Prediction-based Resource Allocation

By: Sourabh Zanwar
Date: 28.07.2023
Supervised by: Marco Pegoraro

Agenda

- Introduction
- Motivation
- Proposed Method
    - Next activity and time prediction
    - Resource allocation
- Experiments
- Results
- Future Work & Limitations

Introduction & Motivation

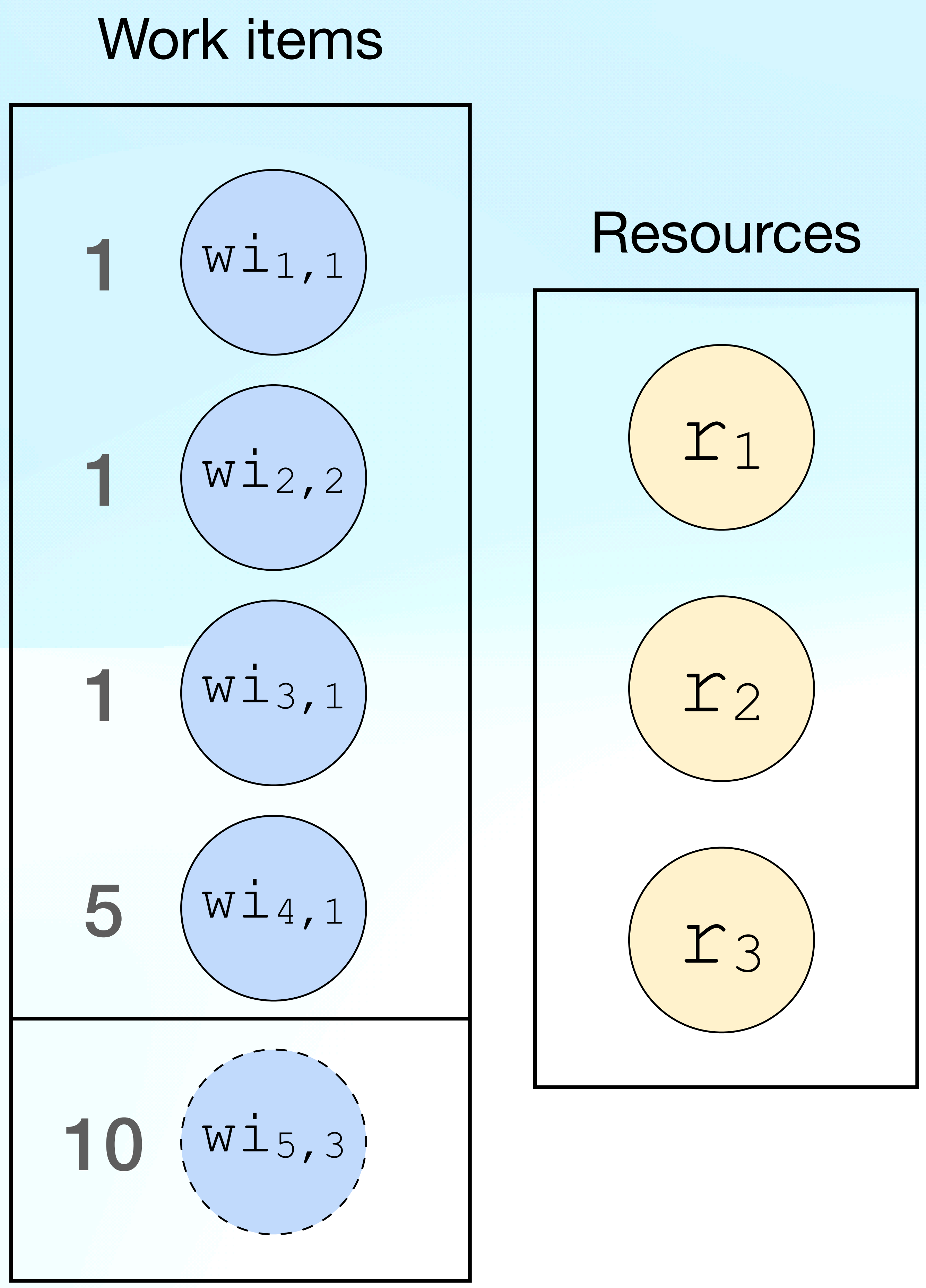
Introduction

- Predictive Business Process Monitoring and Management
    - Efficient scheduling of activities
    - Efficient allocation of resources
- Use Machine Learning to improve Business Processes
- Assessment of the paper Prediction-based resource allocation [1]

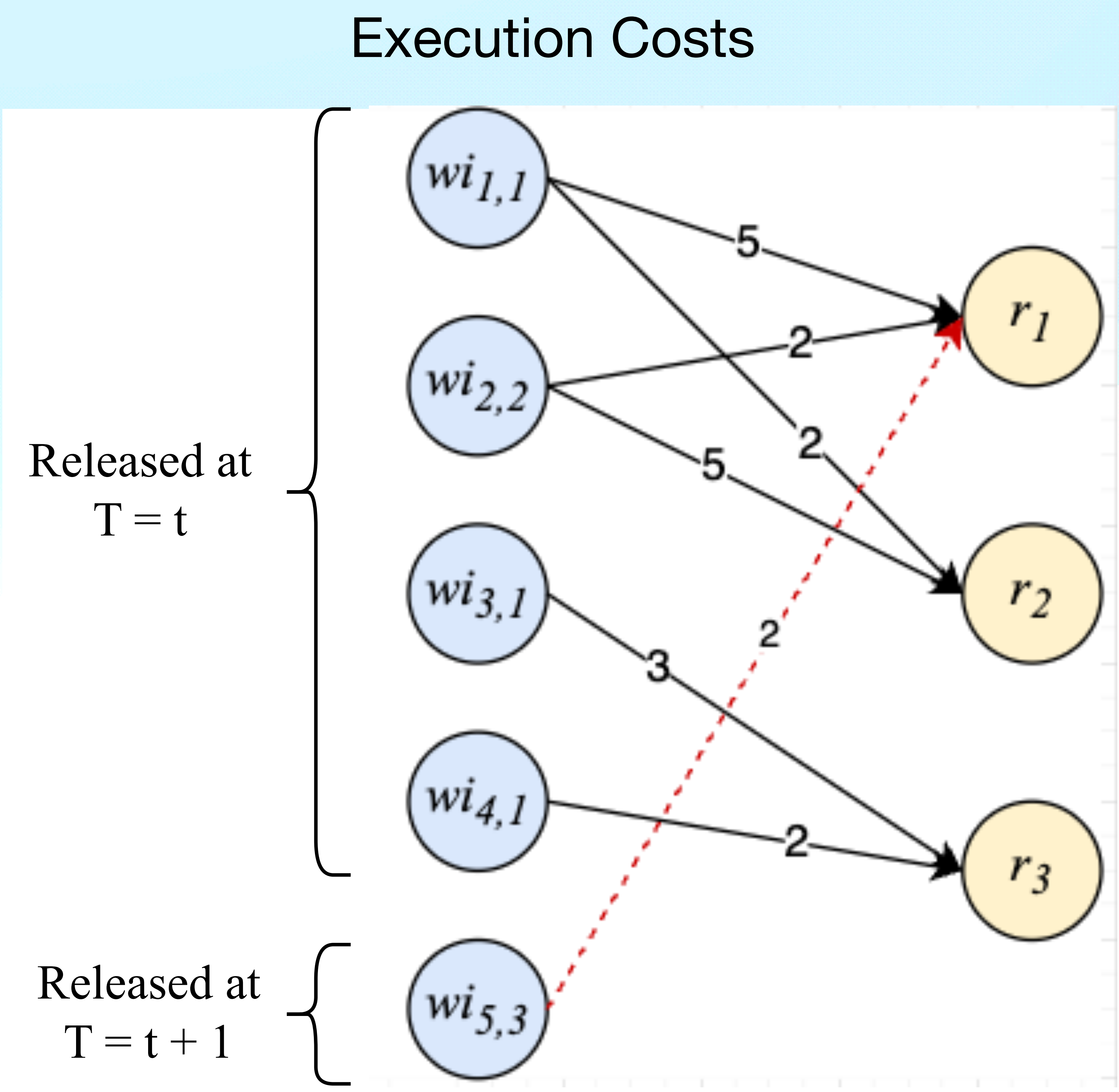
Motivation

- Resource allocation
  - Improved productivity
  - Reduced execution costs
  - Balanced resource usage
- Non-clairvoyant online-over time problem [2]
  - Effectively build prediction models
  - Dispatch resources efficiently

Example



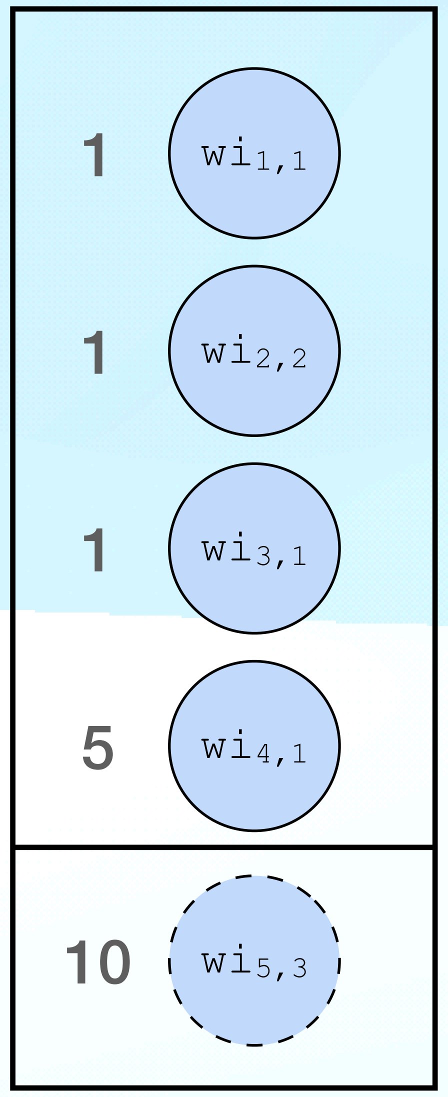
Work items
1
wi1,1
1
wi2,2
1
wi3,1
5
wi4,1
10
wi5,3
Resources
r1
r2
r3

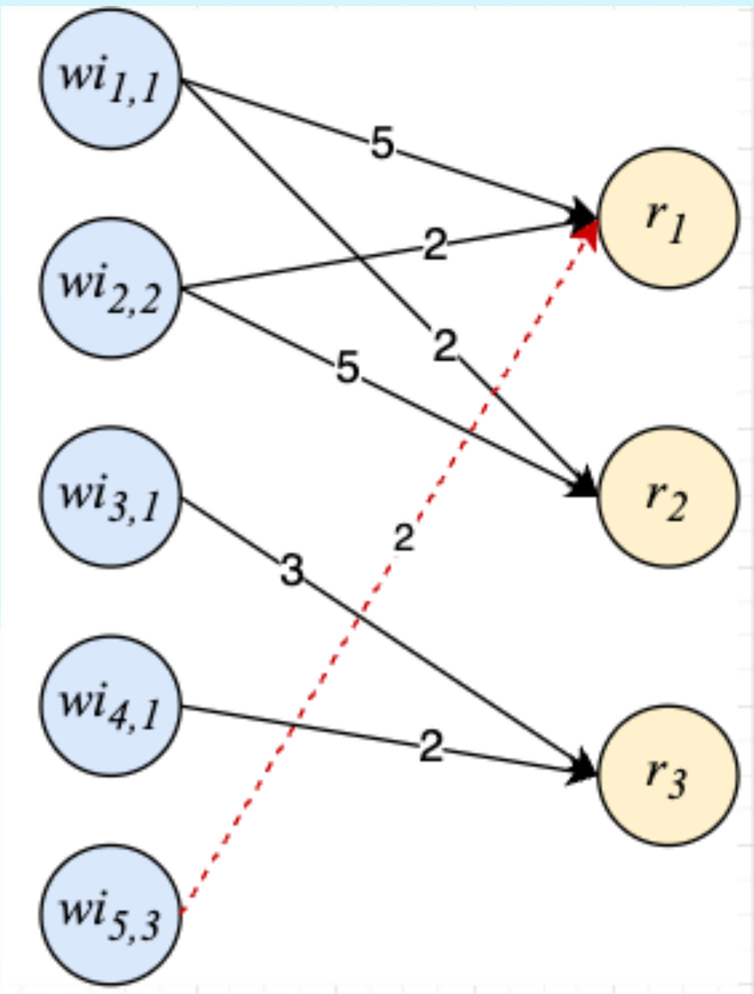


Execution Costs
Released at
T = t
Released at
T = t + 1
wi1,1
wi2,2
wi3,1
wi4,1
wi5,3
r1
r2
r3
5
2
2
5
3
2
2

Baseline

Work items





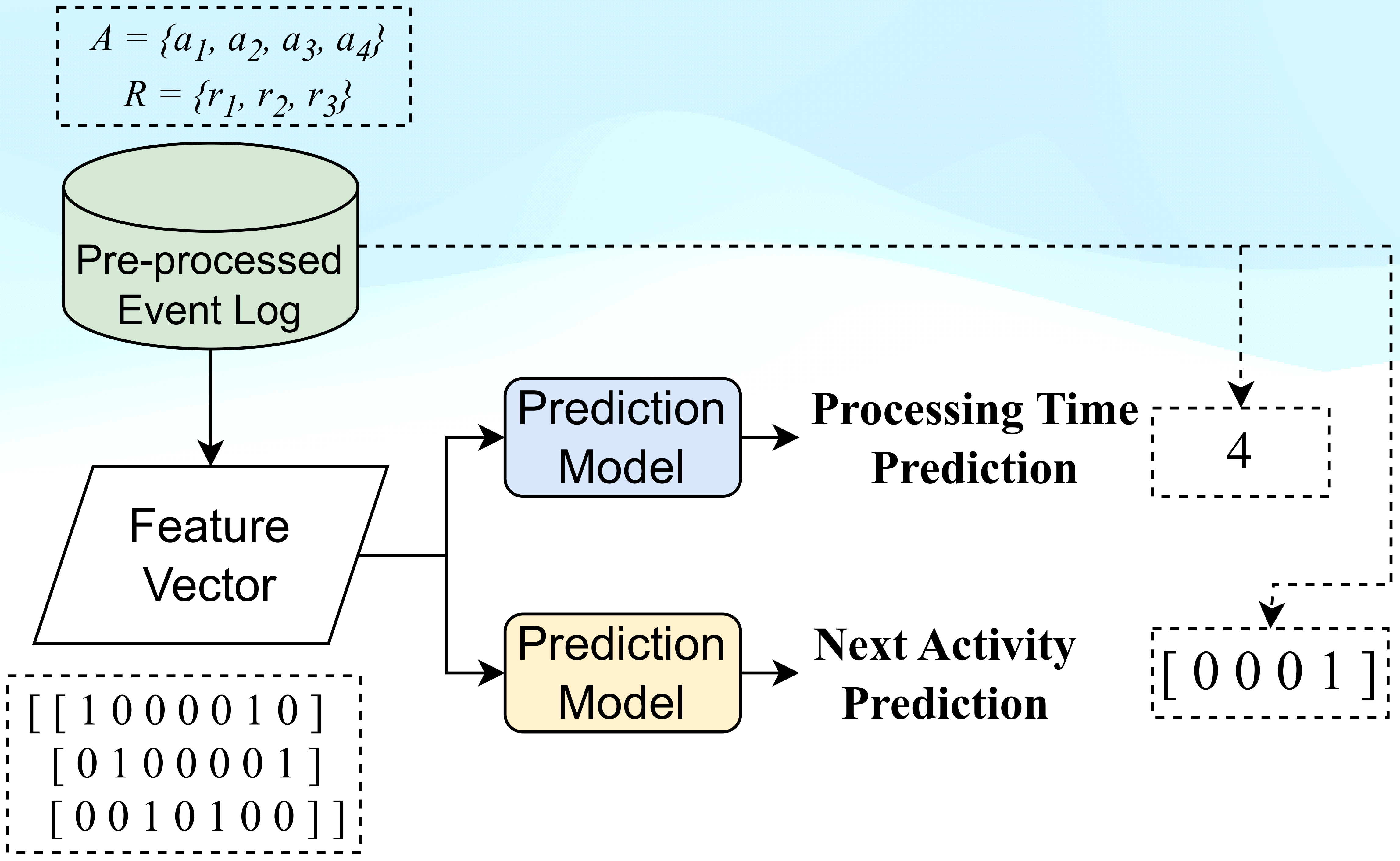
resource	t	t+1	t+2	t+3	t+4	t+5	t+6	$\sum c_i w_i$
r_1	$wi_{1,1}$					$wi_{5,1}$		65
r_2	$wi_{2,2}$							5
r_3	$wi_{4,1}$		$wi_{3,1}$					15

Table. Baseline Resource allocation

Proposed method

Next activity & Time prediction

- Use one-hot encoded resource and activities as input
- Processing time prediction: numerical value
- Next activity prediction: one-hot-encoded activity values



Resource allocation

Algorithm 1 Resource Scheduling algorithm

Input: $\hat{WI}, \hat{R}$
Output: Psuedo-Assignment $\hat{M}$
Produce source node s, sink node t

1

for node $wi_{i,k} \in \hat{WI}$ **do**
  add edge $(s, wi_{i,k}, (0, 1))$
end for

2

for node $r_j \in \hat{R}$ **do**
  add edge $(r_j, t, (0, 1))$
end for

3

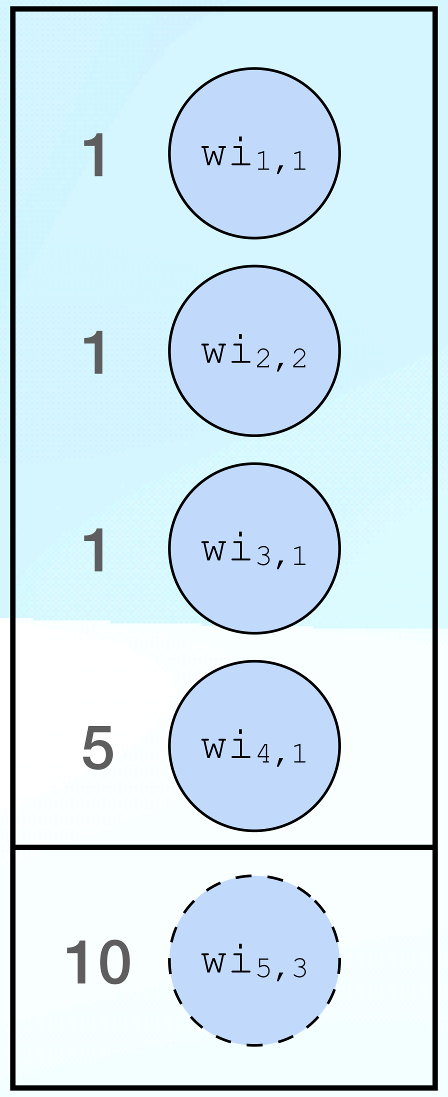
for node $wi_{i,k} \in \hat{WI}$ **do**
  for node $r_j \in \hat{R}$ **do**
    $c \leftarrow (p_{i,k,j} + max(ri_i, rr_j, 0))/w_i$
    add edge $(wi_{i,k}, r_j, (c, 1))$
  end for
end for
$M \leftarrow MinCostMaxFlow(s, t)$
return M

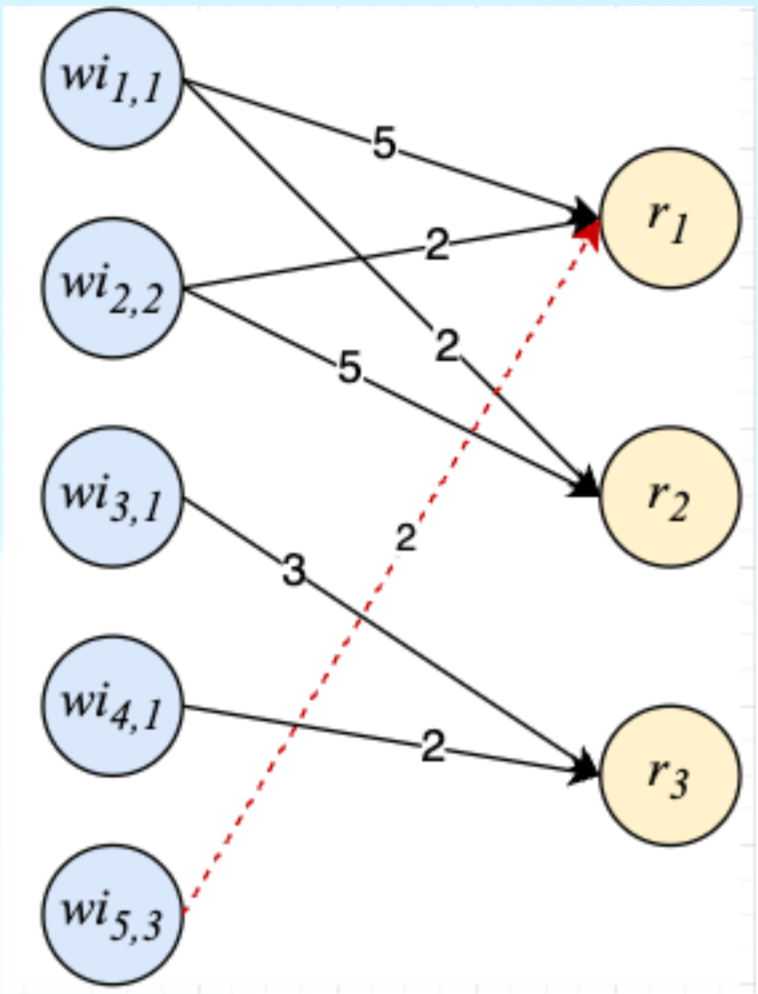
Cost

- $p_{i,k,j}$: processing time for work item $wi_{i,k}$ by resource r_j
- ri_i: remaining time for item i
- rr_j: remaining time for resource r_j to be ready
- w_i: weight of item i

Resource allocation

Work items





<table>
<tr><th>resource</th><th>t</th><th>t+1</th><th>t+2</th><th>t+3</th><th>t+4</th><th>$\sum c_i w_i$</th></tr>
<tr><td>r1</td><td></td><td colspan="2">$wi_{5,1}$</td><td colspan="2">$wi_{2,2}$</td><td></td></tr>
<tr><td>r2</td><td colspan="2">$wi_{1,1}$</td><td></td><td></td><td></td><td></td></tr>
<tr><td>r3</td><td colspan="2">$wi_{4,1}$</td><td colspan="3">$wi_{3,1}$</td><td></td></tr>
</table>

Experiments

Data

- BPIC’2012 Shared Task: Consumer Loan approvals process
- Filtering: events with valid resource and are carried out manually
- Preprocessing: One-hot encoding of Activities and Resources
- Weights and activity time calculation

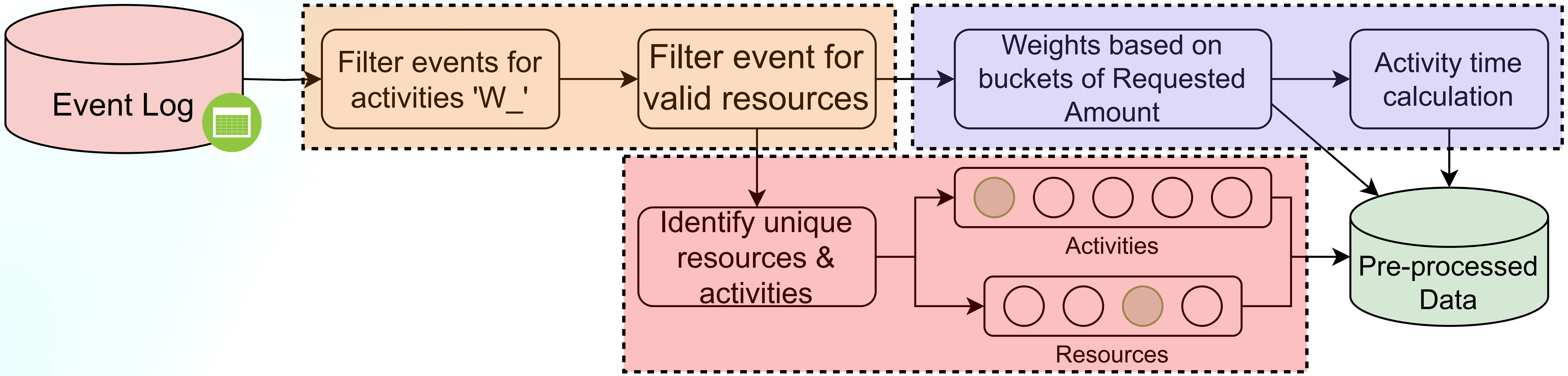


Figure. Data Filtering and preprocessing

Experiments

- Replicate original implementation
    1. LSTM + Minimum cost maximum flow (MCMF)
- Major concern: performance of prediction model
- Train 3 additional models:
    1. BiLSTM + MCMF
    2. GRU + MCMF
    3. CNN + MCMF

Experiments

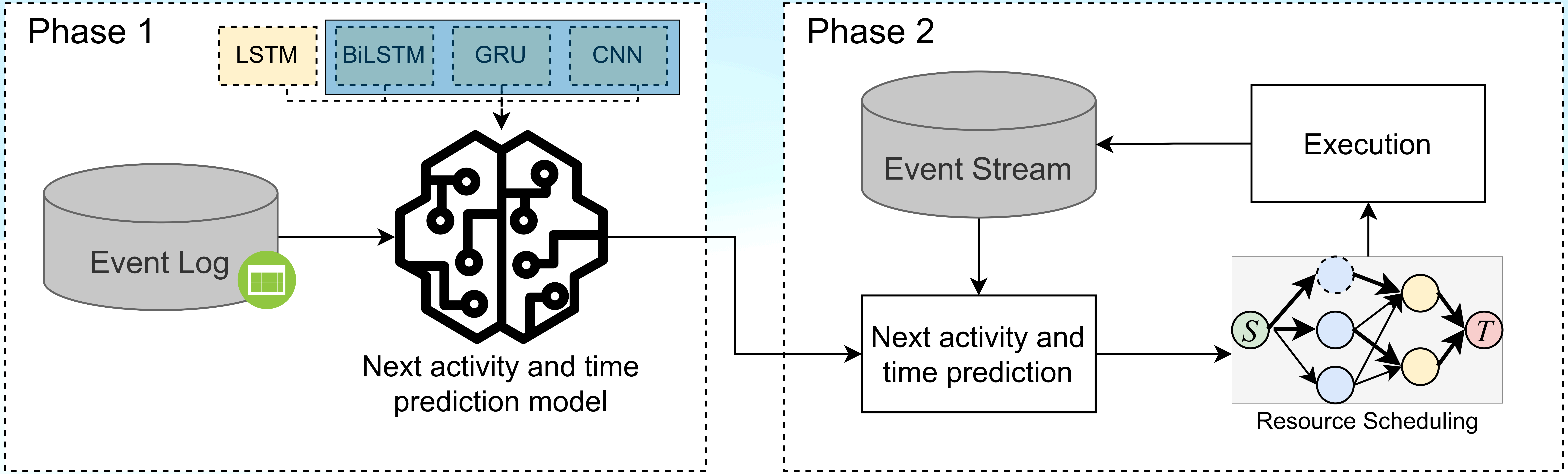


Figure. Experimental Setup

Results

	Method	Weighted Completion	Computation Time	Prediction Time
Suggested in Original paper	Baseline	2695	60	56
	LSTM + MCMF	1823	3151	3145
Additional Prediction Models	BiLSTM + MCMF	1928	3194	3189
	GRU + MCMF	1658	3266	3261
	CNN + MCMF	**807**	3645	3639

Table. Results of our experiments

Results

- CNN capturing spatial patterns from matrix-like data
- challenge conventional assumptions [3]
- suitable approaches for different tasks
- Complex architectures outperformed by simpler architectures like GRU and CNN
- Keeping prediction models simple

	Method	Weighted Completion	% change from baseline
Suggested in Original paper	Baseline	2695	0
	LSTM + MCMF	1823	47%
Additional Prediction Models	BiLSTM + MCMF	1928	39%↓
	GRU + MCMF	1658	62%↑
	CNN + MCMF	**807**	233%↑

Table. Results of our experiments

Future Work & Limitations

- Different resource allocation method:
    - Eg. Ant Colony Optimisation
- Using different real-life datasets
- Limitations:
    - CNN performs best, contrary to studies [3]
    - Prediction time

References

1. G. Park and M. Song, Prediction-based Resource Allocation using LSTM and Minimum Cost and Maximum Flow Algorithm, 2019 International Conference on Process Mining (ICPM), Aachen, Germany, 2019, pp. 121-128, doi: 10.1109/ICPM.2019.00027.

2. M. L. Pinedo, Scheduling: Theory, Algorithms, and Systems, 3rd ed.Springer Publishing Company, Incorporated, 2008.

3. Efrèn Rama-Maneiro, Juan C. Vidal, and Manuel Lama. Deep learning for predictive business process monitoring: Review and benchmark. IEEE Transactions on Services Computing, 16(1):739–756, 2023.

Open to Questions!

Thank you!