Timestamp interestingness measure for time series numerical association rule mining

1st Tilen Hliš
Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
tilen.hlis@um.si

2nd Iztok Fister
Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
iztok.fister@um.si

3rd Uroš Mlakar
Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
uros.mlakar@um.si

4th Iztok Fister Jr.
Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
iztok.fister1@um.si

***Abstract*—Nowadays, Numerical Association Rule Mining methods, where most of them are based on population-based bio-inspired algorithms, represent a fruitful direction for extracting knowledge from transaction databases. Recently, Numerical Association Rule Mining methods have been extended to mine time-series datasets. However, these modified approaches introduce both opportunities and challenges for time-series mining, allowing users to uncover interesting temporal segments within the data. One of the biggest difficulties, when tackling this problem with bio-inspired algorithms, is handling larger time intervals (e.g., hours, or days). To address this issue, we propose a novel interestingness measure designed to limit the scope of time segments. In addition, we present experiments and results that illustrate the potential by application of the new interestingness measure.**

***Index Terms*—Association Rule Mining, evolutionary algorithms, Numerical Association Rule Mining, time series datasets**

I. INTRODUCTION

Association Rule Mining (ARM) is a fundamental data mining method for discovering interesting relationships or associations between data items/attributes in large datasets. In the beginning, after it was introduced by Agrawal et al. [1], the ARM gained prominence through applications like market basket analysis, and it has been applied in domains ranging from retail to health informatics [2]. The ARM problem typically assumes a transactional database with a set of attributes called items, and a set of transactions where each transaction contains a subset of these items [3].

Mathematically speaking, an association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$ is the antecedent itemset and $Y \subseteq I$ is the consequent, with the constraint that $X \cap Y = \emptyset$ [3]. Such a rule suggests that, whenever all items in X appear in a transaction, the items in Y are likely to appear as well. There exist several traditional algorithms for solving ARM problem, e.g. Apriori [1], ECLAT [4], Fp-Growth [5].

The interestingness or quality of a candidate rule is, typically evaluated by metrics such as support and confidence, which are regarded as fundamental measures in ARM [3]. However, the traditional ARM algorithms were originally designed for boolean or categorical attributes, and, therefore, cannot handle continuous numerical attributes directly [6]. To extend traditional ARM algorithms for mining datasets with numeric attributes, Srikant and Agrawal introduced method for mining quantitative association rules in relational tables, which involves discretizing numerical values into intervals so they can be treated as categorical labels/classes [6]. This approach effectively laid the foundation for Numerical Association Rule Mining (NARM), which generalizes traditional ARM algorithms for handling with continuous attribute types.

Unlike traditional ARM algorithms, the NARM algorithms allow attributes in X or Y to be of either categorical $A^{(cat)}$ or numerical $A^{(num)}$ type. Formally, a numerical association rule still has the form $X \Rightarrow Y$, but any numeric attribute $A^{(num)}$ in the rule is represented by a value interval rather than a categorical value. That is, if the attribute is of a numerical type, it appears in the rule as a constraint $A^{(num)} \in [v_1, v_2]$ denoting a range (i.e., an interval) of values, whereas the categorical attribute is drawn as a label from a set of multiple categorical values $A^{(cat)} \in \{v_1, v_2, \dots v_m\}$, and m denotes the size of the categorical set [3].

Nowadays, tackling the NARM problem is mostly utilized using the bio-inspired algorithms (i.e., either Evolutionary Algorithms (EA) or Swarm Intelligence (SI) based algorithms) that are promising in exploring the large search spaces. Some of these approaches are the following: Pujitha et al. in [7] proposed a Quantum Swarm Evolutionary Algorithm (QSE-RM) for mining association rules in large-scale databases, which combines Quantum Evolutionary Algorithms (QEA) with Particle Swarm Optimization (PSO). By leveraging quantum bits to represent solutions, the method enhances population diversity and convergence efficiency compared to traditional Genetic Algorithms (GA) [8]. Mohanty and Champati [9] introduced the GA specifically designed for Fuzzy ARM (FARM). Their approach integrates fuzzy clustering with evolutionary search to optimize rule interestingness and reduce redundancy. Lastly, Fister et al. [10] developed an evolutionary rule-mining framework that maintains specific attributes in numerical datasets. They introduce a new interestingness measure that guides the evolutionary process to preserve key features during rule optimization, thus improving both the interpretability and relevance of the mined rules.

Many contemporary datasets from the real world are essentially time-series datasets, such as audit logs, sensor readings, maintenance reports, and similar records. Consequently, extracting insights or new knowledge from these data can positively impact decision-making in industrial settings, smart agriculture, and related domains. However, traditional NARM approaches are not equipped to mine time-series datasets. To address this limitation, recent papers of Fister et al. [11], [12] have introduced a new paradigm for time-series NARM. While early experiments have been promising, several challenges remain. One notable issue is that the discovered time segments are often overly broad, which can obscure the most interesting events within the data. To overcome this problem, we propose a novel interestingness measure aimed at shrinking the time segment, thereby yielding more informative, time-dependent numerical association rules.

The structure of the remainder of the paper is as follows: Section II outlines the Segmented Interval Time Series NARM framework, segment definitions, and summarization of support, confidence and related interestingness measures. Section III introduces the timestamp interestingness measure (TSM), detailing its formulation and incorporation into the fitness function. Section IV describes the experimental setup, including the particle swarm optimization algorithm and multi-objective fitness function, and evaluates the effect of TSM on rule quality and temporal compactness. Finally, Section V summarizes the findings and points of potential directions for future research.

II. TIME SERIES ASSOCIATION RULE MINING

In this paper, we deal with the Segmented Interval Time Series Numerical Association Rule Mining, which was presented initially in the paper [12]. According to the paper [11], the part sequence of the time series can also be called a segment, derived from the time series matrix Z defined as follows:

$$Z = \begin{pmatrix} ts_1^{(1)}, & \dots & ts_i^{(1)}, & \dots, & ts_j^{(1)} & \dots, & ts_T^{(1)}, \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ ts_1^{(k)}, & \dots & \mathbf{ts}_{t_s}^{(k)}, & \dots, & ts_j^{(k)}, & \dots & ts_T^{(k)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ ts_1^{(l)}, & \dots & ts_i^{(l)}, & \dots, & \mathbf{ts}_{t_e}^{(l)}, & \dots & ts_T^{(l)}, \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ ts_1^{(N)}, & \dots & ts_i^{(N)}, & \dots, & ts_j^{(N)} & \dots & ts_T^{(N)}. \end{pmatrix}, \quad (1)$$

where parameter $t \in [1, T]$ denotes the time series counter, and T the length of each episode, while each element of the time series denotes a set of features:

$$ts = \{F_1, F_2, \dots, F_M\}, \quad (2)$$

and the variable M denotes the total number of features, whose corresponding attributes are typically obtained from various sensors. The segment is determined with its starting point t_s (at the time, when the element $ts_i^{(k)}$ has broken) and ending point t_e (at the time, when the element $ts_j^{(l)}$ has broken). In general, the segment S presents the part of the time series Z within the time interval $[t_s, t_e]$, in other words:

$$S[t_s, t_e] = \left(ts_{t_s}, ts_{t_{s+1}}, \dots, ts_{t_e}\right). \quad (3)$$

Let us mention that the segment S is denoted with a border line in Eq. (1).

A. Interestingness measures

By solving the ARM, including the NARM problems, numerous interestingness measures have been proposed to quantify the strength or usefulness of a rule. In fact, the literature reports over fifty different interestingness metrics beyond the basic ones [3]. Nevertheless, support and confidence remain the two most important and widely used measures for evaluating the mined association rules [3].

The support of an association rule indicates how frequently that pattern occurs in the database. Formally, for the association rule $X \Rightarrow Y$ in the transaction database D, the support is defined as the fraction or percentage of transactions in D that contain both X and Y [3], in other words:

$$\mathit{support}(X \Rightarrow Y) = \frac{|X \cup Y|}{|D|}.$$

The higher support means the association rule $X \Rightarrow Y$ involves the more prevalent combination of items in the dataset.

The confidence of the association rule $X \Rightarrow Y$ reflects the conditional probability of finding Y in a transaction given that the transaction contains X [3]. Equivalently, it measures the proportion of transactions containing X in which Y also appears:

$$\mathit{confidence}(X \Rightarrow Y) = \frac{\mathit{support}(X \cup Y)}{\mathit{support}(X)}.$$

An association rule with confidence $= 80\%$ means that 80% of the transactions that contain X also contain Y.

Support and confidence are typically used together to filter and select the association rule $X \Rightarrow Y$ as follows: a *frequent rule* must exceed the minimum support threshold $\mathit{support}(X \Rightarrow Y) \geq S_{\min}$, ensuring it is statistically significant in the data, and a *strong rule* must exceed a minimum confidence threshold $\mathit{confidence}(X \Rightarrow Y) \geq C_{\min}$, ensuring it is sufficiently predictive [3]. By focusing on rules that meet both criteria, analysts can identify meaningful patterns even in large numerical datasets using NARM.

III. PROPOSED INTERESTINGNESS MEASURE

Purpose: A timestamp measure is devoted for the optimization of time-series segments within time-series:

$$Z = ts_0, \ldots, ts_{t_s}, \ldots, ts_{t_e} \ldots, ts_T,$$

where the time-series segment is defined with its starting and ending timestamps:

$$S[t_s, t_e] = ts_{t_s}, \ldots, ts_{t_e},$$

and T defines the lengths of the time-sequence.

Definition: The timestamp metric is mathematically defined as (Fig. 1):

$$TSM = 1 - \frac{t_e - t_s}{t_T - t_0}, \qquad (4)$$

where t_s and t_e denotes the starting and ending timestamps of the defined time-sequences segment, while t_0 and t_T designate the starting and ending timestamps of the definite time-series sequence.

Timestamp's logic: The logic behind a definition of the proposed timestamp metric is as follows: The timestamp is defined as the number of seconds taken from the particular time start. On personal computers, the starting time is date 1.1.1970, which means that the starting point for calculating time is the mentioned date in each personal computer. The timestamp metric is defined as the ratio between the duration of the time segment and the duration of the time-series. The ratio is subtracted from the one, which means that the TSM metric prefers association rules with the shorter time-series segments within the time-series.

In practical settings, the long segments often correspond to daily-, weekly-, or even monthly-level intervals, whereas the short segments reflect much narrower time windows, typically of duration of one or two hours. When association rules are extracted from long segments, they usually describe broader, more general temporal patterns characteristic of the selected day or period. In contrast, rules obtained from short segments tend to capture specific, time-localized events (e.g., rapid changes in soil moisture after rainfall or drying effects due to sunlight). The TSM metric is designed precisely to favor these short informative intervals by assigning higher scores to rules derived from compact temporal segments.

Usage: The timestamp metric is used as an additional term in a fitness function that is applied for evaluating the quality of the mined association rule. The fitness function is defined as a linear combination of more metrics, where each term is weighted using the predefined weight.

IV. EXPERIMENTS AND RESULTS

The goal of the experimental work was twofold. First, we aimed to examine the influence of the newly proposed TSM on the quality of the mined association rules. Primarily, the purpose of the TSM is to evaluate the temporal consistency of rule activation across the time segments, thereby enhancing the interpretability and robustness of time series association rules. Additionally, we sought to determine whether the inclusion of TSM in the fitness function contributes to generating shorter and more temporally compact rules, i.e., rules describing events that occur within shorter time spans.

A. Experimental Setup

To mine time series association rules, we employed the Particle Swarm Optimization (PSO) bio-inspired algorithm [13]. The PSO was selected due to its demonstrated effectiveness in discovering association rules in various domains, as evidenced by numerous recent studies where it was applied either in its canonical form or as part of hybrid metaheuristic frameworks [14]. The algorithm maintains a population of candidate solutions, that iteratively adjust their positions in the search space according to both their own historical best positions and the global best position identified by the swarm. This mechanism allows PSO to efficiently explore the solution space while maintaining a balance between exploration and exploitation. The experiments focused on PSO, as the main objective was to evaluate the advantages of the proposed TSM measure, and extending the analysis to additional algorithms was beyond the intended scope of this study.

All the experiments were conducted using the *Niapy* library [15], a well-established open-source framework for metaheuristic optimization. The most important control parameters of PSO were set to their default values as defined in the library. Specifically, the population size was set to $Np = 50$, while the maximum number of fitness function evaluations was fixed at $MAX\_FE = 1000$. These parameter values were chosen empirically, based on preliminary calibration runs that indicated a satisfactory trade-off between search capability and computational cost. The inertia weight and acceleration coefficients were kept at their default values in the Niapy library.

B. Fitness Function Design

To guide the optimization process toward high-quality and temporally meaningful rules, a weighted single-objective fitness function was designed as follows:

$$f(\Gamma(\mathbf{x}_i)) = \frac{\alpha \times support(\Gamma(\mathbf{x}_i)) + \beta \times confidence(\Gamma(\mathbf{x}_i)) + \gamma \times TSM(\Gamma(\mathbf{x}_i))}{\alpha + \beta + \gamma} \qquad (5)$$

where $\Gamma(\mathbf{x}_i) \mapsto X \Rightarrow Y$ denotes a genotype/phenotype mapping that serves for transforming the representation of the individual solution in the genotype search space to the corresponding association rule in the phenotype problem space, α, β, and γ represent the relative importance (i.e., weights) of the particular time-series NARM interestingness measures, either $support(X \Rightarrow Y)$, $confidence(X \Rightarrow Y)$, and $TSM(X \Rightarrow Y)$, respectively. By adjusting these weights, different optimization scenarios were evaluated to investigate the contribution of the TSM measure. This formulation ensures that the resulting rules are not only statistically significant (high support and confidence) but also temporally stable according to the proposed TSM measure.

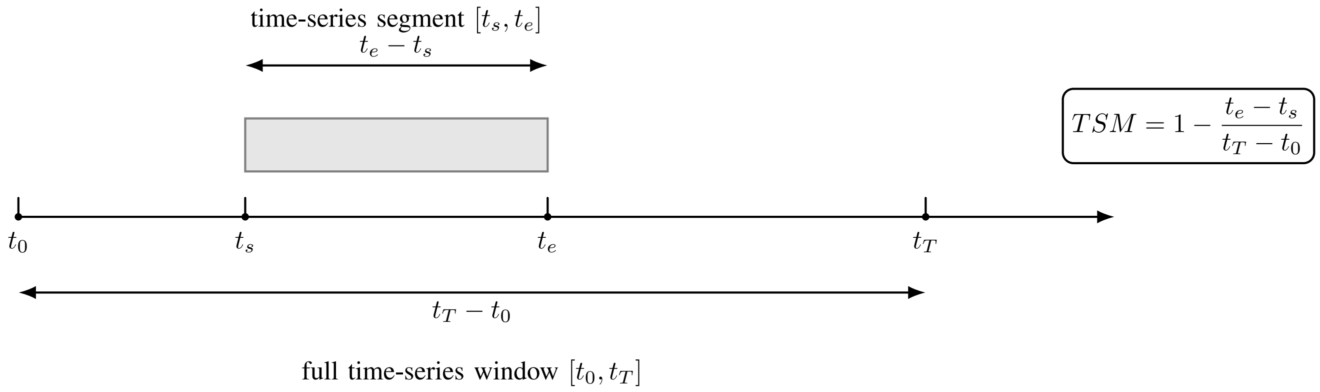

Fig. 1. Timestamp metric depiction.

TABLE I
AVERAGE RESULTS OF 30 INDEPENDENT PSO RUNS FOR DIFFERENT FITNESS WEIGHT CONFIGURATIONS.

Algorithm	Fitness	Support	Confidence	TSM	Antecedent_length	Consequent_length	TimeSpan (s)
$\text{PSO}_{\alpha=0.05\_\beta=0.05\_\gamma=0.9}$	**0.204 ± 0.031**	0.101 ± 0.046	0.344 ± 0.084	**0.656 ± 0.102**	**1.034 ± 0.033**	1.070 ± 0.023	**504785.175 ± 150282.015**
$\text{PSO}_{\alpha=0.25\_\beta=0.25\_\gamma=0.5}$	0.150 ± 0.018	0.106 ± 0.044	**0.388 ± 0.083**	0.650 ± 0.096	1.035 ± 0.030	1.072 ± 0.029	513420.939 ± 140489.773
$\text{PSO}_{\alpha=0.45\_\beta=0.45\_\gamma=0.1}$	0.093 ± 0.020	**0.108 ± 0.054**	0.374 ± 0.096	0.607 ± 0.128	1.045 ± 0.044	**1.070 ± 0.019**	576623.066 ± 187392.719
$\text{PSO}_{\alpha=0.5\_\beta=0.5}$	0.113 ± 0.024	0.091 ± 0.032	0.360 ± 0.076	0.651 ± 0.080	1.110 ± 0.108	1.176 ± 0.054	512635.769 ± 117971.617
$\text{PSO}_{\alpha=0.3\_\beta=0.7}$	0.138 ± 0.030	0.097 ± 0.032	0.353 ± 0.077	0.638 ± 0.076	1.083 ± 0.078	1.204 ± 0.117	531979.903 ± 112063.572

C. Experimental Procedure

Each experimental run of the PSO algorithm produced a set of time-segmented association rules for the selected dataset. Given the stochastic nature of the PSO search process, the results reported in this study represent the average metric values over 30 independent runs.

The following interestingness measures were computed for each mined rule:

- **Support** - proportion of the time segments where the rule holds true.
- **Confidence** - the conditional probability that the consequent occurs given the antecedent.
- **Antecedent_length** and **Consequent_length** - the number of items on each side of the rule.
- **TSM** - the newly proposed Timestamp Interestingness Measure.
- **TimeSpan** - the total duration of the rule's activation window, expressed in seconds.

D. Evaluation and Expected Outcomes

By systematically varying the weights α, β, and γ in the fitness function, we were able to assess how strongly the inclusion of TSM influenced both the *quality* and *temporal compactness* of the mined rules. We hypothesized that the higher values of γ (i.e., greater emphasis on TSM) will result in rules with the shorter activation spans and more consistent temporal behavior, without significantly degrading the classical association rule metrics such as support and confidence.

The resulting data enable a detailed comparison of rule sets obtained under different weighting configurations. Statistical analyses (e.g., mean and Standard Deviation) were performed to evaluate whether the observed differences are systematic and meaningful, thus validating the usefulness of incorporating the TSM measure into the optimization process.

E. Software stack

The experiments were performed using the open-source NiaARMTS framework [1] for time-series numerical association rule mining. NiaARMTS is an extension of the original NiaARM framework [2], providing additional support for extracting rules from time-series data. Although NiaARMTS is a standalone framework, it leverages the NiaPy library [3] for its metaheuristic search algorithms. NiaPy is a micro-framework designed to facilitate rapid prototyping and application of nature-inspired optimization algorithms without needing to implement them from scratch. The implementation of the proposed timestamp interestingness measure was integrated into NiaARMTS's fitness function.

F. Datasets

For the experimental evaluation, we used datasets comprising sensor measurements of the Aloe vera plant health [4]. The data were collected in September 2024 in Slovenia using a Lilygo T-CALL ESP32 microcontroller [11]. The microcontroller was equipped with three sensors: an Adafruit BH1750 light-intensity sensor, a DHT22 air-temperature and humidity sensor, and a soil-moisture hygrometer. Measurements from all sensors were sampled every 5 seconds and transmitted to a cloud platform for storage and processing.

[1] https://github.com/firefly-cpp/NiaARMTS
[2] https://github.com/firefly-cpp/NiaARM
[3] https://github.com/NiaOrg/NiaPy
[4] https://github.com/firefly-cpp/smart-agriculture-datasets

G. Results

The results of the conducted experiments are summarized in Table I. Each configuration of the PSO algorithm corresponds to a distinct weighting scheme in the fitness function, where α, β, and γ represent the weights of the corresponding terms of the fitness function as defined in Eq. (5). For comparative purposes, additional experiments were performed without the TSM measure as well (the lower part of the Table I).

From the results in the Table I, it can be observed that the inclusion of the TSM measure in the fitness function definition led to noticeable improvements in the overall fitness value and the temporal stability of the generated rules. The configuration with the reinforced value of the TSM weight (i.e., $\gamma = 0.9$) achieved the best average fitness (0.204 ± 0.031), accompanied by the increasing the value to **TSM** $= 0.656 \pm 0.102$ in the Table I. This confirms that emphasizing the temporal stability term effectively guides the PSO algorithm towards rules that are more consistent in time.

Interestingly, the configurations with reinforced value of TSM weights also tend to produce rules with slightly shorter antecedent and consequent, suggesting that the additional temporal constraint encourages the formation of simpler rule structures. Moreover, the **TimeSpan** values in the Table I show a clear tendency to decrease when the **TSM** term is dominant, indicating that the rules cover shorter time intervals and are thus more temporally compact.

In contrast, the control configurations without the TSM measure (by $PSO_{\alpha=0.5\_\beta=0.5}$ in the Table I and by $PSO_{\alpha=0.3\_\beta=0.7}$ in the Table I) achieved lower overall fitness values and produced rules with longer antecedents and consequents. These results justify the hypothesis that the proposed TSM interestingness measure contributes to improving both the interpretability and the temporal focus of the mined association rules.

Overall, the experimental findings demonstrate that integrating the TSM measure into the optimization process can significantly affect the structure and quality of the resulting time series association rules. The metric not only enhances temporal stability but also promotes the generation of shorter, more meaningful association rules without a substantial decrease in the traditional association rule measures such as support and confidence.

V. CONCLUSION

In this work, we proposed and evaluated the integration of the TSM into the fitness function for time series association rule mining. The TSM was designed to measure the temporal consistency of rule activation, encouraging the discovery of shorter rules that remain stable across time segments. The experimental results demonstrated that incorporating the TSM interestingness measure into the optimization process using the PSO algorithm leads to notable improvements in both the quality and interpretability of the mined rules. The configurations emphasizing the TSM achieved higher fitness values, improved temporal stability, and produced rules that were generally shorter and more temporally compact. The obtained results confirm that the TSM interestingness measure successfully guides the search toward more meaningful, shorter, and consistent association rules without a significant decrease in traditional ARM interestingness measures (i.e. support and confidence).

The major contribution of this work was to illustrate the benefits and practical implications of the proposed TSM measure within the time-series NARM framework. Due to the need to maintain a focused methodological scope, the evaluation was limited to a single representative optimization algorithm. Future work will extend the provided analysis by embedding TSM within additional metaheuristic optimization algorithms, to assess the robustness and generality of the measure across different search paradigms. Furthermore, the experimental scope was limited to comparisons between TSM-weighted and non-TSM configurations to keep the contribution focused and within the page constraints. Future work will also incorporate additional temporal baselines (e.g., fixed window segmentation or alternative temporal measures) and will extend the evaluation to multiple time-series datasets to further assess the generality of the proposed approach.

ACKNOWLEDGMENT

The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057 and P2-0041).

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work the authors used language tools such as Lumo (the AI assistant from Proton), Grammarly in order to improve the article's readability. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, no. 2, p. 207–216, Jun. 1993. [Online]. Available: https://doi.org/10.1145/170036.170072

[2] W. Altaf, M. Shahbaz, and A. Guergachi, "Applications of association rule mining in health informatics: a survey," *Artificial Intelligence Review*, vol. 47, no. 3, pp. 313–340, 2017.

[3] M. Kaushik, R. Sharma, I. Fister Jr, and D. Draheim, "Numerical association rule mining: a systematic literature review," *arXiv preprint arXiv:2307.00662*, 2023.

[4] M. J. Zaki, "Scalable algorithms for association mining," *IEEE transactions on knowledge and data engineering*, vol. 12, no. 3, pp. 372–390, 2002.

[5] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM sigmod record*, vol. 29, no. 2, pp. 1–12, 2000.

[6] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," in *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, 1996, pp. 1–12.

[7] A. S. Pujitha, S. G. M. Subhani, M. M. A. K. L. Mounika, and M. A. Kumar, "A quantum swarm evolutionary algorithm for mining association rules in large databases," *International Journal of Computer Networks and Wireless Communications*, vol. 13, no. 4, 2023.

[8] M. Mitchell, *An Introduction to Genetic Algorithms*. The MIT Press, 02 1996. [Online]. Available: https://doi.org/10.7551/mitpress/3927.001.0001

[9] B. Mohanty and S. L. Champati, “A novel approach to optimise fuzzy association rule by using evolutionary genetics algorithm,” *International Journal of Modelling, Identification and Control*, 2025. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJMOR.2025.147873

[10] I. Fister, D. Novak, A. Iglesias, and A. Gavez, “A new interestingness measure for maintaining specific attributes in numerical association rule mining,” in *2025 IEEE International Conference on Big Data*, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/11166644

[11] I. Fister Jr, D. Fister, I. Fister, V. Podgorelec, and S. Salcedo-Sanz, “Time series numerical association rule mining variants in smart agriculture,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 12, pp. 16 853–16 866, 2023.

[12] I. Fister Jr, S. Salcedo-Sanz, E. Alexandre-Cortizo, D. Novak, I. Fister, V. Podgorelec, and M. Gorenjak, “Toward explainable time-series numerical association rule mining: A case study in smart-agriculture,” *Mathematics*, vol. 13, no. 13, p. 2122, 2025.

[13] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 1942–1948.

[14] T. Su, H. Xu, and X. Zhou, “Particle swarm optimization-based association rule mining in big data environment,” *IEEE access*, vol. 7, pp. 161 008–161 016, 2019.

[15] G. Vrbančič, L. Brezočnik, U. Mlakar, D. Fister, and I. Fister, “Niapy: Python microframework for building nature-inspired algorithms,” *Journal of Open Source Software*, vol. 3, no. 23, p. 613, 2018.