Case Studies of Causal Discovery from IT Monitoring Time Series

Ali Aït-Bachir[1,*] **Charles K. Assaad**[1,*,†] **Christophe de Bignicourt**[1,*] **Emilie Devijver**[2,*]

Simon Ferreira[1,3,*] **Eric Gaussier**[2,*] **Hosein Mohanna**[1,*] **Lei Zan**[1,2,*]

[1]EasyVista, France
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, France
[3]École Normale Supérieure de Lyon, France
[*]Authors are listed in an alphabetical oder
[†]Corresponding author

Abstract

Information technology (IT) systems are vital for modern businesses, handling data storage, communication, and process automation. Monitoring these systems is crucial for their proper functioning and efficiency, as it allows collecting extensive observational time series data for analysis. The interest in causal discovery is growing in IT monitoring systems as knowing causal relations between different components of the IT system helps in reducing downtime, enhancing system performance and identifying root causes of anomalies and incidents. It also allows proactive prediction of future issues through historical data analysis. Despite its potential benefits, applying causal discovery algorithms on IT monitoring data poses challenges, due to the complexity of the data. For instance, IT monitoring data often contains misaligned time series, sleeping time series, timestamp errors and missing values. This paper presents case studies on applying causal discovery algorithms to different IT monitoring datasets, highlighting benefits and ongoing challenges.

1   INTRODUCTION

Information technology (IT) systems play a crucial role in the success of modern businesses. These systems are utilized for data storage and processing, communication with customers and suppliers, and the automation of various business processes. Given their significance, it is essential to monitor IT systems to ensure their proper functioning and efficiency [Tamburri et al., 2020]. IT monitoring has become increasingly valuable due to improved storage capacity, enabling the collection of extensive observational time series data [Tamburri et al., 2020]. Even though analyzing these large amounts of observational time series data can enhance efficiency and optimize processes [Chatzigiannakis et al., 2009], they also pose a significant challenge for many companies due to their complex nature.

The interest in causal discovery [Spirtes et al., 2000, Pearl et al., 2000, Chickering, 2002, Peters et al., 2017] is growing within the IT monitoring community [Meng et al., 2020, Li et al., 2022, Assaad et al., 2023b, Wang et al., 2023], as knowing causal relations allows for reducing downtime and enhancing the overall performance of IT systems by optimizing their resources and identifying areas for improvement. In addition, causal discovery can help IT professionals to swiftly identify actionable root causes of anomalies and incidents and to take corrective action to eliminate them [Meng et al., 2020, Assaad et al., 2023b]. Moreover, causal discovery can also be used to predict and preempt future issues in IT systems. By analyzing historical data, IT professionals can recognize patterns indicative of future problems and address them proactively. By leveraging causal discovery, IT professionals can enhance the efficiency, performance, and reliability of IT systems, leading to improved business outcomes.

However, analyzing IT monitoring data poses several challenges due to its complexity. IT monitoring data is often collected from multiple sources, resulting in misaligned time series. Additionally, there can also be non-informative time series. For example, there can be sleeping time series (for a period of time) due to users' inactivity on certain servers. Furthermore, timestamp errors can be present, and the low sampling rate of data in IT monitoring systems can complicate the search for causal relationships, as the lag between causes and effects may be relatively small.

This paper presents a case study for applying causal discovery algorithms to different IT monitoring datasets.

This study highlights the potential benefits of utilizing causal discovery techniques in IT monitoring and emphasizes the ongoing challenges and complexities associated with working with such data. These findings stress on the need for further research and development in this area to fully harness the potential of causal discovery algorithms in analyzing IT monitoring data.

The remainder of the paper is organized as follows: Section 2 presents preliminaries and the main algorithms applied to the case studies. Section 3 describes the IT monitoring dataset for each case study and discusses the challenges related to each dataset as well as the background knowledge and theories available to experts at the time of the application. Section 4 presents and discusses the results of causal discovery algorithm in each case study. Finally, Section 5 discusses challenges and points out some aspect of causal discovery from time series that are not included in any of the case studies and Section 6 concludes the paper.

2   SET UP

Causal discovery in time series aims at discovering, from observational data, causal relations within and between d-variate time series $\mathcal{X}$ where, for a fixed t, each $\mathcal{X}_t$ is a vector $(\mathcal{X}_t^1, \cdots, \mathcal{X}_t^d)$ in which each variable $\mathcal{X}_t^p$, such that $p \in \{1, \cdots, d\}$, represents a measurement of the p-th time series at time t.

2.1   CAUSAL GRAPHS FOR TIME SERIES

There are at least four ways to represent time series through a causal graph. The first is called a *full time causal graph* [Assaad et al., 2022a] and represents a infinite graph of the dynamic system, as illustrated in Figure 1a. Note that in this work, we *assume* that the *full time causal graph* is *acyclic*.

Definition 1 (Full time causal graph, Assaad et al. [2022a]). *Let $\mathcal{X}$ be a multivariate discrete-time stochastic process and $\mathcal{G}^f = (\mathcal{V}^f, \mathcal{E}^f)$ the associated full time causal graph. The set of vertices $\mathcal{V}^f$ in that graph consists of the set of components $\mathcal{X}_t^1, \ldots, \mathcal{X}_t^d$ at each time $t \in \mathbb{Z}$. The set of edges $\mathcal{E}^f$ of the graph are defined as follows: for each t, variables $\mathcal{X}_{t-\gamma}^p$ and $\mathcal{X}_t^q$ are connected by a lag-specific directed link $\mathcal{X}_{t-\gamma}^p \to \mathcal{X}_t^q$ if and only if $\mathcal{X}^p$ causes $\mathcal{X}^q$ at time t with a time lag of $0 \le \gamma$ for $p \ne q$ and with a time lag of $0 < \gamma$ for $p = q$.*

It is usually not possible to infer general full time causal graphs as there usually is a single observation for each time series at each time instant. Thus it is common to rely on the so-called *consistency throughout time assumption* [Assaad et al., 2022a] which states that all causal relationships remain constant in direction throughout time. When assuming consistency throughout time and because every causal relation has a maximal time lag γ_{max}, the full time causal graph can be represented through a time window by a finite graph of size $\gamma_{max} + 1$ which we call *window causal graph* [Assaad et al., 2022a].

Definition 2 (Window causal graph, Assaad et al. [2022a]). *Let $\mathcal{X}$ be a multivariate discrete-time stochastic process and $\mathcal{G}^w = (\mathcal{V}^w, \mathcal{E}^w)$ the associated window causal graph with a maximal lag γ_{max}. The set of vertices $\mathcal{V}^w$ in that graph consists of the set of components $\mathcal{X}_{t-\gamma}^1, \ldots, \mathcal{X}_{t-\gamma}^d$ at each time $t - \gamma$ for $0 \le \gamma \le \gamma_{max}$. The set of edges $\mathcal{E}^w$ of the graph are defined as follows: $\mathcal{X}_{t-\gamma}^p$ and $\mathcal{X}_t^q$ are connected by a directed link $\mathcal{X}_{t-\gamma}^p \to \mathcal{X}_t^q$ if and only if $\mathcal{X}_{t-\gamma}^p$ causes $\mathcal{X}_t^q$ in the full time causal graph (in this case then there is also a directed edge between each homologous pairs of nodes $\mathcal{X}_{t-\gamma-i}^p$ and $\mathcal{X}_{t-i}^q$ for $0 \le i \le \gamma_{max} - \gamma$).*

Figure 1b illustrates a window causal graph corresponding to the full time causal graph given in Figure 1a.

In practice, it can be sufficient to know the causal relations between time series as a whole, without knowing precisely the relations between time instants; in addition, in some applications, an expert would like to validate a causal graph before using it, but validating a window causal graph and its temporal lags between causes and effects can be difficult. In these cases, one can use an abstraction of the window graph which usually takes the form of an *extended summary causal graph* [Assaad et al., 2022b] or a *summary causal graph* [Assaad et al., 2022a]. An example of these two abstract graphs are given in Figure 1c and 1d.

Definition 3 (Extended summary causal graph, Assaad et al. [2022b]). *Let $\mathcal{X}$ be a multivariate discrete-time stochastic process and $\mathcal{G}^e = (\mathcal{V}^e, \mathcal{E}^e)$ the associated extended summary causal graph. The set of vertices $\mathcal{V}^e$ in that graph consists of the set of time slices $\mathcal{V}_{t-}^e$ and $\mathcal{V}_t^e$ such that $\mathcal{V}_{t-}^e = \mathcal{X}_{t-}^1, \ldots, \mathcal{X}_{t-}^d$ and $V_t^e = \mathcal{X}_t^1, \ldots, \mathcal{X}_t^d$. The set of edges $\mathcal{E}^e$ are defined as follows:*

- *variables $\mathcal{X}_t^p$ and $\mathcal{X}_t^q$ with $p \ne q$ are connected by a directed link $\mathcal{X}_t^p \to \mathcal{X}_t^q$ if and only if $\mathcal{X}^p$ causes $\mathcal{X}^q$ at time t with a null time lag;*
- *variables $\mathcal{X}_{t-}^p$ and $\mathcal{X}_t^q$ are connected by a directed link $\mathcal{X}_{t-}^p \to \mathcal{X}_t^q$, if and only if $\mathcal{X}^p$ causes $\mathcal{X}^q$ at time t with a strictly positive time lag.*

Definition 4 (Summary causal graph, Assaad et al. [2022a]). *Let $\mathcal{X}$ be a multivariate discrete-time stochastic process and $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the associated summary causal graph. The set of vertices $\mathcal{V}$ in that graph consists of the set of time series $\mathcal{X}^1, \ldots, \mathcal{X}^d$. The set of*
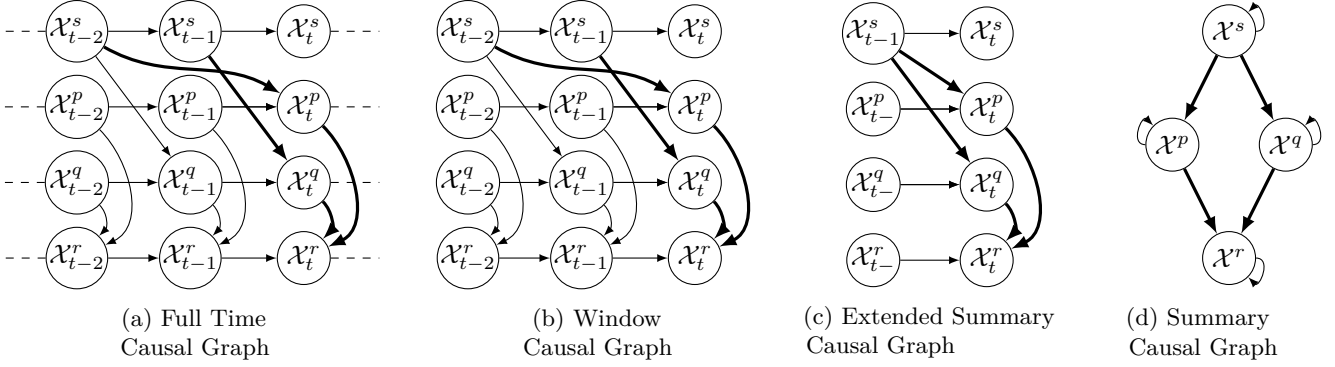
Figure 1: Different causal graphs to respresent a diamond structure with self causes: full time causal graph (a), window causal graph (b), extended summary causal graph (c) and summary causal graph (d). Note that the first one gives more information but cannot be inferred in practice, the second one is a schematic viewpoint of the full behavior, the third one only distinguishes between instantaneous and lagged causal relations, whereas the last one gives an overview of the causal relationships without any reference to time.

edges $\mathcal{E}$ of the graph are defined as follows: variables $\mathcal{X}^p$ and $\mathcal{X}^q$ are connected if and only if there exists some time t and some time lag i such that $\mathcal{X}^p_{t-i}$ causes $\mathcal{X}^q_t$ at time t with a time lag of $0 \leq i$ for $p \neq q$ and with a time lag of $0 < i$ for $p = q$.

Note that the summary causal graph can contain cycles which is not the case for extended summary causal graphs.

2.2 ASSUMPTIONS

Given observational data, on which one can compute correlations and statistical independencies, it is not always possible to infer a causal graph. In addition to the acyclicity of the full time causal graph and consistency throughout time, all the algorithms considered in this work rely on some of the following assumptions:

- Causal Markov condition [Spirtes et al., 2000, Pearl et al., 2000]: every variable is independent of all its nondescendants in the graph conditional on its parents;

- Causal sufficiency [Spirtes et al., 2000, Pearl et al., 2000]: all common causes, i.e., confounders, of all observed variables are observed;

- Minimality [Spirtes et al., 2000]: all adjacent nodes are dependent;

- Faithfulness [Spirtes et al., 2000, Pearl et al., 2000]: all conditional independencies are entailed from the causal Markov condition;

- Semi-parametric model [Peters et al., 2017], which stipulates a general form for the underlying model, as linear models or nonlinear additive noise models;

- Stationarity: the generative process does not change with respect to time.

2.3 ALGORITHMS

Granger Causality is one of the oldest methods proposed to detect causal relations between time series. However, in its standard form [Granger, 1969], it is known to handle a restricted version of causality that focuses on linear relations and causal priorities as it assumes that the past of a cause is necessary and sufficient for optimally forecasting its effect. This approach has nevertheless been improved since then [Granger, 2004, Arnold et al., 2007] through, *e.g*, the use of variable selection tools and result of this method can be represented in the form of a summary causal graph. Namely, GCMVL [Arnold et al., 2007] is multivariate Granger algorithm that use a lasso-based technique for variable selection.

Score-based approaches [Chickering, 2002] search over the space of possible graphs trying to maximize a score that reflects how well the graph fits the data. Recently, a new score-based method called Dynotears[1] [Pamfil et al., 2020] was presented to infer a window causal graph from time series.

Constraint-based approaches, based on the PC algorithm [Spirtes et al., 2000], are certainly one of the most popular approaches for inferring causal graphs. Several algorithms, adapted from non-temporal causal graph discovery algorithms, have been proposed in this family for time series, among which PCMCI is capable of infering a window causal graph and accounts for the effect size. Initially, PCMCI [Runge et al., 2019] was

---

[1]`https://github.com/quantumblacklabs/causalnex/`

		Causal graph	Causal Markov Condition	Causal sufficiency	Faithfulness / Minimality	Semi-parametric model	Linear model	Consistency throughout time	Stationarity	Instantaneous relations	Misaligned time series	Sleeping time series	Timestamp errors	Missing values	Different sampling rate
Algorithms	GCMVL	S/E	✓	✓		✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
	Dynotears	W	✓	✓		✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
	PCMCI$^+$	W	✓	✓	F	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗
	PCGCE	E	✓	✓	F	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗
	VarLiNGAM	W	✓	✓	M	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
	TiMINo	S	✓	✓	M	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗
	NBCB-w	S/W	✓	✓	M	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗
	NBCB-e	S/W	✓	✓	M	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗
	CBNB-w	S/W	✓	✓	M	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗
	CBNB-e	S/W	✓	✓	M	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗
Datasets	MoM	S	✓	✓	?	?	?	✗	✗	✓	✗	✓	✓	✗	✗
	Ingestion	S	✓	✓	?	?	?	✗	✗	✓	✗	✓	✓	✗	✗
	Web activity	S	✓	?	?	?	?	✗	✗	✓	✓	✓	✓	✗	✗
	Antivirus	S	✓	?	?	?	?	✗	✗	✓	✓	✓	✓	✗	✓

Table 1: Summary of the main characteristics of algorithms and different IT monitoring datasets considered in the paper. For causal graphs, S means that the algorithm provides a summary causal graph, E means that the algorithm provides an extended summary causal graph and W means that the algorithm provides a window causal graph; F corresponds to faithfulness and M to minimality. An empty cell mean that the information given in the corresponding column was not discussed by the authors of the corresponding algorithm. A question mark means that the expert of the IT system do not know if the information given in the corresponding column is satisfied for the given dataset.

not able to take into account instantaneous relations but this limitation was recently surmounted with the introduction of PCMCI^{+2}[Runge, 2020]. Another algorithm in this family is PCGCE^3 [Assaad et al., 2022b] which infers an extended summary causal graph by restructuring the data into two slices: one vector for each time series that represents the present and one matrix of each time series that represents the past (up to γ_{max}).

In a different line, approaches based on Structural Equation Models assume that the causal system can be defined by a set of equations that explain each variable by its direct causes and an additional noise. Causal relations are in this case discovered using footprints produced by the causal asymmetry in the data. For time series, the most popular algorithms in this family are VarLiNGAM^4 [Hyvärinen et al., 2008, Hyvärinen et al., 2010], which is an extension of LiNGAM [Shimizu et al., 2011] through autoregressive models that infers a window causal graph, and TiMINo^5 [Peters et al., 2013], which discovers a causal relationship in form of a summary causal graph by looking at independence between the noise and the potential causes.

There exist also hybrid algorithms which combine constraint-based with semi-parametric algorithms. Among hybrid methods, NBCB [Assaad et al., 2021] starts by discovering the causal order between time series through a semi-parametric strategy (which yields a graph that contains the true graph), and then prunes unnecessary edges using a constraint-based strategy. Initially, this method assumes that the summary causal graph is acyclic but this limitation was recently surmounted [Assaad et al., 2023a]. In addition, in the new generalized version, NBCB is considered more like a framework that combines any semi-parametric strategy and constraint-based strategy. In this paper, we con-

---

[2]`https://github.com/jakobrunge/tigramite`
[3]`https://github.com/ckassaad/PCGCE`
[4]`https://github.com/cdt15/lingam`

[5]`http://web.math.ku.dk/~peters/code.html`

sider NBCB-w[6] which combines a restricted version of VarLiNGAM and a restricted version of PCMCI$^+$ as well as NBCB-e[6] which combines a restricted version of VarLiNGAM and a restricted version of PCGCE. NBCB-w infers a window causal graph as it is based on PCMCI$^+$ and NBCB-e infer an extended summary causal graph as it is based on PCGCE. Another hybrid-based framework exists and it is called CBNB [Assaad et al., 2023a]. It can be considered as a backward version of NBCB. In this paper, we consider the two algorithms CBNB-w[6] and CBNB-e[6] from the CBNB framework which can be considered respectively as the backward versions of NBCB-w and NBCB-e.

In Table 1, we classify causal discovery algorithms with respect to the assumptions they rely on in addition to different characteristics.

2.4 HYPER-PARAMETRES

For PCMCI$^+$, PCGCE, NBCB-w, NBCB-e, CBNB-w, and CBNB-e we use the linear partial correlation to find conditional independencies and for PCGCE (as well as for NBCB-e and CBNB-e), as the authors suggested [Assaad et al., 2022b], we reduce the dimensionality of the past slice to 1 using PCA. For TiMINo [Peters et al., 2013] we use the linear time series model and the HSIC test and for VLiNGAM, the regularization parameter in the adaptive Lasso is selected using BIC. For Dynotears, we set all hyperparameters to their recommended values ($\lambda_W = \lambda_A = 0.05$ and $\alpha_W = \alpha_A = 0.01$). For all methods, we set the significance threshold to 0.05 since according to IT monitoring experts the maximal delay between a cause and its effect is of 15 minutes, in our experiments we set the maximal lag γ_{max} according to the sampling rate and to the 15 minutes delay. For instance, for a sampling rate of 1 minute we set γ_{max} to 15 and for a sampling rate of 5 minute we set γ_{max} to 3. In Appendix, we also study how the results change by varying γ_{max}.

2.5 PRE-PROCESSING

Time series in monitoring systems are not always exactly aligned together and come in different sampling rates as the timestamps depend on when the data was collected. In the following we present two pre-processing strategies that we considered for aligning time series:

- Strategy 1: Time series are analyzed in terms of sampling rates and the lowest one is chosen. Afterwards, all the time series are re-sampled according to this lowest sampling rate with the closest value

to the timestamp taken as the new value. Upon re-sampling, missing values can be clearly observed. If missing values are detected, they are filled using simple linear interpolation of Pandas data frames[7].

- Strategy 2: Each raw value x_i is converted into integral value s_i at each point i as follows: $s_i = x_i(t_i - t_{i-1}) + s_{i-1}$. Then all time series are re-sampled such that each re-sampled value x_j at every n (the lowest sampling rate) steps is calculated as follows: $x_j = \frac{s_i - s_{i-n}}{t_i - t_{i-n}}$. The time t_i (of value s_i) is the time that is after the corresponding time to x_j.

3 DATASETS DESCRIPTION

In this section, we present the summary causal graph (the window causal graph and the extended summary causal graph are not available) and the datasets[8] for each case study. All summary causal graphs are constructed either by IT monitoring experts or directly using the system topology. Note that all data points are collected using Nagios[9], an open-source software that monitors systems, networks and infrastructure, and which gives the timestamp according to the collection time which does not necessarily correspond to the real time of the value. In addition, on some of the case studies, the alignment between time series is not guaranteed as data collection is performed by different plugins with different starting times and different sampling rates [Holzinger et al., 2021]. In Table 1, we also classify datasets with respect to the different assumptions needed by causal discovery algorithms and to other different characteristics. In Appendix, we also give additional information on the datasets.
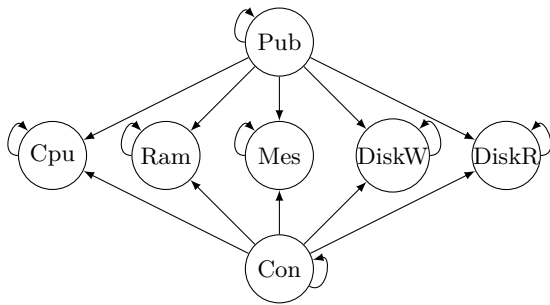
3.1 MOM ACTIVITY DATASETS

First, we consider two Middleware oriented Message datasets which we denote as MoM 1 and MoM 2 and which are defined through the monitoring of an IT pipeline which ingests incoming messages based on a Publish/Subscribe architecture. These datasets contain seven different time series such as MoM 1 consists of 288 timestamps, MoM 2 consists of 364 timestamps, and they are both collected with a one-second sampling rate. Note that there is no overlapping time between these
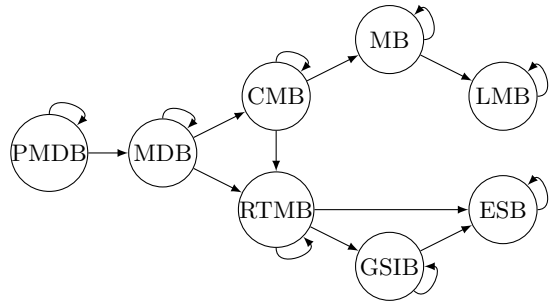
---

[6]https://github.com/ckassaad/Hybrids_of_CB_and_NB_for_Time_Series

[7]https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.interpolate.html

[8]All datasets are available at https://easyvista2015-my.sharepoint.com/:f:/g/personal/aait-bachir_easyvista_com/ElLiNpfCkO1JgglQcrBPP9IBxBXzaINrM5fOILz6wbgoEQ?e=OBTsUY
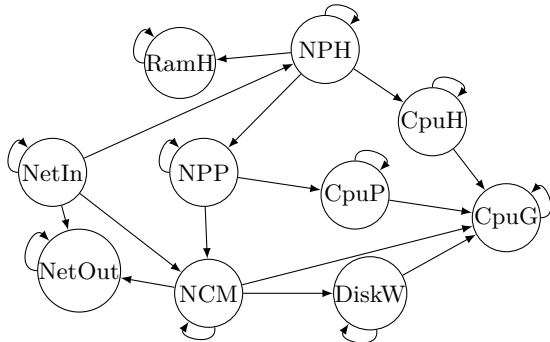
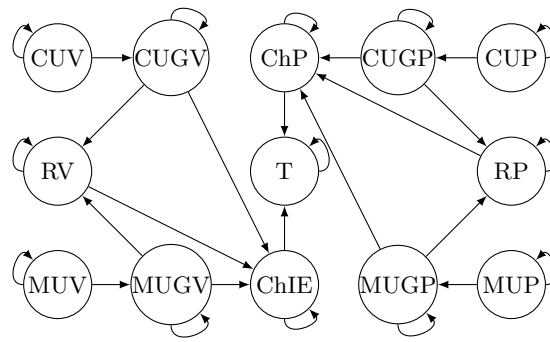[9]https://www.nagios.org/

(a) MoM system.

(b) Ingestion IT monitoring system.

(c) Web-Activity.

(d) Antivirus-Activity.

Figure 2: Summary causal graphs for different datasets: MoM system based on Publish/Subscribe architecture (a), Ingestion IT monitoring system (b), Web-Activity (c) and Antivirus-Activity (d). Those summary causal graphs are constructed either by IT monitoring system experts or directly using the system topology.

datasets. The corresponding summary causal graph is presented in Figure 2a where *Pub* represents the publish rate that monitor the number of messages per seconds; *Con* is the number of consumers; *Mes* represents the number of messages remaining in the queue; *Cpu* represents the percentage of used CPU; *Ram* represents the percentage of used RAM; *DiskW* represents the Disk write in Kbytes/second; *DiskR* represents the Disk read in Kbytes/second. There might exist additional links between *Cpu, Ram, DiskW* and *DiskR* under extreme conditions and abnormal behavior but these relations are excluded from the graph since there is no incident or clear anomaly that is detected for these datasets. Note that, for these datasets, timestamps of different time series are aligned.

3.2 INGESTION ACTIVITY DATASET

We also consider a dataset introduced in Assaad et al. [2023b] which we denote as the Ingestion dataset. This dataset contains eight time series which consist of 991 timestamps collected with a one-minute sampling rate. The corresponding summary causal graph which is constructed using Storm ingestion topology that describes the relations between the inputs and outputs of each Bolt is provided in Figure 2b where *PMDB* represents

the extraction of some information about the messages received by the Storm ingestion system; *MDB* refers to an activity of a process that orients messages to other processes with respect to different types of messages; *CMB* represents the activity of extraction of metrics from messages; MB represents the activity of insertion of data in a database; *LMB* reflects the updates of the last values of metrics in Cassandra; *RTMB* represents the activity of searching to merge data with information coming from the check message bolt; *GSIB* represents the activity of insertion of historical status in database; *ESB* represents the activity of writing data in Elasticsearch. All values are calculated by multiplying the number of messages executed on the specific bolt in a given time window by the average execution latency in the same time window, and then dividing it by the time window which corresponds to 10 minutes. Note that, for this dataset, timestamps of different time series are aligned.

3.3 WEB ACTIVITY DATASET

We consider a dataset that reflects the activity in a web server. This dataset contains ten time series collected with a one-minute sampling rate. The raw data of this case study were initially misaligned. In order to

align them, we use the two pre-processing strategies described in Section 2.5. We denote the dataset pre-processed using Strategy 1 as Web 1 and the dataset pre-processed using Strategy 2 as Web 2. The two processed datasets contains 3000 timestamps. The corresponding summary causal graph is presented in Figure 2c where *NetIn* represents the data received by the network interface card in Kbytes/second; *NetOut* represents the data transmitted out by the network interface card in Kbytes/second; *NPH* represents the number of HTTP processes; *NPP* represents the number of PHP processes; *NCM* represents the number of open MySql connections which are started by PHP processes; *CpuH* represents the percentage of CPU used by all HTTP processes; *RamH* represents the percentage of RAM used by all HTTP processes; *CpuP* represents the percentage of CPU used by all PHP processes; *DiskW represents the Disk write in Kbytes/second*; *CpuG* represents the percentage of global CPU usage.

3.4 ANTIVIRUS ACTIVITY DATASET

Lastly, we consider a dataset which depicts the impacts of antivirus activity in servers. This dataset contains 13 time series such that 3 of them are collected with a one-minute sampling rate and the rest with a five-minutes sampling rate. The raw data of this case study were initially misaligned. In order to align them, we use the two pre-processing strategies described in Section 2.5, leading to the dataset Antivirus 1 for Strategy 1 and Antivirus 2 for Strategy 2. The two processed datasets consist of 1321 timestamps. The corresponding summary causal graph is presented in Figure 2d where *CUV* represents the percentage of CPU usage of antivirus processes in server V; *CUGV* represents the percentage of CPU usage of the global server V; *MUV* represents the percentage of memory usage of antivirus process; *MUGV* represents the percentage of global memory usage of the server; *RV* represents the Disk IO read in Kbytes/second; *ChIE* refers to the required duration in seconds to open an *IE browser* on server V; *CUP* represents the percentage of CPU usage of antivirus processes in server P; *CUGP* represents the percentage of CPU usage of the global server P; *MUP* represents the percentage of memory usage of antivirus process; *MUGP* represents the percentage of global memory usage of the server; *RP* represents the Disk IO read in Kbytes/second; *ChP* represents refers to the required duration in seconds to open a *CITRIX Portal* on server P; *T* represents the global time in seconds required to open a CITRIX portal and open the IE browser.

4 RESULTS

In this section we evaluate the performance of each causal discovery algorithm presented in Section 2 on each dataset presented in Section 3. Since we only have access to the true summary causal graph, we evaluate the detection of oriented edges of the summary causal graph (for algorithms that detect a window causal graphs or an extended summary causal graph, we start by inferring the window causal graphs or the extended summary causal graph then deduce the summary causal graph from it) using the F1-score. Recall that as mentioned in Section 2, γ_{max} is set according to the 15 seconds delay rule for MoM datasets and the 15 minutes delay rule for the other datasets.

All results [10] are presented in Table 2. GCMVL exhibits poor performance on MoM 1, MoM 2, Web 2, Antivirus 1, and Antivirus 2 datasets. However, it shows better performance on Ingestion and Web 1 datasets, achieving at most an F1-score of 0.2. Dynotears demonstrates relatively better performance compared to GCMVL across all datasets, except for the Ingestion dataset. PCMCI$^+$, NBCB-w and CBNB-w achieve the highest F1-score of 0.4 on the MoM 1 dataset along with NBCB-w and CBNB-w. However, PCMCI$^+$ performs poorly on the MoM 2 dataset and on the Ingestion dataset. On Web 1 and Web 2, it respectively achieves better F1-scores of 0.23 and 0.3, and lower F1-scores of 0.04 and 0.11 on Antivirus 1 and Antivirus 2. PCGCE exhibits low performance on the MoM 1 dataset but has better performance on MoM 2, Ingestion, Web 1, and Web 2 with F1-scores of 0.12, 0.12, 0.22, and 0.15, respectively. Remarkably, PCGCE performs well on Antivirus datasets, achieving the highest F1-score of 0.45 for Antivirus 2, along with NBCB-e. VLiNGAM shows poor performance on MoM 1 and MoM 2 datasets. However, it shows better performance on the other datasets and achieves the highest F1-score of 0.29 on Web 1 dataset. TiMINo performs poorly on the majority of datasets but shows better performance on MoM 2 and Ingestion datasets. For all datasets, NBCB-w and CBNB-w (which are two hybrids methods which comines PCMCI$^+$ with VLiNGAM) either outperform PCMCI$^+$ or performs equally to PCMCI$^+$ and they outperform VLiNGAM for only 3 datasets. On the other hand, for most datasets, NBCB-e and CBNB-e (which are two hybrids methods which comines PCGCE with VLiNGAM) outperform PCCGE and VLiNGAM except for Web 1 and Antivirus 2. In addition, in most datasets NBCB-e outperform CBNB-e. Notably, NBCB-e achieves the best F1-scores in most datasets (MoM 2, Ingestion, Web 2, Antivirus 1 and Antivirus 2) and

---

[10]The code is available at `https://github.com/ckassaad/Case_Studies_of_Causal_Discovery_from_IT_Monitoring_Time_Series`

Table 2: Results for real IT monitoring datasets where γ_{max} is set according to the 15 seconds delay rule for MoM datasets and to the 15 minutes delay rule for Ingestion, Web and Antivirus datasets. We report the F1-score.

	MoM 1	MoM 2	Ingestion	Web 1	Web 2	Antivirus 1	Antivirus 2
GCMVL	0.0	0.0	0.2	0.2	0.0	0.08	0.0
Dynotears	0.26	0.2	0.14	0.23	0.3	0.18	0.19
PCMCI$^+$	**0.4**	0.0	0.0	0.23	0.3	0.04	0.11
PCGCE	0.0	0.12	0.12	0.22	0.15	0.3	**0.45**
VLiNGAM	0.0	0.0	0.19	**0.29**	0.18	0.15	0.22
TiMINo	0.0	0.17	0.18	0.0	0.0	0.0	0.0
NBCB-w	**0.4**	0.0	0.13	0.23	0.3	0.14	0.24
NBCB-e	0.13	**0.29**	**0.27**	0.19	**0.42**	**0.31**	**0.45**
CBNB-w	**0.4**	0.0	0.15	0.23	0.3	0.17	0.16
CBNB-e	0	0.24	0.13	0.22	0.29	**0.31**	0.38

has relatively good performance on the other datasets.

Note that all algorithms, except GCMVL and TiMINo, have better performance on the Antivirus dataset when pre-processing Strategy 2 is applied.

Overall, according to the performance of each algorithm, NBCB-e seems to be the best choice across all datasets. However, it is important to note that the best performance achieved (0.45 in Antivirus 2) is far from being satisfactory.

5   DISCUSSION

As shown in the previous section, the results of causal discovery algorithms considered in this work are not satisfactory. Most probably this is due to the violation of the assumptions that these algorithms rely on. These algorithms typically assume *Consistency throughout time* and *Stationarity*, yet IT systems exhibit different states (e.g., normal, warning, critical). Transitions between these states can potentially induce changes in the causal strengths between metrics or even completely alter the underlying causal graph. Consequently, it will be interesting to test methods that relax this assumption such as CD-NOD [Huang et al., 2020], R-PCMCI [Saggioro et al., 2020], LoSST [Kummerfeld and Danks, 2013], and SDCI [Rodas et al., 2021]. On top of that, the linearity assumption is not verified in any of the datasets, thus it would be interesting to test nonlinear methods, for example, by using non-linear independence test in constraint-based methods and non-linear regression models in semi-parametric methods.

It is also assumed throughout this paper that the full time graph is acyclic which coincides well with the summary causal graphs of our case studies. However, in general, considering low sampling rate challenges the legitimacy of this assumption as there might be two lagged causal relation with a lag smaller than the time delay between each two collected data points. Ad-

ditionally, in this work we only focused on continuous data, however, IT systems comprise not only continuous variables but also ordinal variables (*e.g.*, CPU frequency) and nominal variables (*e.g.*, device states: normal, busy, overcharged) which can help improve performance. To address mixed data types, it is worth exploring independent measures and tests for mixed data. Prominent methods include SCPC [Cui et al., 2016], MGVI [Tsagris et al., 2018], MIIC [Cabeli et al., 2020], LH [Marx et al., 2021], RAVK [Rahimzamani et al., 2018], MS [Mesner and Shalizi, 2020], and CMIh [Zan et al., 2022], as they can be integrated directly into constraint-based algorithms. Furthermore, missing data poses a significant challenge in determining the causal graph. This issue could be fixed by methods like MVPC [Tu et al., 2019] and CBR-PC [Gain and Shpitser, 2018]. Finally, there might always be hidden common causes in the system, so using methods based on FCI [Spirtes et al., 2000, Gerhardus and Runge, 2020, Assaad et al., 2022b] might be useful, but in this case, the graph will be in most cases less informative and less interpretable by IT monitoring experts.

6   CONCLUSION

IT systems are crucial for the success of modern businesses, and monitoring them is essential to ensure their proper functioning. Causal discovery techniques offer powerful tools for identifying the root causes of issues, optimizing IT systems, and predicting future problems. However, the analysis of IT monitoring data presents challenges due to its complexity and volume. The case study presented in this paper shows both the potential benefits and ongoing challenges of applying causal discovery algorithms to IT monitoring data. This area should continue to be an active field of research and development, with the aim of improving the efficiency and performance of IT systems in diverse industries and applications.

References

Andrew Arnold, Yan Liu, and Naoki Abe. Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 66–75, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936097. doi: 10.1145/1281192.1281203.

Charles K. Assaad, Emilie Devijver, Eric Gaussier, and Ali Ait-Bachir. A mixed noise and constraint-based approach to causal inference in time series. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 453–468, Cham, 2021. Springer International Publishing. ISBN 978-3-030-86486-6.

Charles K. Assaad, Emilie Devijver, and Eric Gaussier. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819, feb 2022a.

Charles K. Assaad, Emilie Devijver, and Eric Gaussier. Discovery of extended summary graphs in time series. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 96–106. PMLR, 01–05 Aug 2022b.

Charles K. Assaad, Daria Bystrova, Julyan Arbel, Emilie Devijver, Eric Gaussier, and Wilfried Thuiller. Hybrids of constraint-based and noise-based algorithms for causal discovery from time series. *arXiv preprint arXiv:2306.08765*, 2023a.

Charles K. Assaad, Imad Ez-Zejjari, and Lei Zan. Root cause identification for collective anomalies in time series given an acyclic summary causal graph with loops. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 8395–8404. PMLR, 25–27 Apr 2023b.

Vincent Cabeli, Louis Verny, Nadir Sella, Guido Uguzzoni, Marc Verny, and Hervé Isambert. Learning clinical networks from medical records based on information estimates in mixed-type data. *PLoS computational biology*, 16(5):e1007866, 2020.

Vasilis Chatzigiannakis, Symeon Papavassiliou, and Georgios Androulidakis. Improving network anomaly detection effectiveness via an integrated multi-metric-multi-link (m3l) pca-based approach. *Secur. Commun. Networks*, 2:289–304, 2009.

David Maxwell Chickering. Learning equivalence classes of bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002. ISSN 1532-4435. doi: 10.1162/153244302760200696.

Ruifei Cui, Perry Groot, and Tom Heskes. Copula pc algorithm for causal discovery from mixed data. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II 16*, pages 377–392. Springer, 2016.

Alexander Gain and Ilya Shpitser. Structure learning under missing data. In *International conference on probabilistic graphical models*, pages 121–132. PMLR, 2018.

Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12615–12625. Curran Associates, Inc., 2020.

Clive Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–38, 1969. URL https://EconPapers.repec.org/RePEc:ecm:emetrp:v:37:y:1969:i:3:p:424-38.

Clive W. J. Granger. Time series analysis, cointegration, and applications. *The American Economic Review*, 94(3):421–425, 2004. ISSN 00028282. URL http://www.jstor.org/stable/3592936.

Kilian Holzinger, Henning Stubbe, Franz Biersack, Angela Gonzalez Mariño, Abdoul Kane, Francisco Fons Lluis, Zhang Haigang, Thomas Wild, Andreas Herkersdorf, and Georg Carle. Precise real-time monitoring of time-critical flows. CoNEXT '21, page 489–490, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450390989. doi: 10.1145/3485983.3493356.

Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *The Journal of Machine Learning Research*, 21(1):3482–3534, 2020.

Aapo Hyvärinen, Shohei Shimizu, and Patrik O. Hoyer. Causal modelling combining instantaneous and lagged effects: An identifiable model based on non-gaussianity. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages

424–431, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390210. URL http://doi.acm.org/10.1145/1390156.1390210.

Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O. Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(56):1709–1731, 2010.

Erich Kummerfeld and David Danks. Tracking time-varying graphical structure. *Advances in neural information processing systems*, 26, 2013.

Mingjie Li, Zeyan Li, Kanglin Yin, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. Causal inference-based root cause analysis for online service systems with intervention recognition. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3230–3240, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539041.

Alexander Marx, Lincen Yang, and Matthijs van Leeuwen. Estimating conditional mutual information for discrete-continuous mixtures using multi-dimensional adaptive histograms. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 387–395. SIAM, 2021.

Yuan Meng, Shenglin Zhang, Yongqian Sun, Ruru Zhang, Zhilong Hu, Yiyin Zhang, Chenyang Jia, Zhaogang Wang, and Dan Pei. Localizing failure root causes in a microservice through causality inference. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, pages 1–10, 2020. doi: 10.1109/IWQoS49365.2020.9213058.

Octavio César Mesner and Cosma Rohilla Shalizi. Conditional mutual information estimation for mixed, discrete and continuous data. *IEEE Transactions on Information Theory*, 67(1):464–484, 2020.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1595–1605. PMLR, 26–28 Aug 2020.

Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19:2, 2000.

Jonas Peters, D. Janzing, and B. Schölkopf. Causal inference on time series using restricted structural equation models. In *Advances in Neural Information Processing Systems 26*, pages 154–162, 2013.

Jonas Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

Arman Rahimzamani, Himanshu Asnani, Pramod Viswanath, and Sreeram Kannan. Estimators for multivariate information measures in general probability spaces. *Advances in Neural Information Processing Systems*, 31, 2018.

Carles Balsells Rodas, Ruibo Tu, and Hedvig Kjellstrom. Causal discovery from conditionally stationary time-series. *arXiv preprint arXiv:2110.06257*, 2021.

Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 1388–1397. PMLR, 03–06 Aug 2020.

Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, 5(11), 2019. doi: 10.1126/sciadv.aau4996.

Elena Saggioro, Jana de Wiljes, Marlene Kretschmer, and Jakob Runge. Reconstructing regime-dependent causal relationships from observational time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(11):113115, 2020.

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, and Kenneth Bollen. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research*, 12:1225–1248, 2011. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1953048.2021040.

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

Damian A. Tamburri, Marco Miglierina, and Elisabetta Di Nitto. Cloud applications monitoring: An industrial study. *Information and Software Technology*, 127:106376, 2020. ISSN 0950-5849. doi: https://doi.org/10.1016/j.infsof.2020.106376.

Michail Tsagris, Giorgos Borboudakis, Vincenzo Lagani, and Ioannis Tsamardinos. Constraint-based causal discovery with mixed data. *International journal of data science and analytics*, 6:19–30, 2018.

Ruibo Tu, Cheng Zhang, Paul Ackermann, Karthika Mohan, Hedvig Kjellström, and Kun Zhang. Causal discovery in the presence of missing data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1762–1770. PMLR, 2019.

Sheng Wang, Qiang Zhao, Yinghua Han, and Jinkuan Wang. Root cause diagnosis for process faults based on multisensor time-series causality discovery. *Journal of Process Control*, 122:27–40, 2023. ISSN 0959-1524. doi: https://doi.org/10.1016/j.jprocont.2022.12.006.

Lei Zan, Anouar Meynaoui, Charles K. Assaad, Emilie Devijver, and Eric Gaussier. A conditional mutual information estimator for mixed data and an associated conditional independence test. *Entropy*, 24(9), 2022. ISSN 1099-4300. doi: 10.3390/e24091234. URL https://www.mdpi.com/1099-4300/24/9/1234.

A APPENDIX

In the following, we start by presented additional experimental results then present an examination of the datasets we have considered.

A.1 ADDITIONAL RESULTS

Table 3: Results for real IT monitoring datasets for $\gamma_{max} = 15$. We report the F1-score.

	MoM 1	MoM 2	Ingestion	Web 1	Web 2	Antivirus 1	Antivirus 2
GCMVL	0.0	0.0	0.2	0.29	0.0	0.08	0.0
Dynotears	0.26	0.2	0.14	0.24	**0.34**	0.19	0.25
PCMCI$^+$	**0.4**	0.0	0.0	0.22	0.31	0.1	0.13
PCGCE	0.0	0.12	0.12	**0.3**	0.27	0.27	**0.26**
VLiNGAM	0.0	0.0	0.19	0.24	0.17	0.19	0.16
TiMINo	0.0	0.17	0.18	0.0	0.13	0.0	0.07
NBCB-w	**0.4**	0.0	0.13	0.18	0.23	0.13	0.19
NBCB-e	0.13	**0.29**	**0.27**	0.19	0.22	0.22	0.15
CBNB-w	**0.4**	0.0	0.15	0.22	0.29	0.2	0.19
CBNB-e	0.0	0.24	0.13	0.23	0.33	**0.28**	0.22

Table 4: Results for real IT monitoring datasets for $\gamma_{max} = 10$. We report the F1-score.

	MoM 1	MoM 2	Ingestion	Web 1	Web 2	Antivirus 1	Antivirus 2
GCMVL	0.0	0.0	0.0	**0.32**	0.0	0.09	0.0
Dynotears	**0.36**	0.14	0.14	0.22	**0.39**	0.18	0.22
PCMCI$^+$	0.0	0.0	0.0	0.22	0.31	0.07	0.14
PCGCE	0.0	0.0	0.11	0.27	0.24	**0.33**	0.27
VLiNGAM	0.27	0.09	**0.27**	0.22	0.18	0.19	0.16
TiMINo	0.0	0.17	0.17	0.0	0.0	0.06	0.06
NBCB-w	0.15	0.0	0.13	0.19	0.23	0.15	0.25
NBCB-e	0.13	**0.2**	0.18	0.25	0.22	0.26	0.21
CBNB-w	0.15	0.0	0.16	0.22	0.29	0.2	0.21
CBNB-e	0.0	0.12	0.11	0.21	0.26	**0.33**	**0.29**

Table 5: Results for real IT monitoring datasets for $\gamma_{max} = 5$. We report the F1-score.

	MoM 1	MoM 2	Ingestion	Web 1	Web 2	Antivirus 1	Antivirus 2
GCMVL	0.0	0.0	0.0	0.19	0.0	0.08	0.0
Dynotears	0.27	**0.21**	0.14	0.22	0.3	0.18	0.17
PCMCI$^+$	0.0	0.15	0.0	0.17	0.32	0.04	0.11
PCGCE	**0.31**	0.0	0.22	0.21	0.34	0.3	0.36
VLiNGAM	0.0	0.19	**0.25**	0.23	0.2	0.18	0.18
TiMINo	0.0	0.0	0.18	0.0	0.0	0.0	0.0
NBCB-w	0.0	0.12	0.13	0.2	0.23	0.13	0.3
NBCB-e	0.27	0.0	0.11	**0.24**	**0.42**	0.29	**0.38**
CBNB-w	0.0	0.13	0.15	**0.24**	0.29	0.18	0.18
CBNB-e	**0.31**	0.0	0.13	0.15	0.38	**0.33**	0.27

Tables 3, 4, 5, and 6 present the F1-scores for each method using different values of γ_{max} (15, 10, 5, and 3, respectively). Among these methods, GCMVL performs poorly on all datasets, except for Web 1 dataset where it achieves the highest F1-scores of 0.32, when $\gamma_{max} = 10$. Dynotears demonstrates stable performance across various datasets when γ_{max} is varied. It achieves the highest F1-scores on the Web 2 dataset with a large values of γ_{max}, and on the MoM 2 dataset with a small values of γ_{max}.

Table 6: Results for real IT monitoring datasets for $\gamma_{max} = 3$. We report the F1-score.

	MoM 1	MoM 2	Ingestion	Web 1	Web 2	Antivirus 1	Antivirus 2
GCMVL	0.0	0.0	0.14	0.2	0.0	0.08	0.0
Dynotears	0.14	**0.3**	0.14	0.23	0.3	0.18	0.19
PCMCI$^+$	0.0	0.0	0.0	0.23	0.3	0.04	0.11
PCGCE	**0.15**	0.0	0.22	0.22	0.15	0.3	**0.45**
VLiNGAM	0.0	0.0	**0.38**	**0.29**	0.18	0.15	0.22
TiMINo	0.0	0.17	0.18	0.0	0.0	0.0	0.0
NBCB-w	0.0	0.0	0.15	0.23	0.3	0.14	0.24
NBCB-e	0.14	0.0	0.22	0.19	**0.42**	**0.31**	**0.45**
CBNB-w	0.0	0.0	0.16	0.23	0.3	0.17	0.16
CBNB-e	0.0	0.0	0.13	0.22	0.29	**0.31**	0.38

PCMCI$^+$ exhibits poor performance on the MoM, Ingestion, and Antivirus datasets, except for the MoM 1 dataset when γ_{max} is set to 15, where it achieves an F1-score of 0.4. However, PCMCI$^+$ shows better performance on the Web datasets. PCGCE achieves the highest F1-score on the MoM 1 dataset when $\gamma_{max} = 3$ and $\gamma_{max} = 5$ however it F1-score drops to zero for $\gamma_{max} = 10$ and 15. For the MoM 2 dataset, PCGCE has almost always a zero F1-score and for the Ingestion dataset it has relatively a low performance. However, when it comes to the Web and Antivirus datasets, PCGCE consistently exhibits good performance across all values of γ_{max}. VLiNGAM consistently achieves high F1-scores on the Ingestion dataset for all values of γ_{max} except when $\gamma_{max} = 3$, and it performs better on the Web and Antivirus datasets compared to the MoM datasets. TiMINo performs poorly on the majority of the datasets, but it demonstrates stable performance on the Ingestion dataset regardless of the value of γ_{max}. It shows a similar conclusion on the MoM 2 dataset, except when γ_{max} is set to 5. NBCB-w and CBNB-w achieve the best F1-score of 0.4 on the MoM 1 dataset when γ_{max} is set to 15, and it generally performs better on the Web and Antivirus datasets compared to the other datasets. NBCB-e tends to achieve the highest F1-scores in most cases. It achieves the highest F1-scores on the MoM 2 and Ingestion datasets when γ_{max} is large, and on the Web and Antivirus datasets when γ_{max} is smaller, meanwhile, it should be noted that as γ_{max} increases, its performance remains comparative on these datasets. Similarly, CBNB-e has the best F1-score in Antivirus 1 when γ_{max} is set to 10 and 15 and in Antivirus 2 when γ_{max} is set to 10.

In summary, it appears that there is no single method that works well for all datasets. If the value of γ_{max} is unknown, NBCB-e and PCGCE are the recommended choice for the Antivirus datasets, as they consistently performs well across these datasets for all values of γ_{max}. Similarly, NBCB-e, PCGCE and PCMCI$^+$ are the recommended choice for the Web datasets (Dynotears was excluded because as shown in Figures 6 and 7, it gives almost a fully connected graph for the Web datasets). For the Ingestion dataset, VLiNGAM is the best choice. Lastly, Dynotears is a better option for the MoM 1 and MoM 2 datasets due to its stability across different values of γ_{max}.

However, it is important to note that the best performance achieved (0.45 in Antivirus 2) is far from being satisfactory for real world application.

In Figures 3,4,5, 6, 7, 8 and 9 we also give the the inferred graphs that correspond to the results in Table 2 (where γ_{max} is set using the 15 seconds rule for the MoM datasets and using the 15 minutes results for the rest of the datasets). In general, we can say that there is a lot of false positives and that Dynotears tend to give a fully connected graph while constraint-based and hybrid based methods tend to give sparse graphs.

A.2 DATA EXAMINATION

Examination and visualization of time series is useful to observe trends, patterns, and dependencies in the data. By analyzing the data beforehand, we can identify potential behavior change, seasonality, sleeping time series, missing values or other time-dependent effects that may influence the outcomes we are interested in. Abnormal behavior, sleeping time series and misaligned data are a common occurrence in Monitoring data.

A.2.1 MoM datasets

Since MoM dataset was created in a controlled environment, the time series are aligned because sampling is uniformly collected at every second. There are also no missing values in this dataset, and no completely or partially sleeping time series.

A.2.2 Ingestion activity dataset

As mentioned before, all the data are aligned for this dataset with sampling of 1 minute. Moreover, it contained no missing values upon inspection. Figure 12 contains a clear example of behavior change (highlighted in red). This is particularly interesting because the behavior change occurs in all time series approximately in the same region. There are no completely sleeping time series in this dataset, PMDB and RTMB are partially sleeping.

A.2.3 Web activity dataset

Upon examination of the 10 time series, it was observed that the timestamps were not exactly aligned. It is noteworthy to mention that there were no sleeping time series observed in this dataset. However, NPP, NetIn and NetOut are partially sleeping. In terms of sampling, all time series had a sampling of 1 minute. To align all the time series and make them of the same sampling, all the time series were resampled to 5 minutes using either Strategy 1 or Strategy 2. Upon resampling, RamH and CpuG contained missing values, the maximum number of missing values was 1 for both. The missing values were filled using simple linear interpolation of Pandas dataframes. It is important to mention that there were missing values in the raw data, but when sampled to a longer sampling the number of missing values were reduced. Afterwards, they were interpolated.

A.2.4 Antivirus activity dataset

This dataset contained 13 time series in total, and the timestamps were not exactly aligned. Moreover, the raw data contained missing values. There were no completely sleeping time series observed in this dataset, but RP, CUP,RV,CUV and MUP were partially sleeping. In terms of sampling, ChIE, T and ChP had an original sampling of 5 minutes and the rest were 1 minute. To align all the time series and make them of the same sampling, all the time series were resampled to 5 minutes using either Strategy 1 or Strategy 2. Upon resampling, four metrics contained missing values. CUGV had 5 missing values with at most 1 consecutive missing value. CUV, ChP and MUP had 219 missing values. However, there were at most 2 consecutive missing values in these time series, so no large block of missing values was observed. The missing values were filled using simple linear interpolation of Pandas data frames.

Table 7: Summary of different datasets.

|	MoM	Ingestion	Web	Antivirus
Number sleeping time series after pre-processing	0	0	0	0
Number of partially sleeping time series after pre-processing	0	2	3	5
Sampling rate(s) before pre-processing	1 sec	1 min	1 min	1 & 5 mins
Contained missing values before pre-processing	No	No	Yes	Yes
Resampled after pre-processing	1 sec	1 min	5 mins	5 mins
Number of time series with missing values after resampling	0	0	2	4
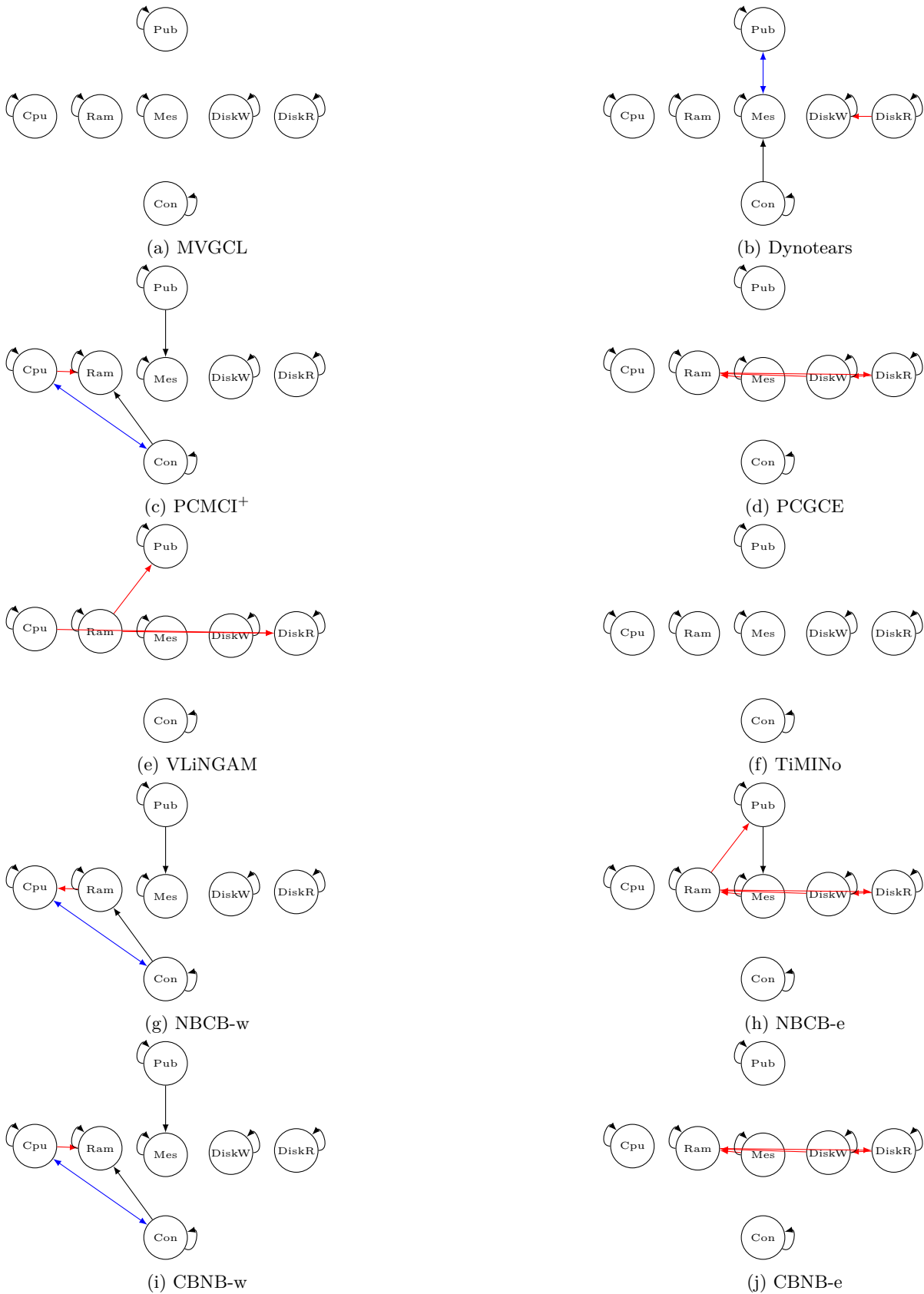
Figure 3: Inferred summary graph from MoM 1 dataset. Red edges correspond to false positive, black edges correspond to true positives and blue edges correspond to a true positive from one side and a false positive from another side.
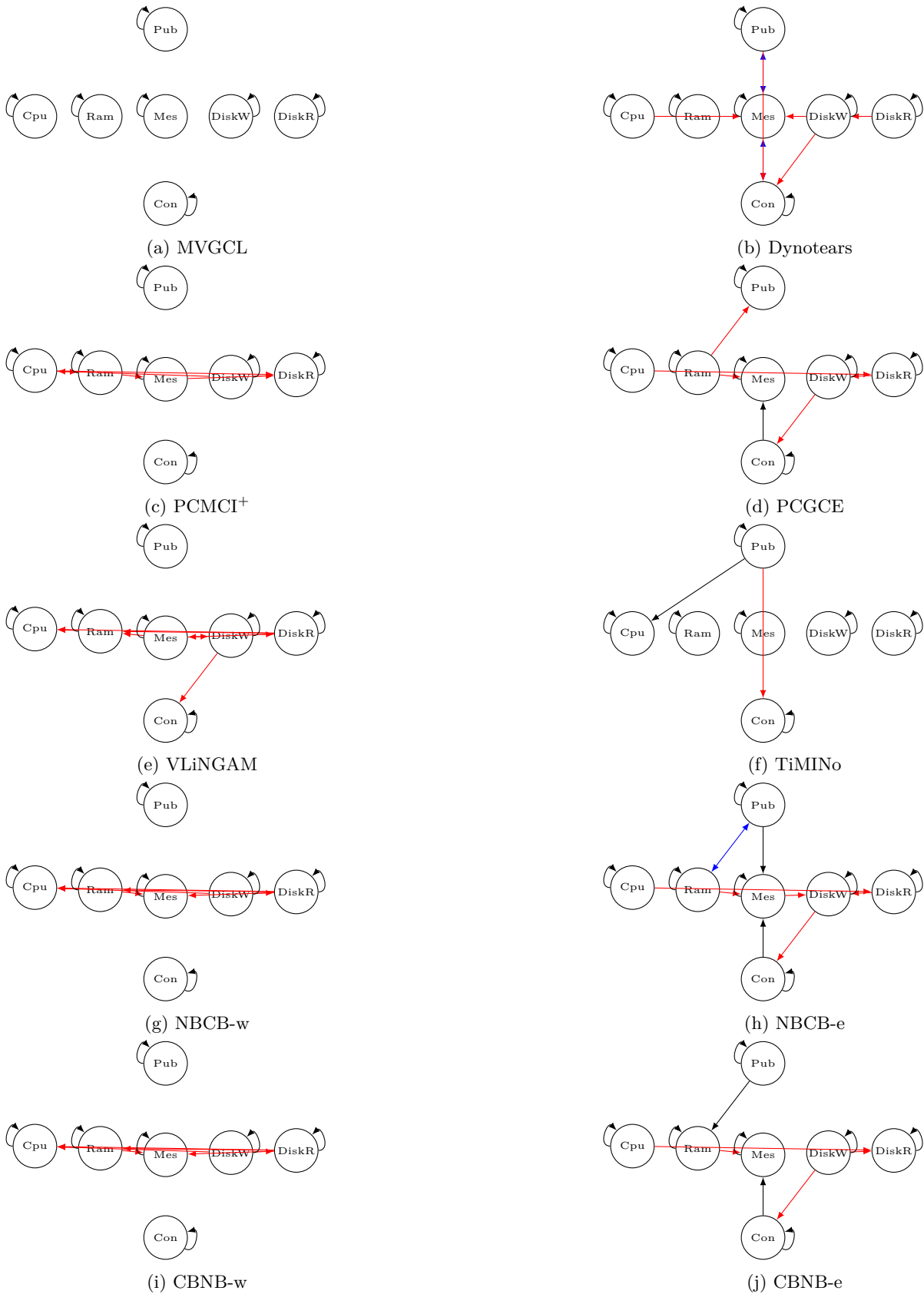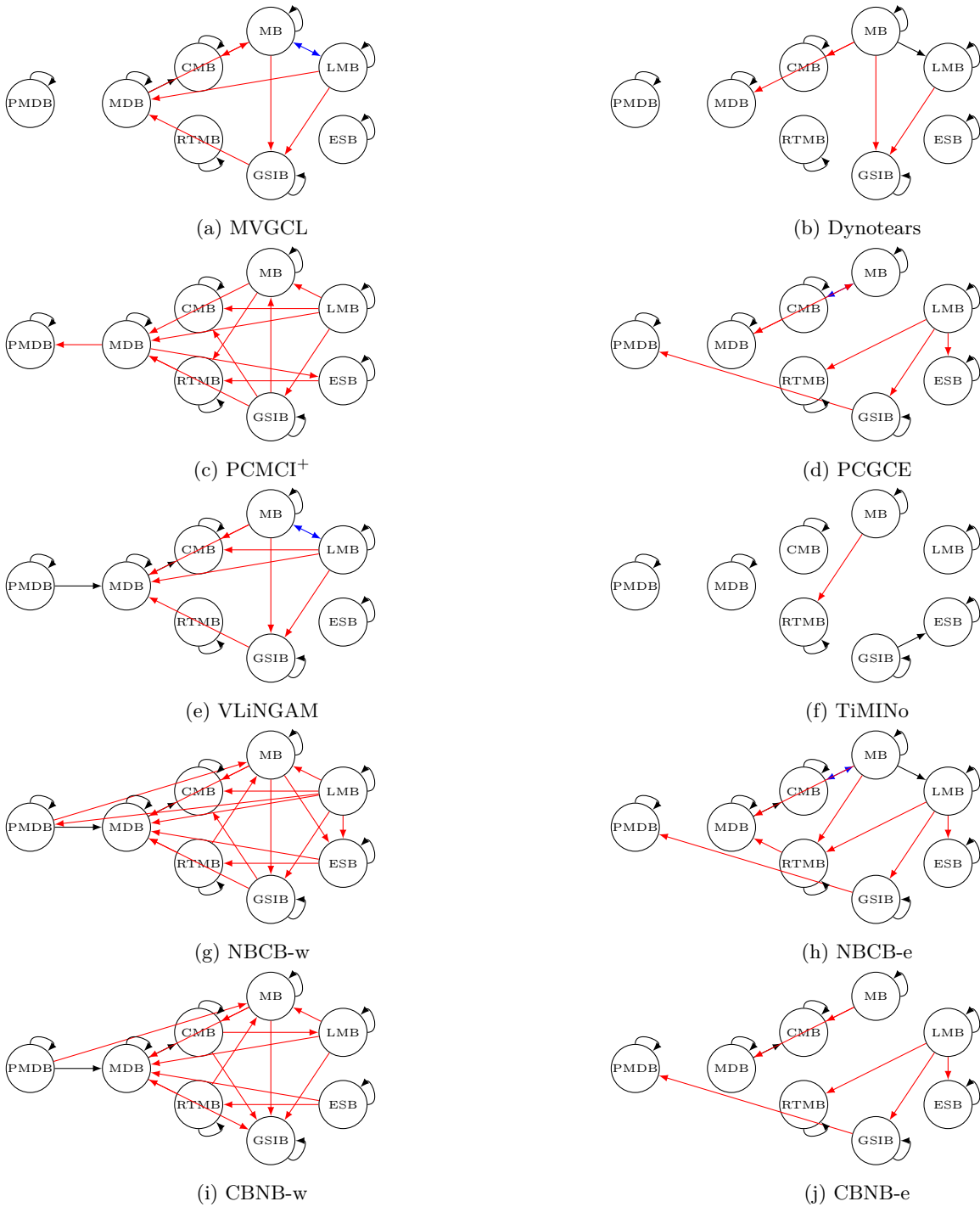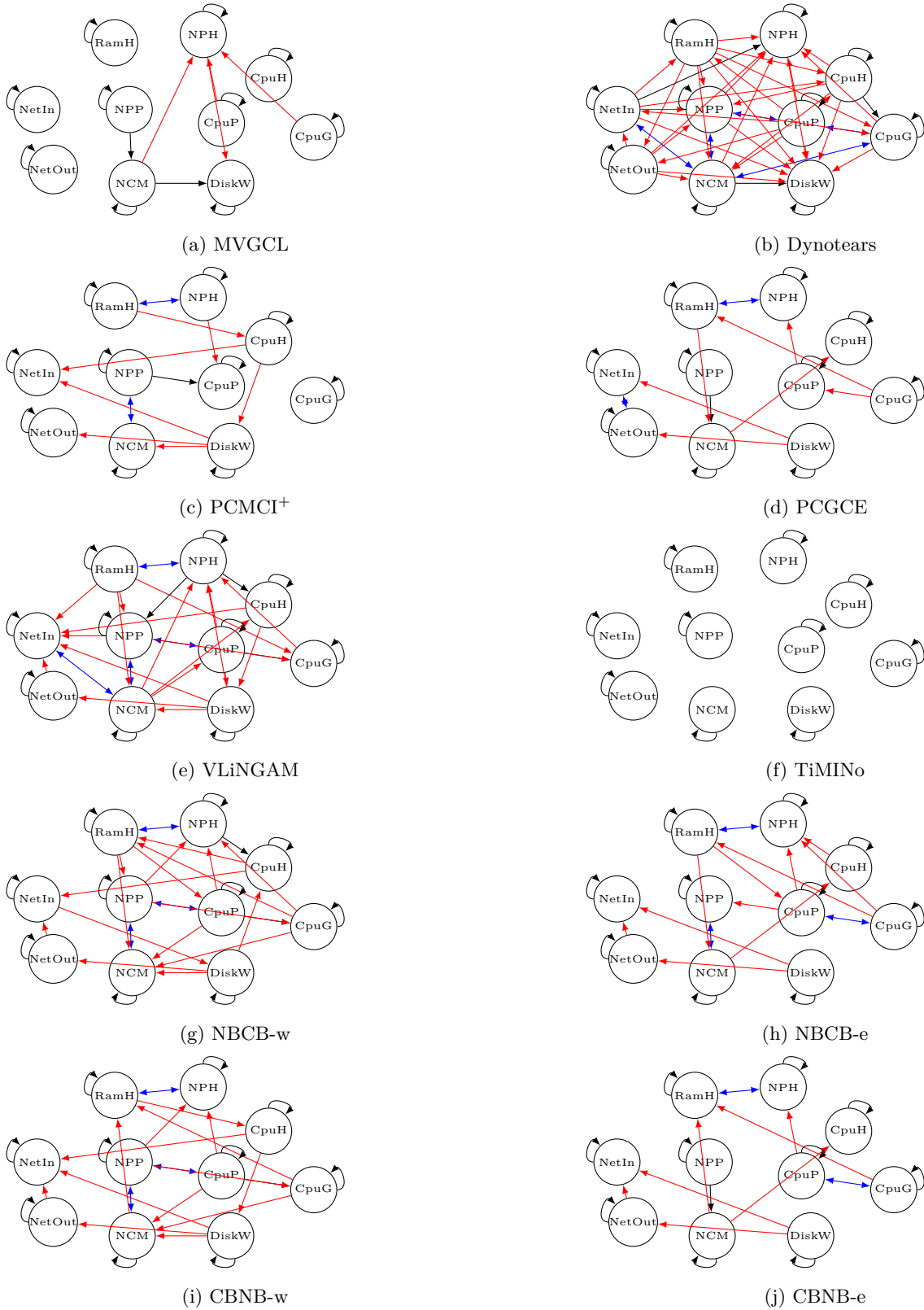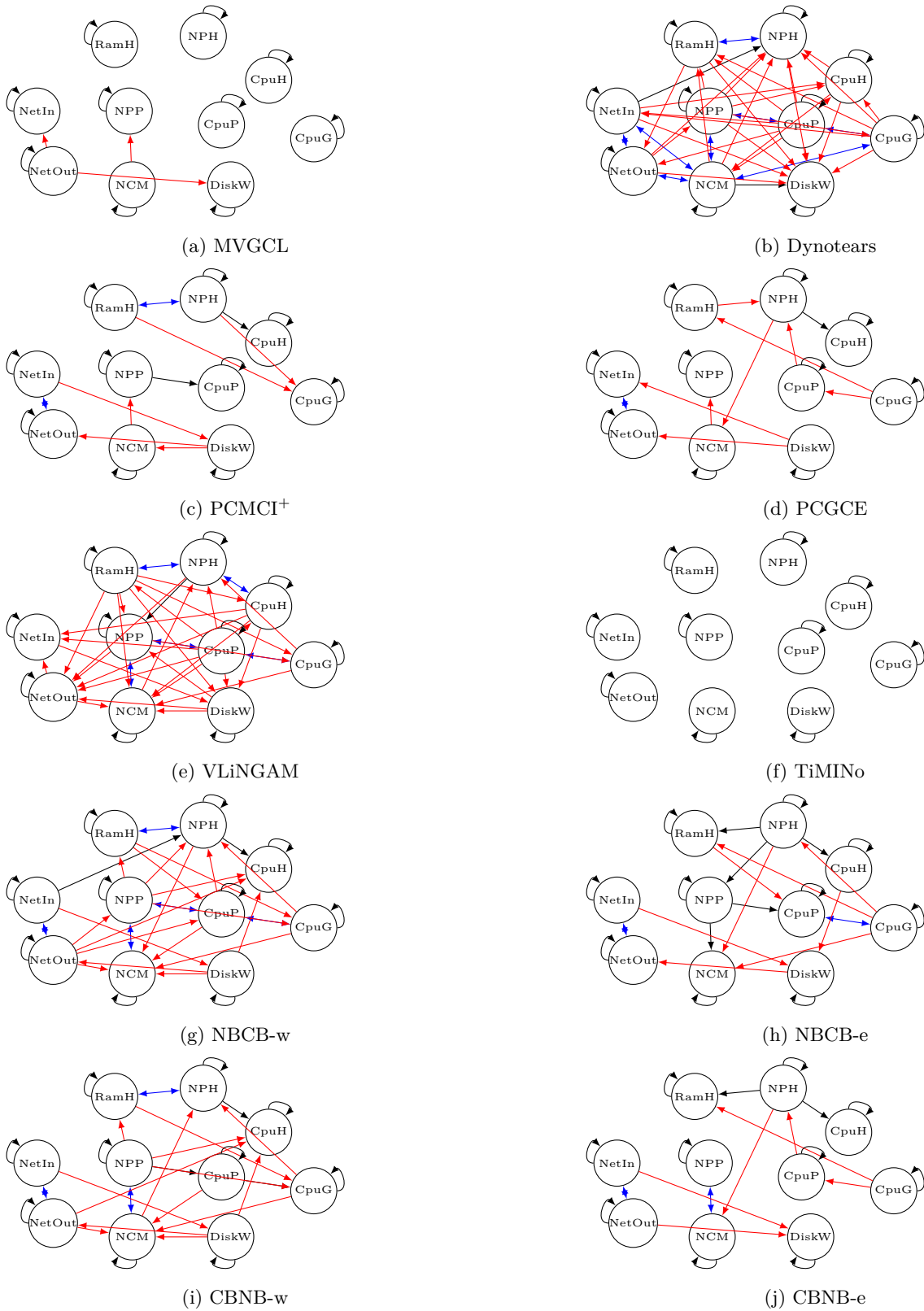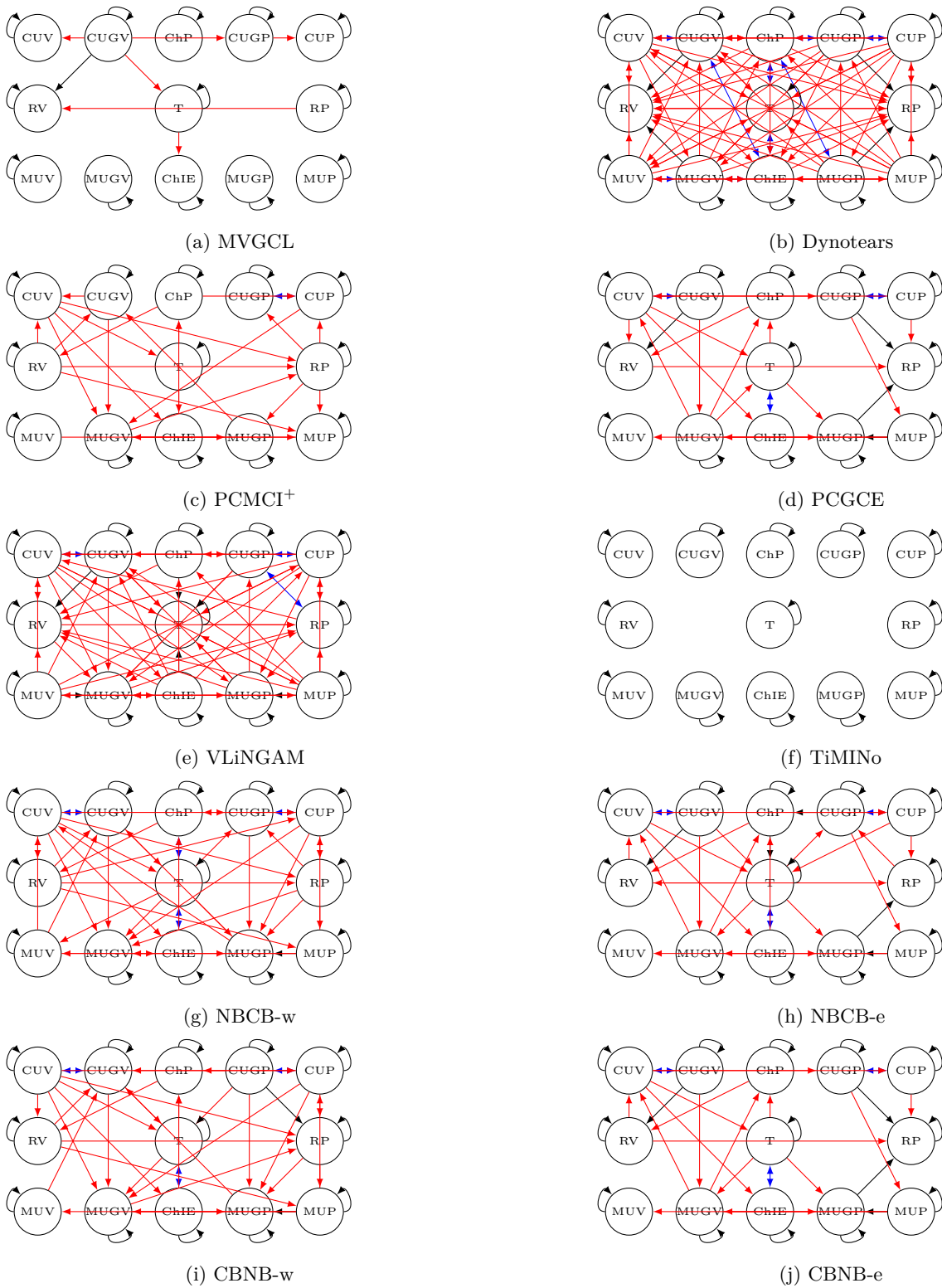
Figure 4: Inferred summary graph from MoM 2 dataset. Red edges correspond to false positive, black edges correspond to true positives and blue edges correspond to a true positive from one side and a false positive from another side.
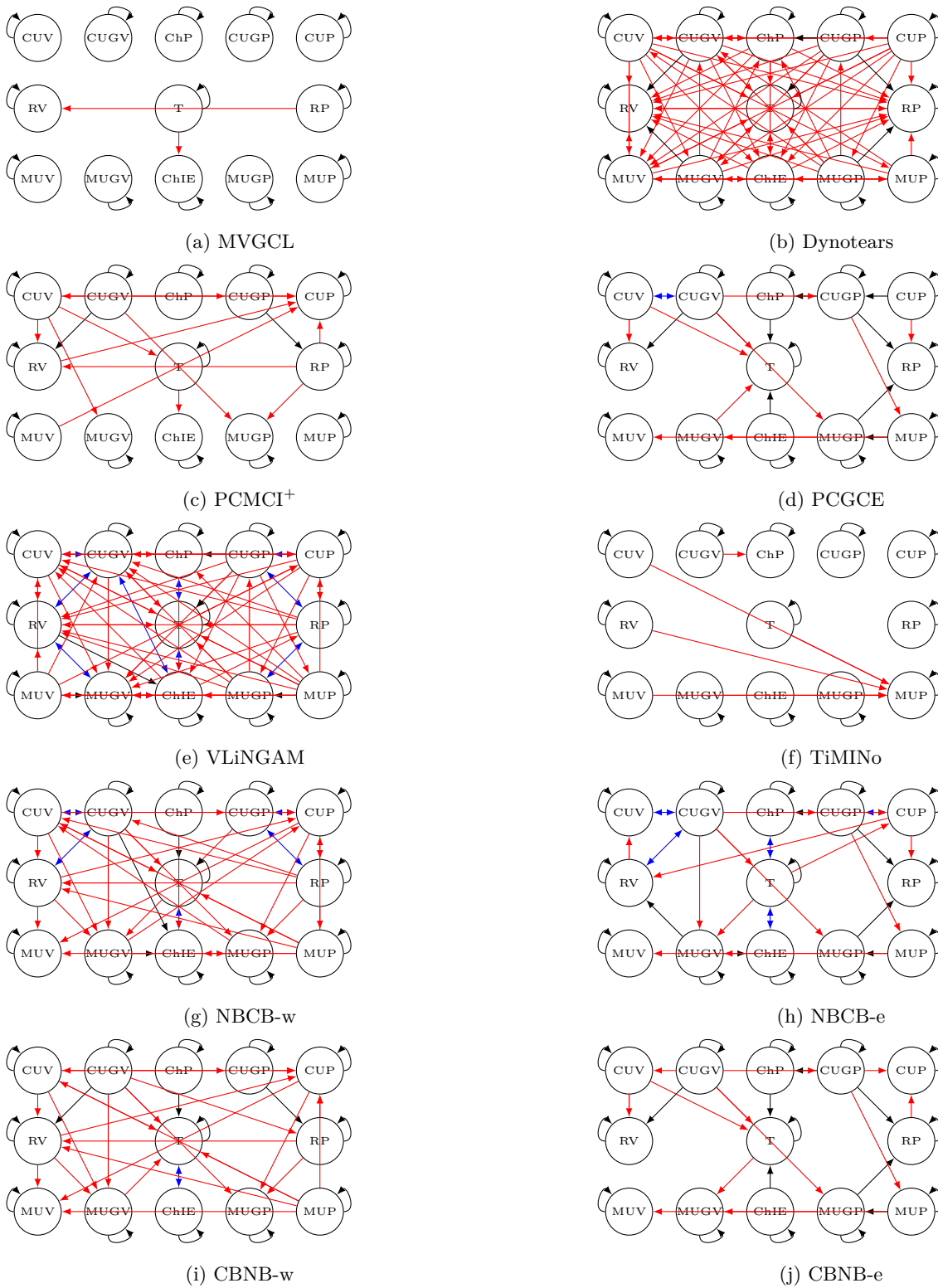
(a) MVGCL

(b) Dynotears

(c) PCMCI$^+$

(d) PCGCE

(e) VLiNGAM

(f) TiMINo

(g) NBCB-w

(h) NBCB-e

(i) CBNB-w

(j) CBNB-e

Figure 5: Inferred summary graph from Ingestion dataset. Red edges correspond to false positive, black edges correspond to true positives and blue edges correspond to a true positive from one side and a false positive from another side.

(a) MVGCL

(b) Dynotears

(c) PCMCI$^+$

(d) PCGCE

(e) VLiNGAM

(f) TiMINo

(g) NBCB-w

(h) NBCB-e

(i) CBNB-w

(j) CBNB-e

Figure 6: Inferred summary graph from Web 1 dataset. Red edges correspond to false positive, black edges correspond to true positives and blue edges correspond to a true positive from one side and a false positive from another side.

(a) MVGCL

(b) Dynotears

(c) PCMCI$^+$

(d) PCGCE

(e) VLiNGAM

(f) TiMINo

(g) NBCB-w

(h) NBCB-e

(i) CBNB-w

(j) CBNB-e

Figure 7: Inferred summary graph from Web 2 dataset. Red edges correspond to false positive, black edges correspond to true positives and blue edges correspond to a true positive from one side and a false positive from another side.

Figure 8: Inferred summary graph from Antivirus 1 dataset. Red edges correspond to false positive, black edges correspond to true positives and blue edges correspond to a true positive from one side and a false positive from another side.

(a) MVGCL

(b) Dynotears

(c) PCMCI$^+$

(d) PCGCE

(e) VLiNGAM

(f) TiMINo

(g) NBCB-w

(h) NBCB-e

(i) CBNB-w

(j) CBNB-e

Figure 9: Inferred summary graph from Antivirus 2 dataset. Red edges correspond to false positive, black edges correspond to true positives and blue edges correspond to a true positive from one side and a false positive from another side.
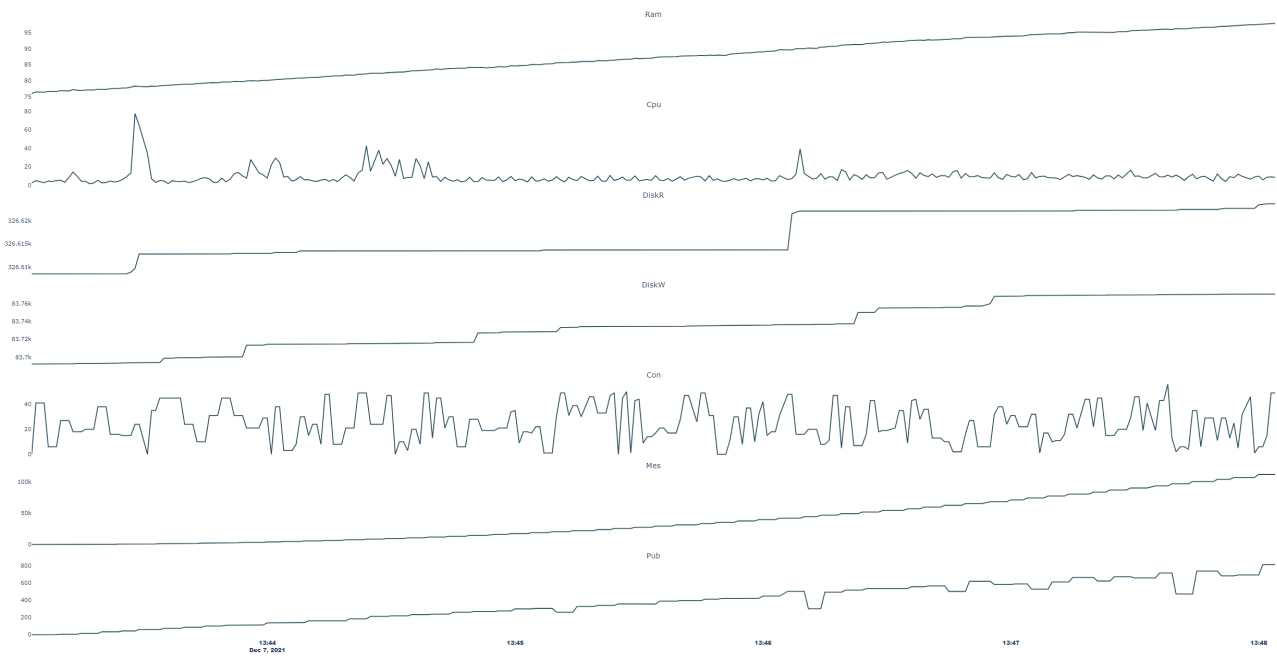
Figure 10: Overview of MoM 1 dataset.
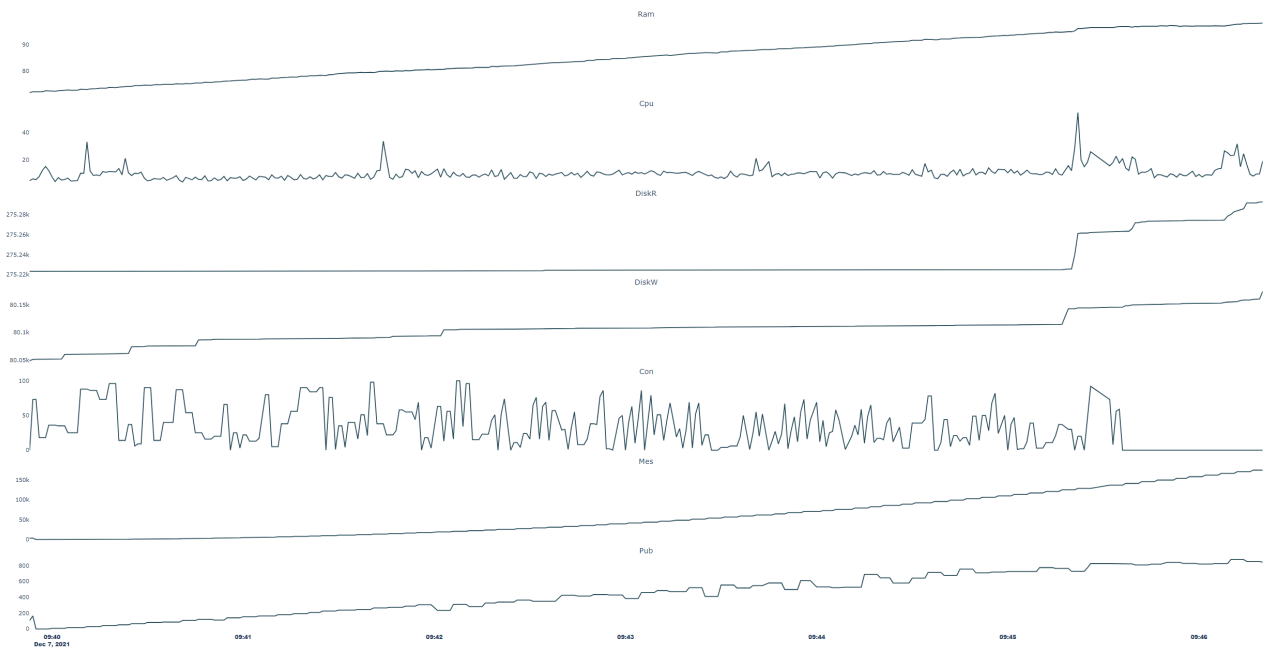
Figure 11: Overview of MoM 2 dataset.
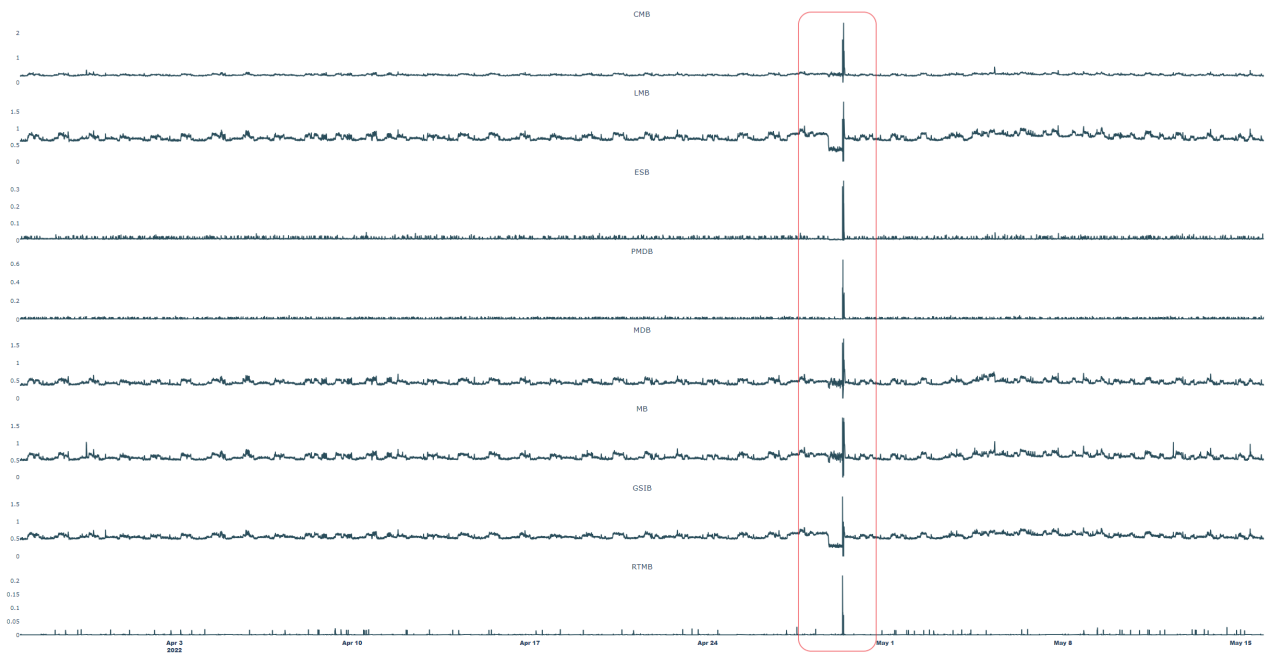
Figure 12: Overview of Ingestion data, behavior change regions approximately highlighted inside the red box.
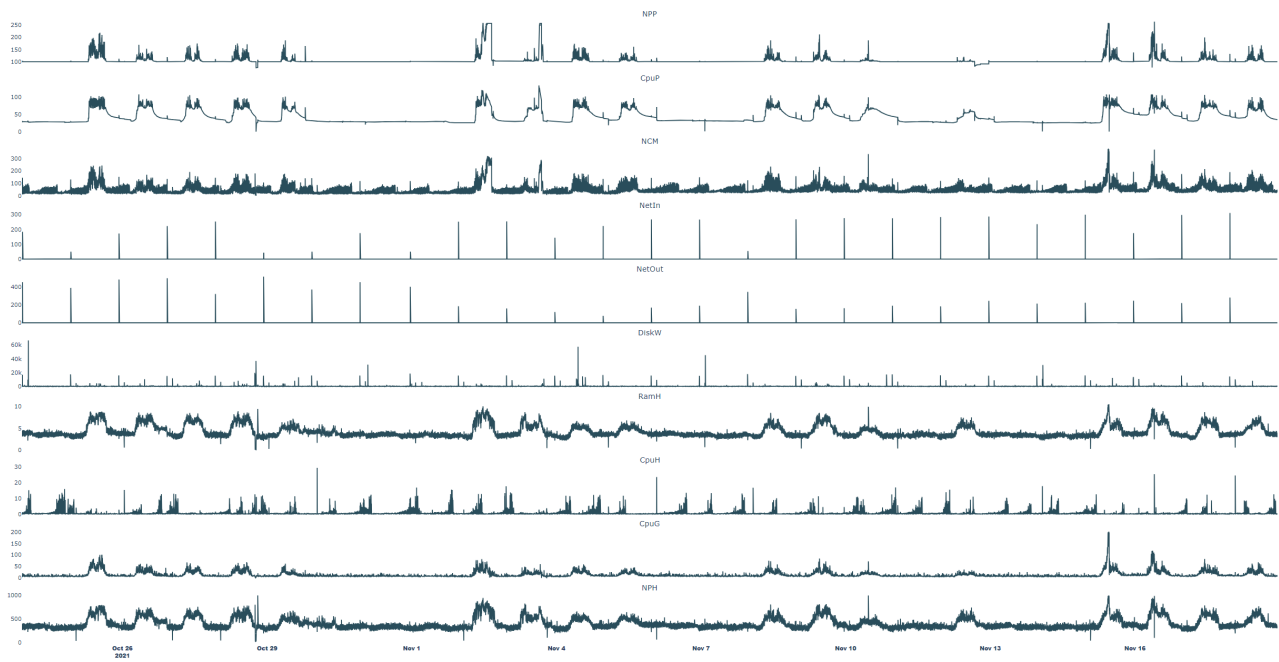
Figure 13: Overview of raw Web data.
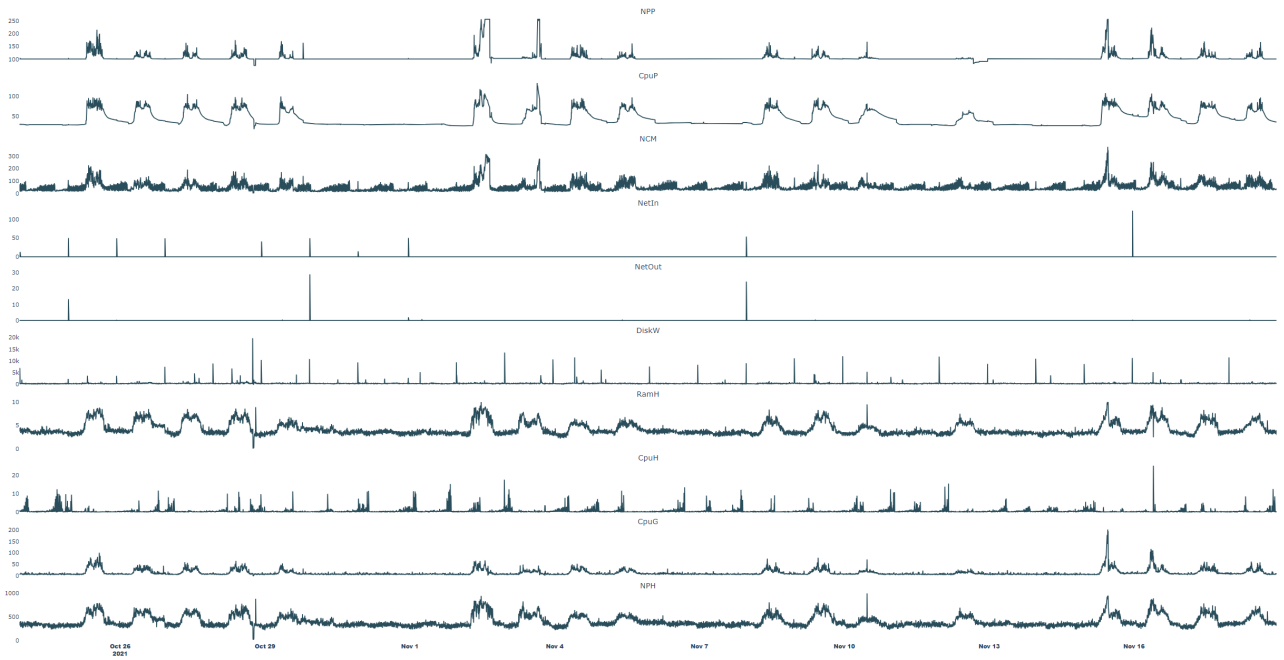
Figure 14: Overview of Web data after pre-processing 1.
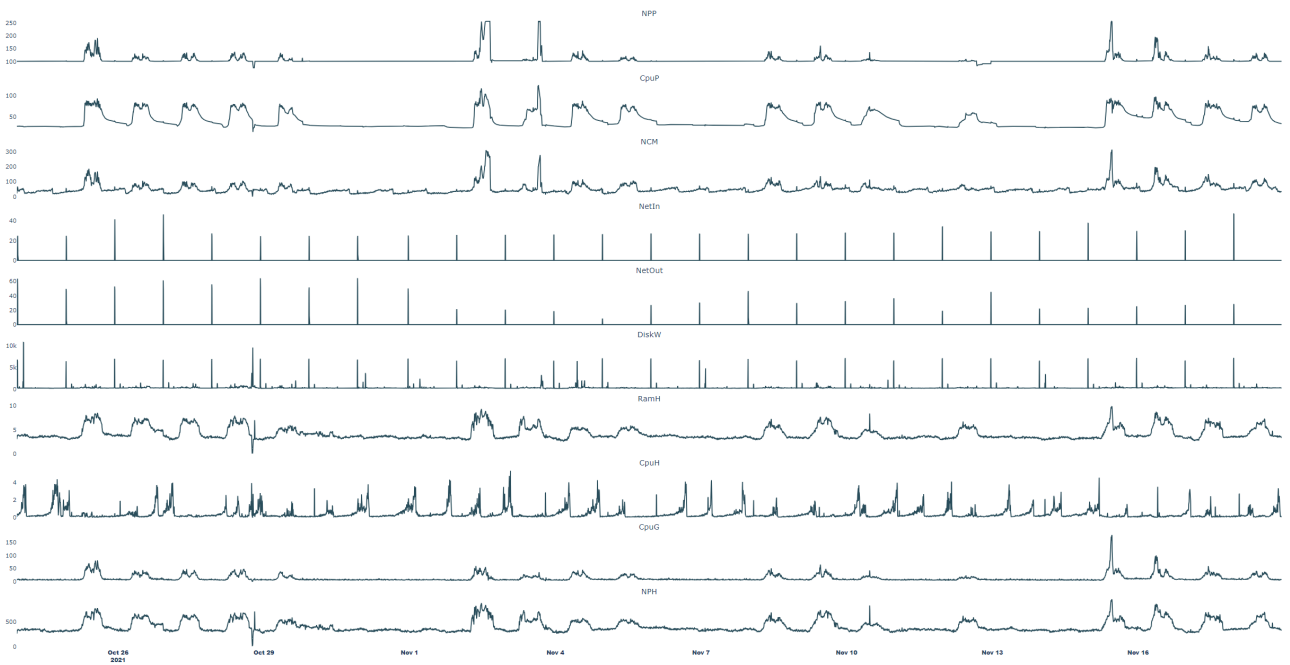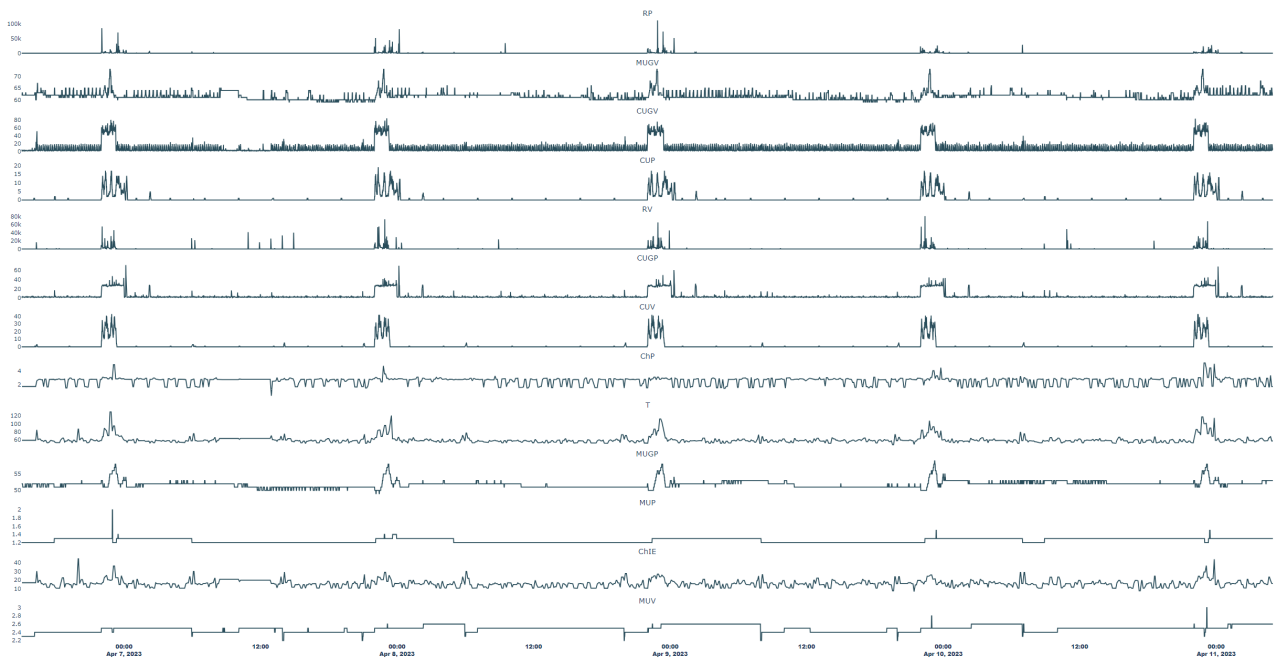

Figure 15: Overview of Web data after pre-processing 2.

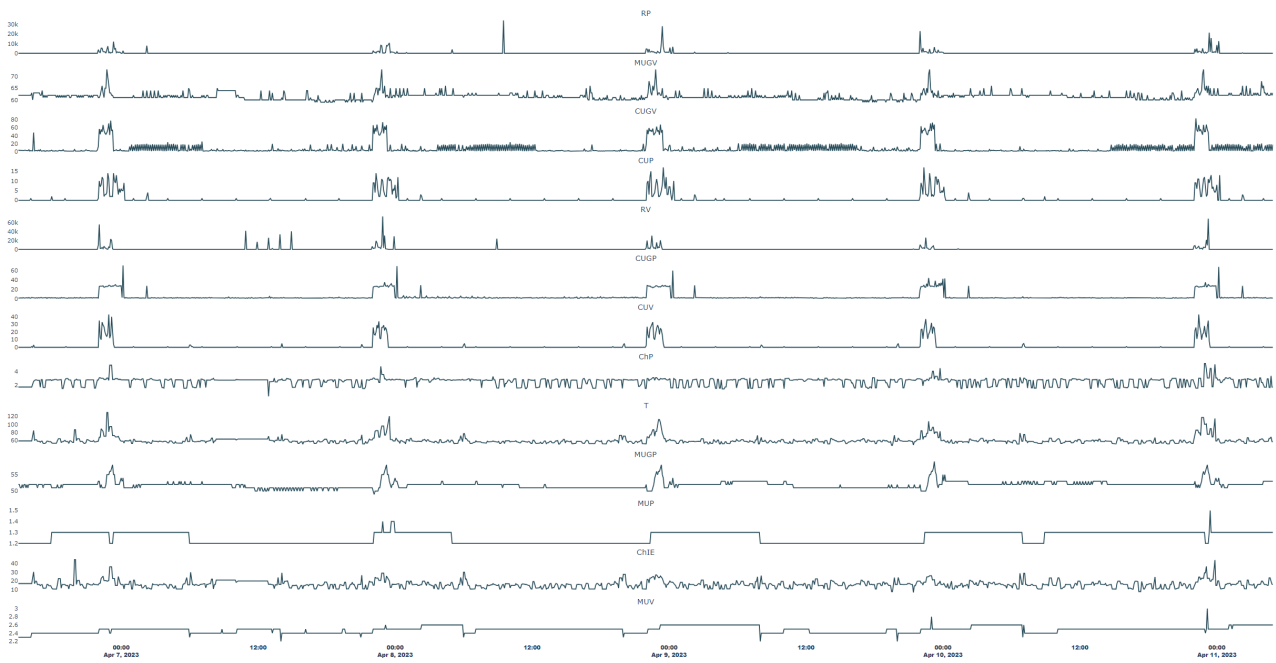Figure 16: Overview of raw Antivirus data.

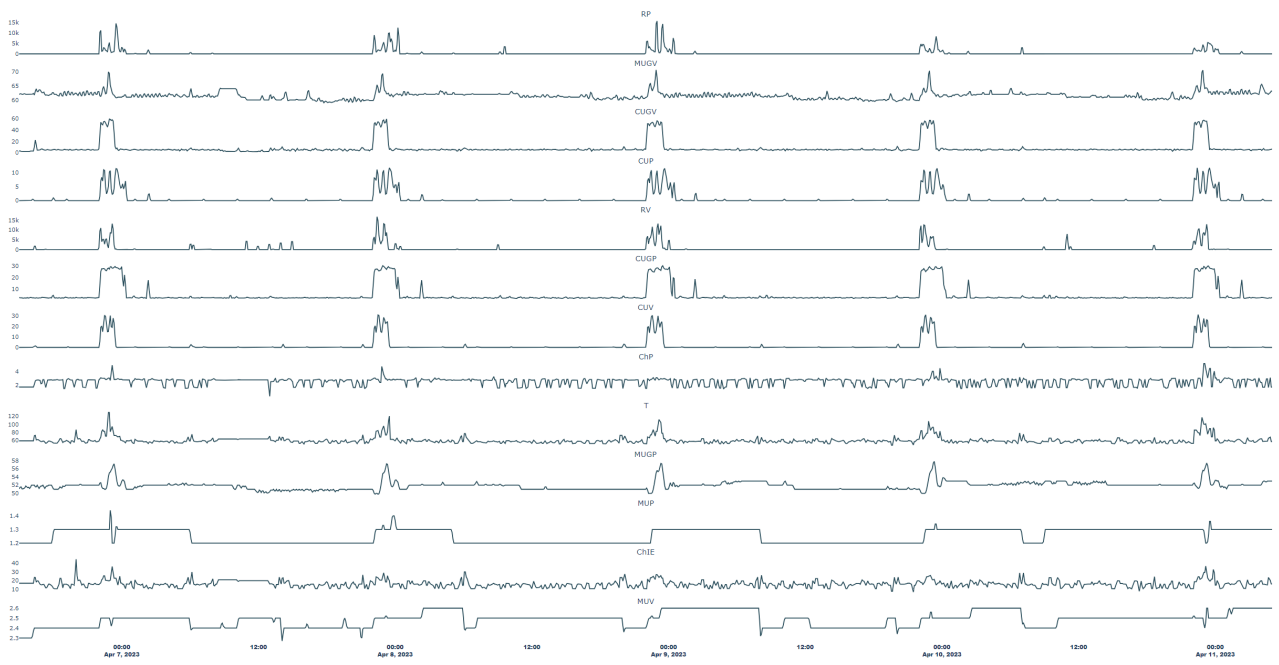Figure 17: Overview of Antivirus data after pre-processing 1.

Figure 18: Overview of Antivirus data after pre-processing 2.