eTRUMiner: Mining Multivariate Temporal Rules from Heterogeneous and Incomplete Time Series*

Eliane Karasawa[1], **Elaine P. M. Sousa**[1]

[1]Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP) – São Carlos – SP – Brazil

eligniechk@gmail.com, parros@icmc.usp.br

Abstract. *This paper introduces eTRUMiner, a novel algorithm for mining multivariate temporal association rules from heterogeneous time series datasets with missing observations. Our approach enables user-defined discretization, discovers frequent patterns, and outputs rules in short or detailed formats, thus enhancing interpretability. The results show that: (i) the choice of discretization method significantly influences rule relevance; (ii) eTRUMiner preserves the rules with high confidence in a dataset with up to 15% missing data; and (iii) the extracted rules can capture plausible causal dynamics. These findings demonstrate eTRUMiner's robustness to incomplete data and its usefulness for exploratory analysis and forecasting in complex temporal domains.*

1. Introduction

The volume of data generated and collected daily is enormous, from our everyday life with IoT and social media usage to our work by-products, for example. In this scenario, machine learning plays a relevant role, with tens of thousands of related papers published in the academic and industrial area. The association rule mining is one of the machine learning tasks capable of extracting knowledge from databases.

Association rules represent causality relationships between antecedent and consequent [Agrawal et al. 1993], which is of great interest due to its simplicity, high explanation potential, and prediction abilities. However, for time series data, patterns may depend not only on co-occurrence but also on lead-lag relationships and seasonality. Therefore, the time of rule occurrences and the temporal distance between antecedent and consequent are valuable information.

Discovering causal, time-dependent relations in multivariate time series is crucial for domains ranging from finance to climate science. The temporal feature of the rules allows us to understand the order and time of occurrence of events [Segura-Delgado et al. 2020]. Hence, mining temporal rules can be an efficient means of obtaining useful information from massive data sources [Han et al. 2011]. For example, a temporal rule on the economic scenario could be "one year after a rise in import, the country's GDP also increases with 69% confidence" [Karasawa and Sousa 2023].

Temporal association rule mining extends the classical formulation by setting an explicit time window Δt to the rules, that is, ($A \Rightarrow C$, Δt), where A is the antecedent of the rule and C is its consequent. Yet most existing methods treat time implicitly as a simple ordering key or require pre-processing to align multiple series of different length and

*Os autores agradecem à CAPES e ao CNPq pelo apoio financeiro.

sampling rate [Romani et al. 2010, Zhao and Zhang 2017]. They also tend to be limited to univariate [Das et al. 1998, Schlüter and Conrad 2011] or complete multivariate time series datasets, which hampers their applicability to modern, heterogeneous and incomplete temporal repositories such as economic indicators, sensor networks, and electronic health records.

We propose eTRUMiner (extended Temporal RUles Miner), the next step of TRUMiner, a previous algorithm designed to mine temporal rules limited to two variables [Karasawa and Sousa 2022]. The eTRUMiner is capable of mining multivariate time series from several data sources and outputs rules with two or more distinct variables, indicating the exact time-span between antecedent and consequent. It can handle heterogeneous time series with missing observations and also missing variables. The algorithm runs with different discretization methods, allowing the user to choose an adequate discretization for the analysis purposes. The rules can be returned in short and extended format, with antecedent, consequent, and temporal feature in both cases. In the extended format, all occurrences of the rule and the corresponding time intervals are detailed.

This paper includes the related work in **Section 2**, background in **Section 3** summarizing the basic concepts of multivariate temporal rule mining, and eTRUMiner description in **Section 4**. **Section 5** describes the datasets, results, and analysis of eTRUMiner execution on international trade data. Finally, in **Section 6**, we present our conclusion and directions for future work.

2. Related Work

Rule-mining research dates to the 1990's, with the landmark Apriori algorithm in [Agrawal et al. 1993] that laid the foundation for later advances. For temporal rules, [Das et al. 1998] proposed the temporal feature as the number of quantized elements between the antecedent and the consequent of the rules, but only as a maximum time span. The MOWCATL algorithm [Harms and Deogun 2004] performs rule extraction considering a time window that can be a fixed or a maximum time between the antecedent and the consequent. However, rule extraction is performed only on predetermined elements of interest. In the Clearminer [Romani et al. 2010], an algorithm for extracting association rules composed of distinct variables from multivariate time series, the temporal factor delimits the maximum time window to generate the rules.

[Schlüter and Conrad 2011] evaluated three discretization methods, introducing a prototype to extract temporal rules from univariate time series. In [Zhao and Zhang 2017] the authors propose an algorithm to mine temporal rules from multivariate series with temporal feature such as the time span between the antecedent and the consequent of the rule. However, it needs to group the obtained patterns in clusters to reduce the generated patterns. TRiER [Amaral and Sousa 2019] extracts temporal exception rules from multivariate time series aiming for the maximum number of variables for each item. In [de Oliveira et al. 2017], the focus is the extraction of association rules in graphs.

In [He et al. 2024], the authors proposed temporal rule mining of multivariate time series using shapelets, however, there is no mention of missing value handling. [Ho et al. 2025] focus on temporal pattern mining, providing information of which variables occurred when, but with no mention of heterogeneous datasets. The work of [Srivastava et al. 2024] proposes using association rules from financial time series for

deep learning forecasting, with missing values treatment being series exclusion or observations imputation. We could not find any research work that simultaneously handles heterogeneous multivariate time series, missing observations and variables, and multivariate temporal rules as eTRUMiner does.

3. Background

Discretization is a relevant pre-processing step for temporal rule mining. The discretization of a multivariate time series can be applied to each variable separately as in a univariate time series and then group the variables to generate the discretized multivariate time series. For example, consider a multivariate time series s with n observations and δ variables where a univariate time series is $s[var_X] = obs_1^X, \ldots, obs_n^X$, with $var_X \in [1, ..., \delta]$. For each variable, the discretization process generates the discretized univariate time series $s'[var_X] = \alpha_{t_1,t_f}^X, \ldots, \alpha_{t_i,t_n}^X$, α_{t_i,t_f}^X representing a quantized symbol from variable X covering $s[var_X]$ from t_i (beginning time) to t_f (ending time).

The transaction is defined as two sets of patterns, where a set is exemplified as $([var_X, \alpha_i^X], \ldots)$, and a temporal feature Δt indicating the time interval between the two sets. It can be represented as

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

Each pattern (e.g. $[var_X, \alpha_i^X]$) contains a variable (e.g. var_X representing variable X) and its respective quantized symbol (e.g. α_i^X), with i indicating the coverage time interval (t_i, t_f) in the discretized time series. Each variable can appear in at most one pattern within a transaction [Karasawa and Sousa 2023]

The set with earlier beginning time on the transaction is termed the "antecedent", while the latter is the "consequent". As the temporal feature indicates the time span, it can be delimited by a temporal threshold w, the maximum time window, chosen by a domain specialist to filter only causal relationships, as detailed in [Romani et al. 2010]. With this delimitation, the maximum number of transactions T generated from a dataset with N time series can be calculated as $\frac{N}{2}.(w+1)(2.L-w)$, where L is the number of discretized observations from each time series.

A multivariate temporal rule is defined as

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

where $([var_X, \alpha_i^X], \ldots)$ is the rule's antecedent and the consequent is $([var_Y, \alpha_j^Y], \ldots)$. The antecedent came from the first set in the transactions while the consequent came from the second, covering up to δ variables. The temporal feature Δt of the rule comes from the temporal feature of the transaction, indicating the time span between the antecedent's and consequent's beginning time.

Each temporal rule is derived from a single transaction and includes at least one pattern from the antecedent set and one from the consequent set. It can contain up to δ distinct patterns, with a minimum of two. The same is valid for the time series variables since each variable is only in one pattern of the transaction.

To evaluate the rules, quality measures from association rules mining, such as support and confidence [Agrawal et al. 1993], can be extended to multivariate tempo-

ral rules. However, it is necessary to integrate the temporal feature and the multivariate characteristic. Support is the frequency of a rule in the whole set of transactions. **Equation 1** shows the multivariate temporal support used in this work, as presented in [Karasawa and Sousa 2023].

$$sup = 100.\frac{freq([var_X, \alpha_i^X], \ldots \Rightarrow [var_Y, \alpha_j^Y], \ldots, \Delta t)}{T} \quad (1)$$

The dividend is the frequency of the rule $([var_X, \alpha_i^X], \ldots \Rightarrow [var_Y, \alpha_j^Y], \ldots, \Delta t)$ and T is the number of transactions obtained from the dataset. Since a transaction also has the temporal feature, even transactions composed of the exact same sets can be distinct. The number of transactions for a temporal feature k, with $k \in [0, ..., w]$, is

$$T_{\Delta t=k} = N.(L - k)$$

The temporal support can vary from 0 to 100, where a rule with support equal to 100 is present in all the transactions obtained from discretized time series. However, in temporal rule mining, this can only occur when $w = 0$, i.e. only the temporal feature $\Delta t = 0$ is allowed. For multiple time spans, the percentage of transactions for a temporal feature k is then

$$T_{\Delta t=k}(\%) = 100.\frac{2.(L - k)}{(w + 1)(2.L - w)}$$

For example, in a time series with 23 discretized elements and a time window $w = 5$, the temporal feature $\Delta t = 0$ is present in 18.7% of the transactions. Therefore, the maximum support that any rule with temporal feature $\Delta t = 0$ can obtain is 18.7. The larger the size of the evaluated time window, the lower the maximum support that can be achieved.

The precision of the rule 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. **Equation 2** measures the confidence in multivariate temporal rules.

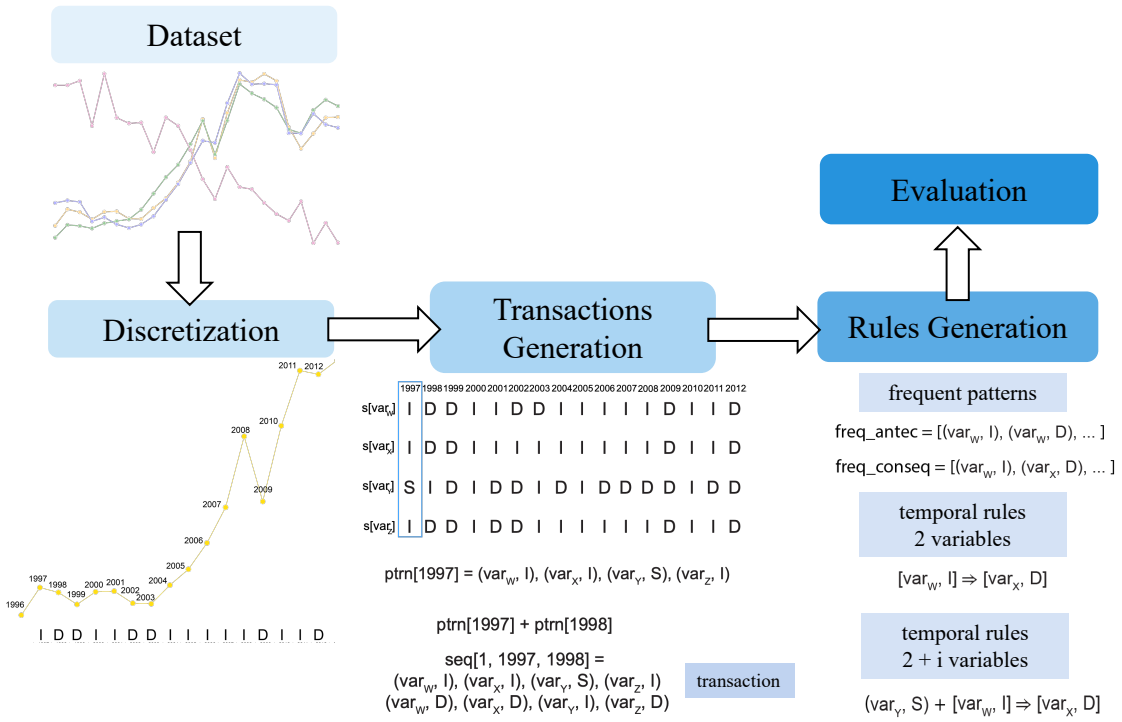
$$conf = 100.\frac{freq([var_X, \alpha_i^X], \ldots \Rightarrow [var_Y, \alpha_j^Y], \ldots, \Delta t)}{freq([var_X, \alpha_i^X], \ldots, \Delta t)} \quad (2)$$

For the output of the rules, we adopt the formats “short” and “extended”. A “short” rule contains the antecedent, the consequent, and the temporal feature. An “extended” rule is more detailed, composed of the short rule and all occurrences in time series. To locate each occurrence of the rule, the series index and the time of the first observation that refers to the beginning of the temporal rule are provided.

4. The eTRUMiner Algorithm

The eTRUMiner (extended Temporal RUles Miner) is an algorithm to mine multivariate temporal rules from time series of different origins. It handles time series with missing observations and missing variables (incomplete series), and distinct duration between variables (heterogeneous series), without pre-processing needs. The algorithm can be

Figure 1. eTRUMiner overview with step samples.



summarized in 4 steps: discretization, transactions generation, rules generation, and rules evaluation. **Figure 1** illustrates eTRUMiner by exemplification of the first three steps.

In the discretization process, each variable of each time series is discretized into a series of quantized symbols composed of: a symbol (eg.: I, D, S), a beginning, and an ending time on the respective time series. The quantized symbols are then grouped into patterns to generate transactions in the transaction generation step. The frequent patterns obtained from the transactions are used to generate the rules, first two-variable rules and then rules with more than two distinct variables. The evaluation step is executed based on user thresholds, such as minimum confidence, to obtain relevant rules.

The entry dataset (S) is a set of multiple univariate time series that may have different origins but can be grouped into domain-meaningful multivariate time series. In **Figure 1**, each color in the dataset representation indicates a distinct variable. The series identifier must coincide in all variables to allow eTRUMiner to identify a multivariate time series. Other entry parameters are the discretization method, the temporal threshold, the minimum support, and the minimum confidence. The temporal threshold limits the maximum temporal feature based on the semantic meaning of the rule. Minimum support and confidence are also used to reduce the algorithmic complexity.

4.1. Discretization

In the discretization process, the choice of discretization method is intrinsically tied to the application domain. Since the eTRUMiner was originally designed for economic data, the chosen methods were a *variation-based* discretization, decis, quartis and SAX [Lin et al. 2003]. Each discretized observation has its symbol, beginning time and ending

time. This representation allows the quantized symbol to represent a sequence of the original observations.

eTRUMiner handles missing observations and missing variables during the discretization step because beginning and ending times of each quantized observation are stored. With this information, transactions and rules are generated using precise time marks. This ability of eTRUMiner simplifies time series pre-processing, and allow diverse datasets to be explored, even with missing observations, missing variables, and distinct duration between series without the need for data imputation.

Figure 2. Discretization process with missing observations.

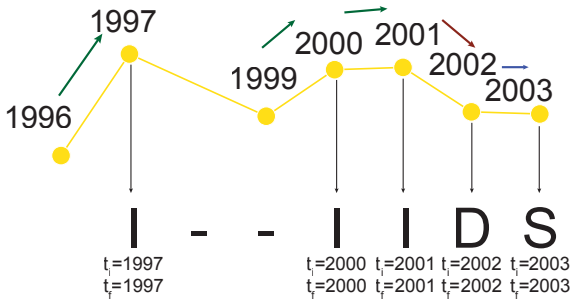


Figure 2 samples a discretization process in which the applied method discretizes each observation in one quantized symbol, so both beginning and ending time equals the observation time. An element *I* refers to an increase behavior between observations, *D* indicates a decrease while *S* stands for a stability behavior. Since the third observation (1998) is missing, the second and third quantized symbols are not generated, with no loss to the method because the beginning and ending times are stored.

4.2. Transactions Generation

After discretizing all time series, the quantized series are used to generate transactions in the next step. Each quantized symbol is stored in the pattern form, linked to a time stamp with its transactions occurrences. The rule generation then is performed over frequent patterns, using minimum support as the threshold and the transaction frequency.

As detailed in **Section 3**, a pattern contains the quantized symbol and its respective variable. In the transactions generation step, patterns are ordered by the beginning time of the quantized symbol (e.g. in **Figure 1** quantized symbols for each variable are aligned for beginning year) and grouped into lists (ptrn[1997] for example in **Figure 1**), the maximum number of elements being the number of distinct variables in the dataset. The set of lists are concatenated in pairs (in **Figure 1** a pair is exemplified as seq[1, 1997, 1998]) complying with the temporal threshold as the maximum time span between the beginning time of the pair for generate the transactions.

For each transaction is stored an identifier and the beginning time of both antecedent and consequent sets. Every pattern that composes a transaction is then counted, considering the temporal feature of the transaction in which it occurs and if it is in the antecedent or consequent of the transaction. The storage of transactions occurrences helps in rules generation and facilitates rules location in the extended representation.

4.3. Rules Generation

Given the minimum support, we can calculate the minimum occurrences of a pattern to be classified as a frequent pattern. It is given by $sup_{min} * T$ where sup_{min} is a user-defined threshold and T accounts for the total number of transactions. Only frequent antecedent and consequent patterns are used to generate temporal rules for each temporal feature up to the temporal threshold w.

In this work, the antecedent frequent patterns are joined to the consequent ones if they are from distinct variables, producing rules with no repeated variable. This joining step generates all possible rules, but only rules with a high intersection frequency between antecedent and consequent occurrences are stored. The minimum occurrences are also used here to filter the relevant rules. Rules with more than one pattern in the antecedent and one in consequent are obtained by the eTRUMiner adding other frequent patterns from distinct variables of the ones already present in the rule.

In the increase pattern step, only the rules generated with the n patterns and with frequency above the minimum occurrences are used to generate rules with the $n + 1$ patterns. The process of rules generation is executed up to the end of the possible variables to increase or the last rules generated with n patterns cannot generate any rule with $n + 1$ pattern with frequency above the minimum. Each distinct rule with minimum frequency or more is stored for the rule evaluation step.

4.4. Rules Evaluation

The evaluation of rules is the last step of eTRUMiner. In this work, support and confidence were implemented as defined in **Equation 1** and **Equation 2** respectively. They are traditional quantitative measures for rules mining, with adaptations to fully incorporate the temporal feature of temporal rules. However, in temporal rules mining, support can be very low, as detailed in **Section 3**.

The support measure utilizes the frequency of the rule, stored on the rules generation step, and the total of transactions generated, obtained from the transaction generation step. The confidence is obtained from the frequency of the rules over the frequency of the transaction that has the same patterns in the antecedent and also the same temporal feature.

Rules are returned if they meet the minimum support and minimum confidence thresholds defined by the user. They are ordered by support and confidence, respectively, and can be returned in a short or extended format. For example, $([IMP, I] \Rightarrow [GDP, I], \Delta t = 0)$ is a sample of the short format while its extended format includes occurrences, for example (bra,1997;bra,2000;bra,2001). This rule indicates that a rise on import volumes is detected with a rise in GDP volumes in the same year. The extended format also shows that there are occurrences of this rule in Brazil on 1997, 2000 and 2001.

4.5. eTRUMiner Implementation

The implementation of eTRUMiner was carried out in C++ using the concept of classes. To construct the dataset from the time series input, the identifier of each time series must be consistent (for example, in the evaluated sample, the Brazil series are referred to “bra” in all variables), maintaining exactly the same denomination in all variables of the same

series. The order of organization between the series is irrelevant, which facilitates the integration of variables from different sources into a single set of multivariate series.

The discretization is performed by variable for each series, allowing different discretizations between variables. However, this analysis was not performed in this work. The implementation of the SAX discretization uses a table containing the coverage range of each symbol, stored on a file. The maximum number of distinct elements in the SAX discretization is 26, which constitutes the size of the alphabet.

The main data structures used in eTRUMiner are vectors and dictionaries. For example, each temporal feature Δt contains a dictionary to store frequent patterns and their occurrences as an antecedent or a consequent. This allows for easy navigation during rules generation, while also avoiding pattern and variable duplication. Since each transaction is stored as an index from a vector, a rule frequency is reduced to the intersection between patterns occurrences.

The memory usage by eTRUMiner is given by $O(3^{\delta}.N.L.w)$ in the worst-case scenario, with each transaction generating up to almost 3^{δ} rules. The threshold w limits the memory usage, δ is the number of variables in the N time series of the dataset that are composed of up to L quantized symbols. For time complexity, O($w.3^{\delta}.\delta^4.L^2$) is the worst case scenario, with rules generation as the costly step. However, this estimate does not consider some heuristics implemented at code level, such as minimum support, order between patterns stored and patterns in the same antecedent or consequent having the same beginning time.

5. Experimental Analysis

The eTRUMiner experiments were performed over multivariate time series from international trade of distinct sources, with missing observations and missing variables, adding up 9.39% of missing values. The dataset covers 232 countries from 1996 to 2020 of import[1], export[1], ECI[2] and GDP[3]. The experiments are presented in three subsections based in their objectives: in **Subsection 5.1** we evaluate the algorithm's performance on the dataset, in **Subsection 5.2** we assess eTRUMiner's ability to handle missing values, and the semantic interpretation of the extracted rules is detailed in **Subsection 5.3**.

5.1. eTRUMiner Performance

We tested, four discretization methods on the dataset: the *variation-based* discretization, decis, quartis, and SAX. For the first three methods, discretization was applied directly to the original data, whereas SAX includes a z-score normalization step before generating quantized symbols. For the discretizations *variation-based* and SAX, the maximum number of distinct quantized symbols is limited to three, while decis and quartis do not have a predefined maximum. In terms of the resulting number of rules, *variation-based* and SAX produce more than ten thousand rules, while decis and quartis generate hundreds of thousands.

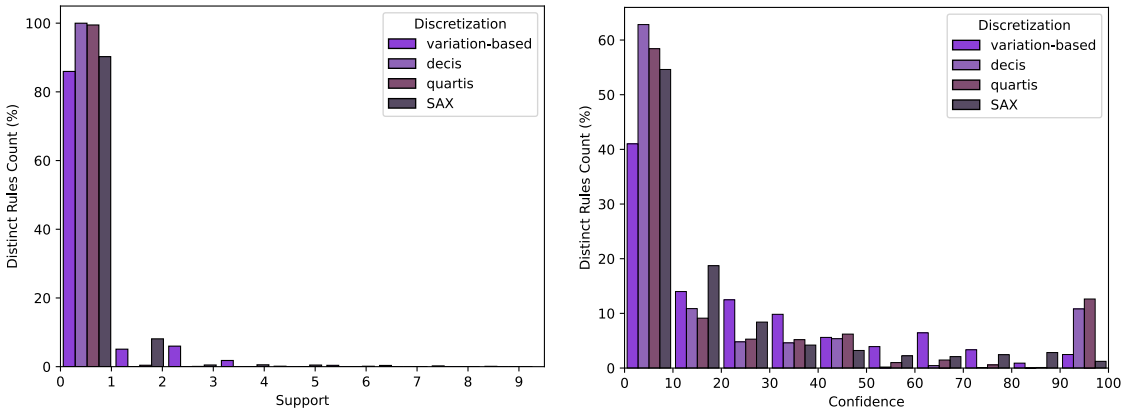
[1]BACII (CEPII) `http://www.cepii.fr/CEPII/en/bdd\_modele/bdd\_modele\_item.asp?id=37`

[2]ECI (Harvard) `https://atlas.cid.harvard.edu/rankings`

[3]GDP (IMF) `https://www.imf.org/en/Publications/WEO/weo-database/2022/April`

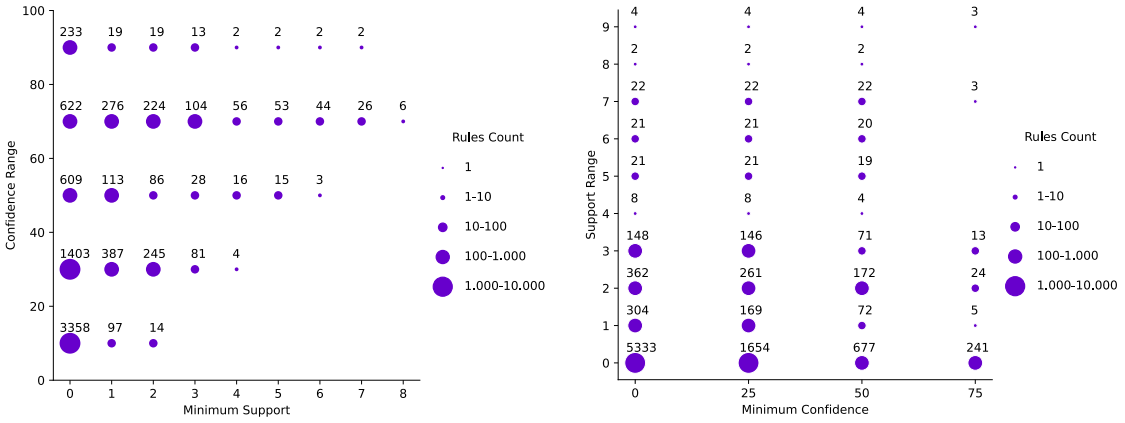
Figure 3 presents the distribution of rules on support and confidence for the evaluated methods. The *variation-based* discretization generates the lowest volume of rules and achieves the highest support value, indicating the lowest dispersion in support values among rules. For confidence measure, the values can not be directly compared across discretization methods, since the confidence of a rule is the percentage of occurrence of the same antecedent in the specific set of rules.

Figure 3. Percentage distribution of temporal rules in support and confidence measures.



In all discretization methods, more than 80% of the rules have support below 1 while the maximum value is 9.32, obtained for *variation-based*. This method also extracted 0.5% of the rules with support between 9 and 10. For the confidence measure, at least 40% of the rules have confidence up to 10, but rules with high confidence (between 90 and 100) have a high probability of low occurrence and therefore are not of interest.

Figure 4. Rules distribution after set threshold for support and confidence.



The high percentage of rules with low support and low confidence strengthens the usage of minimum support and minimum confidence threshold, that can significantly reduce the volume of rules generated and the algorithm processing and memory cost, and also help select more relevant rules. **Figure 4** presents the distribution of the rules

after applying the minimum support and minimum confidence thresholds for time series discretized using the *variation-based* method.

In **Figure 4a**, the x-axis represents the minimum support threshold and the y-axis indicates the confidence interval of the rules. The number above each symbol refers to the number of rules in the respective confidence interval for the minimum support threshold. From this plot, it is possible to observe that rules with lowest support have mostly low confidence, between 0 to 20, as expected. Rising minimum support threshold up to 3 selects most of the rules between confidence 60 to 80, with low loss of rules with confidence between 80 - 100. Rules with low confidence tend to disappear with a high minimum support threshold, indicating that $sup_{min} \geq 3$ is a suitable threshold for this dataset.

Figure 4b shows the minimum confidence threshold for *variation-based* discretization. Rules concentrate between support 0 and 3 for all minimum confidences evaluated. For rules with support above 2, the volume drop begins at $conf_{min} = 50$ and becomes significant only for $conf_{min} = 75$, indicating a great resilience of these rules to the application of a confidence cutoff. Given that a significant reduction in the volume of rules with high support occurs after $conf_{min} = 50$, a confidence cutoff in this range is indicated for the *variation-based* discretization applied to our dataset.

The maximum support obtained by eTRUMiner for our data is 9.32, with up to hundreds of thousands of rules according to discretization method. For the discretization methods we evaluated, the *variation-based* led to the best results, with lower volume of rules generated, rules with the highest support value, and good resilience to minimum support and confidence cuts. The SAX discretization also had good results, but since it tends to smooth missing observations it may not be a suitable method for datasets with missing values.

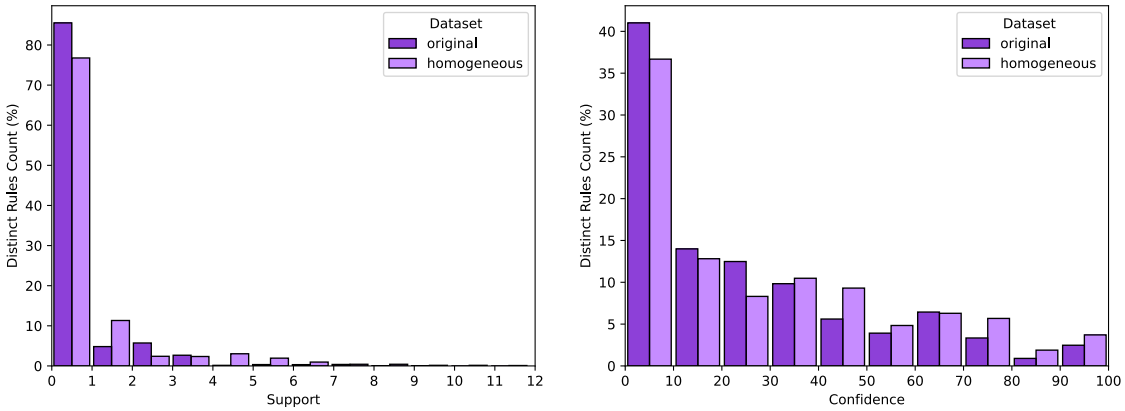
5.2. Missing Values Evaluation

For the analysis of missing values, since the dataset already has missing observations and missing variables, we selected the complete time series to create a new "homogeneous" dataset composed of 116 series with the same four variables from 1996 to 2019. We compared the homogeneous dataset with the original one regarding resulting rules and evaluated the impact of increasing percentages of missingness in the former. For these analyses, we employed the *variation-based* discretization method due to its resilience to missing observations, as shown in the previous experiments.

Although the number of time series is reduced to half compared to the original dataset, the number of total rules decreased only 3%, with 6,225 rules generated from the original data versus 6,039 rules extracted from the homogeneous dataset. However, the maximum support increased from 9.32 to 11.84, indicating that missing values tends to impact quality measures.

Figure 5 presents the distribution of rules in support and confidence ranges of both datasets. Despite the similar distribution behavior, the distribution of rules from homogeneous dataset reaches higher support and confidence values than those of the original dataset. Thus, complete time series data could result in more cohesive rules with better distribution of rules considering these quality measures.

Figure 5. Rules distribution comparative between datasets from *variation-based* discretization.



A more detailed analysis of missing values was performed with observations randomly removed from the homogeneous dataset, ranging from 1 to 15% of removal. The number of generated rules varied from 6,017 with 1% of removal to 4,852 on 15% of missing values, representing a reduction of 20% in the volume of rules. It is noteworthy that no new rules were extracted from the subsets with missing values, due to the choice of the discretization method. The missing values had a negative impact mainly on the quality measurements, with a mean support variation of up to 55% considering coincident rules extracted from the homogeneous dataset and the subset with 15% missing values.

Minimum support and minimum confidence thresholds were applied to help in the analysis of relevant rules. Although confidence thresholds have a mild effect on rules volume reduction, it helps select rules with lower variation on support when compared to their counterparts from the homogeneous dataset. For instance, in the 15% missing values subset, rules mean support dispersion is under 40%, but with a higher cut as minimum support ($sup_{min} = 4$) and minimum confidence $conf_{min} = 50$, 34 relevant, coincident rules were extracted with low variation in quality measures, showing that eTRUMiner preserves relevant rules while being capable of handling missing data.

5.3. Semantic Evaluation

In this section, we discuss some of the relevant rules eTRUMiner extracted from the original dataset regarding their meaning, time of occurrence, and location. eTRUMiner discovered the rule $([GDP, D] \Rightarrow [IMP, I][EXP, I], \Delta t = 5)$, $sup = 3.11, conf = 88.15$, indicating that 5 years after a decrease in GDP a rise in country's imports and exports is often observed. This rule is verified in Japan, Canada and Russia starting in 1998, during the Dotcom bubble crisis (1996-2004), representing an expected behavior after the crisis, as economies undergo restructuring.

During 2005 to 2012, we observe mostly an economic growth season, with relevant rules exhibiting increasing patterns. The strongest rule is $([IMP, I] \Rightarrow [GDP, I], \Delta t = 0)$, $sup = 18.84, conf = 92.62$, indicating that a rise in import is usually seen with a rise in GDP. A similar strong rule has the same antecedent and consequent, but with a temporal feature of one year. This period also covers the Great

Recession, generating rules such as $([IMP, D][GDP, D] \Rightarrow [EXP, D], \Delta t = 0)$, $sup = 3.48, conf = 94.78$, a behavior of economic recession detected in USA, Germany, and Brazil starting in 2009, as expected.

In the following period, 2013 to 2019, the rules still indicate an increase in economic indexes, but they are more dispersed. This characteristic can be an effect of growth in economic disparity, countries with specialized economies, while others are primarly agricultural. For example, the rule $([EXP, I][ECI, I] \Rightarrow [GDP, I], \Delta t = 2)$, $sup = 3.16, conf = 70$ is verified in China in 2014, Japan in 2016 and Canada in the next year, all countries with a high-specialized economy and comprising the largest economies in the world.

6. Conclusion

This paper introduced eTRUMiner, an algorithm capable of mining multivariate temporal rules from heterogeneous time series datasets with missing observations and missing variables. The algorithm verifies the relationships between several variables for different time intervals and can report all occurrences of each rule in individual series. Related algorithms do not deal with problems existing in real data sets such as missing observations, missing variables, and heterogeneous duration across series variables. The rules can be returned in short and extended formats and are composed of two or more variables.

eTRUMiner was applied and evaluated on macroeconomic multivariate time series of 232 countries from 1996 to 2020. Distinct discretization methods were analyzed to determine the impact of missing data. The *variation-based* presented the highest maximum support, the best distribution of support and confidence in distinct rules count, and generated the rules with smallest dispersion in missing data scenario.

The application of eTRUMiner on a homogeneous and complete dataset produces a more concise set of rules, with higher support. Even so, our algorithm is able to mine temporal rules from heterogeneous and incomplete time series with acceptable impact on the quality of returned rules. For the economic analysis, the rules extracted match the expected behaviors.

Future research could focus on incorporating additional evaluation metrics into eTRUMiner, offering new perspectives on the rules extracted. Given eTRUMiner's versatility over discretization methods and evaluation measurements, its application to new datasets is both feasible and encouraged. Moreover, because eTRUMiner supports variable-specific discretization, exploring mixed-method discretization across variables in rules mining is another promising direction.

References

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216.

Amaral, T. and Sousa, E. (2019). Trier: A fast and scalable method for mining temporal exception rules. In *XXXIV Simpósio Brasileiro de Banco de Dados*, pages 1–12. SBC.

Das, G., Lin, K.-I., Mannila, H., Renganathan, G., and Smyth, P. (1998). Rule discovery from time series. In *Proceedings of the 4th ACM KDD*, KDD'98, pages 16–22.

de Oliveira, F., Costa, R., Goldschmidt, R., and Cavalcanti, M. C. (2017). Mineração de regras de associação multirrelação em grafos: Direcionando o processo de busca. In *XXXII Simpósio Brasileiro de Banco de Dados*, pages 270–275. SBC.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Harms, S. K. and Deogun, J. S. (2004). Sequential association rule mining with time lags. *Journal of Intelligent Information Systems*.

He, G., Dai, L., Yu, Z., and Chen, C. L. P. (2024). Gan-based temporal association rule mining on multivariate time series data. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5168–5180.

Ho, V. L., Ho, N., Pedersen, T. B., and Papapetrou, P. (2025). Efficient generalized temporal pattern mining in time series using mutual information. *IEEE Transactions on Knowledge and Data Engineering*, 37(4):1753–1771.

Karasawa, E. and Sousa, E. (2022). Truminer: Mineração de regras temporais em bases de séries multivariadas e heterogêneas. In *XXXVII Simpósio Brasileiro de Bancos de Dados*, pages 403–408. SBC.

Karasawa, E. G. and Sousa, E. P. M. (2023). Mining temporal rules from heterogeneous multivariate time series. *Journal of Information and Data Management*, 14(2).

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD*, DMKD '03, pages 2–11.

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 *2010 ACM Symposium on Applied Computing*, pages 900–905.

Schlüter, T. and Conrad, S. (2011). About the analysis of time series with temporal association rule mining. In *2011 IEEE Symposium on Computational Intelligence in Data Mining*, pages 325–332.

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.

Srivastava, T., Mullick, I., and Bedi, J. (2024). Association mining based deep learning approach for financial time-series forecasting. *Applied Soft Computing*, 155:111469.

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.