management is an important process in the industrial field since it is the process by which the alarms are designed, implemented, and operated. An approach to suppress alarms was proposed by [12]. Authors generated frequent temporal patterns of the alarm signals and used them to analyze the correlations of signals that occurred together. They use this information to define suppression alarm rules.

A method for detecting and quantifying the level correlation between industrial alarms has been shown in [10]. They proposed a technique for generating pseudo-alarm signals and used them for detecting the actual delay in the occurrence of two correlated alarms.

Although most of the related work discussed here produce a graph of causal relationships as output, few of them are capable of both delivering a causal model, root cause detection, and analysis of indirect relationships. Hence, this work aims to contribute to filling this gap.

Table 1
Comparative between approached proposal and existing work.

Article	Causality model	Root cause detection	Indirect relationship analysis	Main techniques
A new method to detect and quantify correlated alarms with occurrence delays [10]	No	No	No	Delays correlation, Pseudo alarm signal generation & Alarm correlation
Online pattern matching and prediction of incoming alarm floods [11]	No	No	No	Chattering and Sequence filters & Incremental dynamic programming strategy
Sequence mining based alarm suppression [12]	No	No	No	Multi-temporal mining method, Bayes classifiers
Detection of frequent alarm patterns in industrial alarm floods [13]	No	No	No	Itemset mining methods
A novel scoring function for Bayesian networks learning [14]	No	No	No	Family transfer entropy & Bayesian networks
Pattern matching of alarm flood sequences [15]	No	No	No	Smith–Waterman algorithm
Data-driven dynamic sensors based on causality analysis [16]	No	No	No	Granger causality & transfer entropy
Process monitoring based on causality analysis and probability inference [17]	Yes	No	No	Independent component analysis, multi-layer perceptrons, mutual information & learning of Bayesian networks.
Data-driven causal inference [18]	Yes	No	Yes	Modified Bauer's transfer entropy
Simplified causality map for root cause diagnosis [19]	No	Yes	Yes	Granger causality
Capturing causality for fault diagnosis based on alarm series [20]	No	Yes	No	Transfer entropy
Cause and effect analysis for decision support in alarm floods [21]	No	Yes	No	Modified transfer entropy
Cause–effect analysis of industrial alarm variables [22]	No	Yes	Yes	Transfer and normalized entropy, normalized direct transfer entropy
Machine learning with knowledge-based modeling for forecasting large, complex, spatiotemporal systems [23]	Yes	Yes	No	Ontologies, prolog rule engine & constraint- and score-based algorithms
Causal analysis for alarm flood reduction [24]	Yes	Yes	No	Chatterting filter, flood periods identification & clustering of flood sequences
Root-cause analysis of occurring alarms based on Bayesian networks [25]	Yes	Yes	No	Bayesian network parameter update method and analysis of alarm root cause
Semiparametric PCA based process fault diagnosis [26]	Yes	Yes	No	PCA semiparametric & Bayesian networks
Process system fault detection [27]	Yes	Yes	Yes	PCA & Bayesian networks
Fault detection and pathway analysis [28]	Yes	Yes	Yes	Dynamic Bayesian networks
Proposed approach	Yes	Yes	Yes	Bauer's transfer entropy & Adapted K2 Algorithm.

3. Fundamental theory

In this section it is presented the necessary background for understanding the concepts, associated with the proposal, to identify causal relationships between industrial alarms. These concepts are namely Transfer Entropy and K2 algorithm for structural learning of Bayesian Networks.

3.1. Transfer entropy

Transfer Entropy was initially proposed by [31]. TE is classified as a technical measure "that shares some of the desired properties of mutual information but takes the dynamics of information transport into account". Through the computation of this measure, it is possible to quantify, dynamically, the exchange of information between two sequences of variables and to know in which direction this exchange is propagated.

Given two random variables I and J, it is needed to know how much information is dynamically transferred from J to I. Assuming I and J are two discrete-time series, TE uses a set of samples from the past of both series to verify whether the future of I is influenced by the past of J.

To check this hypothesis, the vectors $i^k = [i_t, \ldots, i_{t-k+1}]$ and $j^l = [j_t, \ldots, j_{t-l+1}]$ were build. In which the superscript indexes, denominated *time horizons*, define how many samples of the past of the variable are utilized.

In addition to these definitions, it is necessary to set a *prediction horizon* to the variable I. This horizon indicates how far in the future of I will be analyzed and is symbolized by the parameter h.

$$\mathrm{TE}_{J->I} = \sum_{i,i_{t+h},j} p(i_{t+h}, i_t^k, j_t^l) \log \frac{p(i_{t+h} \mid i_t^k, j_t^l)}{p(i_{t+h} \mid i_t^k)} \quad (1)$$

The transfer entropy is computed through Eq. (1) using the joint probability: $p(i_{t+h}, i_t^k, j_t^l)$, and the conditionals probability $p(i_{t+h} \mid i_t^k, j_t^l)$ and $p(i_{t+h} \mid i_t^k)$ to identify a directed exchange of information between the series. In this paper it used the Bauer's form of Transfer Entropy computation, [32], where the prediction horizon, h, is varied.

3.2. Structural learning on Bayesian networks

Bayesian Networks are probabilistic models based on direct acyclic graphs. In these networks, the nodes represent variables of interest, whereas the connections represent relationships of informational or causal dependencies [33]. The dependencies are relative to the node parent set where they are conditional probabilities for a node, given its parents.

The literature describes three approaches for learning the structure of a Bayesian network using data: scoring-based, constraint-based, and hybrid approach [34]. Scored-based algorithms try to find a topology that maximizes the score function. This function tries to find the best topology for the network based on the input data [35]; some examples of algorithms are in [36,37], and [35]. The constraint-based algorithms uses conditional independence tests to evaluate the variables dependence and infer causality, some examples can be found in [38,39], and [40]. In the hybrid approach, both score and constraint strategies are present. Some algorithms that follow this approach can be found in [41,42], and [43].

K2 is a very well-known algorithm for structural learning of Bayesian Networks [37]. It is a score-based approach and requires a prior ordering of the nodes to define the node's network structure. To identify how the network variables affect each other,

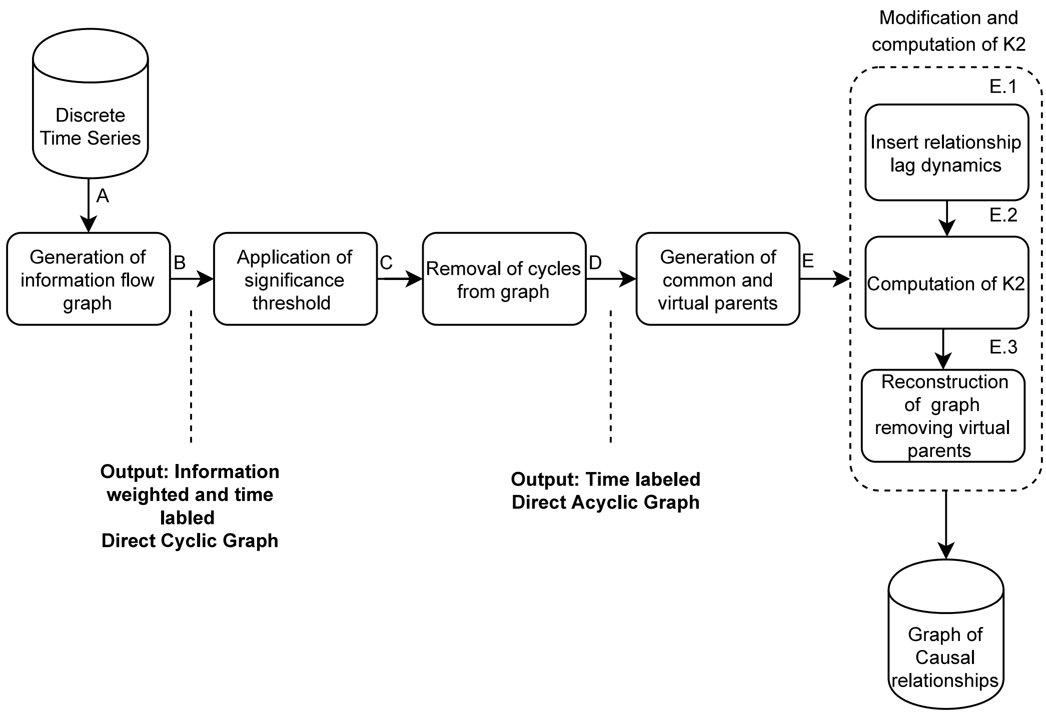

Fig. 1. Block diagram of the presented approach.

K2 proposes an approach that calculates most probable causal structure using a previous ordering of variables. Aiming to do it, the algorithm will try to find the Bayesian structure B_s that maximizes $P(B_s, D)$, where D is a given database and assuming $P(B_s, D)$ as presented in Eq. (2):

$$P(B_s, D) = c \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \qquad N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \tag{2}$$

where: (a) D is a given database; (b) n is the number of nodes in the network; (c) q_i is the number of unique instantiation of the parent set of the node; (d) r_i is the number of all possible values assumed by the node; (e) N_{ijk} is the number of cases in D in which the node is instantiated with its kth value, and the parents of the node are instantiated with the jth instantiation;

Authors argue that for maximizing Eq. (2), it is needed to find the parent set that maximizes the inner product. Therefore, the K2 Algorithm defines as $g(i, \pi_i)$ the function that computes this part of Eq. (2). This function is known as the Cooper–Herskovits metric, and it measures the probability of the parent set π_i be the correct parent set of the node x_i. Eq. (3) shows how the g function is defined:

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{3}$$

4. Proposed approach for detection of causal relationships between industrial alarm variables

Here, the alarm variables are defined as discrete time series. For detecting the causal relationships, the problem was modeled as a graph in which the nodes are the alarm variables, and the arcs are their causal relations.

The central idea of the present approach is to use TE to pair-wise compute an initial graph of causal relationships. Because this graph is usually very dense, this work performs some processing on it and then uses the K2 Algorithm to remove the unnecessary arcs. Here, the K2 algorithm was chosen for its simplicity and for being a well-known algorithm in the literature, but it is possible to use others.

It is important to highlight that the application of TE will provide causal relationships in pairs for each time interval considered. However, only the times associated with their maximum information transferred will be considered. These times will be used as the relationship lags

As the causal graphs based on Bayesian Networks do not have the concept of time, these lags are used to shift the time series at every K2 iteration. Therefore, there is no need to represent time explicitly. Moreover, as no cycles are allowed in K2, a strategy for removing cycles in the causality graph was implemented. For analyzing the ambiguity between multiple paths, distinguishing indirect from direct relationships, this work created a technique that added common and virtual parents to the K2 pre-order.

Therefore, the proposed approach makes the detection of the causal relationships in five stages:

1. Generation of information flow graph
2. Application of Significance Threshold
3. Removal of Cycles
4. Generation of Common and Virtual Parents
5. Modification and Computation of K2

To give an overall perspective of how these stages are connected, Fig. 1 represent the flow of the proposed method. For the sake of understanding, throughout this section, a graphical example is provided. It concerns the outputs of the five stages performed by the presented method. It is important to note that the output graph of each stage is the input graph of the following one.

4.1. Generating the information flow graph

Given a set of N discrete-time series, this stage computes a $N \times N$ matrix, where each position, where each position except

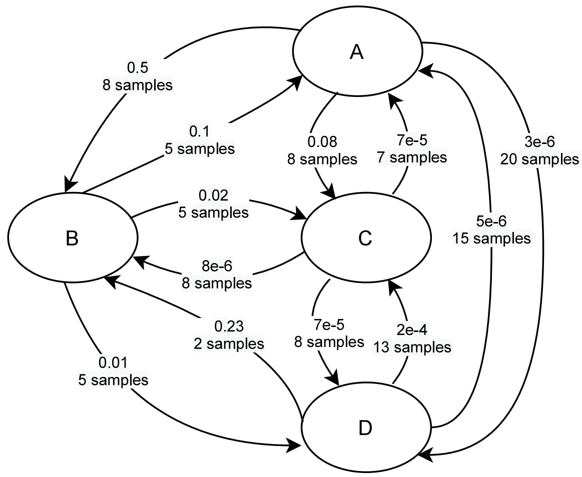

Fig. 2. Information flow graph.

for the ones in the diagonal, presents the transfer entropy from one node to another.

To the horizons time (k and l) it was used a fixed setting. Whereas for the prediction horizon (h), a maximum value was chosen. This was made to identify the information delay of each relationship, that is, the displacement where the highest amount of information was transferred. This form of computing transfer entropy was proposed in [32].

Algorithm 1 shows the computation of the transfer entropy for two series I and J of size N. The output of this stage is shown in Fig. 2

Algorithm 1: Generate graph of transferred entropies and relation delays

Input : A dataset composed of the discrete-time series,
maximum prediction horizon - h,
time horizon - k,
time horizon - l

Output: Direct cyclic graph of transferred entropies

```
foreach time series J ∈ data set do
    foreach time series I ∈ data set do
        for n ← 1 to h do
            if J ≠ I then
                TE_{J->I} ← Σ_{i,i_{t+n},j} p(i_{t+n}, i_t^k, j_t^l) log [p(i_{t+n}|i_t^k, j_t^l) / p(i_{t+n}|i_t^k)]
                entropies ← entropies ∪ {TE_{J->I}}
                TE_Graph ← TE_Graph ∪
                {get_maximum(entropies), h)}
            end
        end
    end
end
Returns a graph representing relationships entropy and delay
```

$$TE_{J->I} \leftarrow \sum_{i,i_{t+n},j} p(i_{t+n}, i_t^k, j_t^l) \log \frac{p(i_{t+n}|i_t^k, j_t^l)}{p(i_{t+n}|i_t^k)}$$

4.2. Application of significance threshold

A statistical threshold was applied to extract significant entropy values. Therefore the insignificant values were replaced with zero.

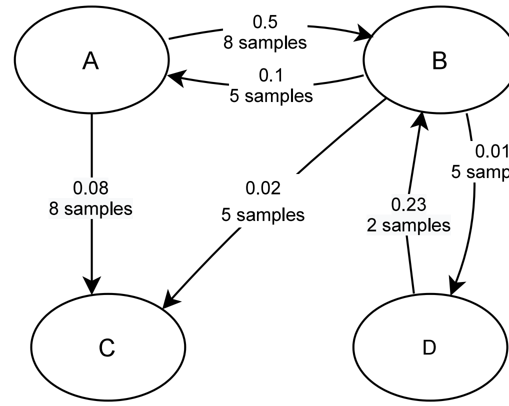

Fig. 3. Example of graph generated on the second stage of proposed methodology.

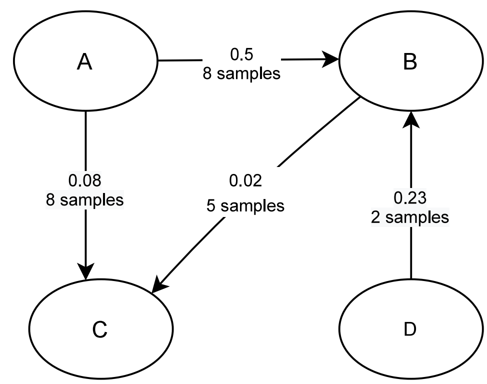

Fig. 4. Example of graph generated on the third stage of proposed methodology.

Given that the distribution of the measured entropies can vary according to the set of time series, it is advisable that the threshold choice is defined through the analysis of the series distribution.

In the empirical studies made in this work, the threshold around the 80% percentile was shown to be suitable. Additionally, studies as the ones performed in [16], the Transfer Entropy distribution average had also shown to be plausible choice.

As the result of this stage, point **C** in Fig. 1, a graph similar to the one given in Fig. 3 is produced. Except for auto-cycles, this graph does not possess any restriction regarding its relationships.

4.3. Removal of cycles

To prepare the graph, generated on the previous stage, to be used by the K2 Algorithm the proposed method does the removal of cycles in the third stage. The approach used to perform this task consists of favoring the strongest relationships; that is, it adds first the ones with the highest amount of information.

Algorithm 2 presents the needed steps to execute this removal. Initially, an empty graph is considered as the target; the graph without cycles. Subsequently, the relationship with the highest amount of entropy is selected and added to the target.

To ensure that the addition of the next relationships will not generate cycles, the algorithm does an exploration at all the genealogy of the node-cause. This exploration is presented in Algorithm 3 and consists of a recursive search, in which the nodes

Algorithm 2: Removal of Graph Cycles

Input: TE graph - G_{TE}
Output: TE graph without cycles G'_{TE}

Assuming $G_{TE} =< E, N >$ *where* $E =< node_c, node_e, weight >$ *as the input*
and $G'_{TE} =< E', N' >$ *where* $E =< node'_c, node'_e, weight' >$ *as the output*
Initialization
 # *The weight of the edge with maximum value of TE*
 $E_{max} \leftarrow max(E)$
 # *A set of nodes that a node cannot have as its son*
 $forbidSons \leftarrow \emptyset$
while *weight*(E_{max}) > 0 **do**
 $forbidSons \leftarrow$ **get_node_genealogy**($node_c$, G'_{TE})
 # *get_node_genealogy is defined in Algorithm 3*
 if $node_e$ **not in** *forbidSons* **then**
  $G'_{TE} \leftarrow G'_{TE} \cup \{E_{max}\}$
 end
 select the next E_{max} *of the graph* G_{TE}
end
* *the subscripts **c** and **e** stands for cause and effect*

"above" the node-cause are marked as in the genealogy of this node. If the node-effect is not present in the genealogy of the node-cause, then no cycle is generated and the relationship is added to the graph. The algorithm repeats this operation until all relationships are analyzed. Fig. 4 shows the output of this stage, assuming the graph shown in Fig. 3 as the input.

Algorithm 3: Get node genealogy

Input : A node which genealogy will be computed - node, TE Graph without cycles - G'_{TE} ,
Output: A set with the node genealogy (parents and ancestors) - genealogy

Assuming $G'_{TE} =< E', N' >$ *where* $E' =< node'_c, node'_e, weight' >$
get_node_genealogy (*node*, G'_{TE}):
 $A_{node} \leftarrow \{n_a \mid e \in E' \text{ and } e =< node'_c = n_a, node'_e = node >\}$
 foreach *node* $\in A_{node}$ **do**
  $A_{node} \leftarrow A_{node} \cup$ **get_node_genealogy** (*node*, G'_{TE})
 end
 return A_{node}
Subscripts **c and **e** stands for cause and effect*

4.4. Generation of common and virtual parents

Some nodes may relate to others in a direct and indirect form in the same graph. When this happens, it is not possible to assert which paths represent the real causal relationship.

To determine the most probable flow in which the information is propagated, this work separates these two kinds of relationships in the fourth stage. This is done by modeling a new graph where the indirect relationships are represented as direct relationships.

The delay of these relationships is defined as the cumulative sum of the delays corresponding to the intermediary paths. In this context, the node-cause of the direct relationship will be called *common parent*, whereas the node-cause of the indirect relationship will be called *virtual parent*.

Algorithm 4: Generation of Common and Virtual Parents

Input : A node which the common and virtual parents will be computed - node, A Graph without cycles - G'_{TE}
Output: A map structure containing all the common and virtual parents of a node and the cumulative delay (lag) correspondent to the respective path among the parent and the node. - ParentsMap

Assuming $G'_{TE} =< E', N' >$ *where* $E' = \{< node'_c, node'_e, weight'_{TE}, delay'_{TE} >\}$
$ParentsMap \leftarrow \{< node_c, sum\_list >\}$
gen_common_virtual_parents (*node*, G'_{TE}, *lagCumSum*, *ParentsMap*)
 $A_{edge} \leftarrow \{e \mid e \in E' \text{ and } node'_e = node\}$
 foreach *edge* $\in A_{edge}$ **do**
  $lagCumSum \leftarrow lagCumSum + lag(edge)$
  $ParentsMap(node'_c) \leftarrow ParentsMap(node'_c) \cup \{lagCumSum\}$
  gen_common_virtual_parents ($node'_c$, G'_{TE}, *lagCumSum*, *ParentsMap*)
  $lagCumSum \leftarrow lagCumSum - lag(edge)$
 end
 return ParentsMap
Subscripts **c and **e** stand for cause and effect*

In this stage, only the lag of the relationships is taken into account. The graphs shown as examples in this subsection will only exhibit the lag attribute as the edge's labels.

To understand what a virtual parent is, note that in the graph shown in Fig. 4 the node C is caused by the node A through two ways: 1 - passing through B ($A \rightarrow B \rightarrow C$); 2 - directly from A: ($A \rightarrow C$). A similar situation occurs with the relationship $D \rightarrow B \rightarrow C$, but this time only the indirect connection exists, from node D to node C.

Thus, the nodes A, and D are said virtual parents of the node C, and the nodes A and B are common parents. Note that in this case the node A appears both in the list of virtual and common parents. This happens because it relates to node C, through different paths. Fig. 5 shows the common and virtual parents of node C.

The node *A_13* results from $A \rightarrow B \rightarrow C$ while the node *D_7* results from $D \rightarrow B \rightarrow C$. The numeral 13 and 7 indicate the cumulative lag for of each relationship respectively.

After computing common and virtual parents for each node in the graph. This stage gather all of the parents to form the K2 pre-order. The Algorithm 4 shows how the computing of virtual and common parents is done. Table 2 shows the common and virtual parents computed for each node.

4.5. Computation of K2

To compute the K2 Algorithm, three entries are needed: a set of n variables (also called nodes), a pre-order, and a dataset of cases. This work proposes two modifications to the K2 Algorithm, regarding respectively the pre-order and dataset of cases:

1. Insertion of virtual nodes
2. Insertion of temporal notion

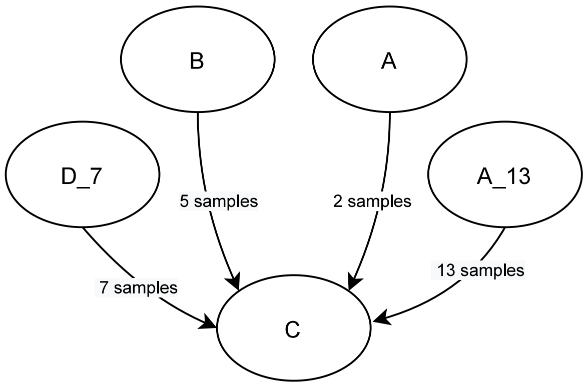

Fig. 5. Parents of node *C*, where *D_7* and *A_13* are virtual and nodes *B* and *A* are common parents.

Table 2
Common and virtual parents for all nodes.

Node	Common parents	Virtual parents
A	–	–
B	A	–
C	A and B	A_13 and D_7
D	–	–

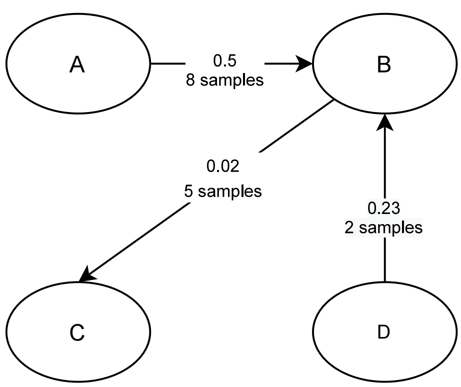

Fig. 6. Example of result after method application.

4.5.1. Insertion of virtual nodes

Regarding the first modification, instead of using only the original nodes in the database, virtual nodes will also be considered in the prior order of K2. As earlier stated, virtual nodes are ancestors that are transformed into parents. This strategy aims to detect the real flow of information on the graph when there are ambiguities.

For example, regarding the node C in Fig. 4 the relationship: $A \rightarrow B \rightarrow C$ is considered an *indirect relationship* , and the relationship: $A \rightarrow C$ is considered *direct*. These two relationships may constitute an ambiguity. Thus, this modification will help the algorithm to identify which path better describes the relationship between the nodes. It is worth stressing that the lag of the virtual relationship is the cumulative sum of the lag between the node and its ancestors.

4.5.2. Insertion of temporal notion

The second modification aims to add the notion of time to the K2 algorithm. In the original K2 algorithm, a row of the dataset of cases is considered an instance of an event. Thus each row represents the state of the variables, and the order between them is irrelevant.

When working with time series, time plays an important role, and the order of events is very significant. Thus, in this work, the K2 is adapted so that the lag between each relationship is reflected in algorithm computation.

This process consists of shifting the *effect-time series* time series of the common and virtual parents according to the delay of the relationship they belong to. This way, the time series are placed in the same time reference.

Algorithm 5 does the generation of this database for each iteration. The version of K2 presented at this work is described by Algorithm 6 and was named as K2-Adapted. It is important to stress that the original data is not modified; only a copy of it is used to generate the new database.

Algorithm 5: Generate dataset of K2 iteration

Input : A dataset with all node cases - database,
a node for which the dataset will be generated - node,
A map structure containing for each node an ParentsMap - TreeMap

Output: a dataset with new columns added aligned according to the delay (lag) between the node and its parents (common/virtual) - dataset

if $List[node] \neq \emptyset$ **then**
  foreach $key, value in TreeMap[node]$ **do**
    foreach lag **do**
      $column \leftarrow$ **shift**(key, -lag)
      $database \leftarrow database \cup \{column\}$
    end
  end
end
**see definition of ParentsMap in algorithm 4*

4.5.3. Reconstruction of the graph

Because virtual parents were added to the original graph, the result delivered by the adapted K2 adapted does not necessarily represent only the correct paths. This occurs because the indirect relationships were modeled as direct in the process of generation of virtual parents.

Hence, a reconstruction of the graph is needed. This reconstruction will retrieve the indirect relations, together with its respective delays. Fig. 6 shows how it would be a possible result delivered by the method application.

5. Case study

A case study was used to apply and evaluate the performance of the proposed method, which focuses on the detection of causal relationships between industrial alarm variables. The case study was realized using the simulator TEP, which is a well-established benchmark for research in monitoring, control, and fault-detection of industrial processes.

The activity flow of this case study includes some previous steps to the application of the proposed method. It starts with the TEP simulation, where the process data was generated, and subsequently used for generating the alarm variables.

Secondly, because of the characteristics of the industrial process, a mean average filter was applied on the alarm data in order to reduce the noisy and false alarms. Finally, this work applied the methodology to the data. Fig. 7 shows the sequence of steps needed for the reproduction of this experiment.

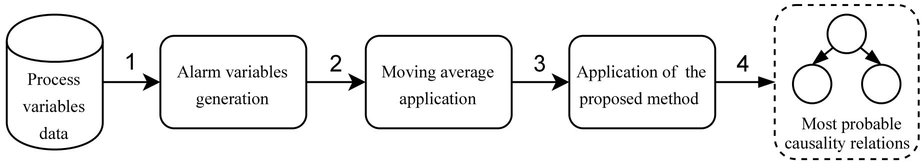

Fig. 7. Case study activity flow.

Algorithm 6: K2 Adapted

Input: a set of n nodes - $(x_1 \ldots x_n)$
an ordering for the nodes ,
a dataset of n columns and m cases - database
Output: Bayesian Network Topology

Be π_i the parents of node i
for $i \leftarrow 1$ **to** n **do**
  $\pi_i \leftarrow \emptyset$
end
for $i \leftarrow 1$ **to** n **do**
  dataSetIteration ← generateDataSet(database)
  #Generation of the database according to delays between nodes. - See Algorithm 5
  $P_{old} \leftarrow f(i, \pi_i)$
  #f function is computed using Eq. (2), based on the dataset of the iteration.
  while *True* **do**
    Select the node $x_j \in pred_{x_i} \setminus \pi_i$ *that maximizes:* f(i,$\pi_i \cup \{x_j\}$)
    # where $pred_{x_i}$ *is the set of nodes that precedes* x_i
    $P_{new} \leftarrow$ f(i,$\pi_i \cup \{x_j\}$)
    $sigma \leftarrow P_{new} > P_{old}$
    if *sigma = True* **then**
      $P_{old} \leftarrow P_{new}$
      $\pi_i \leftarrow \pi_i \cup x_j$
    end
    if ***not***$(pred_{x_i} = \emptyset)$ **then**
      $pred_{x_i} \leftarrow pred_{x_i} \setminus x_j$
    end
    if *(not sigma or* $pred_{x_i} = \emptyset$*)* **then**
      break
    end
  end
  returns reconstructed graph
end

5.1. Tennessee eastman process

The TEP is a plant-wide industrial process based on an actual chemical process. It is composed of eight components: A, B, C, D, E, F, G, and H, which through exothermic reactions produce two products and two byproducts. Eq. (4) shows the reactions that involve these components and the products that are obtained.

$$\begin{array}{rl} A(g) + C(g) + D(g) \rightarrow G(liq), & \text{Product 1} \\ A(g) + C(g) + E(g) \rightarrow H(liq), & \text{Product 2} \\ A(g) + E(g) \rightarrow F(liq), & \text{Byproduct} \\ 3D(g) \rightarrow 2F(g), & \text{Byproduct} \end{array} \tag{4}$$

The process is composed of five major units: the reactor, the product condenser, the vapor–liquid separator, the recycle compressor, and the product stripper. It has a set of 12 manipulated variables and 41 measurement variables, listed in Tables 3–5 of [8].

Table 3
Experimental setup of the simulation of TEP.

Parameters	Settings
Simulation time	760 h
Disturbance length	1 h
Stabilization length	14 h
Disturbance	A feed loss
Number of disturbance activation	50 times
Sampling frequency	100 samples per hour

Table 4
Description of process variable analyzed.

Variable	Description
x_1	A Feed (stream 1)
x_2	D Feed (stream 2)
x_3	E Feed (stream 3)
x_6	Reactor feed rate (stream 6)
x_7	Reactor pressure
x_8	Reactor level
x_9	Reactor temperature
x_{21}	Cooling water outlet temperature

To simulate real failure situations in the plant, the TEP provides a set of 20 disturbances [8]. The authors divided these disturbances into two categories: set-point change and load change, which are often used to perform and evaluate control strategies, studies on plant behavior, relations between variables, among others.

5.2. Experimental setup

To produce the dataset, a 760 h simulation was conducted. The simulation was composed of 50 activations of the disturbance #6, ***A Feed Loss***, which is an interruption to the supply of component A on stream 1 in the plant. This setup was chosen to produce a consistent scenario in which the failure effect could be observed.

The fault activation mainly affects the reactor unit of the process. However, because of the interconnections of the plant, it can also influence other parts of the process. The full experimental setup of this experiment is shown in Table 3.

Because the reactor is the most affected unit, this work is focused on analyzing only the process variables present in this unit and the variables related to the feed stream. The description of these variables is shown in Table 4. The scheme of the plant, obtained from [8] exposing the localization of the variables, is shown in Fig. 8.

The generation of the alarm variables was made by the application of a statistical threshold, the well-known 3σ. For the high-level alarm, it was used the Eq. (5). Similarly, the low-level alarm variables were computed according to Eq. (6). In these equations, μ' is the mean of the process variable under normal conditions, and σ' is the standard deviation. The alarm settings, for each type of alarm, are shown in Table 5.

$$H_a = \begin{cases} 1, & V_p(t) \geq \mu' + 3\sigma' \\ 0, & V_p(t) < \mu' + 3\sigma' \end{cases} \tag{5}$$

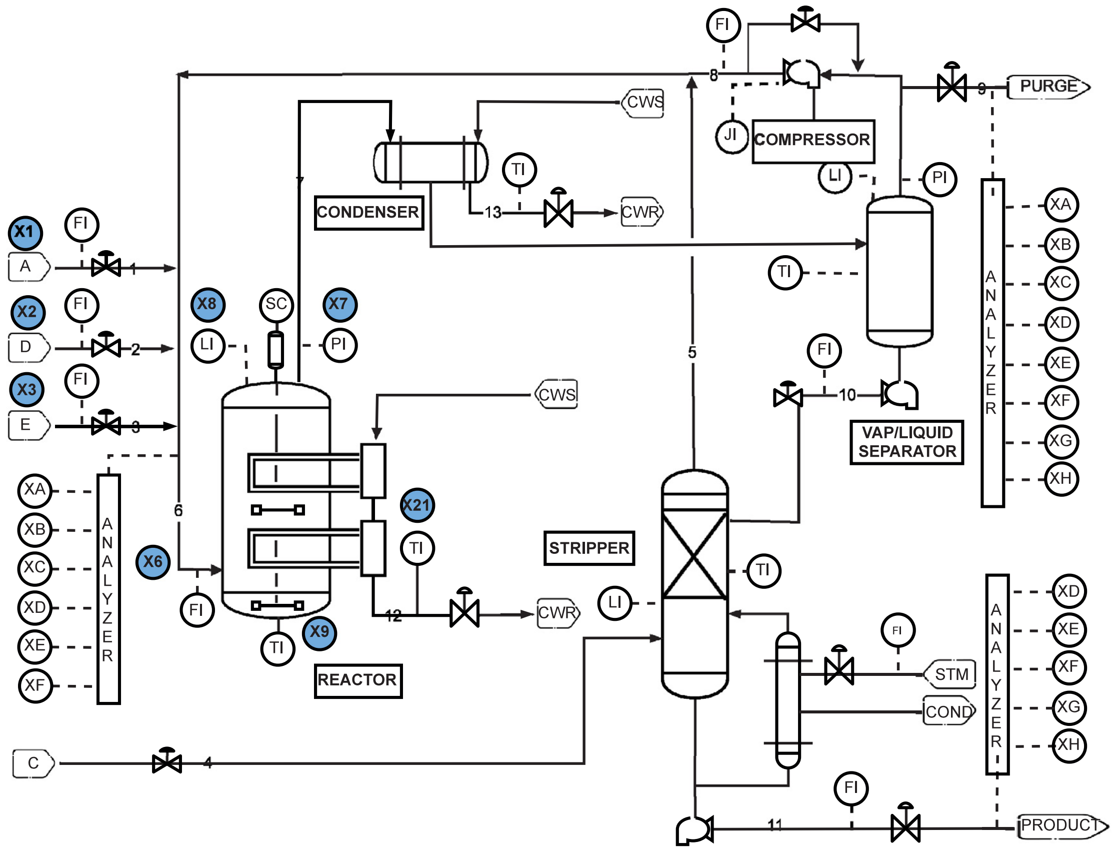

Fig. 8. TEP schematic adapted from [8].

Table 5
Alarm settings configured according to the 3σ rule.

Process variable	Low threshold	High threshold
x_1	0.249	0.284
x_2	3597.394	3711.109
x_3	4371.271	4514.069
x_6	46.954	48.235
x_7	2796.971	2803.033
x_8	63.494	66.505
x_9	122.867	122.932
x_{21}	102.409	102.527

$$L_a = \begin{cases} 1, & V_p(t) \le \mu' - 3\sigma' \\ 0, & V_p(t) > \mu' - 3\sigma' \end{cases} \quad (6)$$

5.3. Processing the data

Before applying the method it was necessary to apply a filter on the alarm data. This operation aimed to reduce the effect of chattering/false alarms.

The technique utilized to filter this type of alarm was the moving average, which creates a new series with each sample being calculated as the average of the corresponding set of samples of the alarming series. The size of the average is defined by the user.

To keep the alarm variables binary if the average computed is greater than 0.5, then it is rounded to 1; otherwise it is set as 0. For example, suppose a moving average of 3 samples is computed for the following alarm sequence: [1, 1, 0, 0, 1, 1], the result of the filter application would be:

[−, −, 0.66, 0.33, 0.33, 0.66]

Table 6
Settings for the application of the methodology.

Parameter	Settings
Moving average length	5 samples
Prediction Horizon h (range)	1–50 samples (1–30 mins)
Time Horizon k	1 sample
Time Horizon l	1 sample
Transfer Entropy threshold	percentile 83%

The rounded average vector will be [1, 0, 0, 1]. Note that as 3 samples average was computed the two first samples of the data were lost.

5.4. Settings used in the method execution

The settings for the *moving average* filter, as well as the parameters for the computation of the TE and the threshold of significance, are shown in Table 6. The parameterization of the moving average was chosen based on the study made by [9], in which the selected setting obtained the best result when working with a similar scenario.

To define the prediction horizon band, an analysis of the dynamics of the process was made, taking into account the approximated time that the variables took to be entirely affected by the disturbance.

For this case study, the parameters k and l were set as 1, as in Schreiber's example in [31], which is the considered the first-order form of TE. However, it is stressed that these parameters can differ in other applications of the proposed method.

Because the K2 algorithm is a non-parametric method, is not needed to define any parameter for its usage. Additionally, as this

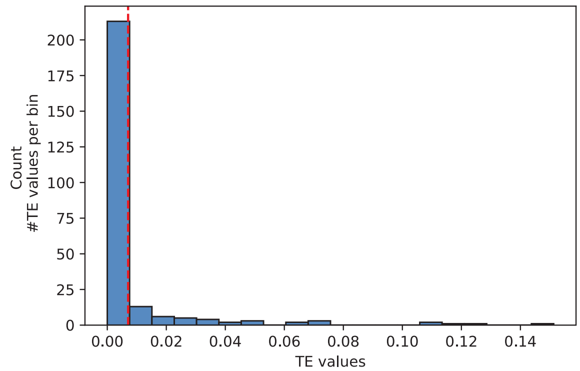

Fig. 9. Histogram of transferred entropies.

work does not compute inference on the network, there are no conditional probability tables.

To choose an adequate threshold for selecting the most relevant relationships, a histogram with the entropy data was plotted. This histogram is shown in Fig. 9 and it is possible to notice the most of the values are very near to zero entropy.

Besides, the shape of the distribution is strongly heavy-tailed. Hence, the most relevant values (i.e., the values representing a higher level of information) are located in the tail of the distribution. Thus, to address these concerns the threshold of significance for this case was defined as the 83% percentile of the data. This value covered the distribution spectrum of most variables, and it is placed at the mean of the data [16]. This threshold is indicated by the red-dashed vertical line in Fig. 9.

6. Results and discussion

This section presents and discusses the results obtained by the application of the method. To give a better understanding of this case study, an overview of the behavior of the control plant studied in this experiment is initially made.

For a better understanding, the intermediary results, the graphs delivered in the stages: significance threshold application, removal of cycles, and the result of the complete application are shown in this section.

6.1. Overview of the process variables and behavior of the system

To understand how the variables behave when the disturbance is applied, Fig. 10 shows the process variables trend. For each variable, the horizontal lines indicate the alarm threshold for the abnormal situation, with the top line indicating the threshold for the high-level alarm and the bottom line indicating it for the low-level alarm.

Knowing that the disturbance causes a loss on the A feed and that this component is part of the reaction to form the Product 1 (see Eq. (4)), it is expected that the first variable to expose the effects of the disturbance will be the variable x_1, which concerns the monitoring of the A feed. Thus, as shown in Fig. 10, the process variable x_1 falls below the normal level, consequently producing a low alarm.

Because the reactions produced in the reactor must maintain the correct amount of each component, the amount of the components D and E cannot be changed to compensate for the diminishing of the A component.

Thus, the variables x_2 and x_3 are not expected to be affected by the disturbance. Looking at the graphic of these variables (second and third graphics on Fig. 10) it is possible to note that they sometimes exceed the high and low threshold, causing low and high alarm activation at these moments. However, this behavior is probably related to the measuring noise of the variables, which is internally simulated on TEP.

Because the A feed was cut off and all TEP reactions are exothermic, the reactor temperature (variable x_9) tends to decrease since the reactions occur in this component. The graphic shown for this variable confirms this behavior. As a consequence of the decrease of reactor temperature, its level (*variable* x_8) and pressure (*variable* x_7) also decrease. Moreover, the Reactor Feed Rate (*variable* x_6), which is directly connected to stream 4, has its value diminished. That occurs because the feed rate is directly proportional to the feed on the streams.

The graphics of these three last variables, in Fig. 10, confirms the behavior discussed by showing that the process variables values go under the low threshold at the moment of disturbance application.

The graphic of the variable x_{21}, last graphic on Fig. 10, shows that there is a growth in its value during the disturbance application. This variable measures the cooling water outlet temperature of the reactor. When the temperature of the reactor increases, the cooling system injects water at a lower temperature to dissipate the heat in the reactor. Hence, when this water comes out of the cooling duct, its temperature tends to be higher.

However, because of the disturbance activation, the reactor temperature tends to drop quickly, which forces the controller of the plant to "close" the outlet of the cooling water in an attempt to maintain the reactor temperature and stop it from lowering the heat. Therefore, during the disturbance, the cooling water temperature increases, causing a high alarm, as shown in the graphic of the variable x_21.

6.2. Intermediary results of the method application

In this case study, a moving average filter [44] was applied to the alarm data. Although data processing is not an obligated step, it was made for reducing the chattering and false alarms of the industrial process. Because the graphs obtained after threshold application and removal of cycles were too dense, it was opted not to exhibits the delays of the relationships in the figure.

To extract the most relevant relationships, the proposed method applies a threshold to the output of the TE application, based on the data distribution. The threshold reduced the number of arcs by 48.78%. The result achieved in this phase is shown in Fig. 11.

Although the graph is consistently less dense in here, it still does not represent the true flows of information. Thus, the method applies a removal of the cycles based on the highest values of entropy. Fig. 12 shows the graph obtained at this stage.

It is possible to note that although the cycles had already been removed, it remains a dense graph. It possesses relationships that possibly indicate the same information but through different paths.

For example, the variable x_1_low is connected to the variable x_9_high through four different paths, which is expected behavior because the computation of TE is pairwise, consequently producing an almost complete graph.

Consequently, by the analysis of this graph, it is not possible to infer which one is the path that better represents the flow of information. Thus, the application of K2-Modified intends to "prune" these spurious relationships, revealing the best parent set for each node.

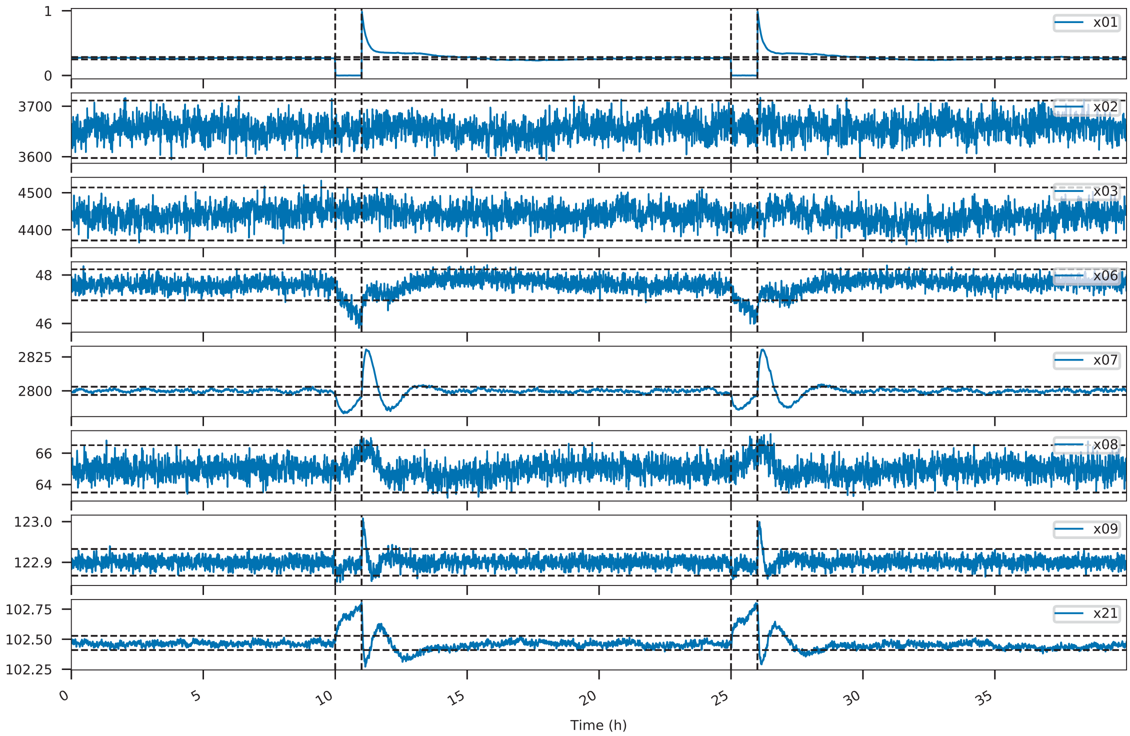

Fig. 10. Process variables overview.

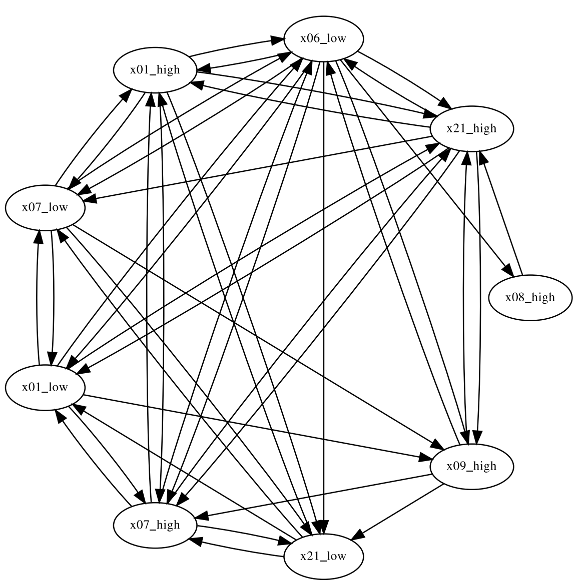

Fig. 11. Graph threshold application, with reduction of 48.78% in the number of arcs.

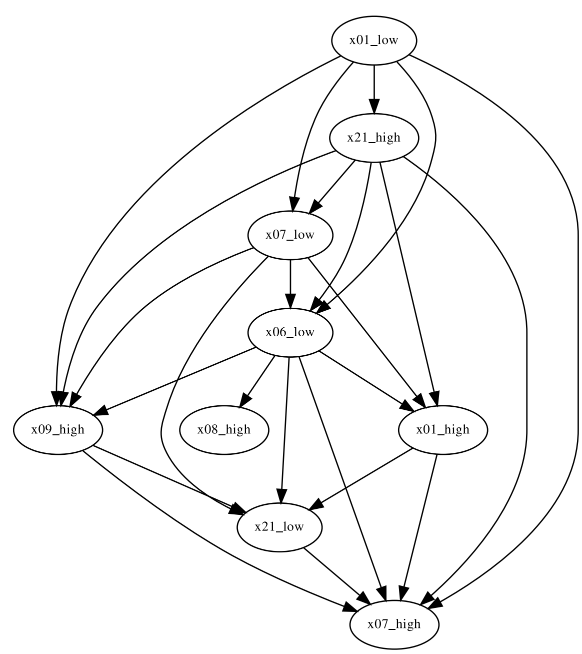

Fig. 12. Graph after removal of cycles.

6.3. Result analysis

The result obtained by applying the proposed method is shown in Fig. 13. The settings used in this case study are shown in Table 6.

Starting the analysis from the root node, the first relationship (x_1_*low* $\rightarrow$ x_{21}_*high*) indicates an increase in the cooling water temperature (alarm x_{21}_*high*), caused by the loss on the A supply. Because the A feed was cut off by the disturbance is expected that the alarm that monitors the drop of feed of component A (x_1_*low*) was the first to be triggered. Moreover, assuming that the disturbance had already been on for a reasonable time, this relationship reflects the behavior of the plant's controllers, which retains the cooling water to increase the reactor temperature.

It is essential to highlight that although this relationship holds a delay of 49 samples (approximately 29 min), this period of time does not indicate that the activation of the alarm variable x_{21}_*high* occurred 29 min after the activation of the variable x_1_*low*. This delay indicates the moment with the highest amount of information transferred.

The next relationship between the alarms x_{21}_*high* and x_6_*low* (reactor cooling water temperature and feed rate) can be an

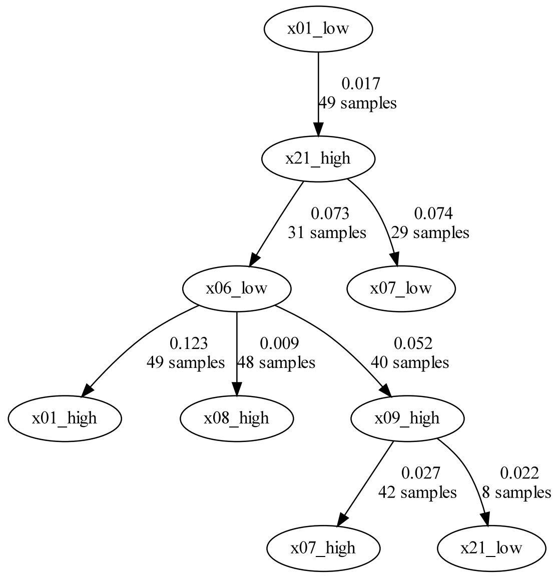

Fig. 13. Result delivered by the method application.

indicator that the elevation of reactor temperature may have led the system to diminish the feed rate because increasing it would elevate the temperature.

The relationship between x_{21}_*high* and x_7_*low* does not seem to be correctly identified. One possible explanation for this misunderstanding is that this relationship is derived from the process dynamic. What happens here is that the process variable x_{21} (Cooling water outlet temperature) is faster than the variable x_7 (Reactor pressure). Thus, the occurrence of one precedes the other, leading the method to infer a causality relationship.

Returning to the edges derived from the alarm variable x_6_*low* (Reactor feed rate), there are three causal relationships. The first one relates this variable with the variable x_9_*high* (Reactor Temperature), which reaffirms the existence of an increase in the reactor temperature and causes a high-level alarm.

The second (x_6_*low* $\rightarrow$ x_1_*high*) results from the controller's behavior. To restrain the disturbance, when the amount of A component diminishes, the controller forces an abrupt increase on the feed, generating an overshoot, and therefore causing an increase in the level of the reactor.

The third relationship relates the low-level alarm from the variable x_6 with the high-level alarm from the variable x_8_*high* (Reactor Level), which is expected because pressure and temperature are directly proportional.

The last set of relationships relate the alarm x_9_*high* with the alarms x_7_*high* and x_{21}_*low*. The first relationship (x_9_*high* $\rightarrow$ x_7_*high*) regards to the elevation of the reactor temperature, which cause the pressure on it to increases.

Moreover, the relationship between the alarms x_9_*high* and x_{21}_*low* can be interpreted as a side effect from the release of the cooling water, when the plant controllers increase the amount of component A in the reactor.

It is important to acknowledge that because a real plant-wide simulation, from a real chemical industry, was used, there are no templates nor ground truth to evaluate metrics such as precision, F1-Score, Structural Hamming Distance, or Balanced Scoring Function. However, the graph was analyzed for specialists of the problem, and the relationships found are shown to be consistent according to the applied disturbance.

It was made available in GitHub platform[2] all of the results of this paper, the codes developed, as well as the database used in this work.

7. Conclusions

This work proposed a new method to identify the causal relationships between industrial alarm variables using transfer entropy and a new version K2 algorithm. Because the TE is a pairwise method, to generate a graph assuming each value of entropy as an indicator of a causal relationship produces an almost complete graph.

Thus, to provide an accurate graph it was applied a set of processing on the data delivered by TE, which was then used as a pre-order for the K2 algorithm. To handle the dynamics imposed by the time series some modifications were done in the K2 algorithm, leading to an adapted version: K2-Adapted.

By the analysis of the results obtained in this work, it was possible to see that the proposed method produced a graph of causal relationships consistent with the structure of the industrial plant and the behavior of the process variables analyzed.

As this work proposed a machine-learned previous order as well as added modifications on the K2 Algorithm, the proposed method has intrinsically a greater complexity than TE and K2 if they were used individually.

The proposed one takes into account the transfer entropy complexity $O(2^{(k+l+h)})$ plus the complexity of original K2 Algorithm $O(m \cdot p^4 \cdot r)$ [37] with the addition of the complexity of K2 modifications and post-processing of the data.

Finally, the study of the TE brought up some concerns that will be addressed in future work. Such as a careful analysis of the effects of scaling the time horizons k and l, and on the choice of the prediction horizon h. Another future work proposed is a search of the available methods of structural learning in Bayesian networks to provide a comparison with the result provided by this work.

CRediT authorship contribution statement

Rute Souza de Abreu: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Supervision, Writing – original draft, Validation, Writing – review & editing. **Yuri Thomas Nunes:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Supervision, Validation, Writing – review & editing. **Luiz Affonso Guedes:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Validation, Writing – review & editing. **Ivanovitch Silva:** Formal analysis, Investigation, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil – Finance Code 001, and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil - Process number: 142359/2019-0.

[2] Github link for the developed codes: https://git.io/JtnGt.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jprocont.2021.09.001.

References

[1] P. Goel, A. Datta, M.S. Mannan, Industrial alarm systems: Challenges and opportunities, J. Loss Prev. Process Ind. 50 (2017) 23–36.
[2] M. Bransby, J. Jenkinson, The Management of Alarm Systems, Citeseer, 1998.
[3] D.H. Rothenberg, Alarm Management for Process Control: A Best-Practice Guide for Design, Implementation, and Use of Industrial Alarm Systems, Momentum Press, 2009.
[4] J. Zheng, C. Zhao, Enhanced canonical variate analysis with slow feature for dynamic process status analytics, J. Process Control 95 (2020) 10–31.
[5] A. Bezerra, I. Silva, L.A. Guedes, D. Silva, G. Leitão, K. Saito, Extracting value from industrial alarms and events: A data-driven approach based on exploratory data analysis, Sensors 19 (12) (2019) 2772.
[6] I. Silva, L.A. Guedes, P. Portugal, F. Vasques, Reliability and availability evaluation of wireless sensor networks for industrial applications, Sensors 12 (1) (2012) 806–838.
[7] J. Wang, F. Yang, T. Chen, S.L. Shah, An overview of industrial alarm systems: Main causes for alarm overloading, research status, and open problems, IEEE Trans. Autom. Sci. Eng. 13 (2) (2015) 1045–1061.
[8] J.J. Downs, E.F. Vogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (3) (1993) 245–255.
[9] R.S. de Abreu, A Methodology for Detection of Causal Relationships between Discrete Time Series on Systems (Master's thesis), Universidade Federal do Rio Grande Norte, Natal, Rio Grande do Norte, Brasil, 2019.
[10] W. Hu, J. Wang, T. Chen, A new method to detect and quantify correlated alarms with occurrence delays, Comput. Chem. Eng. 80 (2015) 189–198.
[11] S. Lai, F. Yang, T. Chen, Online pattern matching and prediction of incoming alarm floods, J. Process Control 56 (2017) 69–78.
[12] G. Dorgo, J. Abonyi, Sequence mining based alarm suppression, IEEE Access 6 (2018) 15365–15379.
[13] W. Hu, T. Chen, S.L. Shah, Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods, IEEE Trans. Ind. Electron. 65 (9) (2018) 7290–7300.
[14] Q.-Q. Meng, Q.-X. Zhu, H.-H. Gao, Y.-L. He, Y. Xu, A novel scoring function based on family transfer entropy for Bayesian networks learning and its application to industrial alarm systems, J. Process Control 76 (2019) 122–132.
[15] Y. Cheng, I. Izadi, T. Chen, Pattern matching of alarm flood sequences by a modified Smith–Waterman algorithm, Chem. Eng. Res. Des. 91 (6) (2013) 1085–1094.
[16] L. Cao, F. Yu, F. Yang, Y. Cao, R.B. Gopaluni, Data-driven dynamic inferential sensors based on causality analysis, Control Eng. Pract. 104 (2020) 104626.
[17] X. Chen, J. Wang, J. Zhou, Process monitoring based on multivariate causality analysis and probability inference, IEEE Access 6 (2018) 6360–6369.
[18] Y. Shu, J. Zhao, Data-driven causal inference based on a modified transfer entropy, Comput. Chem. Eng. 57 (2013) 173–180.
[19] Y. Liu, H.-S. Chen, H. Wu, Y. Dai, Y. Yao, Z. Yan, Simplified Granger causality map for data-driven root cause diagnosis of process disturbances, J. Process Control 95 (2020) 45–54.
[20] J. Su, D. Wang, Y. Zhang, F. Yang, Y. Zhao, X. Pang, Capturing causality for fault diagnosis based on multi-valued alarm series using transfer entropy, Entropy 19 (12) (2017) 663.
[21] W. Hu, T. Chen, S.L. Shah, M. Hollender, Cause and effect analysis for decision support in alarm floods, IFAC-PapersOnLine 50 (1) (2017) 13940–13945, http://dx.doi.org/10.1016/j.ifacol.2017.08.2215, URL: https://www.sciencedirect.com/science/article/pii/S2405896317328872. 20th IFAC World Congress.
[22] W. Hu, J. Wang, T. Chen, S.L. Shah, Cause-effect analysis of industrial alarm variables using transfer entropies, Control Eng. Pract. 64 (2017) 205–214.
[23] A. Wikner, J. Pathak, B. Hunt, M. Girvan, T. Arcomano, I. Szunyogh, A. Pomerance, E. Ott, Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems, Chaos 30 (5) (2020) 053111.
[24] V. Rodrigo, M. Chioua, T. Hagglund, M. Hollender, Causal analysis for alarm flood reduction, IFAC-PapersOnLine 49 (7) (2016) 723–728.
[25] J. Wang, Z. Yang, J. Su, Y. Zhao, S. Gao, X. Pang, D. Zhou, Root-cause analysis of occurring alarms in thermal power plants based on Bayesian networks, Int. J. Electr. Power Energy Syst. 103 (2018) 67–74.
[26] Y. Wang, Y. Liu, F. Khan, S. Imtiaz, Semiparametric PCA and bayesian network based process fault diagnosis technique, Can. J. Chem. Eng. 95 (9) (2017) 1800–1816.
[27] M.T. Amin, S. Imtiaz, F. Khan, Process system fault detection and diagnosis using a hybrid technique, Chem. Eng. Sci. 189 (2018) 191–211.
[28] M.T. Amin, F. Khan, S. Imtiaz, Fault detection and pathway analysis using a dynamic Bayesian network, Chem. Eng. Sci. 195 (2019) 777–790.
[29] H. Lee, C. Kim, S. Lim, J.M. Lee, Data-driven fault diagnosis for chemical processes using transfer entropy and graphical lasso, Comput. Chem. Eng. 142 (2020) 107064.
[30] L. Ma, J. Dong, K. Peng, Root cause diagnosis of quality-related faults in industrial multimode processes using robust Gaussian mixture model and transfer entropy, Neurocomputing 285 (2018) 60–73.
[31] T. Schreiber, Measuring information transfer, Phys. Rev. Lett. 85 (2) (2000) 461.
[32] M. Bauer, J.W. Cox, M.H. Caveness, J.J. Downs, N.F. Thornhill, Finding the direction of disturbance propagation in a chemical process using transfer entropy, IEEE Trans. Control Syst. Technol. 15 (1) (2006) 12–21.
[33] J. Pearl, Bayesian Networks, UCLA: Department of Statistics, 2011.
[34] N. Friedman, I. Nachman, D. Peér, Learning Bayesian network structure from massive datasets: the sparse candidate algorithm, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 1999, pp. 206–215.
[35] L.M.d. Campos, A scoring function for learning Bayesian networks based on mutual information and conditional independence tests, J. Mach. Learn. Res. 7 (Oct) (2006) 2149–2187.
[36] R.R. Bouckaert, Probabilistic network construction using the minimum description length principle, in: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Springer, 1993, pp. 41–48.
[37] G.F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Mach. Learn. 9 (4) (1992) 309–347.
[38] P. Spirtes, C.N. Glymour, R. Scheines, D. Heckerman, Causation, Prediction, and Search, MIT Press, 2000.
[39] G.F. Cooper, A simple constraint-based algorithm for efficiently mining observational databases for causal relationships, Data Min. Knowl. Discov. 1 (2) (1997) 203–224.
[40] E.V. Strobl, A constraint-based algorithm for causal discovery with cycles, latent variables and selection bias, Int. J. Data Sci. Anal. 8 (1) (2019) 33–56.
[41] I. Tsamardinos, L.E. Brown, C.F. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, Mach. Learn. 65 (1) (2006) 31–78.
[42] J.M. Ogarrio, P. Spirtes, J. Ramsey, A hybrid causal search algorithm for latent variable models, in: Conference on Probabilistic Graphical Models, 2016, pp. 368–379.
[43] M.L. Wong, K.S. Leung, An efficient data mining method for learning Bayesian networks using an evolutionary algorithm-based hybrid approach, IEEE Trans. Evol. Comput. 8 (4) (2004) 378–404, http://dx.doi.org/10.1109/TEVC.2004.830334.
[44] N.J. Salkind, L.A. Shaw, Statistics for People Who (Think They) Hate Statistics using R, Sage publications, 2019.