universität wien

VU Business Intelligence II

Replication Study: Mining Behavioral Patterns for Conformance Diagnostics

Group Member:

- Khalifa Nikzad (12437813)
- Fatemehsadat Sepehrihosseini (12350114)
- Azizullah Massomy(12244794)
- Shogofa Nawrozy (12314943)
- Irdi Kuka (12407107)

Winter semester 2025

Paper Overview – Context

- Process mining compares real executions (logs) with process models
- Deviations are common in real-life processes
- The challenge:
  How can we detect and explain deviations in a meaningful way?

Paper Overview: Problem & Idea

- Problem:
  - Many common deviations
  - Hard to explain them clearly
- Authors’ idea:
  - Use **behavioral constraints**
  - Automatically discover patterns that explain deviations
- Key novelty (simple):
  - Not only *detect* deviations
  - Also *explain* them in a structured way

Paper Overview: How It Works

- Input:
  - Process model
  - Event log
- Steps:
  - Instantiate candidate constraints
  - Check which constraints hold in the model
  - Minimize constraints
  - Use them to explain deviations

What do we mean by a Constraint?

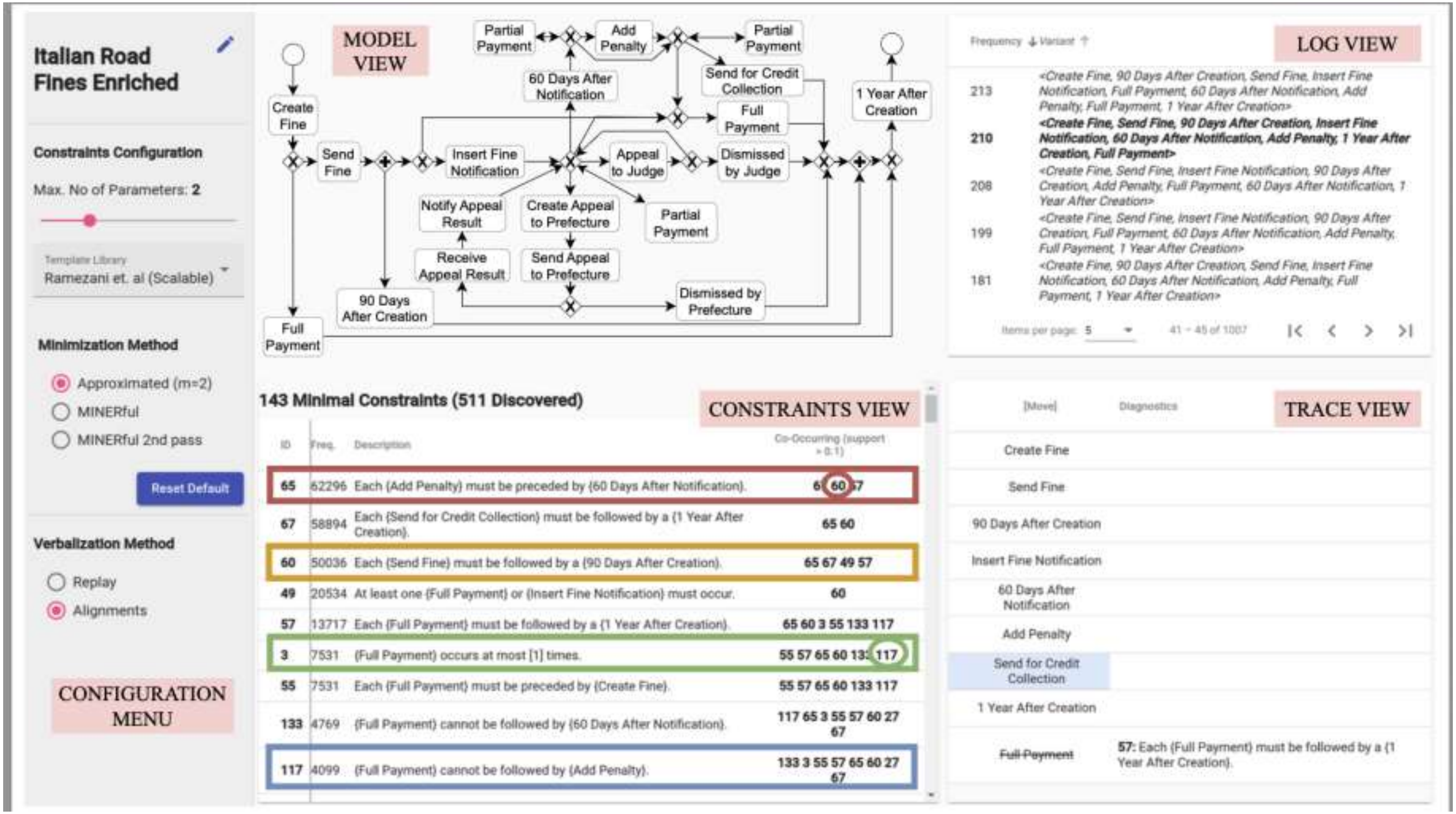
- A **constraint** is a **rule about how activities should happen** in a process.
- It describes **expected behavior**, not a single trace.
- Constraints are used to:
    - detect deviations
    - explain why a process execution is wrong

Simple example

- Activity A: Send fine
- Activity B: Pay fine

Constraint:
“If Send fine happens, then Pay fine should eventually happen.”

If this rule is broken → we found a **deviation**.



Italian Road Fines Enriched

Constraints Configuration

Max. No of Parameters: **2**

Template Library: Ramezani et. al (Scalable)

Minimization Method

● Approximated (m=2)
○ MINERful
○ MINERful 2nd pass

Reset Default

Verbalization Method

○ Replay
● Alignments

Frequency ↓	Variant ↑
213	*<Create Fine, 90 Days After Creation, Send Fine, Insert Fine Notification, Full Payment, 60 Days After Notification, Add Penalty, Full Payment, 1 Year After Creation>*
210	***<Create Fine, Send Fine, 90 Days After Creation, Insert Fine Notification, 60 Days After Notification, Add Penalty, 1 Year After Creation, Full Payment>***
208	*<Create Fine, Send Fine, Insert Fine Notification, 90 Days After Creation, Add Penalty, Full Payment, 60 Days After Notification, 1 Year After Creation>*
199	*<Create Fine, Send Fine, Insert Fine Notification, 90 Days After Creation, Full Payment, 60 Days After Notification, Add Penalty, Full Payment, 1 Year After Creation>*
181	*<Create Fine, 90 Days After Creation, Send Fine, Insert Fine Notification, 60 Days After Notification, Add Penalty, Full Payment, 1 Year After Creation>*

Items per page: 5 41 – 45 of 1007

143 Minimal Constraints (511 Discovered)

ID	Freq.	Description	Co-Occurring (support > 0.1)
65	62296	Each {Add Penalty} must be preceded by {60 Days After Notification}.	**67 60 57**
67	58894	Each {Send for Credit Collection} must be followed by a {1 Year After Creation}.	**65 60**
60	50036	Each {Send Fine} must be followed by a {90 Days After Creation}.	**65 67 49 57**
49	20534	At least one {Full Payment} or {Insert Fine Notification} must occur.	**60**
57	13717	Each {Full Payment} must be followed by a {1 Year After Creation}.	**65 60 3 55 133 117**
3	7531	{Full Payment} occurs at most [1] times.	**55 57 65 60 133 117**
55	7531	Each {Full Payment} must be preceded by {Create Fine}.	**55 57 65 60 133 117**
133	4769	{Full Payment} cannot be followed by {60 Days After Notification}.	**117 65 3 55 57 60 27 67**
117	4099	{Full Payment} cannot be followed by {Add Penalty}.	**133 3 55 57 65 60 27 67**

[Move]	Diagnostics
Create Fine	
Send Fine	
90 Days After Creation	
Insert Fine Notification	
60 Days After Notification	
Add Penalty	
Send for Credit Collection	
1 Year After Creation	
~~Full Payment~~	**57:** Each {Full Payment} must be followed by a {1 Year After Creation}.

Role of Constraints in the Evaluation

Table 4 — Performance

- Constraints are treated as **candidates**
- The system:
  - generates many possible rules
  - removes invalid and redundant ones
- Goal: measure **scalability and runtime**

Table 5 — Diagnostics

- Constraints are used as **explanations**
- The system checks:
  - which constraints are violated
  - how many deviations are explained
- Goal: measure **explanation quality**

Key idea

- Table 4 → How expensive is it to find rules?
- Table 5 → How useful are these rules to explain deviations?

Evaluation Setup in the Paper

- Two datasets:
  - Road Fines (RF)
  - BPI-15
- Different constraint sizes:
  - k = 2, 3, 4
- Two template sets:
  - ALL templates
  - Γ-invariant (SCALABLE) templates
- **Paper evaluates:**
  - Runtime & scalability (Table 4)
  - Diagnostic quality (Table 5)

Replication Setup & Experience

What the authors provided

- Executable Python implementation
- Event logs and constraint libraries in JSON format

How we replicated the experiments

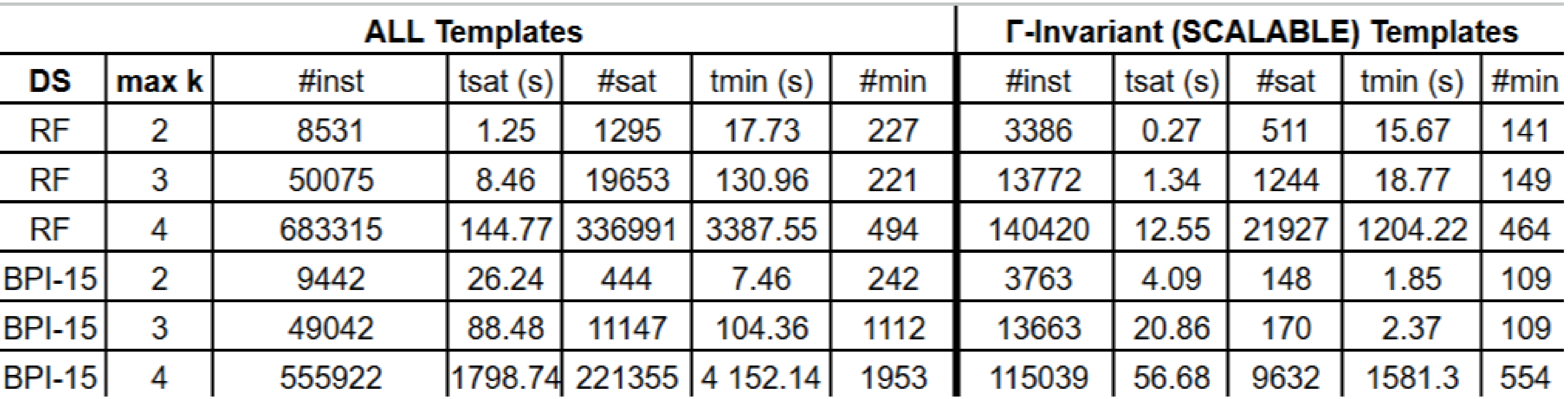
- Used the same datasets and template repositories
- Ran the experiments for k = 2, 3, and 4
- Collected both performance and diagnostic outputs

Main challenges

- Experiments become very slow for larger k
- High memory and computation cost
- Careful setup was needed to avoid re-running expensive steps

Replication Results (Table 4)

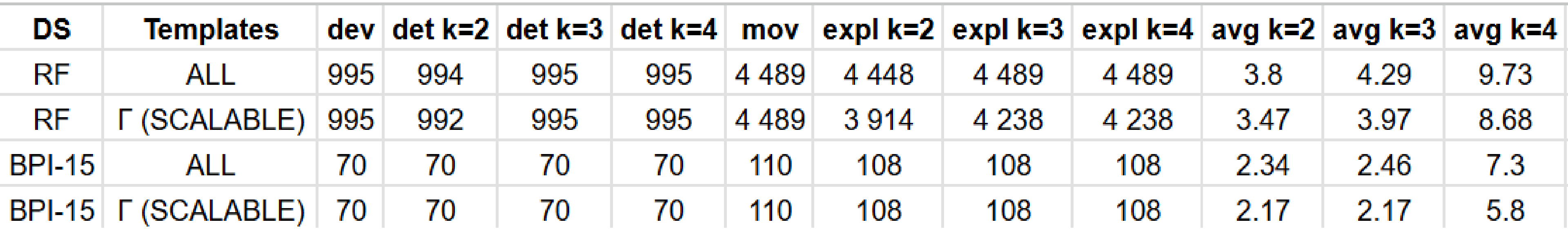
- As k increases:
  - Number of constraints increases rapidly
  - Runtime increases strongly
- Γ-invariant templates:
  - Much faster
  - Much fewer constraints



		ALL Templates					Γ-Invariant (SCALABLE) Templates				
DS	**max k**	#inst	tsat (s)	#sat	tmin (s)	#min	#inst	tsat (s)	#sat	tmin (s)	#min
RF	2	8531	1.25	1295	17.73	227	3386	0.27	511	15.67	141
RF	3	50075	8.46	19653	130.96	221	13772	1.34	1244	18.77	149
RF	4	683315	144.77	336991	3387.55	494	140420	12.55	21927	1204.22	464
BPI-15	2	9442	26.24	444	7.46	242	3763	4.09	148	1.85	109
BPI-15	3	49042	88.48	11147	104.36	1112	13663	20.86	170	2.37	109
BPI-15	4	555922	1798.74	221355	4 152.14	1953	115039	56.68	9632	1581.3	554

Replication Results (Table 5)

- Diagnostic behavior matches the paper
- More constraints → richer explanations
- Γ-invariant templates:
  - Slightly fewer explanations
  - Still good diagnostic quality



DS	Templates	dev	det k=2	det k=3	det k=4	mov	expl k=2	expl k=3	expl k=4	avg k=2	avg k=3	avg k=4
RF	ALL	995	994	995	995	4 489	4 448	4 489	4 489	3.8	4.29	9.73
RF	Γ (SCALABLE)	995	992	995	995	4 489	3 914	4 238	4 238	3.47	3.97	8.68
BPI-15	ALL	70	70	70	70	110	108	108	108	2.34	2.46	7.3
BPI-15	Γ (SCALABLE)	70	70	70	70	110	108	108	108	2.17	2.17	5.8

Additional investigation (beyond the paper)

- We intentionally kept the diagnostic verification step enabled during all runs
- This allowed us to measure **end-to-end execution cost**, not only discovery time
- We analyzed how this affects runtime across:
  - different k values
  - ALL vs Γ-invariant templates
- This highlights a practical trade-off between:
  - fast constraint discovery
  - full diagnostic execution

Conclusion & Limitations

Replication outcome

- We successfully replicated the quantitative evaluation (Tables 4 and 5)
- The main scalability and diagnostic trends reported in the paper were confirmed

Limitations and future work

- Experiments were run on different hardware
- Large k values lead to very long runtimes
- Future work could include:
  - applying the approach to new datasets
  - comparing with other conformance tools

Work Distribution

All members contributed to the project. We jointly suggested and selected the paper, read and discussed it, planned the replication approach, reviewed the results, and refined the final presentation. Primary responsibilities were:

- Azizullah Massomy– Coordination, environment setup, and troubleshooting replication issues.
- Fatemehsadat Sepehrihosseini – High-level paper summary and writing up the technical approach.
- Khalifa Nikzad – Running the replication, organizing outputs, and trying extra experiments for more insights.
- Irdi Kuka – Analyzing results and comparing them with the paper's reported findings.
- Shogofa Nawrozy – Final slide structure, assignment requirement checks, and final presentation polishing.
- Link to repository: GitHub