Causal analysis of time series

Polster–Prieto Florian, ENSEA, florian.polster-prieto@ensea.fr

19/08/2022

Abstract

This report introduces the major concepts of temporal causal analysis as well as models and algorithms able to discover causal relationships in time series. We first describe the graphs used in causal analysis and the associated definitions before presenting different methods and their characteristics. We then present causal discovery algorithms, and a way to evaluate them. To do so, we introduce suitable datasets and metrics before evaluating the algorithms on a specific dataset. We then try to analyze a real dataset with the introduced algorithms by taking into account our evaluation. The results of the analysis has limited informative value, due to the scarcity of the samples, the required interpolation of the data and the applied differentiation step.

Contents

Presentation of the company

I made my internship in the ETIS-lab[9], which is a joint research department between CYU Cergy Paris University, ENSEA Graduate School of Electrical Engineering and CNRS/INS2I. ETIS stands for "Equipes Traitement de l'Information et Systèmes", which can be translated as "Teams Information Processing and Systems". The internship started the 09 May 2022 and ended on the 2 September 2022, and was in the field of Causal analysis of time series. ETIS is a research lab specialized in theory of information and information processing. The main areas of work of the lab are re-configurable chip systems, data analysis, image indexing, developmental robotics, information theory, telecommunications and artificial intelligence. It is currently directed by Prof. Olivier Romain assisted by Dr. Veronica Belmega. It is divided in 4 groups: CELL (Smart Embedded Systems), MIDI (Multimedia Indexing & Data Integration), NEURO (Bio inspiration for robots, autonomous AI) and ICI (communications, information theory, signal processing and imaging). The governance scheme of the lab is depicted in Figure1.

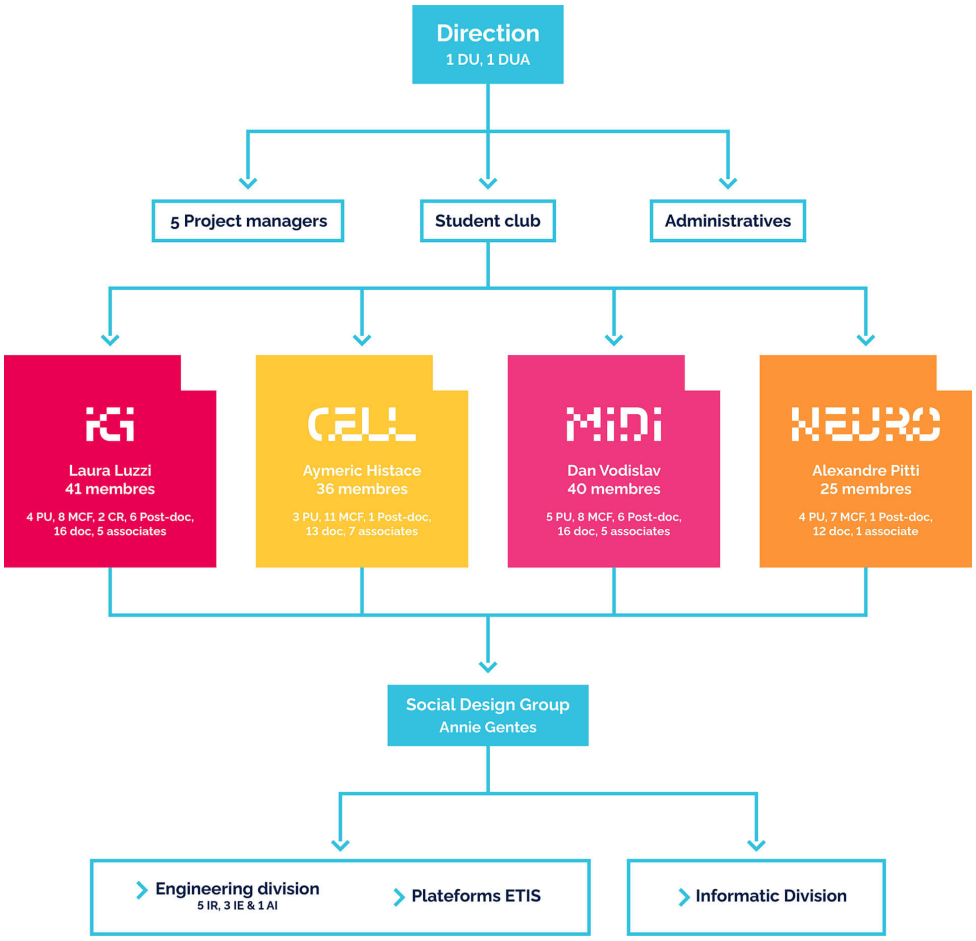


Figure 1: ETIS governance scheme

The ETIS lab has premises at two different locations: at the ENSEA school (6 avenue du Ponceau, Cergy, where ICI, CELL and MIDI groups work) and at the CY University, called Saint Martin site (2 avenue Adolphe-Chauvin, Pontoise, where NEURO and MIDI groups work). The lab possesses 62 servers, for a total of 1230 CPUs and 10 GPUs, which are available from every computer through a double ssh tunnel. During my internship, I worked with two members of the MIDI team, M. Vassilis CHRISTOPHIDES and M. Michele LINARDI.

1 Introduction

1.1 Basic Notations

This report is based on the methodology of previous studies. As such, we will use the same notations and definitions. Here are the standard notations used in the context of temporal causal analysis[2].

Notation	Description
X,Y,X^p,X^q	Random variables
$\mathscr{X}$	Multivariate time series $\mathscr{X}^1$,...,$\mathscr{X}^d$ of size d
$\mathscr{X}^p$	p-th time series of $\mathscr{X}$
$\mathscr{X}_t^p$	Time series $\mathscr{X}^p$ at timestamp t
$X \perp\!\!\!\perp Y$	X and Y are independent
$X \not\perp\!\!\!\perp Y$	X and Y are not independent
$X \to Y$	X is a cause of Y and Y is an effect of X
$X \not\to Y$	X is not a cause of Y
$X \leftrightarrow Y$	X and Y have a common confounder
$X - Y$	There is an undirected link between X and Y
$X \circ\!\!-\!\!\circ Y$	The connection between X and Y is unoriented in both directions
$Par(X, G)$	Set of parents of X in the graph G
$Hom(\mathscr{X}_{t-i}^p, \mathscr{X}_t^q, G)$	Set of vertices pairs $(\mathscr{X}_{k-i}^p, \mathscr{X}_k^q)$ homologous to $(\mathscr{X}_{t-i}^p, \mathscr{X}_t^q)$
$Sepset(X, Y)$	Separation set of X and Y

Table 1: Notations used in this report.

A separation set between two variables is the smallest subset making the two variables independent. A set of homologous vertices $(\mathscr{X}_{k-i}^p, \mathscr{X}_k^q)$ represent the same dependency as the pair $(\mathscr{X}_{t-i}^p, \mathscr{X}_t^q)$ shifted in time.

1.2 Directed Acyclic Graphs

The main objective of causal analysis is to build a graph representing the links between given variables. In this context, we use Directed Acyclic Graphs (noted DAG), which represent the variables as vertices and the links as edges. An edge ($\to$) depicts a link between a cause (or parent) and its consequence, while two unconnected vertices are said to be conditionally independent. A DAG can be represented under the assumption of causal sufficiency.

Definition 1: Causal sufficiency A set of variables is *causally sufficient* if all common causes of all variables are observed.

DAGs introduce two basic structures, depicted on Figure 1: *confounders* and *colliders*. A *confounder* X^r corresponds to a variable that is a common cause of two other variables X^p and X^q, while a *collider* X^u is caused by two uncorrelated variables X^s and X^t.

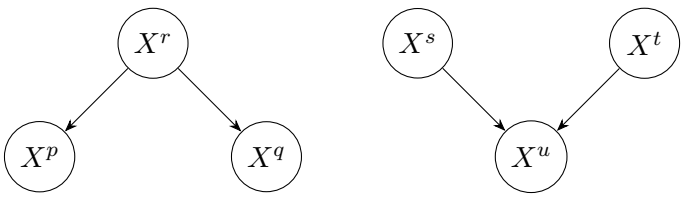


Figure 2: Confounder (left) and Collider (right)

A DAG encapsulates correctly (to some degree) a given probability distribution if it satisfies the following assumptions. Note that *Faithfulness* is stronger than *Minimality* but may not be achieved by some methods.

Definition 2: Markov condition *Necessary and sufficient condition for a random distribution to be compatible with a DAG.* Every variable is independent of all its non-descendants (in the DAG) and conditionnaly independent in its parents.

Definition 3: Minimality condition For a given DAG of the probability distribution X, the graph satisfies the minimality condition if X is not compatible with any proper subgraph of the DAG.

Definition 4: Faithfulness A DAG and a compatible probability distribution X are faithful to one another if all and only the conditional independence relations given in X satisfy the Markov condition on the DAG.

1.3 Maximal Ancestral Graphs

When the causal sufficiency assumption is not satisfied, the causal relations between the variables can not be represented trough a DAG, hence the need to introduce Maximal Ancestral Graphs (noted MAG). MAGs are an extension of DAGs as they include the possible hidden relations that are not observed in the considered data. These hidden relations are either *hidden confounders* or *hidden effects*. A hidden confounder is a common cause of two variables, and is depicted as a double arrowed edge (Figure 2 left). A hidden effect is a hidden variable that induces a dependence between two observed variables, and is depicted as a non arrowed edge (Figure 2 right). A hidden effect is not necessarily a collider, as the dependencies between variables are not restricted to their parents.

Figure 3: Hidden confounder (left) and hidden effect (right)

1.4 Causal graphs for time series

To represent the causal dependencies between time series, we consider the *temporal priority condition* to render the data asymmetrical in time, which can be useful for orienting the dependencies in time.

Definition 3: Temporal priority condition A causal relation between two variables satisfies the temporal priority if the cause happens before the effect.

However, due to the sampling frequency of the data, it is possible that two events occurring at two different time instants are discovered as simultaneous. The concept of a lag specific to each connection has to be introduced to represent correctly the different variables: $\mathscr{X}^p_{t-i} \to \mathscr{X}^r_t$ implies that $\mathscr{X}^p$ at time $t-i$ causes $\mathscr{X}^r$ at time t with a lag of $i \geq 0$. In the specific case where $p = r$, the time lag i has to be strictly positive.

Full Time Causal Graphs

A full time causal graph has the same architecture as a DAG and represents the entirety of the data at every timestamp $t \in \mathbb{Z}$. In practice, it is either not possible to represent such graphs (due to a lack of data) or inconvenient to do so. It is however the only consistent way of representing the causal relations of a dataset if it does not satisfies the assumption of consistency throughout time. The example in Figure 3 shows that each time series causes itself with a time lag of 1, $\mathscr{X}^a$ causes $\mathscr{X}^c$ with a lag of 0 and $\mathscr{X}^b$ causes $\mathscr{X}^a$ and $\mathscr{X}^c$ with a respective time lag of 1 and 2.

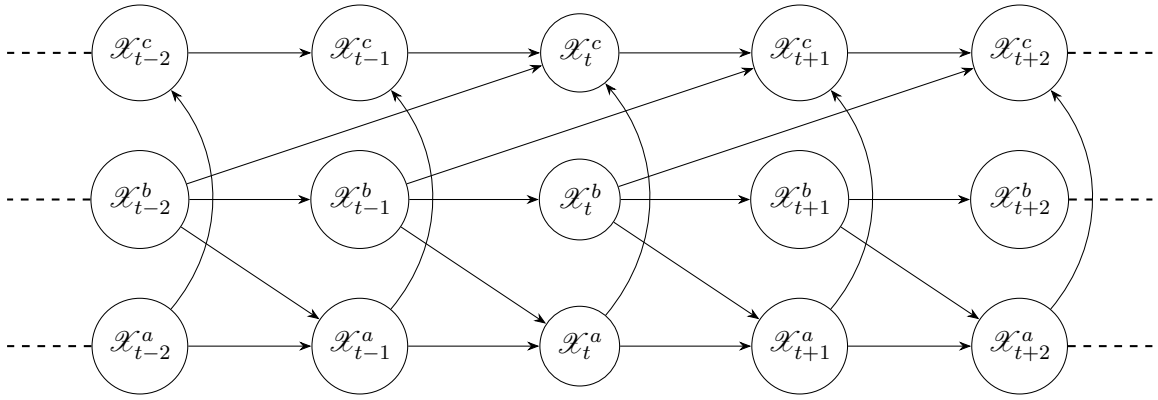


Figure 4: Full Time Causal Graph

Definition 4: Consistency throughout time A causal graph is said to be consistent throughout time if all the causal relations remain constant in direction throughout time.

Window Causal Graphs

This kind of graph can only be represented under the assumption of consistency throughout time. It is a representation of the full time causal graph over a time window equal to the size of the maximal lag τ relating two time series. Figure 4 is the window graph related to the same time series as Figure 3, thus having a size of $\tau = 2$. This kind of graph is finite by definition and encapsulates every causal relation in the case of consistency throughout time.

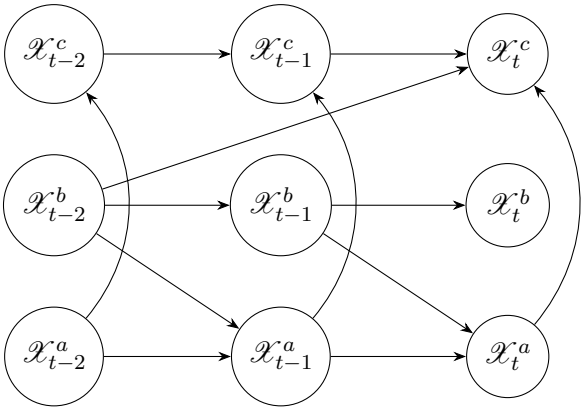


Figure 5: Window Causal Graph

Summary Causal Graphs

A window causal graph can be further compressed if the time lags between the variables are not considered. In this case, either a summary causal graph or an extended[3] summary causal graph can be built. The summary causal graph holds no time information, and is thus less impacted to errors due to lag estimations. Its extended version makes the difference between instantaneous relations (with a lag τ of 0) and other relations by representing only the past and the present vertices. This renders the graph acyclic, which is not necessarily the case for a summary graph. Figure 5 depicts the same example as Figure 3 and 4 for a summary causal graph (left) and the extended version (right).

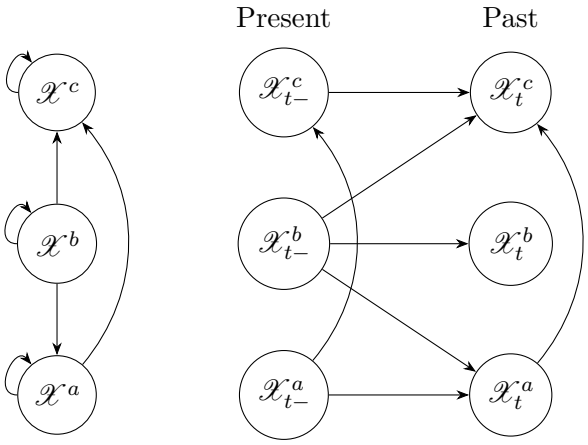


Figure 6: Summary causal graph (left) and extended summary causal graph(right)

2 Models of causal inference

2.1 Granger Causality[4]

In the case of a bivariate time series $\mathscr{X} = (\mathscr{X}^r, \mathscr{X}^p)$, the time series $\mathscr{X}^p$ Granger causes $\mathscr{X}^r$ if past values of $\mathscr{X}^p$ provide unique, statistically significant information about future values of $\mathscr{X}^r$. To test if $\mathscr{X}^p$ Granger causes $\mathscr{X}^r$, two different models have to be computed. The restriced model Mres predicting $\mathscr{X}^r_t$ with past samples of $\mathscr{X}^r$ is defined as:

$$\mathscr{X}^r_t = a_{r,0} + \sum_{i=1}^{\tau} a_{r,i}\mathscr{X}^r_{t-i} + \xi^r_t \qquad \text{(Mres)}$$

The full model Mfull predicts $\mathscr{X}^r_t$ with respect of the past samples of $\mathscr{X}^r$ and $\mathscr{X}^p$, and is defined as :

$$\mathscr{X}^r_t = a_{r,0} + \sum_{i=1}^{\tau} a_{p,i}\mathscr{X}^p_{t-i} \sum_{i=1}^{\tau} a_{r,i}\mathscr{X}^r_{t-i} + \xi^r_t \qquad \text{(Mfull)}$$

ξ^r_t are uncorrelated random variables of mean zero, $(a_{r,i})_{1\leq i\leq\tau}$, $(a_{p,i})_{1\leq i\leq\tau}$ are real coefficients and τ is the optimal lag value. Both models are then passed trough a statistical test to determine which one fits better the data. If Mfull is more accurate, we can deduce that $\mathscr{X}^p$ Granger causes

$\mathscr{X}^r$. To extend this method to a multivariate time series $\mathscr{X} = (\mathscr{X}^1, ..., \mathscr{X}^d)$, we can consider the same models on a vectorial scale, thus obtaining better results while being extremely costly.

This approach does not work on non-linear or non-stationary processes as it uses a linear autoregression model. Furthermore, the instantaneous relations can not be depicted correctly, as using $\mathscr{X}_t^p$ to regress $\mathscr{X}_t^r$ does not allow to find the cause and the consequence. Despite these downsides, Granger causality is considered as a valuable tool.

2.2 Dynamic Bayesian Networks

Structural Equation Model[11]

This model (also called Noise-Based model), describes a causal system by a set of equations where each equation represents each variable in function of its direct causes and additional noise: $X^r = f_r(X^p, \xi^r)$. We focus in our case on the Linear non-Gaussian models (LiNGAM), where $X^p \perp\!\!\!\perp \xi^r$ and f_r is linear. From $X = AX + \xi$ representing the whole system X in a vector, we obtain $X = B\xi$ with $B = (I - A)^{-1}$. The causal relations are then deduced through the strength of the coefficients in the matrix A.

The extension to time series, called VarLiNGAM, is made by computing the matrices $A_i, \forall i \in (1, ..., d)$. To do so, we compute two autoregressive models, one with possible instantaneous influence (SVAR) and the other wihtout (VAR):

$$\mathscr{X}_t = \sum_{i=0}^{\tau} A_i \mathscr{X}_{t-i} + e_t \qquad \text{(SVAR)}$$

$$\mathscr{X}_t = \sum_{i=0}^{\tau} M_i \mathscr{X}_{t-i} + e_t \qquad \text{(VAR)}$$

The A_0 matrix is deduced with the LiNGAM method, and the A_i are deduced with the help of $A_i = (I - A_0)M_i$. This method is capable of learning instantaneous relations without a high computational cost, but relies on strong assumptions such as linearity.

Constraint-Based Model[24]

This model aims at building a graph skeleton based on conditional dependencies, and orients it with a set of rules. It exploits the collider structure as it is the only one that can be oriented without ambiguity. The most used method is the Peter-Clark algorithm (called PC) and has 4 basic rules for orientation:

PC Rule 0: For every triple $X - Y - Z$ such that X and Z are not adjacent and $Y \notin Sepset(X, Z)$, orient the triple as $X \to Y \leftarrow Z$.

PC Rule 1: In a triple $X \to Y - Z$ such that X and Z are not adjacent, orient $Y - Z$ as $Y \to Z$.

PC Rule 2: If there exist a direct path from X to Y and an edge between X and Y, then orient $X \to Y$.

PC Rule 3: Orient $X - Y$ as $X \to Y$ whenever there are two paths $X - A \to Y$ and $X - B \to Y$.

Those rules are used in the traditional PC algorithm, and may not be used in its improved versions. The main problem of the PC algorithm is that is it order dependent and not necessarily stable.

3 Causal Discovery Algorithms

In this section, we will present three algorithms suited for temporal causal discovery to evaluate them in the next section and compare them beforehand.

3.1 PCMCI+

PCMCI+[22] is based on two different phases based on the lag between the variables: a lagged skeleton phase and a contemporaneous skeleton phase. The first algorithm determines $\hat{B}_t^-(\mathscr{X}_t^j)$, and the second one outputs the predicted graph using the output of the first algorithm. $\hat{B}_t^-(\mathscr{X}_t^j)$ represents at most all lagged parents of all contemporaneous ancestors of $\mathscr{X}_t^j$, and in the best case only the parents of $\mathscr{X}_t^j$. I represents the statistical test value helping to rewove links if its value falls bellow the significance threshold α_{PC}. This conditional independence test is chosen among 3 provided by the tigramite[23] package: Parcorr, GPDC and CMIknn. We used GPDC as it is suited to non-linear relations, whereas Parcorr is only able to detect linear relations. CMIknn is also well suited for non-linear relations, but we had convergence issues while testing the datasets with it.

PCMCI+ Algorithm 1

Input: Time series $\mathscr{X}$, max lag τ_{max}, threshold α_{PC}, conditional independence test CI

1 for all $\mathscr{X}_t^j$ in $\mathscr{X}_t$ do
2 Initialize $\hat{B}_t^-(\mathscr{X}_t^j) = \mathscr{X}_t^-$ and $I^{min}(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j) = \infty \quad \forall \mathscr{X}_{t-\tau}^i \in \hat{B}_t^-(\mathscr{X}_t^j)$
3 Let $p = 0$
4 while any $\mathscr{X}_{t-\tau}^i \in \hat{B}_t^-(\mathscr{X}_t^j)$ satisfies $|\hat{B}_t^-(\mathscr{X}_t^j) \backslash \{\mathscr{X}_{t-\tau}^i\}| \geq p$ do
5 for all $\mathscr{X}_{t-\tau}^i$ in $\hat{B}_t^-(\mathscr{X}_t^j)$ satisfying $|\hat{B}_t^-(\mathscr{X}_t^j) \backslash \{\mathscr{X}_{t-\tau}^i\}| \geq p$ do
6 $S =$ first p variables in $\hat{B}_t^-(\mathscr{X}_t^j) \backslash \{\mathscr{X}_{t-\tau}^i\}$
7 $(p - value, I) \leftarrow CI(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j, S)$
8 $I^{min}(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j) = min(|I|, I^{min}(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j))$
9 if $p - value > \alpha_{PC}$ then mark $\mathscr{X}_{t-\tau}^i$ for removal
10 Remove non-significant entries and sort $\hat{B}_t^-(\mathscr{X}_t^j)$ by $I^{min}(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j)$ from largest to smallest
11 Let $p = p + 1$
12 return $\hat{B}_t^-(\mathscr{X}_t^j)$ for all $\mathscr{X}_t^j$ in $\mathscr{X}_t$

Algorithm 2

Input : Time series $\mathscr{X}$, max lag τ_{max}, threshold α_{PC}, conditional independence test CI, $\hat{B}_t^-(\mathscr{X}_t^j)$ for all $\mathscr{X}_t^j$ in $\mathscr{X}_t$

Form graph G with lagged links from $\hat{B}_t^-(\mathscr{X}_t^j)$ for all $\mathscr{X}_t^j$ in $\mathscr{X}_t$ and fully connect all contemporaneous variables
Initialize contemporaneous adjacencies $\hat{A}(\mathscr{X}_t^j) = \{\mathscr{X}_t^i \neq \mathscr{X}_t^j \in \mathscr{X}_t \quad where \quad \mathscr{X}_t^i \circ\!\!-\!\!\circ \mathscr{X}_t^j \text{ in G}$
Initialize $I^{min}(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j) = \infty$ for all links in G
Let p=0
While any adjacent pairs $(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j)$ for $\tau \geq 0$ in G satisfy $|\hat{A}(\mathscr{X}_t^j) \backslash \{\mathscr{X}_{t-\tau}^i\}| \geq p$ do
    Select new adjacent pairs $(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j)$ for $\tau \geq 0$ satisfying $|\hat{A}(\mathscr{X}_t^j) \backslash \{\mathscr{X}_{t-\tau}^i\}| \geq p$
    while $(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j)$ are adjacent in G and not all $S \subseteq \hat{A}(\mathscr{X}_t^j) \backslash \{\mathscr{X}_{t-\tau}^i\}$ with $|S| = p$ have been considered do
        Choose new $S \subseteq \hat{A}(\mathscr{X}_t^j) \backslash \{\mathscr{X}_{t-\tau}^i\}$ with $|S| = p$
        Set $Z = (S, \hat{B}_t^-(\mathscr{X}_t^j) \backslash \{\mathscr{X}_{t-\tau}^i\}, \hat{B}_{t-\tau}^-(\mathscr{X}_{t-\tau}^i))$
        $(p - value, I) \leftarrow CI(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j, Z)$
        $I^{min}(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j) = min(|I|, I^{min}(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j))$
        if $p - value > \alpha_{PC}$ then
            Delete link $\mathscr{X}_{t-\tau}^i \to \mathscr{X}_t^j$ for $\tau > 0$ (or $\mathscr{X}_t^i \circ\!\!-\!\!\circ \mathscr{X}_t^j$ for $\tau = 0$) from G
            Store (unordered) sepset$(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j) = S$
    Let $p = p + 1$
    Re-compute $\hat{A}(\mathscr{X}_t^j)$ from G and sort by $I^{min}(\mathscr{X}_{t-\tau}^i, \mathscr{X}_t^j)$ from largest to smallest
Return G, sepset

The output of PCMCI+ is a window graph of size τ_{max}, which we convert into a weighted summary causal graph for comparability purposes. The original code of PCMCI+ is provided at [23]. In this implementation, the optimal significance threshold is computed through a maximization step (among [0.001, 0.005, 0.01, 0.025, 0.05]) if none is given in input. The tigramite package allows us to find the optimal τ_{max} value for each multivariate time series with a plot of the autocorrelation and the intercorrelation of the variables shifted in time.

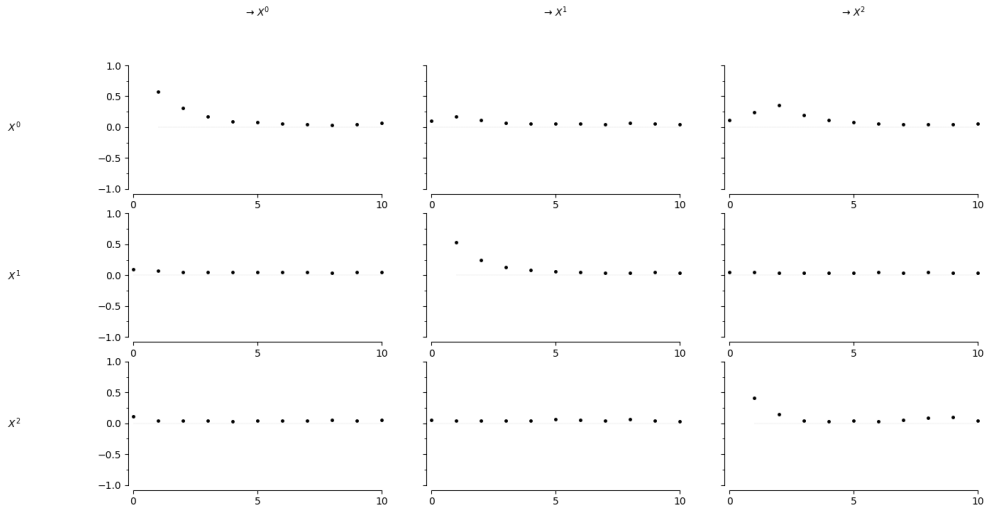


Figure 7: Plot of the output of the get_lagged_dependencies function for 3 variables

With this plot, we select τ_{max} as being the maximal τ value among the τ values that achieve a maximum in every graph (and without taking into account graphs that remain constant). In the example above, τ_{max} would be 2, found in the graph $X^0 \to X^2$.

3.2 LPCMCI

LPCMCI[10] follows the same principles as PCMCI+ but is able to discover hidden confounders in the data, represented as a double arrowed link between two variables. It outputs a partial ancestral graph $C(G)$, which may contain edges of the type $\circ\!\!\rightarrow$ and $\circ\!\!-\!\!\circ$. Algorithms S2 and S3 can be found at [22] and cover the same functions as PCMCI+.

LPCMCI Algorithm
Input: Time series $\mathscr{X}$, max lag τ_{max}, significance α_{PC}, conditional independence test CI, non-negative integer k
1 Initialize $C(G)$ as complete graph with $\mathscr{X}^i_{t-\tau} \circ\!\!\rightarrow \mathscr{X}^j_t$ (for τ in $0 < \tau < \tau_{max}$) and $\mathscr{X}^i_{t-\tau} \circ\!\!-\!\!\circ \mathscr{X}^j_t$ (for $\tau = 0$)
2 for $0 \leq l \leq k-1$ do
3 Remove edges and apply orientations using Algorithm S2
4 Repeat line 1, orient edges as $\mathscr{X}^i_{t-\tau} ?\!\!\rightarrow \mathscr{X}^j_t$ if $\mathscr{X}^i_{t-\tau} * \!\!\rightarrow \mathscr{X}^j_t$ was in $C(G)$ after line 3
5 Remove edges and apply orientations using Algorithm S2
6 Remove edges and apply orientations using Algorithm S3
7 Return PAG= $C(G)$

The output of LPCMCI is also a window causal graph of size τ_{max}. The implementation can be found at [23]. The τ_{max} value is determined the same way as in the PCMCI+ algorithm.

3.3 TCDF

TCDF[17] is a neural network algorithm capable of temporal causal discovery. For a given multivariate time series $\mathscr{X}$ of size d, the algorithm predicts each time series $\mathscr{X}^i$ using a convolutional neural network N_i. Each network N_i has the same dilated depthwise architecture (with a dilation coefficient c and a sliding kernel of size K), but outputs an unique prediction $\mathscr{X}^i$ (depending on every other variable), its kernel weights W_i and its attention score a_i (vector of size d). This last output helps the algorithm select the potential causes $\mathscr{X}^p$ of the variable $\mathscr{X}^i$: these $\mathscr{X}^p$ are the variables where the attention score is larger than the gap (applied to the softmax function) between two elements of the sorted vector a_i. The potential causes of $\mathscr{X}^i$ are then tested on the network N_i. To do so, $\mathscr{X}^p$ is replaced with a random variable $\mathscr{X}^p_{bis}$ of same mean and same variance ($\mathscr{X}^p$ is shuffled in practice). $\mathscr{X}^i$ is then computed using N_i and $\mathscr{X}^p_{bis}$. If the loss of the network does not increases (compared to the first prediction of the network multiplied by the significance α), $\mathscr{X}^p$ is considered as a cause and a link $\mathscr{X}^p \rightarrow \mathscr{X}^i$ is added. The implementation of the algorithm can be found at [16]. The output of the algorithm we use is a weighted summary graph, where the weights of the connections stand for the time lag of the causal relations. The significance level used in our evaluation is 0.8 according to the authors of the method. This algorithm is able to detect hidden confounders, and labels them as a double connection with a lag of 0. One of the main drawback of the method is the fact that τ_{max} depends on the hyperparameters of the networks, and thus can not be set manually. This implies that the algorithm outputs different result when computed with different values of τ_{max}.

TCDF Algorithm

Input: d-dimensional time series $\mathscr{X}$ of length T, L the hidden number of hidden layers, kernel size K, dilation coefficient c, number of epochs, loss function and learning rate, significance threshold α

Compute the maximum lag value $\tau_{max} = 1 + (K-1)\sum_{l=0}^{L} c^l$
Form an empty window graph G of size τ_{max}
for $q \in 1, ..., d$ do
    fit N_q: each $\mathscr{X}_t^q$ is predicted using $(\mathscr{X}_{t-i})_{1 \leq i \leq \tau}$
    Compute the attention scores a_q and the weights W_q
    Sort the attention scores a_q with decreasing order
    Compute the biggest attention score s_q associated to the largest gap between two consecutive elements of a_q
    for $p \in 1, ..., d$ do
        if $Softmax(a_{q,p}) > s_q$ then
            $i = argmax(W_{q,p})$
            Add edge $\mathscr{X}_{t-i}^p \to \mathscr{X}_t^q$ to the graph
            for $(\mathscr{X}_{j-i}^p, \mathscr{X}_j^q) \in Hom(\mathscr{X}_{t-i}^p, \mathscr{X}_t^q, G)$ do
                add edge $\mathscr{X}_{j-i}^p \to \mathscr{X}_j^q$ to the graph
    for $\mathscr{X}_{t-i}^p \in Par(\mathscr{X}_t^q, G)$ do
        Compute the loss of N_q on $\mathscr{X}$ where $\mathscr{X}_{t-i}^p$ is permuted
        diff = loss value of $\mathscr{X}^p$ at epoch 0 − loss value of $\mathscr{X}^p$ at last epoch for
        testdiff = loss value of $\mathscr{X}^p$ at epoch 0 − loss value of $\mathscr{X}_{bis}^p$ at last epoch
        if (testdiff)>(diff $*$ significance):
            Remove edge $\mathscr{X}_{t-i}^p \to \mathscr{X}_t^q$ from G
            for $(\mathscr{X}_{j-i}^p, \mathscr{X}_j^q) \in Hom(\mathscr{X}_{t-i}^p, \mathscr{X}_t^q, G)$ do
                remove edge $\mathscr{X}_{j-i}^p \to \mathscr{X}_j^q$ from the graph
Return the graph G

3.4 Comparison of the algorithms

Here is a table comparing qualitatively the three algorithms.

	Type of Causal Graph	**Instantaneous relations discovery**	**Hidden confounders discovery**	**Model used**
PCMCI+	Window	Yes	No	Constraint based
LPCMCI	Window	Yes	Yes	Constraint based
TCDF	Summary	Yes	Yes	Granger causality

The three algorithms are supposed to be tested on stationary data, and perform differently on the same dataset. TCDF needs theoretically more timestamps as PCMCI+ and LPCMCI as it contains neural networks. PCMCI+ and LPCMCI on the other hand tend to diverge when given too much timestamps or variables, but can give good results with fewer timestamps than TCDF. Both constraint-based algorithms can however remain uncertain about some edges, and represent them as unoriented edges $X \circ\!\!-\!\!\circ Y$. In this case, we consider the edges as undirected $X - Y$, which counts as two connections and lowers the precision of the predicted graph (except if the predicted link is a hidden confounder in the ground truth graph).

4 Datasets

To evaluate properly the presented algorithms, we have to use appropriate datasets. In this section, we present few datasets that could be used to evaluate our algorithms. The core principle for this evaluation is the ground truth graph. It represents the true causal relations in the dataset, and is compared to the output of the algorithms. As finding the ground truth graph of real data is complex, we focus in our evaluation on synthetical data, i.e. data that was generated with specific and known dependencies.

4.1 Artificial Time Series

This dataset can be found at [12]. It consist of 6 different architectures, for a total of 60 different multivariate time series. It was generated using non-linear functions for the relations between different time series and linear functions for self causation, according to this formula :

$$\forall q, \mathscr{X}_0^q = 0; \forall t > 0, \mathscr{X}_t^q = a_{t-1}^{qq}\mathscr{X}_{t-1}^q + \sum_{\substack{(p,\gamma) \\ \mathscr{X}_{t-\gamma}^p \in Par(\mathscr{X}_t^q)}} a_{t-\gamma}^{pq} f(\mathscr{X}_{t-\gamma}^p) + 0.1\xi_t^q$$

where $\gamma > 0$, a_t^{pq} are random coefficients chosen in $\mathscr{U}([-1; -0.1] \cup [0.1; 1])$ for all $1 \leq p \leq d$, $\xi_t^q \sim \mathscr{N}(0, \sqrt{15})$ and f is a nonlinear function chosen uniformly between absolute value, tanh, sine and cosine. The different architectures are following.

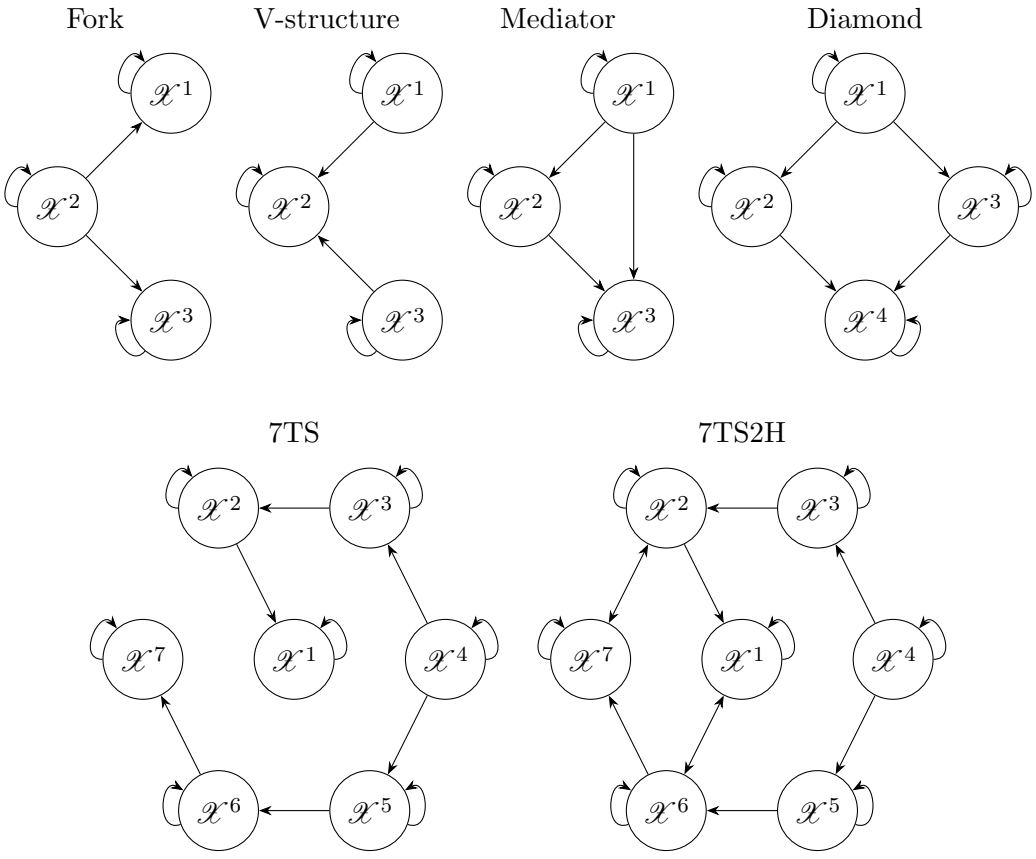


Figure 8: Architectures of the Artificial time series

Each architecture form has 10 different multivariate time series, and each one of them has 4000 evenly spaced timestamps, is stationnary and has a maximal time lag of 1. 7TS2H is the

only architecture with hidden confounders, which are depicted with double arrowed edges, but not provided.

4.2 Functional Magnetic Resonance Imaging (FMRI)

The FMRI dataset contains 28 multivariate time series containing simulated neural activity with non linear relations. The data is supposed to be causally sufficient (implying no hidden confounders), stationary and provides a different groundtruth for every dataset. The provided time series have either 5, 10 or 15 variables, between 50 and 5000 timestamps, and have in general pentagonal architectures. Figure 9 depicts an example of one of the time series provided in the dataset.

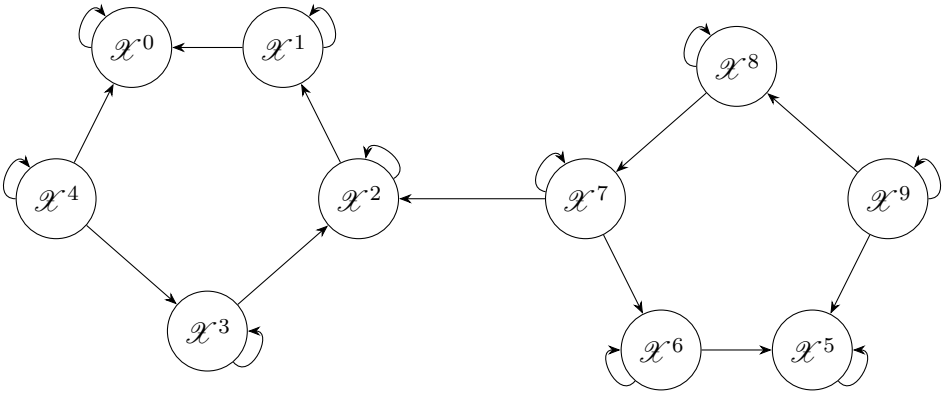


Figure 9: Simulation 17 of the FMRI dataset

The multivariate time series of this datasest have between 10 and 33 causal relations with unknown lag, which is an issue for our evaluation as we want to evaluate the capability of the algorithms to discover the lag of the infered causal relations. It can be found at [1].

4.3 Finance dataset

The Finance dataset can be found at [13] and contains 10 different architectures, each having 25 variables. Each time series has 4000 timestamps and no hidden confounders. More details for each time series of the dataset are provided in this table:

Label of multivariate time series	**Number of relations**	**Lag values**
TS (1)	0	no lag
TS (2) to (6)	20	1
TS (7)	40	1
TS (8)	20	{1,2,3}
TS (9)	40	{1,2,3}
TS (10)	7	{1,2,3}

Table 2: Finance dataset

4.4 Comparative table of the datasets

	Artificial Dataset	**FMRI Dataset**	**Finance Dataset**
Number of multivariate time series	60	28	10
Number of variables	{3,4,7}	{5,10,15}	25
Number of causal relations	{5,6,8,13,15}	{10,12,13,21,33}	{7,20,40}
Time series length	4000	50-5000	4000
Delays	1	unknown	1-3
Hidden confounders	Yes	No	No
Type of relations	Linear and non linear	Non linear	Linear
Self causations	Yes	Yes	Yes

Table 3: Caption

5 Metrics

We will now introduce metrics[14, 5] used in our evaluation and provide an example of the behavior of the metrics. We try to evaluate in this study the predicted graph regarding its structure and the predicted lags. As such, we focus on two main metrics that encapsulate separately both parameters. However, as we encounter degenerative cases for one of the metrics, we introduce and use other metrics to be able to evaluate these particular cases.

In the following definitions, both the ground truth graph G and the predicted graph $\hat{G}$ of a multivariate time series $\mathscr{X}$ are supposed to have the same set of vertices. This supposition leads to an evaluation on the edges of the graph only. For the following metrics, we consider undirected edges $X - Y$ and unoriented edges $X \circ\!\!-\!\!\circ Y$ as a double arrowed edge $X \leftrightarrow Y$ to preserve the maximum of information given by the algorithms.

True positive: A true positive edge TP is an edge that is both in the predicted graph and in the ground truth graph.

False positive: A false positive edge FP is an edge on the predicted graph that is not on the ground truth graph.

False negative: A false negative edge FN is an edge on the ground truth graph that is not on the predicted graph.

True negative: A true negative TN is the absence of an edge on the predicted graph and on the ground truth graph.

With the help of these definitions, we can define Precision, Recall, F1-score and false positive rate.

Precision: Precision is defined as the fraction of relevant instances among the retrieved instances, and is computed as:

$$Precision = \frac{TP}{TP + FP}$$

Recall: Recall is defined as the fraction of relevant instances that where retrieved, and is computed as:

$$Recall = \frac{TP}{TP + FN}$$

F1-score: The F1-score is the mean of precision and recall, and is computed as:

$$F1 = \frac{2TP}{2TP + FP + FN}$$

False positive rate: The false positive rate, or FPR, is the probability of falsely evaluating an edge and is computed as:

$$FPR = \frac{FP}{FP + TN}$$

Adjacency matrices[25] To present the next definitions, we have to first define the concept of adjacency matrices. For a graph of linear relations, the coefficients of the adjacency matrix B are the one linking the variables together

$$B = \{b_{ij}\} \quad with \quad x_i = \sum_j b_{ij} x_j + e_i$$

In our case, the relations are non linear, and we will thus express the adjacency matrix in two different ways, depending on the metrics we use. The first expression of the adjacency matrix B is to represent it in a binary way: $b_{ij} = 1$ represents $X^i \to X^j$ (variable X^i causes X^j with i the line index and j the column index) and $b_{ij} = 0$ represents $X^i \not\to X^j$.
The other expression of B (called weighted adjacency matrix) takes into account the lag instead of a 1. However, to differentiate a connection with a time lag of 0 and no connection, we chose to replace the 0 coefficient with 0.5.

Normalized Mean Squared Error The NMSE allows us to evaluate the distance of the predicted graph from the ground truth regarding the lags. It uses the weighted adjacency matrix and is computed as:

$$NMSE = \frac{1}{N} \sum_{\substack{\text{elements of} \\ \text{the matrix}}} (B_{true} - \hat{B})^2$$

With B_{true} the ground truth graph, $\hat{B}$ the predicted graph and N the number of variables.

Structural Intervention Distance[20] The SID is a distance evaluating the structure of two graphs by comparing their intervention distributions. It is defined as the number of intervention distributions falsely estimated from X^i to X^j (with $i \neq j$) by $\hat{G}$. The intervention distribution from X to Y is defined as:

$$p(Y|do(X = \hat{x})) = p(y) \quad \text{if Y is a parent of X}$$

$$p(Y|do(X = \hat{x})) = \sum_{pa_x} p(y|\hat{x}, pa_x) p(pa_x) \quad \text{if Y is not a parent of X}$$

With pa_x the set of parents of X. The implementation from the author of this metric can be found at [21], and takes as inputs the non-weighted adjacency matrices of both graphs. As the function is not symmetric in its arguments, we compute it with $SID = SID(G_{gt}, \hat{G})$. However, it does not take into account self causations (diagonal of the adjacency matrix), and is not able to compute a value if the predicted graph $\hat{G}$ has any double arrowed edge. To bypass this issue,

we consider only for this metric every undirected (and unoriented) edge as a simple edge, as the majority of the double arrowed edges are induced by undirected edges. To do so, we convert a double arrowed edge to a simple edge by taking into account its presence in the ground truth graph: if the only relation between X and Y in the ground truth is $X \to Y$, $X \leftrightarrow Y$ becomes $X \leftarrow Y$. This consideration greatly penalizes unoriented edges, which is a tolerable trade-off as the predicted undirected edge does not allow to conclude on the orientation of the real edge. However, this modification is not able to cover every degenerative case. If $X \leftrightarrow Y$ is both in the ground truth and in the predicted graph, we can not modify the predicted edge as the prediction is correct. We are also unable to correct the predicted edge $X \leftrightarrow Y$ if there is no causal relation between X and Y. These two cases will still lead to errors in the computation of the SID, and will not be treated further with this metric.

Example of the metrics

We now propose an example for the metrics introduced above. To do so, we compare the ground truth graph G with the predicted graph $\hat{G}_1$, the only differences being the inversion of the edge between X_1 and X_2 and the lag prediction. The following matrices are the adjency matrices of G (left) and $\hat{G}_1$ (right):

$$\begin{array}{c} \\ X_1 \\ X_2 \\ Y_1 \\ Y_2 \\ Y_3 \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & Y_1 & Y_2 & Y_3 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ X_1 \\ X_2 \\ Y_1 \\ Y_2 \\ Y_3 \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & Y_1 & Y_2 & Y_3 \end{array} \\ \begin{pmatrix} 0 & 1 & 5 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0.5 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

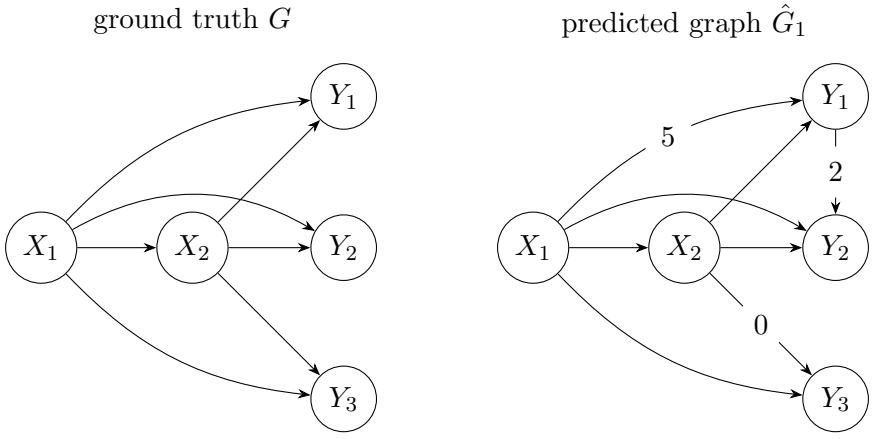


Figure 10: For the sake of clarity, the edges have a lag of 1 when not labeled

In this example, every edge in the ground truth graph is correctly predicted and one edge is added wrongly, leading to $TP = 7$, $FP = 1$, $FN = 0$ and $TN = 18$ (as we take self causation into account). As such, we have:

$$Precision = \frac{7}{7+1} = 0.875, \quad Recall = \frac{7}{7+0} = 1, \quad F1 = \frac{2*7}{2*7+1+0} = 0.93$$

$$\text{and} \quad FPR = \frac{1}{1+18} = 0.05$$

For the NMSE, we obtain when computing the difference between the two matrices:

$$\begin{pmatrix} 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{leading to } NMSE = \frac{(-4)^2 + (-2)^2 + (0.5)^2}{5} = 4.05$$

To measure the SID, we compute for each graph the intervention distribution of each possible edge (except for self causation) and compare them. We use for this task following assumptions[26, 27]:

$$\mathbb{P}(X|Y) = \frac{\mathbb{P}(X,Y)}{\mathbb{P}(Y)} \qquad p(x_i) = \sum_j p(x_i, y_j)$$

$$\text{If } (A \perp\!\!\!\perp B)|C \text{ we have } \mathbb{P}(A|B,C) = \mathbb{P}(A|C) \text{ and } \mathbb{P}(A,B|C) = \mathbb{P}(A|C)\mathbb{P}(B|C)$$

We have for example when comparing the intervention distributions from Y_2 to Y_3:

$$\begin{aligned} \mathbb{P}_{\hat{G}_1}(Y_3|\hat{Y}_2) &= \sum_{x_1,x_2,y_1} p(y_3|\hat{y}_2, x_1, x_2, y_1)p(x_1, x_2, y_1) \\ &= \sum_{x_1,x_2,y_1} \frac{p(y_3, \hat{y}_2, x_1, x_2, y_1)}{p(\hat{y}_2, x_1, x_2, y_1)} p(x_1, x_2, y_1) \\ &= \sum_{x_1,x_2,y_1} \frac{p(y_3, \hat{y}_2, x_1, x_2, y_1)}{p(\hat{y}_2|x_1, x_2, y_1)} \end{aligned}$$

$$\begin{aligned} \mathbb{P}_G(Y_3|\hat{Y}_2) &= \sum_{x_1,x_2} p(y_3|\hat{y}_2, x_1, x_2)p(x_1, x_2) \\ &= \sum_{x_1,x_2} \frac{p(y_3, \hat{y}_2, x_1, x_2)}{p(\hat{y}_2, x_1, x_2)} p(x_1, x_2) \\ &= \sum_{x_1,x_2} \frac{p(y_3, \hat{y}_2, x_1, x_2)}{p(\hat{y}_2|x_1, x_2)} \\ &= \sum_{x_1,x_2} \frac{p(y_3, \hat{y}_2, x_1, x_2)}{p(\hat{y}_2|x_1, x_2, y_1)} \\ &= \sum_{x_1,x_2,y_1} \frac{p(y_3, \hat{y}_2, x_1, x_2, y_1)}{p(\hat{y}_2|x_1, x_2, y_1)} = \mathbb{P}_{\hat{G}_1}(Y_3|\hat{Y}_2) \end{aligned}$$

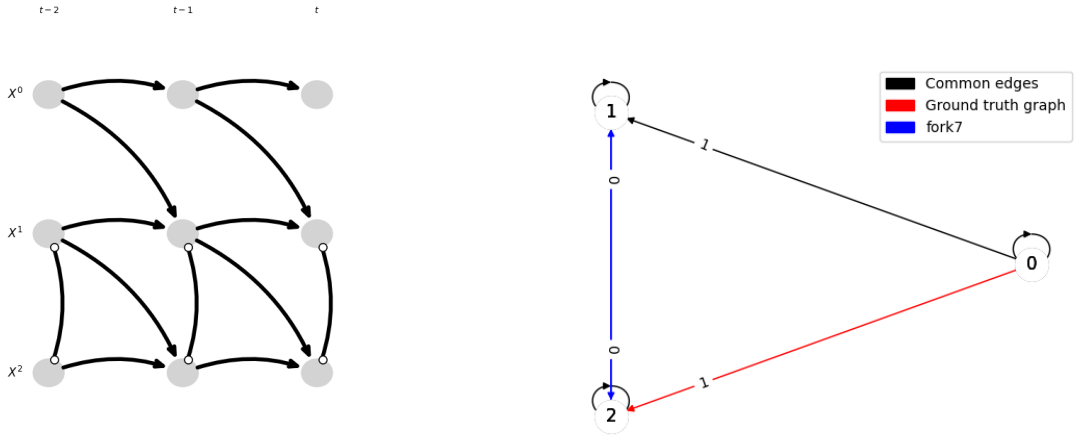
We thus have the same intervention distribution for this particular edge. When doing the same computations over every possible edge in the graphs, we find that the SID is equal to 0. In contrast to other metrics, SID evaluates the information given by the predicted graph regarding the parents of the variables. In our example, the addition of the edge does not change the existing parents of the variables. Additionally, predicting the inversion of an edge (in a graph $\hat{G}_2$) of the ground truth graph (instead of adding an edge in $\hat{G}_1$) would lead to a much higher SID value, as $\hat{G}_2$ would have added and suppressed parents for some variables.

6 Evaluation

The best way to properly evaluate the algorithms would be to test their behavior on every possible dataset to cover all the possibilities that could be encountered with real data. However, for practical reasons, we limited our evaluation to the Artificial Time Series[8] dataset since it is the most complete among the presented datasets. We use for the introduced algorithms the hyperparameters used in other studies[2, 18, 23]. For PCMCI+ we used the first 500 and 1000 timestamps, and only 500 for LPCMCI as the algorithms had issues to converge with more timestamps. The implementation and the results from this study can be found on this github: https://github.com/Flopp88/Causal-Analysis-of-Time-Series.

6.1 PCMCI+

As the provided algorithm selects α_{PC} with a maximization step and the tigramite[23] package allows us to find the optimal value for a dataset, we present here only the results for 500 and 1000 timestamps. As PCMCI+ outputs a window causal graph, we convert it to a weighted summary graph to compare it afterwards with the output of the TCDF. As can be seen in Figure 11, we take only the minimal time lag of all present edges in the summary graph, as we want to consider a single lag value per edge. To enhance readability, the predicted summary graph and the ground truth graph are combined on a single plot. Blue edges are only present in the predicted graph (thus being False positives), red edges are only present in the ground truth graph (thus being False negatives) and black edges are True positives.



(a) Predicted window causal graph

(b) Final weighted summary graph

Figure 11: Conversion of the fork7 PCMCI+ prediction for 500 timestamps

The results are presented as the mean±standard deviation over the 10 datasets of an architecture, and an additionnal column for the whole dataset at once.

	Fork	**Mediator**	**V-structure**	**Diamond**	**7TS**	**7TS2H**	**Dataset**
NMSE	0.53 ± 0.19	0.52 ± 0.26	0.44 ± 0.16	0.58 ± 0.24	1.22 ± 1.51	1.78 ± 3.67	0.85 ± 1.64
SID	2 ± 1.7	1.6 ± 1.5	0.9 ± 0.74	2.7 ± 2.06	10.14 ± 6.3	27 ± 4.2	6.23 ± 9.14
Precision	0.93 ± 0.12	0.96 ± 0.1	0.98 ± 0.06	1 ± 0	0.91 ± 0.09	0.96 ± 0.05	0.95 ± 0.08
Recall	0.82 ± 0.11	0.78 ± 0.16	0.82 ± 0.15	0.84 ± 0.12	0.88 ± 0.06	0.72 ± 0.12	0.81 ± 0.13

Table 4: PCMCI+ for 500 timestamps

	Fork	**Mediator**	**V-structure**	**Diamond**	**7TS**	**7TS2H**	**Dataset**
NMSE	0.47 ± 0.19	0.44 ± 0.25	0.54 ± 0.21	0.43 ± 0.21	0.24 ± 0.19	0.26 ± 0.18	0.40 ± 0.22
SID	1.78 ± 2.1	0.8 ± 1.23	0.88 ± 1	1.4 ± 1.9	7.3 ± 4.1	22.5 ± 0.7	3.07 ± 5.17
Precision	0.93 ± 0.15	0.98 ± 0.08	0.92 ± 0.16	1 ± 0	0.92 ± 0.07	0.95 ± 0.08	0.95 ± 0.1
Recall	0.86 ± 0.16	0.87 ± 0.17	0.82 ± 0.18	0.9 ± 0.11	0.92 ± 0.05	0.82 ± 0.12	0.86 ± 0.14

Table 5: PCMCI+ for 1000 timestamps

We add the Precision and Recall for a better comparison of the behavior as the algorithm predicts numerous graphs leading to SID errors: 7 out for 500 points and 14 for 1000 points (out of 60 graphs). We can see for this algorithm that having more timestamps slightly enhances the quality of the output. Additionally, the algorithm has a slightly better performance regarding precision versus recall. For 1000 points, the average precision over the whole dataset is at 0.95, while the average recall is at 0.86. The average precision value is the same for 500 points and the average recall is at 0.81, meaning the algorithm might be able to find every causal relation if given enough timestamps, but the output may still have wrong edges. As expected, the recall on the architecture with hidden confounders is worse, since PCMCI+ is not able to detect them. We can also see that the lags are close to the ones of the ground truth (for 1000 timestamps).

6.2 LPCMCI

For this algorithm, we tested multiple values of the hyperparameter α_{PC} proposed by the authors for 500 timestamps: {0.001,0.005,0.01,0.05}. The results are presented in the tables below:

	Fork	**Mediator**	**V-structure**	**Diamond**	**7TS**	**7TS2H**	**Dataset**
NMSE	1.6 ± 0.62	1.43 ± 0.72	1.37 ± 1.01	1.8 ± 0.86	1.55 ± 0.44	1.34 ± 0.45	1.52 ± 0.70
SID	4.8 ± 1.4	4.4 ± 2.12	3.2 ± 1.69	8.9 ± 2.51	32.9 ± 7.05	35.6 ± 7.76	15.0 ± 14.6
Precision	0.78 ± 0.15	0.76 ± 0.19	0.83 ± 0.18	0.74 ± 0.16	0.70 ± 0.12	0.77 ± 0.15	0.76 ± 0.16
Recall	0.62 ± 0.15	0.62 ± 0.19	0.66 ± 0.13	0.63 ± 0.14	0.59 ± 0.10	0.48 ± 0.09	0.60 ± 0.15

Table 6: LPCMCI for $\alpha_{PC} = 0.001$

	Fork	**Mediator**	**V-structure**	**Diamond**	**7TS**	**7TS2H**	**Dataset**
NMSE	1.54 ± 0.63	1.43 ± 0.65	1.6 ± 0.83	1.76 ± 0.83	1.52 ± 0.29	1.26 ± 0.44	1.52 ± 0.63
SID	4.6 ± 1.35	4.4 ± 2.01	3.7 ± 1.42	8.8 ± 2.39	31.7 ± 9.44	35.67 ± 8.23	14.46 ± 14.35
Precision	0.79 ± 0.14	0.74 ± 0.17	0.79 ± 0.14	0.75 ± 0.15	0.69 ± 0.1	0.77 ± 0.16	0.75 ± 0.14
Recall	0.66 ± 0.13	0.63 ± 0.19	0.66 ± 0.1	0.64 ± 0.14	0.61 ± 0.11	0.49 ± 0.11	0.61 ± 0.14

Table 7: LPCMCI for $\alpha_{PC} = 0.005$

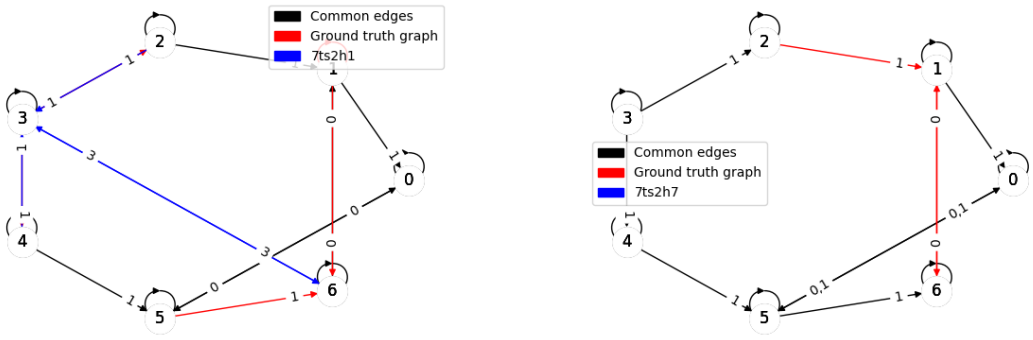
	Fork	**Mediator**	**V-structure**	**Diamond**	**7TS**	**7TS2H**	**Dataset**
NMSE	0.57 ± 0.32	0.47 ± 0.19	0.5 ± 0.24	0.68 ± 0.33	0.62 ± 0.43	0.95 ± 0.9	0.63 ± 0.47
SID	1.5 ± 1.58	0.9 ± 0.57	0.9 ± 1.1	2.8 ± 1.55	8 ± 5.1	25.25 ± 2.87	5.88 ± 8.61
Precision	0.96 ± 0.1	1 ± 0	0.98 ± 0.06	0.96 ± 0.07	0.92 ± 0.09	0.93 ± 0.09	0.96 ± 0.08
Recall	0.84 ± 0.16	0.85 ± 0.09	0.86 ± 0.16	0.84 ± 0.11	0.83 ± 0.09	0.64 ± 0.08	0.81 ± 0.14

Table 8: LPCMCI for $\alpha_{PC} = 0.01$

	Fork	Mediator	V-structure	Diamond	7TS	7TS2H	Dataset
NMSE	1.73 ± 0.6	1.74 ± 0.84	1.51 ± 0.86	1.75 ± 0.96	1.53 ± 0.63	1.59 ± 0.49	1.64 ± 0.73
SID	4.89 ± 1.05	4.5 ± 2.22	3.6 ± 1.65	8.67 ± 3	35.67 ± 10.69	37.78 ± 4.09	14.28 ± 15.03
Precision	0.73 ± 0.14	0.68 ± 0.19	0.77 ± 0.16	0.74 ± 0.16	0.65 ± 0.1	0.72 ± 0.12	0.71 ± 0.15
Recall	0.68 ± 0.1	0.62 ± 0.19	0.7 ± 0.11	0.68 ± 0.17	0.66 ± 0.16	0.55 ± 0.10	0.65 ± 0.15

Table 9: LPCMCI for $\alpha_{PC} = 0.05$

The results show a better behaviour for $\alpha_{Pc} = 0.01$. Precision and recall are added to the tables as the results of the SID for this algorithm are different from PCMCI+. LPCMCI predicts undirected edges where a hidden relation should be. As such, the modified degenerative cases are often considered as wrongly oriented and will therefore increase the SID metric, while getting a result for this metric (for $\alpha_{Pc} = 0.01$), only 3 graphs are not compatible with SID). It is interesting to note that LPCMCI has a worse recall on the time series with hidden confounders than on other architectures. The hidden confounders were detected only twice (for $\alpha_{Pc} = 0.01$) by this algorithm, and are represented on Figure 12. In comparison with PCMCI+, LPCMCI seems to have more trouble identifying hidden confounders. This is however not the case and is linked with the fact that PCMCI+ considers more often a possible relation (inplace of a hidden confounder) but labels it as an unoriented edge as it stays uncertain of the orientation. For the same amount of points, LPCMCI (with $\alpha_{Pc} = 0.01$) scores better than PCMCI+.



(a) 7TS2H1 prediction

(b) 7TS2H1 prediction

Figure 12: Prediction of hidden confounders for LPCMCI with $\alpha_{PC} = 0.01$

6.3 TCDF

We tested the TCDF algorithm while modifying its hyperparameters. For this evaluation, we fix the hyperparameters to:

Epochs = 1000
Kernel size K = 4
Number of hidden layers L = 1
dilation coefficient c = 4
Significance $\alpha = 0.01$
Learning rate = 0.01
Loss function = Adam

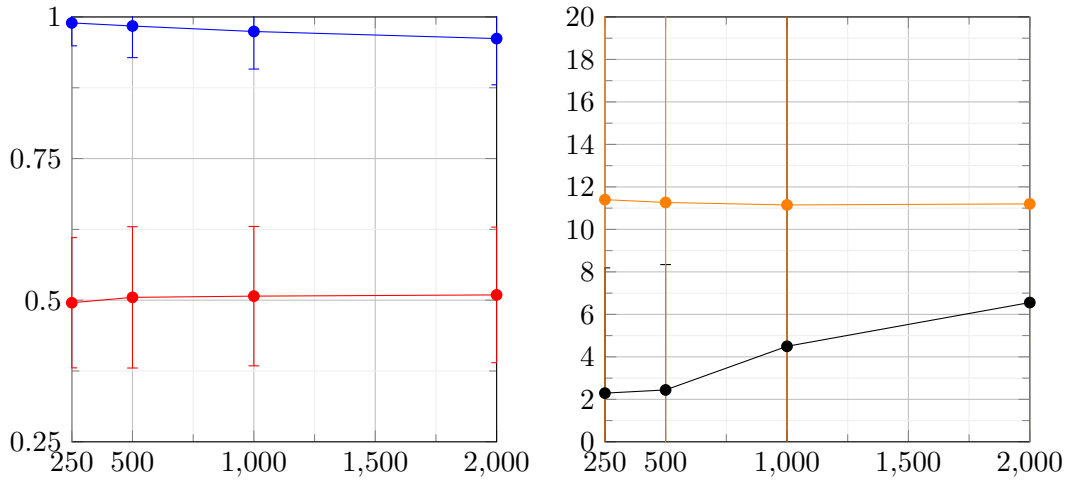
Table 10: Base hyperparameters for the evaluation

We then modified each hyperparameter separately (except for the learning rate and the loss function) to obtain the precision, recall, NMSE and SID of the predicted graphs. The plots represent the mean±standard deviation of the metrics over the 60 multivariate time series of the dataset. This algorithm does only predict oriented edges ($\leftarrow$, $\rightarrow$ or $\leftrightarrow$) and has thus less problems of SID computation errors as the other algorithms. In fact, SID could not be computed on only 4 graphs out of 1200. Before analyzing the results, we can add that this selection of hyperparameters lead to a maximum lag of:

$$\tau_{max} = 1 + (K-1)\sum_{l=0}^{L} c^l = 1 + (4-1)(4^0 + 4^1) = 16$$

This maximum lag value could seem excessive in comparison of our dataset (which has a maximum lag value of 1), but K and c are supposed to have the same value[17], K has to be high enough to have a decent convolution computation and we decided to take one hidden layer instead of 0 (implying $\tau_{max} = 16$ instead of $\tau_{max} = 4$) to slightly improve the quality of the structure. We also tested different number of timestamps for this time series (100, 500, 1000 and 4000) but the metrics were constant over the range of the tested timestamps, and we thus chose to use the whole dataset.

The first hyperparameter we will discuss is the number of epochs, depicted in Figure 13. As we can see, increasing the number of epochs only slightly affects the results. The decrease in precision and the increase in NMSE can be explained as the over-fitting of the neural network during the prediction phase.

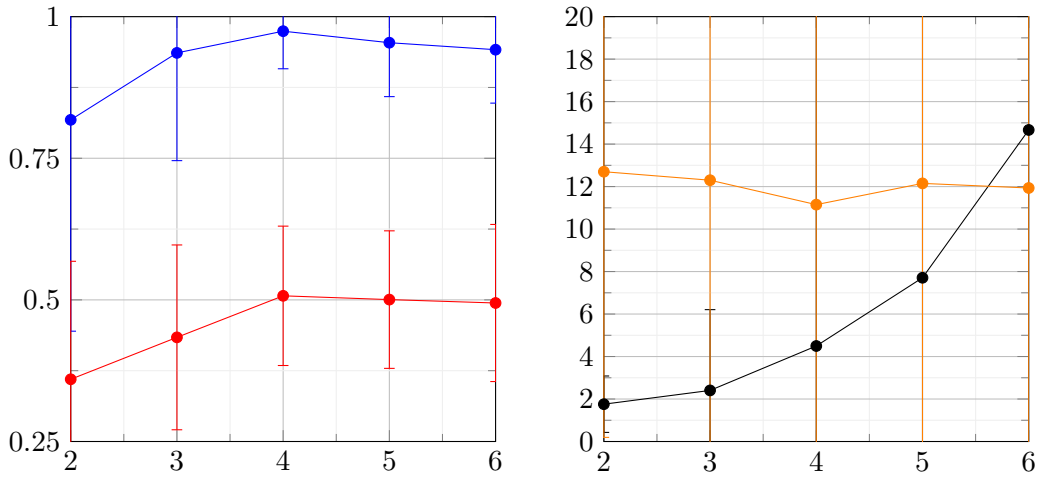


(a) Precision in blue and Recall in red

(b) SID in orange and NMSE in black

Figure 13: Metrics for a modification of the epochs

The kernel size K, depicted in Figure 14, influences heavily the quality of the prediction. The increasing NMSE value is linked with the fact that τ_{max} increases linearly (from 6 to 26 with a step of 5 in our case) with K, allowing the algorithm to predict higher lag values. On the other hand, the kernel size has to be high enough to let the network predict correctly the graph. For this dataset, a kernel size of K=4 is the most optimal value for the structure of the prediction.

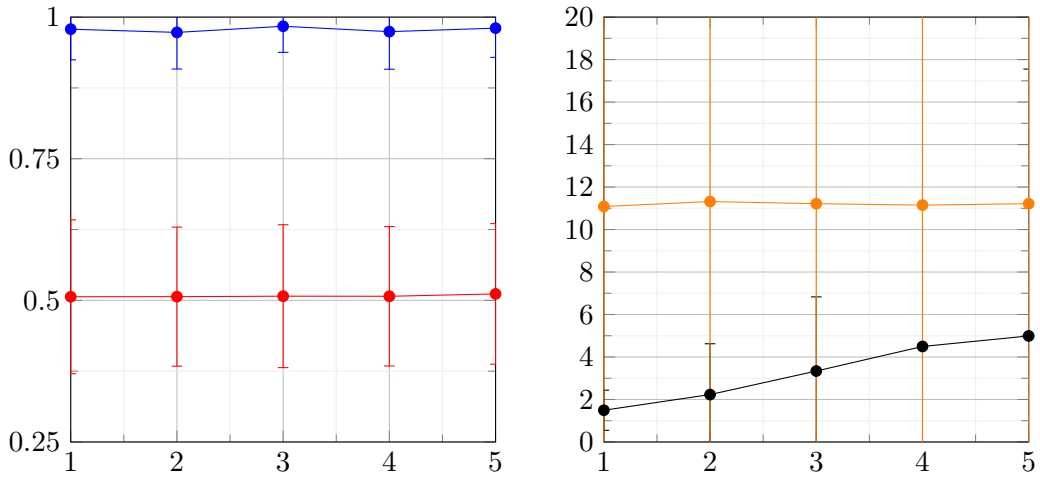


(a) Precision in blue and Recall in red

(b) SID in orange and NMSE in black

Figure 14: Metrics for a modification of the kernel size K

Figure 15 shows us that the dilation coefficient has no impact on the structure of the graph, as Precision, Recall and SID are stable. It does however have an impact on the quality of the predicted delays. This can be explained by the fact that τ_{max} grows linearly with c (from 7 to 19 with a step of 3 in our case), allowing the algorithm to predict higher lag values. As we chose K=4 previously, we fix the value of the dilation coefficient to c=4 as its impact is small.

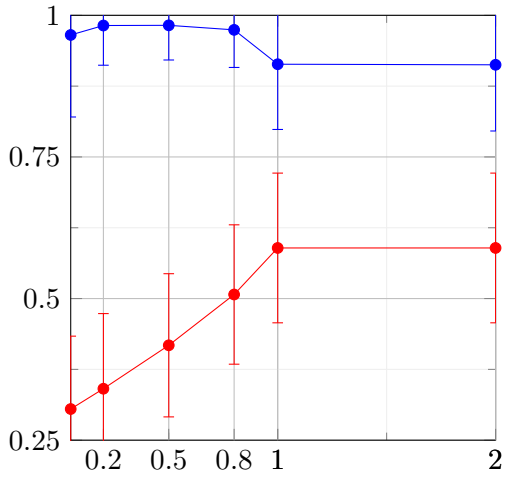


(a) Precision in blue and Recall in red

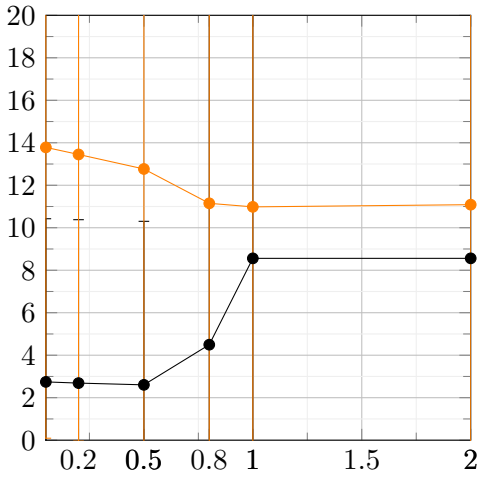
(b) SID in orange and NMSE in black

Figure 15: Metrics for a modification of the dilation coefficient c

The significance threshold, in Figure 16, plays a major role in the quality of the predicted graph. The plotted results concur with the prediction made from the implementation: a low value of α leads to only the variables with a strong realtions passing trough the "loss condition", thus leading to to a high precision (on the edges with Precision and on the lags with NMSE) but a mostly incomplete structure (with Recall and SID). On the other hand, a high significance threshold lets less strongly dependent variables pass the "loss condition", leading to more edges in the predicted graph, represented as an increase of Recall and a decrease in Precision. The overall quality of the graph is however improved as the SID diminishes with a higher α value. Additionnaly, there is no difference between $\alpha = 1$ and $\alpha = 2$. We chose for this hyperparameter a value of 0.8 as it is a good trade off between the precision of the edges, the recall and the quality of the time lags.

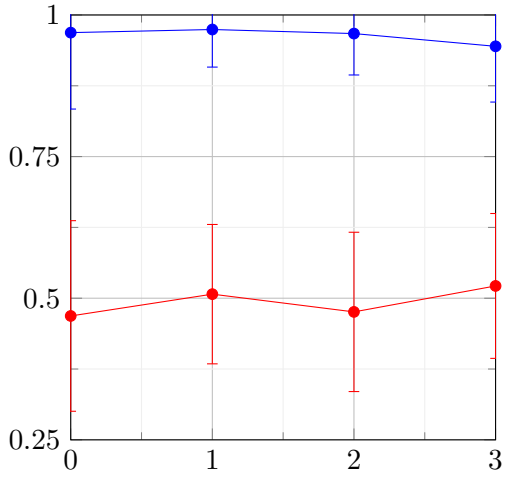


(a) Precision in blue and Recall in red

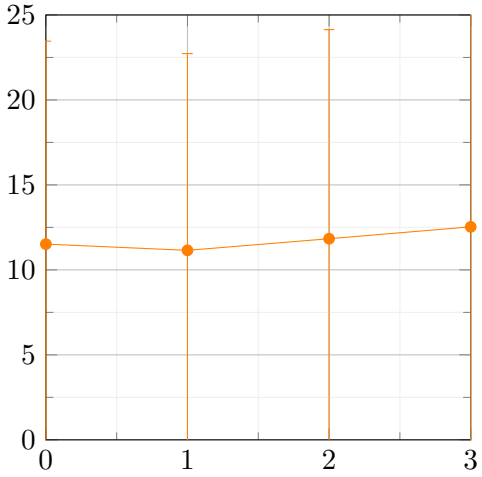


(b) SID in orange and NMSE in black

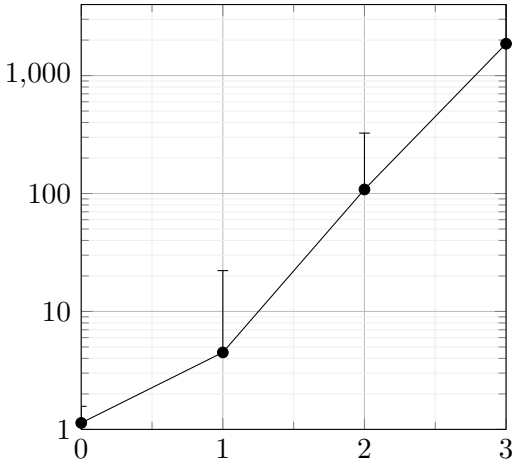
Figure 16: Metrics for a modification of the significance threshold α from 0.05 to 2



(a) Precision in blue and Recall in red



(b) SID



(c) NMSE with logarithmic y scaling

Figure 17: Metrics for a modification of the number of hidden layer L

Figure 17, shows the impact of the hidden layers on the metrics. In our case, $\tau_{max} = 4^{L+1}$, implying that it allows the lag values to go up to 256. exlaining the high NMSE values. We chose L=1 to achieve the best Precision, Recall and SID while having a relatively low NMSE value.

The table of the results for our hyperparameters is presented bellow.

	Fork	**Mediator**	**V-structure**	**Diamond**	**7TS**	**7TS2H**	**Dataset**
NMSE	1.325 ± 0.12	1 ± 0	0.735 ± 0.21	20.11 ± 41.45	1.92 ± 1.81	1.89 ± 2.81	4.50 ± 17.71
SID	7 ± 2.54	19.3 ± 5.83	32.3 ± 3.13	1.8 ± 0.42	3.3 ± 0.82	3.2 ± 1.40	11.15 ± 11.58
Precision	1 ± 0	1 ± 0	1 ± 0	0.92 ± 0.11	0.95 ± 0.09	0.98 ± 0.05	0.97 ± 0.07
Recall	0.37 ± 0.09	0.52 ± 0.05	0.58 ± 0.15	0.55 ± 0.10	0.53 ± 0.09	0.50 ± 0.13	0.51 ± 0.12

Table 11: TCDF for our base hyperparameters

In comparison with PCMCI+ and LPCMCI, TCDF has a slightly better Precision, especially on graphs with 3 variables. TCDF scores however way worse in average than the two other algorithms for the NMSE. This gap occurs mainly because of the difference of the maximum lag between the algorithms. With the tigramite function, the input τ_{max} given in input of the algorithms is small (mean of 2.3 ± 1.3 for the 60 time series) compared to $\tau_{max} = 16$ for TCDF. This leads to more errors on the lag prediction as our significance value is high and lets weak relations pass through the "loss condition". However, this issue can not be fixed directly as the TCDF algorithm does not allow us to control the τ_{max} value expect by modifying the hyperparameters, thus completely reworking the behaviour of the algorithm. This gap lets us conclude that TCDF's lag prediction are not as accurate as the other algorithms. The fact that the SID is high for TCDF compared to the other algorithms while the Precision and Recall have acceptable values indicates where the TCDF has issues for the predictions. As the SID does not take into account the self causations, we can conclude that the True Positives in the TCDF results correspond mainly to self causations, i.e. $X^i \to X^i$. This means that TCDF has more difficulty to infer consistently non linear relations, as the database is constructed with linear relations for self causations and non linear functions for relations between two variables. In the same way, TCDF did not discover any complete hidden confounder between two variables as the relations are also non linear. Its best result regarding hidden confounders are depicted in Figure 18. It shows a partially correct hidden confounder detection between 6 and 1, which are $\mathscr{X}^7$ and $\mathscr{X}^2$. The hidden confounder is however not correctly discovered as TCDF discovers only the relation $\mathscr{X}^7 \to \mathscr{X}^2$ and not $\mathscr{X}^7 \leftarrow \mathscr{X}^2$.

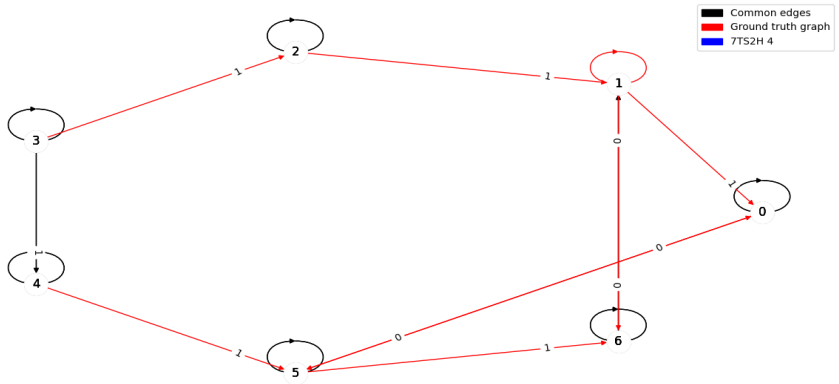


Figure 18: 7TS2H4 prediction for TCDF

7 Supply Chain Data Analysis: Causal Discovery in European Cereal Trade

We now try to discover causal relations on a real multivariate time series. We used in this experiment two separate datasets and merged them into one. The datasets can be found at [6] and [7]. They represent respectively the prices of cereal and the trade of cereal among states of the EU, from 2009 to 2022.The dataset containing the price is made out of 7 columns: Marketing year, Reference period (date in an interval of 7 days), Member state, Product name, Market name (city where the data was collected), Stage name (means of delivery) and Price per tonne. The import/export dataset consists of 11 columns: Flow (to specify import or export), Marketing year, 2 columns referring to the month(word and number), Member state, (trading) Partner, Product group, Product Code, Quantity in tonnes and Value in thousand euros. To correctly analyze the data, we first had to preprocess it.

7.1 Preprocessing of the datasets

The goal is to provide valid variables to the algorithms. To do so, we had to convert both datasets to the same architecture. We chose to define a variable as following: the price per tonne depending on the country of the data, the type of cereal and the type of data (import, export or price in the country). The import/export dataset had to be reshaped to be expressed in euro per tonne. The time series have timestamps every week for the prices, and every month for the import/export. We chose to take the mean over every month of the price dataset for each variable to make the datasets match. We also had to merge different markets specific to each country for the price dataset by taking the mean over every different market at the same date. After the previous considerations, equally long variables had to be created over the same period of time. As some variables lacked samples, we kept every variable that had more than 75 samples and chose to keep the variables over the period between 01/12/2013 and 01/03/2022, as it was a good trade-off between the number of samples and the number of variables that overlapped. This lead us to 112 variables, with some missing timestamps. To fill the blanks, we interpolated the variables with the scipy.interp1d function [8]. As the data is not linear, it would be more appropriate to use cubic interpolation, but this kind of interpolation outputs negative price variables. We thus used linear interpolation to obtain consistent results. We finally had to pass the variables through the pandas.diff function [19] to render the data stationary, as the algorithm work under the assumption of stationary time series (example in Figure 19). This removed however the first sample of our time series, and leaves us with 112 variables of 99 timestamps.

7.2 Experiment on the algorithms

We now try to feed the three algorithms with our multivariate time series. TCDF is able to compute its results while PCMCI+ and LPCMCI are unable to converge for 112 variables (we used for the constraint-based algorithms $\tau_{max} = 10$, with respect to the get_lagged_dependencies function). As the graph is extremely overloaded, we provide the written output of the TCDF algorithm in Figure 20.

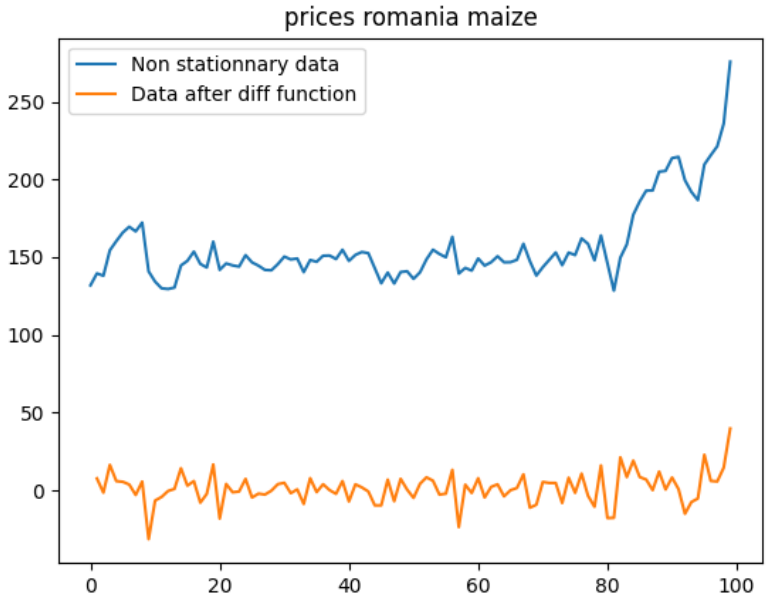


Figure 19: Romania Maize prices before and after diff function

```
import romania barley causes import poland maize with a delay of 7 time steps.
export austria barley causes import croatia wheat with a delay of 6 time steps.
import slovenia barley causes import slovenia barley with a delay of 1 time steps.
import finland wheat causes import lithuania maize with a delay of 0 time steps.
export france rye causes import germany wheat with a delay of 5 time steps.
export bulgaria maize causes export france barley with a delay of 0 time steps.
export france barley causes export romania wheat with a delay of 0 time steps.
import bulgaria maize causes export romania wheat with a delay of 0 time steps.
import slovenia barley causes export italy rye with a delay of 8 time steps.
export finland oats causes export lithuania barley with a delay of 0 time steps.
import finland wheat causes export portugal maize with a delay of 8 time steps.
export italy maize causes export austria oats with a delay of 6 time steps.
import bulgaria maize causes export austria wheat with a delay of 0 time steps.
export czechia barley causes export austria wheat with a delay of 0 time steps.
```

Figure 20: TCDF prediction for the whole dataset

We then tried to filter the dataset with the previous results to see if they were convincing and if PCMCI+ and LPCMCI could find them. To do so, we took every variable appearing in the previous TCDF result and built a dataset out of them. This narrows the dataset to 22 variables, represented in Table 12.

import slovenia barley	export austria barley	export lithuania barley	export france rye
export france barley	export austria oats	export austria wheat	export portugal maize
export czechia barley	export romania wheat	export italy rye	export finland oats
import finland wheat	import croatia wheat	import lithuania maize	import germany wheat
export italy maize	import romania barley	prices croatia maize	import poland maize
import bulgaria maize	export bulgaria maize		

Table 12: Dataset filtered with the results of TCDF

We then run the algorithms with the filtered database and obtain results presented in Figure 22, 23 and 21. For the filtered data, TCDF predicts a lot more relations than for the previous configuration, which can be explained with the fact that the neural networks N_i predict each variable differently as they do not have the same inputs (the attention scores may be in general closer, leading to a lower attention score and thus more selected variables). When comparing the filtered results with the previous result, we see that the only shared edge is (import slovenia barley) $\rightarrow$ (import slovenia barley), predicted with a lag of 1 except for the filtered TCDF output, which predicted a time lag of 5. Regarding our previous evaluation, we can assume that the discovered relation is part of the ground truth graph with a time lag of 1, as TCDF performs in average worse when predicting time lags (regarding the NMSE values). This relation is the only one shared with the 4 predictions.

```
import slovenia barley causes import slovenia barley with a time lag of 1
import slovenia barley causes import slovenia barley with a time lag of 1
import slovenia barley causes export france rye with a time lag of 9
import germany wheat causes export austria barley with a time lag of 6
export austria barley causes export portugal maize with a time lag of 9
export portugal maize causes export portugal maize with a time lag of 1
export romania wheat causes export romania wheat with a time lag of 1
export czechia barley causes export romania wheat with a time lag of 10
import romania barley causes export romania wheat with a time lag of 9
export italy rye causes export italy rye with a time lag of 1
import finland wheat causes export italy rye with a time lag of 6
export austria oats causes export austria oats with a time lag of 1
export finland oats causes export austria oats with a time lag of 10
export italy maize causes export finland oats with a time lag of 6
import finland wheat causes export finland oats with a time lag of 10
import lithuania maize causes import lithuania maize with a time lag of 2
export lithuania barley causes export france rye with a time lag of 9
import bulgaria maize causes import bulgaria maize with a time lag of 1
export austria wheat causes export lithuania barley with a time lag of 10
import bulgaria maize causes export lithuania barley with a time lag of 4
export france barley causes export bulgaria maize with a time lag of 2
import croatia wheat causes import croatia wheat with a time lag of 1
import romania barley causes import croatia wheat with a time lag of 6
import lithuania maize causes import germany wheat with a time lag of 4
prices croatia maize causes prices croatia maize with a time lag of 1
prices croatia maize causes prices croatia maize with a time lag of 1
prices croatia maize causes prices croatia maize with a time lag of 4
import poland maize causes export france barley with a time lag of 9
export bulgaria maize causes export france barley with a time lag of 9
export france barley causes export france barley with a time lag of 1
export italy maize causes export italy maize with a time lag of 1
export italy maize causes export italy maize with a time lag of 1
```

Figure 21: Prediction of LPCMCI for the filtered dataset

```
export austria barley causes export france rye with a time lag of 13
export france rye causes export france rye with a time lag of 1
import lithuania maize causes export france rye with a time lag of 11
export austria barley causes import romania barley with a time lag of 5
import romania barley causes import romania barley with a time lag of 13
import lithuania maize causes import lithuania maize with a time lag of 5
export bulgaria maize causes import lithuania maize with a time lag of 15
import croatia wheat causes import lithuania maize with a time lag of 8
export bulgaria maize causes export portugal maize with a time lag of 8
import lithuania maize causes export portugal maize with a time lag of 7
import poland maize causes export austria wheat with a time lag of 3
export austria wheat causes export france barley with a time lag of 7
export italy rye causes export finland oats with a time lag of 15
export france rye causes export finland oats with a time lag of 6
import finland wheat causes import croatia wheat with a time lag of 8
export austria oats causes export romania wheat with a time lag of 15
export romania wheat causes export romania wheat with a time lag of 2
import slovenia barley causes import finland wheat with a time lag of 8
import finland wheat causes import finland wheat with a time lag of 1
export france rye causes import finland wheat with a time lag of 1
import bulgaria maize causes export italy maize with a time lag of 0
export italy maize causes export italy maize with a time lag of 6
prices croatia maize causes export bulgaria maize with a time lag of 1
import slovenia barley causes export bulgaria maize with a time lag of 13
export austria oats causes export bulgaria maize with a time lag of 0
import finland wheat causes import slovenia barley with a time lag of 15
import germany wheat causes import slovenia barley with a time lag of 14
import slovenia barley causes import slovenia barley with a time lag of 5
import romania barley causes export austria oats with a time lag of 14
export bulgaria maize causes export austria barley with a time lag of 15
prices croatia maize causes export austria barley with a time lag of 9
export france rye causes export austria barley with a time lag of 1
import lithuania maize causes import poland maize with a time lag of 14
export finland oats causes import poland maize with a time lag of 8
import finland wheat causes export czechia barley with a time lag of 11
import croatia wheat causes export czechia barley with a time lag of 4
export portugal maize causes prices croatia maize with a time lag of 13
import finland wheat causes prices croatia maize with a time lag of 14
export italy maize causes import bulgaria maize with a time lag of 4
export italy rye causes export lithuania barley with a time lag of 0
import germany wheat causes export lithuania barley with a time lag of 4
import croatia wheat causes export italy rye with a time lag of 8
```

Figure 22: Prediction of TCDF for the filtered dataset

```
export lithuania barley causes export lithuania barley with a time lag of 1
export portugal maize causes import finland wheat with a time lag of 6
export portugal maize causes import poland maize with a time lag of 6
export austria oats causes export austria oats with a time lag of 1
export italy rye causes export italy rye with a time lag of 1
export czechia barley causes export france barley with a time lag of 5
import lithuania maize causes import lithuania maize with a time lag of 1
export italy maize causes export italy maize with a time lag of 1
import croatia wheat causes import croatia wheat with a time lag of 1
```

Figure 23: Prediction of PCMCI+ for the filtered dataset

The three filtered predictions share however other relations, represented in table 13. The cells of the table represent the predicted lag of the algorithms, and a blank cell implies that the specific realtion was not discovered by the algorithm.

Relation	**TCDF**	**Filtered TCDF**	**Filtered PCMCI+**	**Filtered LPCMCI**
import slovenia barley causes import slovenia barley	1	5	1	1
export italy maize causes export austria oats	6	6		
import bulgaria maize causes import bulgaria maize		6	1	
import germany wheat causes import croatia wheat		13	5	
export lithuania barley causes export lithuania barley			1	1
import lithuania maize causes import lithuania maize			1	1
export italy rye causes export italy rye		2		1
export czechia barley causes export czechia barley		1		1
export romania wheat causes export italy maize		13		9
export italy maize causes export italy maize		1		1

Table 13: Shared relations over the different algorithms

7.3 Conclusion

Due to the preprocessing of the data and the assumptions we made, the inferred results should be interpreted with care. As we interpolated the data with a linear function (which applied the mean between two points), we could have build non existing linear dependencies. From the final relations, the following variables were interpolated: (import slovenia barley) for 49 samples, (import bulgaria maize) for 2 samples, (export lithuania barley) for 1 sample, (import lithuania maize) for 2 sample, (export italy rye) for 64 sample, (export czechia barley) for 2 sample. It is safe to assume that the self-dependencies discovered in (import slovenia barley) and (export italy rye) were influenced by our interpolation. We can thus not consider them as part of the ground truth graph, as there is some uncertainty left. Regarding our previous evaluation, we assume

that PCMCI+ and LPCMCI predict better the lags than TCDF. Additionally, we suppose that the algorithms behave on this dataset the same as for the Artificial dataset[12]. With these considerations, we consider the intersection of two prediction as part of the ground truth graph of the dataset. The final relations we extract from this database are consequently:

export italy maize causes export austria oats with a lag of 6
import bulgaria maize causes itself with a lag of 1
import germany wheat causes import croatia wheat with a lag of 5
export lithuania barley causes itself with a lag of 1
import lithuania maize causes itself with a lag of 1
export czechia barley causes itself with a lag of 1
export romania wheat causes export italy maize with a lag of 9
export italy maize causes itself with a lag of 1

Table 14: Final relations extracted from the database

The results may however not be representative of the real behavior of this variables, as the initial data was not stationary. We passed the time series through the diff[19] to tackle this issue, but it may not have rendered the data stationary. Source [15] implies that differentiating may not be able to achieve this goal, suggesting that some variables could still be non-stationary. PCMCI+ and LPCMCI require stationary data as an input, but TCDF has not yet been tested with this kind of data. We can however assume that non-stationary data is not well suited for the algorithm, as the returned Summary causal graph needs the assumption throughout time to be represented (see 1.4). This uncertainty could lead to blatant errors in the outputs. The retrieved relations are mostly self causations, except for three of them. They are particularely noteworthy as they seem rather incoherent due to the fact that they link different countries and cereal. This could imply the existence of hidden confounders, which are not infered by our algorithms.

8 Bibliography

[1] Evaluation of network modelling methods for fmri. https://www.fmrib.ox.ac.uk/datasets/netsim/index.html.

[2] C. K. Assaad, E. Devijver, and E. Gaussier. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819, 2022.

[3] C. K. Assaad, E. Devijver, and E. Gaussier. Discovery of extended summary graphs in time series. *38th Conference on Uncertainty in Artificial Intelligence*, 2022.

[4] A. Barrett, L. Barnett, and A. Seth. Multivariate granger causality and generalized variance. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 81:041907, 04 2010. DOI: 10.1103/PhysRevE.81.041907.

[5] L. Cheng, R. Guo, R. Moraffah, P. Sheth, K. S. Candan, and H. Liu. Evaluation methods and measures for causal learning algorithms. *JOURNAL OF IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE*, 0(0), 2022.

[6] E. Commission. Cereal prices. https://agridata.ec.europa.eu/extensions/DashboardCereals/ExtCerealsPrice.html#, 2022.

[7] E. Commission. Cereal trades. https://agridata.ec.europa.eu/extensions/DashboardCereals/CerealsTrade.html#, 2022.

[8] T. S. community. scipy.interpolate.interp1d. https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html, 2008-2022.

[9] ETIS. Etis. https://www.etis-lab.fr/etis/#, 2022.

[10] A. Gerhardus and J. Runge. High-recall causal discovery for autocorrelated time series with latent confounders. *34th Conference on Neural Information Processing Systems*, 2020.

[11] A. Hyvärinen, K. Zhang, S. Shimizu, and P. O. Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(56): 1709–1731, 2010. URL http://jmlr.org/papers/v11/hyvarinen10a.html.

[12] A. Karim, D. Emilie, and G. Eric. Basic causal structures with additive noise. https://dataverse.harvard.edu/dataverse/basic_causal_structures_additive_noise, 2020.

[13] S. Kleinberg. Simulated financial time series. http://www.skleinberg.org/data.html, 2012.

[14] R. Moraffah, P. Sheth, M. Karami, A. Bhattacharya, Q. Wang, A. Tahir, A. Raglin, and H. Liu. Causal inference for time series analysis: Problems, methods and evaluation. Februay 2021.

[15] R. Nau. Stationarity and differencing. https://people.duke.edu/~rnau/411diff.htm, 2020.

[16] M. Nauta. Tcdf implementation. https://github.com/M-Nauta/TCDF.

[17] M. Nauta. Temporal causal discovery and structure learning with attention-based convolutional neural networks. August 2018.

[18] M. Nauta, D. Bucur, and C. Seifert. Causal discovery with attention-based convolutional neural networks. January 2019.

[19] T. pandas development team. pandas.dataframe.diff. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.diff.html#pandas-dataframe-diff, 2008-2022.

[20] J. Peters and P. Bühlmann. Structural intervention distance (sid) for evaluating causal graphs. *ResearchGate*, 2013.

[21] J. Peters and P. Bühlmann. Sid. https://cran.r-project.org/src/contrib/Archive/SID/, 2015.

[22] J. Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. *36th Conference on Uncertainty in Artificial Intelligence*, 2020.

[23] J. Runge, E. Gillies, E. V. Strobl, and S. Palachy-Affek. Tigramite package. https://github.com/jakobrunge/tigramite.

[24] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search, 2nd Edition*. 01 2000.

[25] Wikipedia. Adjacency matrix. https://en.wikipedia.org/wiki/Adjacency_matrix, .

[26] Wikipedia. Conditional independence. https://en.wikipedia.org/wiki/Conditional_independence, .

[27] Wikipedia. Law of total probability. https://en.wikipedia.org/wiki/Law_of_total_probability, .