




# Mining Temporal Rules from Heterogeneous Multivariate Time Series

Eliane G. Karasawa   [ Instituto de Ciências Matemáticas e de Computação (USP) | *elig-niechk@gmail.com* ]

Elaine P. M. Sousa  [ Instituto de Ciências Matemáticas e de Computação (USP) | *parros@icmc.usp.br* ]

 Instituto de Ciências Matemáticas e de Computação (ICMC), Universidade de São Paulo (USP), São Carlos, SP, Brazil.

Received: 10 March 2023 • Published: 22 December 2023

**Abstract** This paper presents TRUMiner (Temporal RULes Miner), an algorithm to mine temporal rules from multivariate time series considering pairs of variables. It provides extended multivariate temporal rules that point the occurrence of the mined patterns in the original time series. Furthermore, TRUMiner can be used with any discretization method and deals with missing data and heterogeneous time series datasets, including different number of variables per time series and distinct number of observations per variable. We evaluated the algorithm on international trade multivariate data from several sources. Results show the relevance of extended rules and the algorithm applicability to heterogeneous time series, simplifying data integration and pre-processing.

**Keywords:** Multivariate Temporal Rules, Time Series, Data Pre-processing, Data Mining

## 1 Introduction

Data mining is a relevant field, with tens of thousands of related papers published last year in IEEE only. Its relevance is given by the capacity to discover knowledge from several sources of massive data. In this context, association rules mining presents great interest due to its simplicity, high explanation potential, and prediction abilities. It provides valuable knowledge as rules represent causality relationships between antecedent and consequent [Agrawal *et al.*, 1993]. Seeking additional information related to the uncovered knowledge, new studies incorporate a temporal feature to the rules, allowing an understanding of events' order and time of occurrence [Segura-Delgado *et al.*, 2020].

Temporal rules mining is commonly applied to time series, a type of data inherent in various domains such as economy, health, biology, and geography. Therefore, mining temporal rules is an efficient means to obtain useful information from these massive data sources [Han *et al.*, 2011]. An example of a temporal rule on the economic scenario is “one year after a rise in import, the country's GDP also rises with 69% of confidence”.

Real multivariate time series datasets may come from distinct sources and have missing observations and missing variables. Therefore, well-known approaches to mine temporal rules usually require data integration and pre-processing [Romani *et al.*, 2010; Zhao and Zhang, 2017]. Moreover, most of them focus on univariate time series [Das *et al.*, 1998; Harms and Deogun, 2004; Schlüter and Conrad, 2011] with some optimizations to reduce time complexity [Shokoohi-Yekta *et al.*, 2015; Xue *et al.*, 2016; Buaton *et al.*, 2021]. We thus propose TRUMiner (Temporal RULes Miner), previously introduced in Karasawa and Sousa [2022], to cope with these issues. In this paper, we include background and problem formaliza-

tion, explore more details about the algorithm and present additional experimental analysis.

TRUMiner mines multivariate time series from several data sources. It can handle heterogeneous time series with missing observations and also missing variables. The algorithm can run with different discretization methods, allowing the user to choose an adequate discretization for the analysis purposes. TRUMiner returns multivariate temporal rules, considering pairs of variables, in short and extended format, with antecedent, consequent, and temporal feature in both cases. In the extended one, it adds all occurrences of the rule and the corresponding time intervals in every time series.

This paper is organized as follows: **Section 2** summarizes background concepts of multivariate temporal rule mining and presents the main symbols we use throughout this article (**Table 1**). **Section 3** integrates related work, while the TRUMiner algorithm is detailed in **Section 4** with the presentation of each process step. **Section 5** describes datasets, results, and analysis of TRUMiner execution over economic trade data. Finally, in **Section 6**, we present our conclusion and directions for future work.

## 2 Background

Association rules mining seeks to explain and predict data behavior. It is an implication  $A \Rightarrow C$  where  $A$  is the rule antecedent, and  $C$  is its consequent.  $A$  and  $C$  are itemsets from transactions in a database, and both belong to a set of items  $I$  such that  $A \cap C = \emptyset$  and  $A, C \neq \emptyset$ . As a particular case of association rules, temporal rules are defined as a pair  $(A \Rightarrow C, \Delta t)$ , where  $\Delta t$  is a characteristic of the rule named the temporal feature of  $A \Rightarrow C$  [Chen and Petrounias, 2000]. In our approach, the temporal feature describes the exact time

difference between  $A$  and  $C$ .

Temporal rules mining can be applied to time series, a set of observations sorted in time [Box *et al.*, 2015], often real measurements with variation in regular time intervals [Mitsa, 2010]. A discrete multivariate time series  $s$  in a dataset  $S$  is

$$s = obs_1, \dots, obs_n$$

with  $obs_i$  being the  $i$ -th  $\delta$ -dimensional observation,  $i \in [1, \dots, n]$ . We define  $\delta$  as the number of variables comprising the multivariate time series in  $S$ .

Data pre-processing is commonly applied in time series to obtain understandable association patterns more efficiently [Han *et al.*, 2011], with data discretization as a typical method. Discretization over a univariate time series  $s[var_X] = obs_1^X, \dots, obs_n^X$  of variable  $X$  generates a discretized time series  $s'[var_X] = \alpha_{t_1, t_f}^X, \dots, \alpha_{t_i, t_n}^X$  with  $L$  quantized symbols.

Each symbol  $\alpha$  has its beginning and ending time represented here as a subscript pair  $t_i, t_f$ , respectively. The quantized symbol can represent from a fraction of an observation up to multiple observations that underwent discretization. In multivariate time series, the discretization process is applied to each variable individually.

The quantized symbols of distinct variables from one time series are combined to obtain the transactions in the multivariate scenario. Therefore, a transaction is

$$([var_X, \alpha_i^X], \dots, [var_Y, \alpha_j^Y], \dots), \Delta t,$$

with each element containing a variable (e.g.  $var_X$  standing for variable  $X$ ) and its respective quantized symbol without  $t_i$  and  $t_f$  (e.g.  $\alpha_i^X$  where  $i$  represents the symbol position on the discretized time series). The transaction also has the temporal feature  $\Delta t$  that indicates the exact time interval between the element with the first beginning time and the one with the last beginning time. The temporal feature can be delimited by a threshold  $w$  that indicates its maximum time span.

From the transactions, a multivariate temporal rule is

$$([var_X, \alpha_i^X], \dots) \Rightarrow ([var_Y, \alpha_j^Y], \dots), \Delta t$$

where  $([var_X, \alpha_i^X], \dots)$  is the rule's antecedent and the consequent is  $([var_Y, \alpha_j^Y], \dots)$ . Antecedent and consequent are from a single transaction and can add up to  $\delta$  variables. The beginning time of the consequent symbols cannot be before the beginning time of the antecedent. The temporal feature  $\Delta t$  of the rule comes from the transaction temporal feature.

Quality measures from association rules mining, such as support and confidence, can be extended to evaluate multivariate temporal rules by integrating the temporal feature and the multivariate characteristic. Support is the frequency of a rule in the whole transactions' set. **Equation 1** presents multivariate temporal support used in this work, based on Romani *et al.* [2010]. The dividend is the frequency of the rule  $([var_X, \alpha_i^X], \dots \Rightarrow [var_Y, \alpha_j^Y], \dots, \Delta t)$  and  $T$  is the number of transactions obtained from the dataset.

$$sup = \frac{freq([var_X, \alpha_i^X], \dots \Rightarrow [var_Y, \alpha_j^Y], \dots, \Delta t)}{T} \quad (1)$$

The rule's precision is measured by confidence, given by the frequency of the rule over the frequency of all transactions that generate rules with the same antecedent and temporal feature. Based on Romani *et al.* [2010], the **Equation 2** measures the confidence in multivariate temporal rules.

$$conf = \frac{freq([var_X, \alpha_i^X], \dots \Rightarrow [var_Y, \alpha_j^Y], \dots, \Delta t)}{freq([var_X, \alpha_i^X], \dots, \Delta t)} \quad (2)$$

Although evaluation measures for multivariate temporal rules are extensions of association rules measures, values of support are inherently lower due to the transaction generation method in the multivariate approach.

**Table 1.** Table of symbols.

Symbol	Description
$A$	Rule antecedent
$C$	Rule consequent
$\Delta t$	Temporal feature
$A \Rightarrow C, \Delta t$	Temporal rule
$S$	Dataset with $N$ multivariate time series
$N$	Number of multivariate time series in $S$
$s$	Discrete multivariate time series
$\delta$	Number of variables in the dataset $S$
$obs_i^X$	$i$ -th observation of variable $X$ from time series $s$
$s[var_X]$	Discrete univariate time series of variable $X$
$s'[var_X]$	Discretized univariate time series from $s[var_X]$
$\alpha_{t_i, t_f}^X$	Quantized symbol from variable $X$ with beginning time $t_i$ and ending time $t_f$
$L$	Number of quantized symbols in $s'[var_X]$
$w$	Maximum time span of rules
$sup$	Support of a temporal rule
$conf$	Confidence of a temporal rule

### 3 Related Work

There have been several research work on temporal rules mining, but there is no consensus on standard terminology or temporal feature usage in the mining process. Here, we list some of the main contributions involving time series.

In Das *et al.* [1998], the authors propose the discretization of time series by forming subsequences using sliding windows and grouping. The temporal feature of the rule is obtained from the number of existing elements between antecedent and consequent. The MOWCATL algorithm [Harms and Deogun, 2004] performs rule mining from predetermined elements of interest and a fixed or maximum time window between antecedent and consequent.

Clearminer [Romani *et al.*, 2010] finds association rules in multivariate series, with antecedent and consequent coming from different variables. The algorithm allows delimiting the maximum time window to generate rules and returns detailed rules, containing a sample of the initial, intermediate, and

final observation of the antecedent and consequent and their initial and final times.

Schlüter and Conrad [2011] introduced a prototype to mine temporal rules from univariate time series. Three discretization methods are evaluated: SAX [Lin *et al.*, 2003] and two variations of a clustering-based method.

The algorithm proposed by Zhao and Zhang [2017] mines temporal rules from multivariate series with min-max normalization and groups the obtained patterns in clusters to reduce generated patterns. TRiER [Amaral and Sousa, 2019] extracts temporal exception rules from multivariate series aiming for the maximum number of variables for each item. In de Oliveira *et al.* [2017], the focus is association rule mining in graphs.

Our temporal rule mining solution, the TRUMiner, is based on the Clearminer approach [Romani *et al.*, 2010], but we added multiple built-in discretization methods and allow easy integration of other ones, as desired in Schlüter and Conrad [2011], to enable various types of analysis for one dataset. The extended rules returned are also complete, indicating every exact rule occurrence in each original time series. As with Zhao and Zhang [2017], the TRUMiner discovers multivariate temporal rules but with no need to group patterns in the process. Finally, TRUMiner can handle heterogeneous datasets, with missing observations and missing variables.

## 4 TRUMiner

The TRUMiner (Temporal Rules Miner) aims to discover multivariate temporal rules from multivariate time series. Due to the high growth in time and memory complexity for each variable added to the rule, in this work we consider pairs of variables. Given a dataset

$$S = s_1, \dots, s_N$$

with  $N$  multivariate time series ( $s$  as defined in **Section 2**), we describe a transaction from a discretized time series  $s'$  as a pair of time ordered elements ( $[var_X, \alpha_i^X], [var_Y, \alpha_j^Y]$ ), with temporal feature  $\Delta t$ . Each element is composed of a variable and its quantized symbol, so each variable appears only in one transaction element. From the generated transactions, a multivariate temporal rule is

$$[var_X, \alpha_i^X] \Rightarrow [var_Y, \alpha_j^Y], \Delta t$$

where the antecedent of the rule ( $[var_X, \alpha_i^X]$ ) is composed of quantized symbol  $\alpha_i^X$  in the variable  $X$  ( $var_X$ ), and the consequent ( $[var_Y, \alpha_j^Y]$ ) consists of symbol  $\alpha_j^Y$  in  $var_Y$ . Antecedent and consequent have each a beginning time  $t_i$  and an ending time  $t_f$ , and the temporal feature  $\Delta t$  is the difference between the  $t_i$  of the consequent and the antecedent.

The temporal feature fully integrates the multivariate temporal rule and is therefore considered for rule evaluation. Support and confidence are straightforwardly computed for a pair of elements as presented in **Equations 1 and 2**, respectively.

Typically, two elements combined in a rule have maximum support of 100%. However, TRUMiner can generate up to  $2 * (w + 1)$  transactions for each specific element from

a pair of variables, where  $w$  is the maximum temporal feature threshold to obtain rules. **Figure 1** exemplifies transactions with an element sample  $[var_X, \chi]$  where  $\chi$  is a specific quantized symbol. As each transaction comprises only a pair of elements, it can be either the first or the second element in the pair. In each case,  $[var_X, \chi]$  can be involved in up to  $(w + 1)$  transactions with distinct temporal feature, totalizing  $2 * (w + 1)$  transactions for this specific element. Thus, multivariate temporal rules from pair of variables often present low support compared to association rules.

$$\begin{array}{ll} ([var_X, \chi], [var_Y, ?], \Delta t = 0) & ([var_Y, ?], [var_X, \chi], \Delta t = 0) \\ ([var_X, \chi], [var_Y, ?], \Delta t = 1) & ([var_Y, ?], [var_X, \chi], \Delta t = 1) \\ \dots & \dots \\ ([var_X, \chi], [var_Y, ?], \Delta t = w) & ([var_Y, ?], [var_X, \chi], \Delta t = w) \end{array}$$

**Figure 1.** Transactions for a specific element -  $[var_X, \chi]$ .

The rules mined represent relationships between pairs of variables from multivariate time series. The antecedent and consequent are from distinct variables, and the temporal feature is the elapsed time between the antecedent's beginning and the consequent's beginning. **Figure 2** shows an overview of TRUMiner and **Algorithm 1** summarizes its main operations. The process can be split into three major steps: discretization, transactions generation, and rules evaluation.

The input dataset is composed of distinct sets of univariate time series, one set for each variable (e.g. a csv file), possibly coming from different sources. The only requirement is to identify each time series consistently, regardless of the order of the collected data. For instance, we always use "bra" to refer to Brazil variables. Thus, TRUMiner eases the integration of variables from multiple data sources into multivariate time series. Moreover, as detailed in **Section 4.1**, our algorithm can deal with missing variables and observations, and variable-length time series, thus simplifying the necessary pre-processing.

### 4.1 Discretization

**Figure 2 (a)** illustrates the discretization process that transforms the observations of time series  $s_i$  variable  $X$  (denoted by  $s_i[var_X]$ ) into a series of quantized symbols ( $s'_i[var_X]$ ) where  $s_i$  is a multivariate time series from a dataset  $S$  with  $i \in [1, \dots, N]$  and  $N$  being the number of series. The quantized symbols' type and corresponding time span depend on the discretization method.

TRUMiner deals with missing observations and variable length series by storing  $t_i$  and  $t_f$  (respectively, the beginning and ending time of the symbol), allowing quantized symbols with distinct time span. When the series has missing observations, the algorithm is able to generate transactions with existing quantized symbols since it has only to verify if the beginning and ending time of each symbol to compose the transaction satisfy the criteria of maximum time span and minimum beginning time. Thus, the beginning and ending time of the symbols are used to generate coherent rules, allowing to use heterogeneous datasets.

The TRUMiner approach allows any discretization method to be used (**Algorithm 1 lines 1 - 3**), including ones

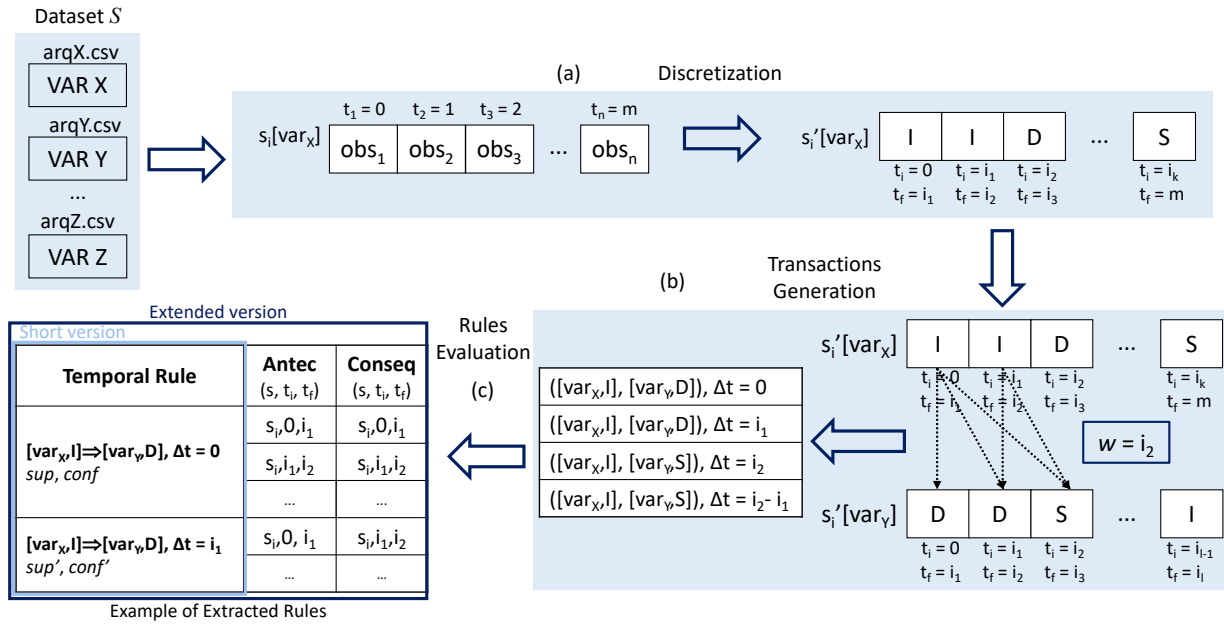


Figure 2. TRUMiner overview and running example.

with normalization. In this work, we implemented and evaluated decis, quartis, SAX, and a *variation-based* discretization. Decis and quartis allow a greater detailing of the observations behavior, informing the increase or decrease percentage from the previous observation. The SAX method [Lin *et al.*, 2003] is a well-known approach applied in several domains, including economy.

The *variation-based* discretization generates symbols from the variation between two or more successive observations. The symbols can be increase ( $I$ ), decrease ( $D$ ), and stability ( $S$ ), defined according to user parameters: the maximum number of observations to generate exactly one symbol, the maximum difference between two observations to indicate stability symbol, the minimum number of observations to be discretized in a stability symbol, and the minimum variation between two symbols of increase or decrease to maintain both symbols. The last two thresholds select only relevant behavior in the time series.

Figure 2 (a) shows the *variation-based* discretization, where a symbol (e.g.  $I$ ) can represent up to a set of successive observations (e.g.  $obs_1, obs_2$ ) with beginning time  $t_i$  (e.g.  $t_i = 0$ ) and ending  $t_f$  (e.g.  $t_f = i_1$  with  $i_1$  an arbitrary time). The beginning time of the following symbol (e.g. another  $I$ ) coincides with the ending time of the previous symbol (e.g.  $t_i = i_1$ ), but it is not an obligation and does not necessarily have the same time span as the previous symbol. So the values  $m, i_k, i_{l-1}, i_l$  are arbitrary times, and they may or may not be coincident, but the value  $t_f$  of a symbol must always be equal or bigger than its  $t_i$ . The decis and quartis discretizations are suitable for analysis since they allow a better understanding of *variation-based* results.

## 4.2 Transactions Generation

Figure 2 (b) illustrates the transactions generation process carried by TRUMiner between  $var_X$  and  $var_Y$  of the dis-

cretized time series  $s'_i$ . The value  $w = i_2$  sets as  $i_2$  the maximum beginning time of the second symbol after the beginning of the first symbol. The first transaction generated in the example is  $([var_X, I], [var_Y, D])$ ,  $\Delta t = 0$  with the first element from the first symbol in  $s'_i[var_X]$ , the second element is the first symbol from  $s'_i[var_Y]$ , and the temporal feature is the difference between  $t_i = 0$  of the first element and  $t_i = 0$  of the second element.

Given the pair of variables, for each series that contains both variables, each quantized symbol generated by the discretization process is combined to each quantized symbol from the other variable. In this process, it is verified if  $t_i^Y \geq t_i^X$  and  $t_i^Y \leq (t_i^X + w)$  where  $t_i^X$  is the beginning time of the first symbol and  $t_i^Y$  is the beginning time of the second symbol. For instance, the last symbol of  $s'_i[var_Y]$  has  $t_i \geq i_2$ , so the transaction  $([var_X, I], [var_Y, I])$ ,  $\Delta t = i_{l-1}$  with the first symbol from time series  $s'_i[var_X]$  with  $t_i = 0$  will not be generated.

After reaching  $\Delta t = i_2$ , no new transaction is accounted for the first symbol  $I$  from  $var_X$ , so the following symbol in  $var_X$  (e.g.  $I$ ) is evaluated. It is relevant to highlight that the frequency of a transaction with a fixed first element considers the related temporal feature. For example, given the transaction  $([var_X, I], [var_Y, D])$ ,  $\Delta t = i_1$ , the  $freq([var_X, I], \Delta t = i_1)$  is accounted for each  $I$  in  $var_X$  that composes a transaction of type  $([var_X, I], ())$  with temporal feature  $\Delta t = i_1$ . The transaction generation step is executed for the pair  $(var_X, var_Y)$  and also for  $(var_Y, var_X)$ . Algorithm details are presented in Algorithm 1: lines 4 - 17.

## 4.3 Support and Confidence

TRUMiner obtains the rules directly from the transactions generated. The rule's antecedent is the first element of the transaction, while the consequent is the second. Multivariate temporal rules are evaluated through adapted support

and confidence, as detailed in **Section 2 (Equations 1 and 2)**. A rule's frequency is the number of rules generated with the same antecedent, consequent, and temporal feature (**Algorithm 1: lines 18 - 20**).

The rules are returned above a minimum support ( $sup_{min}$ ) and a minimum confidence ( $conf_{min}$ ). TRUMiner returns rules ordered by support and confidence, with the antecedent variable and its symbol, the consequent variable and symbol, and the temporal feature between antecedent and consequent (e.g.  $([var_X, I] \Rightarrow [var_Y, D], \Delta t = 0)$  in **Figure 2 (c) Short version**). The extended rules include the time series identifiers ( $s$ ) in which each rule is verified, as well as the beginning and ending time of each antecedent and consequent occurrences (e.g.  $([var_X, I] \Rightarrow [var_Y, D], \Delta t = 0)$ , series  $s_i$ , antecedent and consequent  $t_i = 0$  and  $t_f = i_1$  in **Figure 2 (c) Extended version**) (**Algorithm 1: lines 21 - 28**).

#### Algorithm 1 TRUMiner Algorithm

**Input:** Dataset  $S$ , Discretization method  $disc$ , Array of pair of variables  $p$ , Temporal threshold  $w$ , Minimum Support  $min_{sup}$ , Minimum Confidence  $min_{conf}$   
**Output:** Temporal rules (short or extended) ordered above  $min_{sup}$  and  $min_{conf}$

```

1: for each variable  $v$  in  $S$  do
2:   Discretize in  $disc$  each time series  $s_i[v]$  in  $s'_i[v]$ 
3: end for
4: for each pair of variables  $var_X, var_Y$  in  $p$  do
5:   for each discretized time series  $s'_i$  do
6:     for each symbol  $\alpha_{t_i^k, t_f^j}^X$  in  $var_X$  do
7:       for each symbol  $\alpha_{t_i^k, t_f^j}^Y$  in  $var_Y$  do
8:         if  $(t_i^k < t_f^j)$  or  $(t_f^j > (t_f^j + w))$  then
9:           Break
10:        end if
11:         $\Delta t \leftarrow (t_i^k - t_f^j)$ 
12:        Generate transaction with  $\alpha_{t_i^k, t_f^j}^X, \alpha_{t_i^k, t_f^j}^Y$ 
13:        Increase number of total transactions  $T$ 
14:      end for
15:    end for
16:    Repeat process for  $var_Y, var_X$ 
17:  end for
18:  for each transaction  $tr$  do
19:    Generate rule  $r$ 
20:    Evaluate  $sup$  and  $conf$ 
21:    Keeps rules  $sup \geq min_{sup}$  and  $conf \geq min_{conf}$ 
22:  end for
23:  Sort rules by  $sup$  and  $conf$ 
24:  for each rule  $r$  do
25:    for each discretized time series  $s'_i$  do
26:      Evaluate variables  $var_X, var_Y$  if  $r$  occurs
27:    end for
28:  end for
29: end for

```

## 4.4 Final Remarks

TRUMiner was written in C++ using the concept of classes. The low-level language allows better memory usage, avoid-

ing the lack of main memory, which is fundamental for rule mining. The use of classes makes code easier to understand, reuse and upgrade. The overall algorithm memory complexity varies according to the rule's format (extended or short). However, the constant usage is  $\delta \cdot N \cdot n + \delta \cdot n + N$  for data, with  $\delta \cdot N \cdot n$  to store observations,  $\delta \cdot n$  to maintain the time of each observation in each variable, and  $N$  to store the time series identifiers (**Figure 3 (a) Data**). The quantized data is illustrated in **Figure 3 (b) Quantized Data** where the memory usage is given by  $\delta \cdot (3 \cdot N \cdot L)$  for all quantized data, with the constant three from the storage of the quantized observation (e.g.  $I$ ), the beginning time (e.g.  $0$ ) and the ending time (e.g.  $i_1$ ).

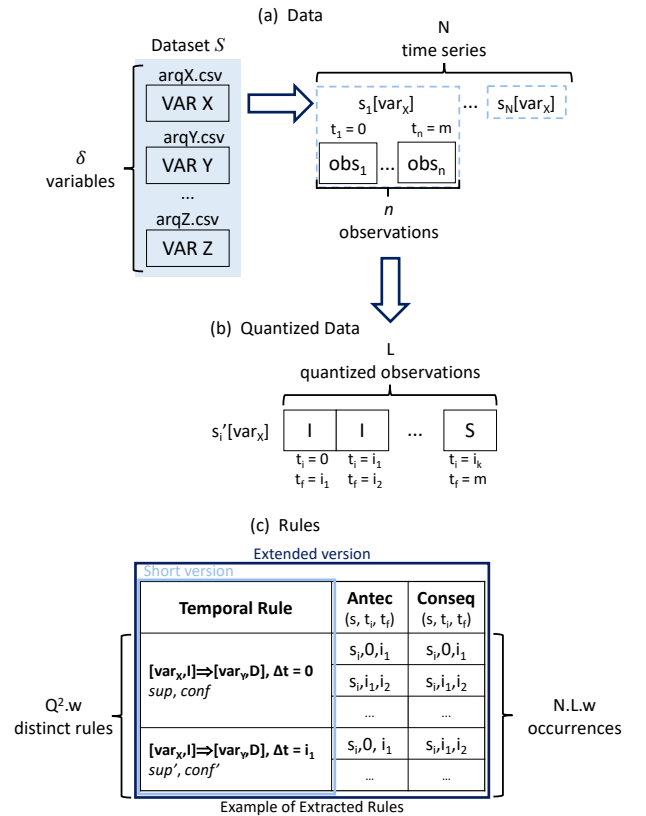


Figure 3. Memory complexity.

For the rules, the memory complexity can be visualized in **Figure 3 (c) Rules**, directly related to the number of distinct quantized symbols and the number of pairs of variables to mine. Considering  $Q$  the number of distinct quantized symbols, the algorithm could generate up to  $Q^2 \cdot w$  distinct rules, so in the short format, the maximum memory usage to mine all possible pairs of variables is  $7/2 \cdot \delta \cdot (\delta - 1) \cdot Q^2 \cdot w$ , with  $\delta \cdot (\delta - 1)/2$  being the number of pairs of variables, and the constant seven from the storage of the variables and quantized observations of antecedent and consequent, temporal feature, support and confidence. In the extended format, is also needed to store the rules occurrences for antecedent and consequent for each pair of variables, so we have up to  $7/2 \cdot \delta \cdot (\delta - 1) \cdot Q^2 \cdot w + N \cdot \delta \cdot (\delta - 1) \cdot L \cdot w$  where the last term is the memory to store the occurrences. Simplifying, we have  $\delta^2 \cdot w \cdot Q^2$  for short and  $\delta^2 \cdot w \cdot (Q^2 + N \cdot L)$  for extended rules.

Since the algorithm works for each pair of variables

(present in the series), the worst-case scenario is all series with all the variables and no missing observations mining for all pair of variables possible. The transaction is generated through all quantized symbols (denoted by  $L$ ) in one variable while combining with quantized symbols of another variable in maximum  $w$  of time span. So, the complexity of TRUMiner can be up to  $N \cdot \delta \cdot (\delta - 1) \cdot L \cdot w$ , and the transaction generation is the most costly process, as expected.

The TRUMiner is capable of mining temporal rules from multivariate time series and can handle missing observations, missing variables and data from multiple sources. Multiple discretization methods are already implemented, and it is possible to include new ones.

However, TRUMiner is designed to mine rules from each time series and not between series, requiring an adaptation for that purpose. Also, as the algorithm works for pairs of variables, patterns between more than two variables can not be discovered. Finally, although TRUMiner uses a temporal threshold  $w$  to directly regulate the algorithm's running time, it is still close to a brute force approach.

## 5 Experimental Analysis

To evaluate TRUMiner, we performed experiments on a real dataset of economic data, integrated into multivariate time series with missing observations and missing variables. In the following sections, we present more details about the dataset and results obtained by TRUMiner.

### 5.1 Dataset

The experimental analysis carried out explores international economic trade data, with the following indices per country: import and export values<sup>1</sup>, Gross Domestic Product (GDP)<sup>2</sup>, and Economic Complexity Index (ECI)<sup>3</sup> which characterizes the country's technological production capacity.

**Table 2.** Missing data for each variable in dataset.

Variable	Number of series	Number of M.O.
Import	19 (8.33%)	167
Export	19 (8.33%)	166
GDP	42 (21.43%)	146
ECI	4 (3.01%)	21

The dataset consists of annual series with four variables: import (IMP) and export (EXP), both containing observations of 228 countries from 1996 to 2020, GDP with the same interval coverage and observations from 196 countries, and ECI with observations from 133 countries from 1996 to 2019. This dataset presents missing observations (M.O.) and missing variables as shown in **Table 2**. The percentual of series with no missing observation in each variable varies from 80% to 90% of the series total.

The analysis of this dataset allows a better understanding of the relationship between economic variables and may aid in the prediction of economic behavior. Also, mining multivariate temporal rules from a heterogeneous dataset is an improvement on existing related algorithms.

### 5.2 Results

This section presents the results obtained from TRUMiner over an economic trade multivariate dataset. In all experiments, the temporal threshold ( $w$ ) to generate the transactions was 5 years, which is the approximated time of an economic cycle<sup>4</sup> [Zarnowitz and Ozyildirim, 2006]. Rules with temporal feature above the economic cycle may have reduced significance and low acceptance in the economic field. We present the experiments performed and its main objectives as follows.

- **Experiment 1:** Discretization methods evaluation.
- **Experiment 2:** Evaluation of *variation-based* discretization results.
- **Experiment 3:** Impact analysis of missing data.
- **Experiment 4:** Analysis of extended rules.
- **Experiment 5:** Support cut evaluation.
- **Experiment 6:** Discussion on specific countries rules mining.

The **Experiment 1** aims to evaluate the discretization methods in the heterogeneous dataset. For experiments to evaluate discretization methods,  $min_{sup}$  and  $min_{conf}$  were set to 0. Regarding discretization parameters, the *variation-based* method used 0.1 for maximum to classify as stable, 0.0001 for minimum variation to consider for quantization, 0 for maximum value to be a stability symbol, and 0 for minimum variation to generate a symbol. The alphabet size of SAX discretization was set to 3.

A summary of results obtained for the dataset without any data pre-processing for all discretization methods we tested is shown in **Table 3**. While the number of distinct rules in *variation-based* and SAX discretizations is limited due to the number of distinct symbols allowed, for decic and quartic, the observed number of distinct rules is in a higher magnitude order. The confidence in **Table 3** is from the rule with maximum support in each method.

Temporal rules of multivariate series with the temporal feature fully participating have low support due to their combinatorial nature. For each pair of variables the average of rules reaches 30,000. The most frequent and confident rule on the pair (GDP, EXP) is  $([EXP, I] \Rightarrow [GDP, I], \Delta t = 0)$ ,  $sup = 4.29$ , and  $conf = 81.44$ . This rule indicates that worldwide, 81% of the times that a country's exports rise, its GDP also rises in the same period. Despite the low support, the rule has high reliability due to its high confidence.

The support distribution of all rules for each discretization method is presented in **Figure 4**. While *variation-based* has temporal rules with support up to 5, indicating its usefulness, the decic discretization has less than ten temporal rules with

<sup>1</sup>BACII (CEPII) [http://www.cepii.fr/CEPII/en/bdd\\_modele/bdd\\_modele\\_item.asp?id=37](http://www.cepii.fr/CEPII/en/bdd_modele/bdd_modele_item.asp?id=37)

<sup>2</sup>GDP (IMF) <https://www.imf.org/en/Publications/WEO/w eo-database/2022/April>

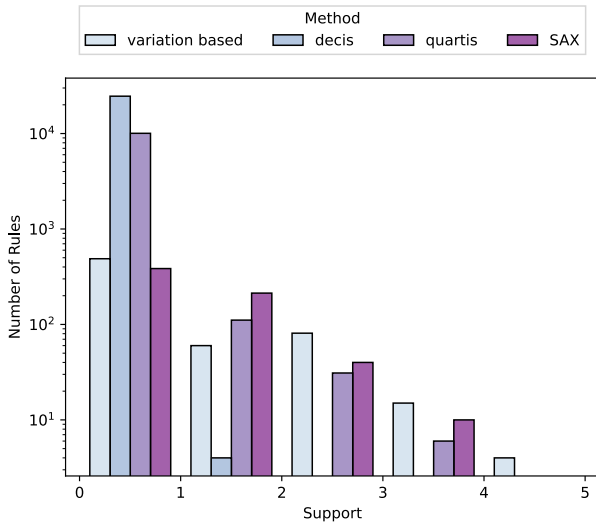
<sup>3</sup>ECI (Harvard) <https://atlas.cid.harvard.edu/rankings>

<sup>4</sup>EABCN <https://eabcn.org/dc/chronology-euro-area-business-cycles>,  
NBER <https://www.nber.org/research/data/us-business-cycle-expansions-and-contractions>



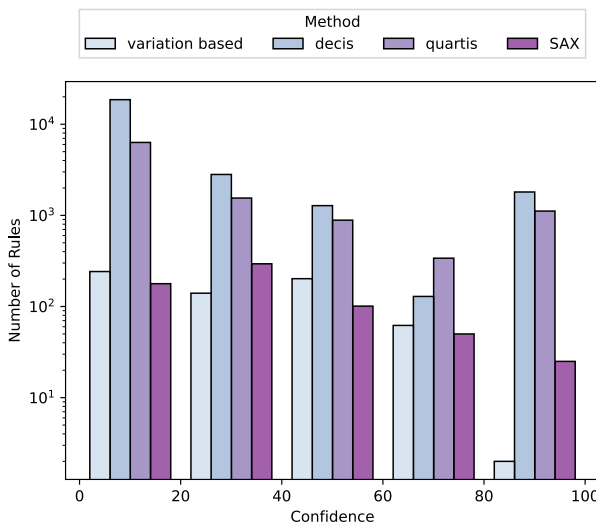
**Table 3.** Main results for each discretization.

Discretization Method	Distinct Rules	Maximum Support	Confidence
Variation-based	648	4.43	81.94
Decis	24,630	1.20	59.02
Quartis	10,203	3.64	80.22
SAX	648	3.32	91.34



**Figure 4.** Support distribution of rules in each discretization method.

support greater than 1. Tens of thousands of distinct rules with low support are verified in decis due to its low granularity for discretization. The exaggerated generation of quantized symbols leads to high quantities of distinct rules, all having low occurrence, and possibly without meaning.



**Figure 5.** Confidence distribution of rules in each discretization method.

**Figure 5** shows the confidence distribution for discretization methods in TRUMiner. While *variation-based* and SAX discretization have the expected behavior, with a decreasing tendency from 0 to 100, decis and quartis have a peak in the higher 20% of confidence. Usually, this occurs for low-frequency transactions, with most occurrences as a natural data oscillation.

The pair (IMP, GDP) generates the rule with the highest support (main rule) of all pairs of variables in each discretization. **Table 4** presents this rule, its support and confidence for each case. The symbol  $I_1$  in decis and quartis discretizations means an increase of one decis or quartis, respectively. While the rule in decis discretization implies that an increase up to 10% of import is followed by an increase up to 10% in GDP in the same year, the values of increase in quartis discretization are up to 25%. In SAX discretization, the symbol  $a$  indicates that the observations mean is the lowest portion of the observations distribution in the variable. This SAX rule implies that in the same period, when the mean observations of import are the lowest, the GDP is also its lowest.

**Table 4.** Main rule from (IMP, GDP) for each discretization.

Discretization	Temporal Rule
Variation-based	$(IMP, I) \Rightarrow (GDP, I), \Delta t = 0$ $sup = 4.43, conf = 81.94$
Decis	$(IMP, I_1) \Rightarrow (GDP, I_1), \Delta t = 0$ $sup = 1.20, conf = 59.02$
Quartis	$(IMP, I_1) \Rightarrow (GDP, I_1), \Delta t = 0$ $sup = 3.64, conf = 80.22$
SAX	$(IMP, a) \Rightarrow (GDP, a), \Delta t = 0$ $sup = 3.32, conf = 91.34$

In *variation-based*, decis and quartis discretization, the rules imply that an increase in import leads to an increase in GDP. Also, in all the rules presented in **Table 4**, the consequent occurs in the same year of the antecedent. The temporal features equal to 0 is expected since it is the time interval with the highest number of transactions. Furthermore, the quartil rule may imply that the observed increase in *variation-based* discretization rule is usually up to an increase of 25% in antecedent and consequent.

In **Experiment 2**, we evaluate the *variation-based* discretization, which showed better results for the analyzed dataset, with stronger rules, i.e., the highest support and high confidence. Hereafter we present further results from TRUMiner with *variation-based* discretization.

**Table 5** shows the main rule for each pair of variables using *variation-based* discretization. All those rules have the symbol  $I$  (increase) in antecedent and consequent, in accordance with the crescent tendency of the economic series. They also have temporal feature equal to 0, as expected. For rules with temporal feature  $\Delta t = 0$ , the antecedent and consequent could be interchangeable, but we presented the case with higher confidence which could indicates the most relevant of the two rules.

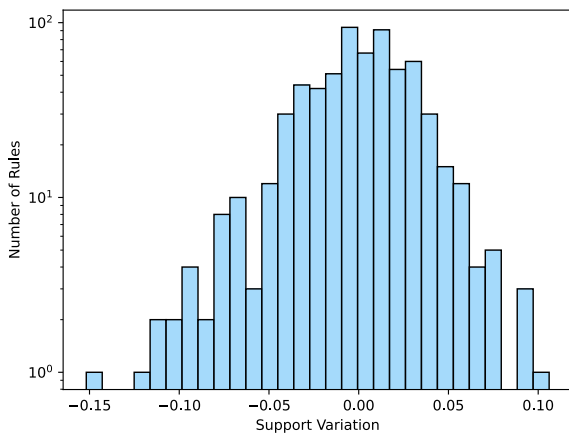
**Experiment 3** evaluates TRUMiner ability to handle missing observations. We executed one analysis focusing on Brazil’s series and another on multiple countries. In the anal-

**Table 5.** Main rules for *variation-based* discretization.

Pair of variable	Temporal Rule
IMP, EXP	$(EXP, I) \Rightarrow (IMP, I), \Delta t = 0$ $sup = 3.83, conf = 75.33$
IMP, ECI	$(ECI, I) \Rightarrow (IMP, I), \Delta t = 0$ $sup = 2.55, conf = 62.10$
IMP, GDP	$(IMP, I) \Rightarrow (GDP, I), \Delta t = 0$ $sup = 4.43, conf = 81.94$
EXP, ECI	$(ECI, I) \Rightarrow (EXP, I), \Delta t = 0$ $sup = 2.45, conf = 59.81$
EXP, GDP	$(EXP, I) \Rightarrow (GDP, I), \Delta t = 0$ $sup = 4.29, conf = 81.44$
ECI, GDP	$(ECI, I) \Rightarrow (GDP, I), \Delta t = 0$ $sup = 2.64, conf = 64.61$

ysis of the Brazil series, we used the variables GDP and IMP, with 0 to 4 observations (16%) randomly removed from import. The ten most frequent rules found in the case without missing observations match at least 60% of the ten most frequent rules in all of the cases with missing observations. This result indicates robustness against missing observations and the possibility of mining without pre-processing.

For deeper evaluation, a subset of the original dataset (described in Section 5.1) was assembled with no missing variables or missing observations. The new dataset is composed of over 100 series from 1996 to 2019 comprising IMP, EXP, ECI, and GDP. We mined the rules with *variation-based* discretization and then removed 5% of the dataset before a new run of TRUMiner.

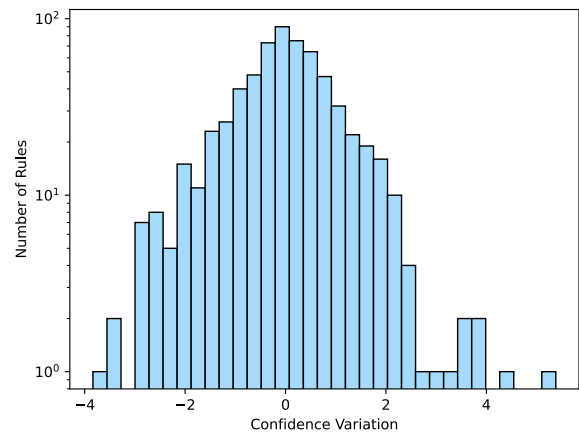


**Figure 6.** Support variation in missing observation analysis.

To evaluate the impact of missing data on overall results, the rules mined from the datasets with and without missing observations were matched and computed their variation of support and confidence. **Figure 6** shows support variation for coincident rules. All the rules were found in both datasets. The maximum support measured in the original dataset is 4.99 and the maximum variation of 0.15 represents only around 3% of discrepancy.

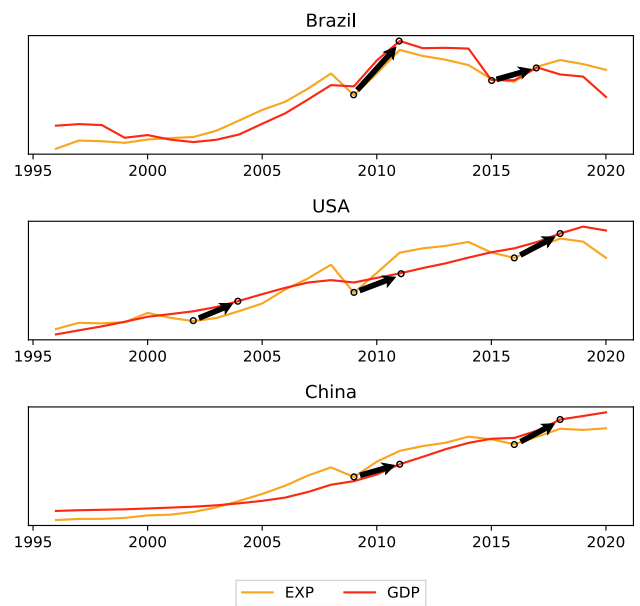
The confidence variation for the matching rule is shown in **Figure 7**. The data distribution is a Gaussian with 0 as the peak, indicating that most rules present the same confidence measure. These results reinforce the TRUMiner’s capacity to

handle missing observations. Regarding missing variables, since the rules are generated given the pair of variables, if one of the two is missing in one specific multivariate time series, TRUMiner does not generate any rules for this series. The other series that compose the dataset and have at least one observation for each variable for the analyzed pair (considering the temporal feature) have rules generated from.



**Figure 7.** Confidence variation in missing observation analysis.

In **Experiment 4**, we focused on the extended rules returned by TRUMiner. This format allows to easily locate the rule in the time series where it happened and relate the observation(s) to the quantized symbol. For example,  $([EXP, D] \Rightarrow [GDP, I], \Delta t = 2)$  indicates that GDP increases two years after exports decrease with evaluation measures of  $sup = 1.45$  and  $conf = 72.72$ .



**Figure 8.** Multivariate temporal rules of Brazil, USA and China.

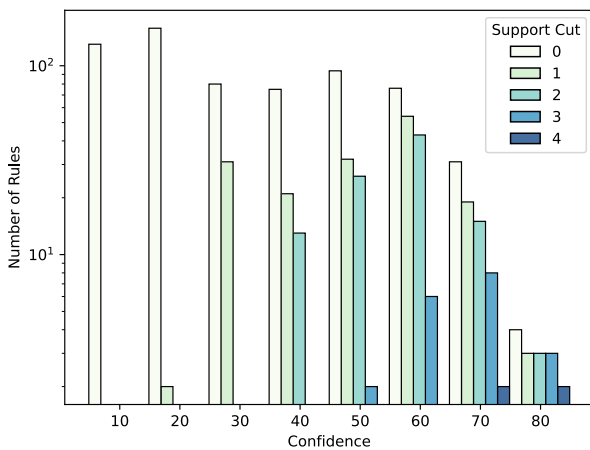
**Figure 8** illustrates some occurrences of this rule in the annual time series of Brazil, USA, and China from 1996 to 2020 for EXP and GDP. For better visualization, we normal-



ized the plotted variables with z-score. The arrows represent the rule occurrences, connecting observations (small circles) related to the antecedent and the consequent of the rule, respectively. It takes place as follows: Brazil in (2009, 2011), (2015, 2017), United States in (2002, 2004), (2009, 2011), (2016, 2018), and China in (2009, 2011), (2016, 2018).

The details provided by the extended rules also demonstrate that TRUMiner can handle varying-length time series. For instance, despite the shorter coverage period for ECI (1996-2019), rule  $([ECI, D] \Rightarrow [GDP, D], \Delta t = 1)$  has an occurrence in Brazil between 2019 (ECI) and 2020 (GDP), which can only be detected by algorithms with this capability. In the semantic analysis, this rule indicates that a decrease in ECI is followed by a decrease in GDP after one year.

In **Experiment 5**, we evaluated the TRUMiner response to a support cut. **Figure 9** shows the confidence distribution of rules from *variation-based* discretization with no support cut and  $sup_{min}$  from 1 to 4. With higher cut, more low confidence rules are discarded since the transactions that generate these rules have low occurrence in the dataset. Some higher confidence rules are also removed, but their exclusion presumably implies data randomness detected by the algorithm.

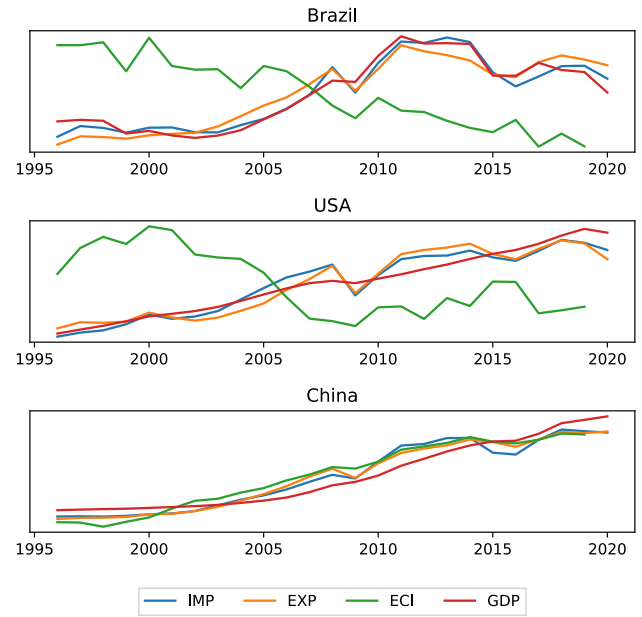


**Figure 9.** Confidence distribution of rules with support cut.

A  $sup_{min} = 2$  for *variation-based* discretization is the minimum value for this dataset that considerably reduces the number of the rules with low confidence, where the minimum confidence verified in **Figure 9** is 40%. It is also the value that preserves the maximum number of rules with higher confidence, returning more than 100 distinct rules.

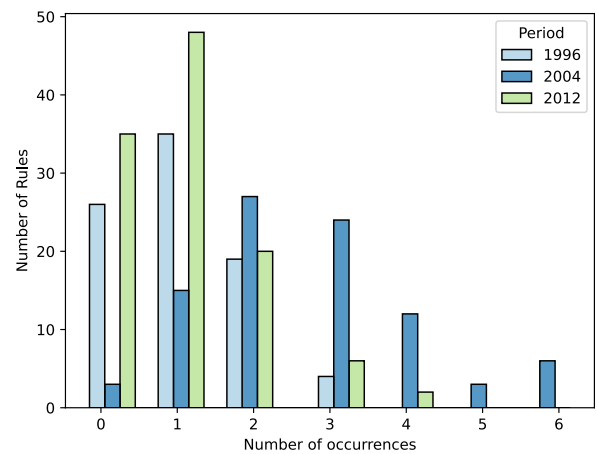
**Experiment 6** provides further results focused on Brazil, United States and China. We divided the original dataset into three periods with 8 years each, respectively: 1996 to 2003, 2004 to 2011, and 2012 to 2019. From **Figure 10**, we observe three world economic crises: a slight decrease in 1998-1999 and two major variations in 2009 and 2015. It is also possible to see COVID-19 reflexes in 2020, with a stagnation in China’s growth and negative growth in economic indexes from Brazil and USA.

Analyzing the rules from 2012 to 2019 in Brazil, United States and China, we verified frequent rules involving a decrease in import, export, or ECI, such as  $([EXP, D] \Rightarrow$



**Figure 10.** Normalized time series of Brazil, USA and China.

$[IMP, D], \Delta t = 0)$  with  $sup = 3.63$  and  $conf = 70.00$ . This rule indicates a possible decrease in import values at the same time which a decrease is verified in export.



**Figure 11.** Occurrence distribution of rules in Brazil for 3 distinct periods.

From the rules with support above 2 we selected those occurring in Brazil and counted every event. **Figure 11** shows the distribution of rules for the three periods. While the last period has generated 111 distinct strong rules, the first and the second periods reach only 80% of this quantity, indicating a worldwide diffuse behavior of the economic indexes in the last decade. It is also from 2012 to 2019 that all the strong rules with the most occurrences are related to a decrease in the economic indexes, mainly in import, export, and ECI.

## 6 Conclusion

Multivariate time series mining is a promising area due to the ability to obtain new and relevant information indexed to its

time interval. The temporal rules allow a better explanation of the patterns found and the possibility of making predictions about the data. Existing methods are often limited to univariate datasets and analysis, and require extensive series pre-processing.

TRUMiner algorithm aims to simplify the temporal rule mining process, as it can deal with data from multiple sources, time series of varying lengths, and with missing variables and observations. The multivariate temporal rules returned can be analyzed in short and extended formats, with the latter presenting the rules' occurrences in the time series and their time interval. Although the nature of the problem implies low support values, the results are promising for the analysis of heterogeneous multivariate time series.

The next step of our research is to generate multivariate temporal rules with three or more distinct variables efficiently. Another possible work is to optimize the TRUMiner with the use of big data techniques to treat bulky datasets.

## Acknowledgements

We would like to thank the Universidade de São Paulo (USP), particularly Instituto de Ciências Matemáticas e de Computação (ICMC) for all the support provided during this research.

## Funding

This research was partially funded by National Council for Scientific and Technological Development (CNPq), National Council for the Improvement of Higher Education (CAPES) and São Paulo Research Foundation (FAPESP).

## Authors' Contributions

Sousa contributed to the conception and evaluation of this study. Karasawa performed the experiments. Karasawa is the main contributor and writer of this manuscript. All authors read and approved the final manuscript.

## Availability of data and materials

The datasets generated and/or analysed during the current study are available in CEPII, IMF and Atlas of Economic Complexity.

## References

- Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216. DOI: 10.1145/170036.170072.
- Amaral, T. and Sousa, E. (2019). Trier: A fast and scalable method for mining temporal exception rules. In *XXXIV SBBD*, pages 1–12. SBC.
- Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.
- Buaton, R., Zarlis, M., Mawengkang, H., and Efendi, S. (2021). Find the best rule of time series data mining with cluster analysis. *Journal of Physics: Conference Series*, 1830(1):012020. DOI: 10.1088/1742-6596/1830/1/012020.
- Chen, X. and Petrounias, I. (2000). Discovering temporal association rules: Algorithms, language and system. In *16th ICDE*, pages 306–306. IEEE.
- Das, G., Lin, K.-I., Mannila, H., Renganathan, G., and Smyth, P. (1998). Rule discovery from time series. In *4th ACM KDD*, volume 98, pages 16–22.
- de Oliveira, F. A., Costa, R. L., Goldschmidt, R. R., and Cavalcanti, M. C. (2017). Mineração de regras de associação multirrelação em grafos: Direcionando o processo de busca. In *SBBD (Short Papers)*, pages 270–275.
- Han, J., Kamber, M., and Pei, J. (2011). *Data mining: Concepts and techniques. (3rd ed)*, Morgan Kaufman.
- Harms, S. K. and Deogun, J. S. (2004). Sequential association rule mining with time lags. *Journal of Intelligent Information Systems*, 22(1):7–22.
- Karasawa, E. and Sousa, E. (2022). Truminer: Mineração de regras temporais em bases de séries multivariadas e heterogêneas. In *XXXVII SBBD*, pages 403–408. SBC. DOI: 10.5753/sbbd.2022.226199.
- Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *8th ACM SIGMOD, DMKD '03*, page 2–11. DOI: 10.1145/882082.882086.
- Mitsa, T. (2010). *Temporal data mining*. CRC Press.
- Romani, L. A. S., de Avila, A. M. H., Zullo, J., Chbeir, R., Traina, C., and Traina, A. J. M. (2010). Clearminer: a new algorithm for mining association patterns on heterogeneous time series from climate data. In *ACM, SAC '10*, page 900–905. DOI: 10.1145/1774088.1774275.
- Schlüter, T. and Conrad, S. (2011). About the analysis of time series with temporal association rule mining. In *2011 IEEE CIDM*, pages 325–332. IEEE.
- Segura-Delgado, A., Gacto, M. J., Alcalá, R., and Alcalá-Fdez, J. (2020). Temporal association rule mining: An overview considering the time variable as an integral or implied component. *WIREs Data Mining and Knowledge Discovery*, 10(4):e1367.
- Shokoohi-Yekta, M., Chen, Y., Campana, B., Hu, B., Zakaria, J., and Keogh, E. (2015). Discovery of meaningful rules in time series. In *21th ACM SIGKDD, KDD '15*, page 1085–1094. DOI: 10.1145/2783258.2783306.
- Xue, R., Zhang, T., Chen, D., Le, J., and Lavasani, M. (2016). Sensor time series association rule discovery based on modified discretization method. In *2016 First IEEE ICCCI*, pages 196–202. DOI: 10.1109/CCI.2016.7778907.
- Zarnowitz, V. and Ozyildirim, A. (2006). Time series decomposition and measurement of business cycles, trends and growth cycles. *Journal of Monetary Economics*, 53(7):1717–1739. DOI: <https://doi.org/10.1016/j.jmoneco.2005.03.015>.
- Zhao, Y. and Zhang, T. (2017). Discovery of temporal association rules in multivariate time series. In *International Conference on Mathematics, Modelling and Simulation Technologies and Applications, 2017, Xiamen*, pages 294–300.