University of Maribor

Faculty of Electrical Engineering and Computer Science

TIMESTAMP INTERESTINGNESS MEASURE FOR TIME SERIES NUMERICAL ASSOCIATION RULE MINING

IEEE 24th World Symposium on Applied Machine Intelligence and Informatics (SAMI 2026)

Authors:

Tilen Hliš (tilen.hlis@um.si), Iztok Fister (iztok.fister@proton.me), Uroš Mlakar (uros.mlakar@um.si) and Iztok Fister Jr. (iztok.fister1@um.si)

PRESENTATION AGENDA

1. Motivation and problem definition
2. Timestamp Interestingness Measure (TSM)
3. Fitness integration
4. Experimental setup
5. Results
6. Conclusion and future work

MOTIVATION

1. Many real datasets are time series (sensors, logs, monitoring):
   - time series appear everywhere: sensor readings, audit logs, industrial monitoring, smart agriculture.
2. Time-series NARM can discover **explainable temporal patterns:**
   - numerical association rule mining is attractive because it yields explainable rules instead of black-box predictions.
3. Challenge:
   - **time segments are often too broad.**

WHAT IS NARM?

- Extension of classical Association Rule Mining (**ARM**):
  - $X \Rightarrow Y, \quad A_{num} \in [v_1, v_2]$.
- Designed for **numerical data.**
- Conditions are **value intervals**, not single values.
- Example:
  - **Temperature ∈ [20, 25] ⇒ Humidity ∈ [60, 70].**

- **ARM:**
  - “If item A appears, then item B appears.”

- **NARM:**
  - “If attribute A is in this value range, then attribute B is in that value range.”

CORE PROBLEM

- Classical interestingness measures:
  - **support**, **confidence.**
    - These measures describe statistical strength.

- They evaluate *strength*, not *temporal focus.*

- Result:
  - rules may be correct but **temporally uninformative**‼️

CLASSICAL INTERESTINGNESS MEASURES

- Support:

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

- Confidence:

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

- Support → how **frequent** the rule is.
- Confidence → how **reliable** the rule is.

- No notion of **time compactness**‼️

OUR GOAL

1. Prefer rules from **shorter time segments.**

2. Keep classical quality (support/confidence).

3. Add a simple temporal term for optimization.

Our goal is **not to replace** support and confidence, because they still matter. The goal is to **add a temporal preference**: if two candidate rules are similar in support and confidence, we want the method to prefer the one that describes a narrower time window, because it is usually more actionable and easier to interpret‼

HOW WE MEASURE THE TEMPORAL LENGTH OF A RULE??

- A time series is a signal (temperature, humidity,...) measured over time, spanning from t_0 to t_T.
- A rule is not valid at all times, but only when its numerical conditions are satisfied.
- This defines an activation interval where the rule holds, from t_s to t_e.
- The temporal length of a rule is the duration of this activation interval.
- We define this duration as **TimeSpan.**
- A short TimeSpan indicates a focused, event-like rule.
- A long TimeSpan indicates a general, long-lasting behavior.

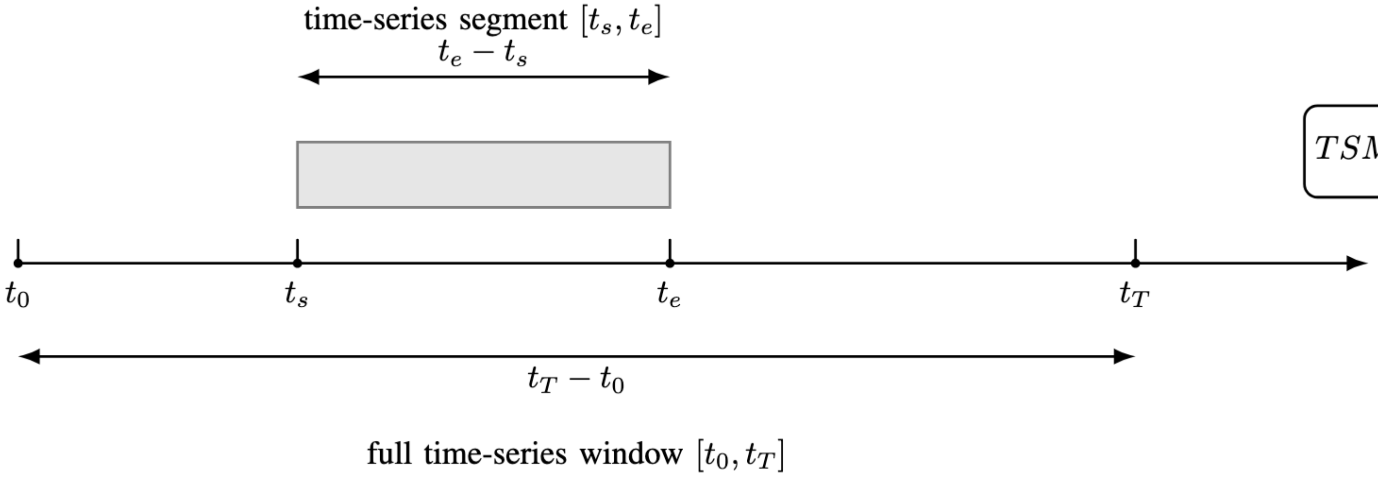
HOW WE MEASURE THE TEMPORAL LENGTH OF A RULE??

- Each rule has its own **activation interval in time.**

- Full time series:
  - **$[t_0 , t_T]$.**

- Rule is active only on:
  - **$[t_s , t_e]$.**

- Rule TimeSpan:
  - **TimeSpan = $t_e - t_s$.**



$$\boxed{TSM = 1 - \frac{t_e - t_s}{t_T - t_0}}$$

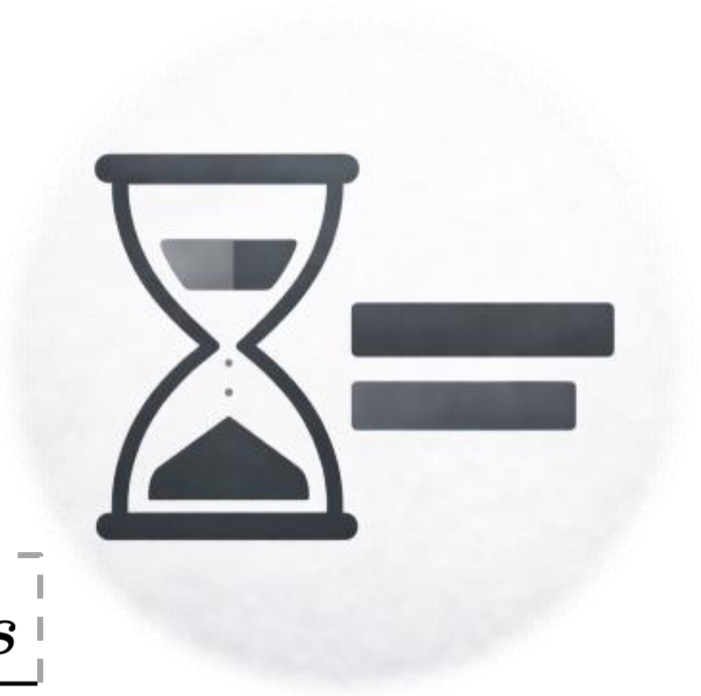
TIMESTAMP INTERESTINGNESS MEASURE (TSM)

- Measures **temporal compactness** of a rule.

- Compares rule activation length to full time series length.

- Penalizes long activation intervals.

- Rewards short, focused temporal patterns.



$$TSM = 1 - \frac{t_e - t_s}{t_T - t_0}$$

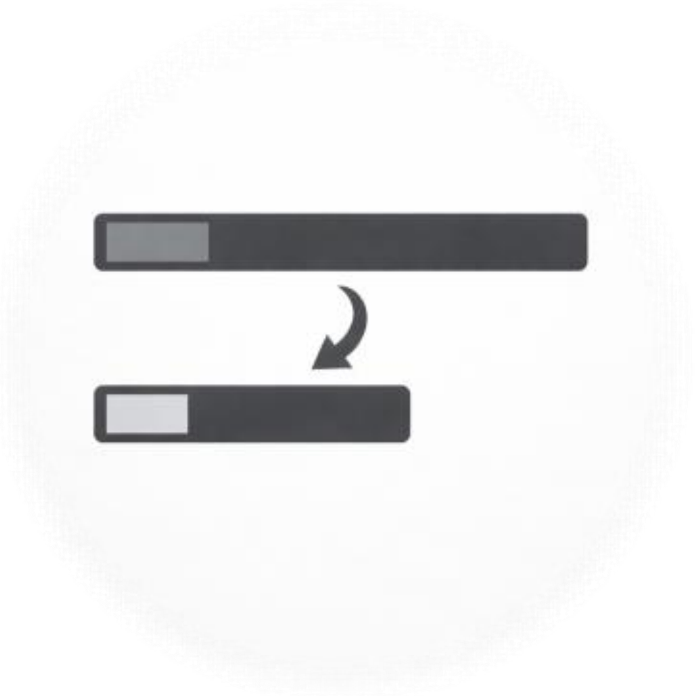
Where:

- t_e - t_s → rule TimeSpan
- t_T – t_0 → full time-series duration

TSM: BEHAVIOR

What does TSM actually do??

- Long activation interval → **low TSM.**
- Short activation interval → **high TSM.**
- Encourages **compact, time-localized rules.**



FITNESS FUNCTION: INTEGRATION

- The fitness function represents the **overall quality of a candidate rule**.
- It is the value that the **optimization algorithm tries to maximize**.
- A **higher** fitness value means a **better** and **more useful rule**.
- In our approach, fitness combines **statistical quality** and **temporal compactness**.

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

- **Support** measures how **often** the rule appears in the data.
- **Confidence** measures how **reliable** the implication is.

- TSM measures how **compact** the rule is in time.
- α, β, and γ control the **importance** of each component.
- Fitness can be directly optimized using metaheuristic algorithms such as PSO.

EXPERIMENTAL SETUP

- We **evaluate** the proposed **fitness** function using Particle Swarm Optimization (**PSO**).
- The goal is to analyze **how the TSM influences** the discovered rules.
- Different values of $\boldsymbol{\gamma}$ were used to control the importance of temporal compactness in fitness.
- The experiments were performed using our open-source **NiaARMTS** framework.

Dataset:

- Real-world smart agriculture dataset.
- Aloe Vera plant monitoring with multiple numerical sensor attributes.
- Sampling interval: 5 seconds.

RESULTS – EFFECT OF WEIGHTS ON RULE PROPERTIES

1. The best overall performance is achieved when **TSM dominates the fitness (γ = 0.9)**.
2. Higher γ leads to **higher TSM values** and **shorter TimeSpan**.
3. Rules become **temporally shorter and structurally simpler**.
4. Support and confidence remain in comparable ranges.

Weights (α, β, γ)	Fitness	Support	Confidence	TSM	Antecedent length	Consequent length	TimeSpan (days)
0.05, 0.05, **0.9**	**0.204**	0.101	0.344	**0.656**	**1.03**	1.07	**5.84**
0.25, 0.25, 0.5	0.150	0.106	**0.388**	0.650	1.03	1.07	5.94
0.45, 0.45, 0.1	0.093	**0.108**	0.374	0.607	1.05	**1.07**	6.67
0.5, 0.5, –	0.113	0.091	0.360	0.651	1.11	1.18	5.93
0.3, 0.7, –	0.138	0.097	0.353	0.638	1.08	1.20	6.16

DISCUSSION AND FUTURE WORK

Discussion:

- **TSM introduces an explicit temporal component** into numerical association rule evaluation.
- Emphasizing TSM leads to **higher TSM values and shorter TimeSpan** of rules.
- The **best fitness values are achieved when TSM is emphasized**.
- **Support and confidence remain stable**, so statistical quality is preserved.
- Rules become **more temporally compact and easier to interpret**.

Future work:

- Extend evaluation to **other optimization algorithms** beyond PSO.
- Validate the method on **additional time-series datasets**.
- Further analyze the effect of **different fitness weight configurations**.

THANK YOU FOR YOUR ATTENTION