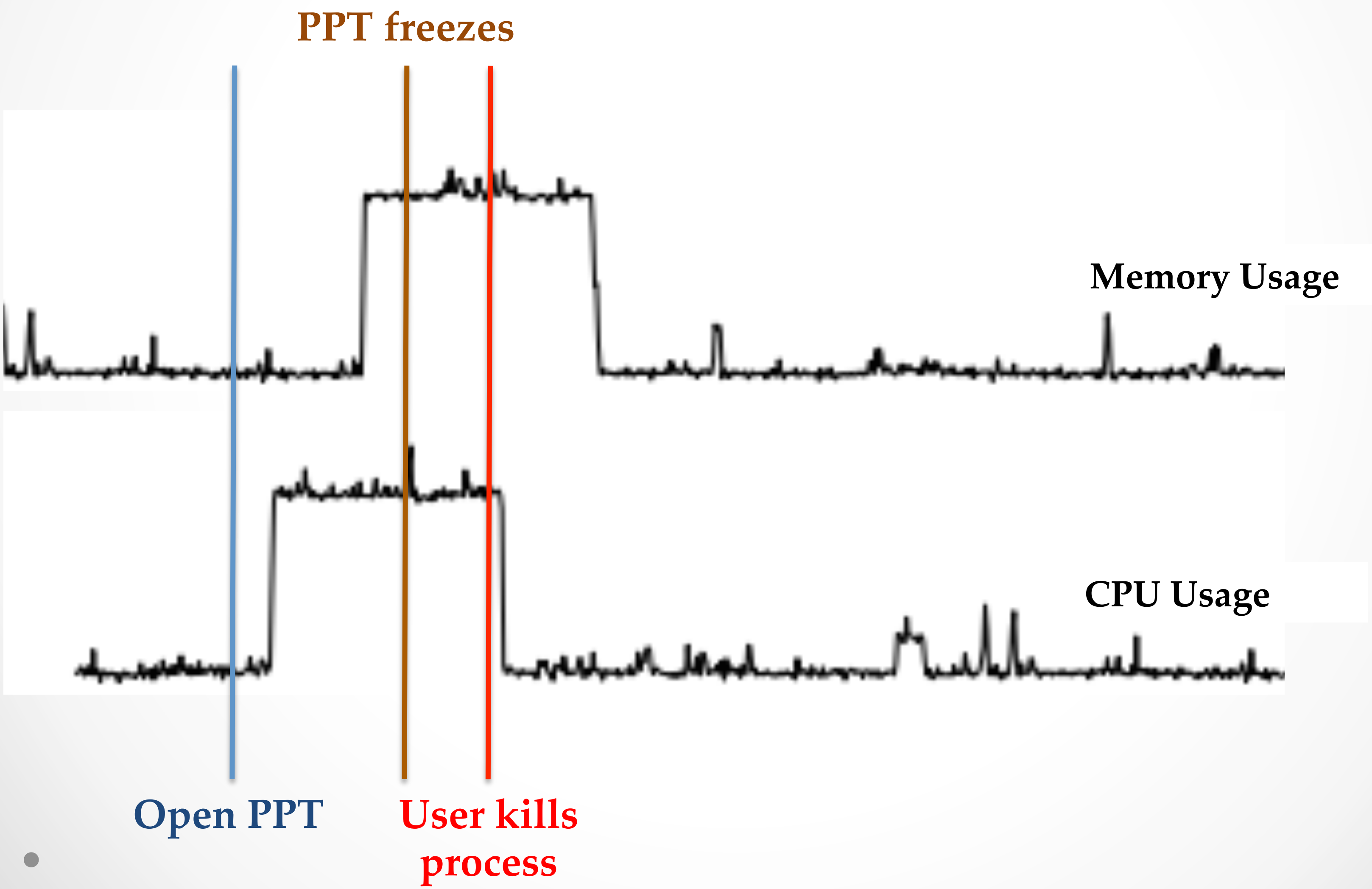
Correlating Events with Time Series for Incident Diagnosis

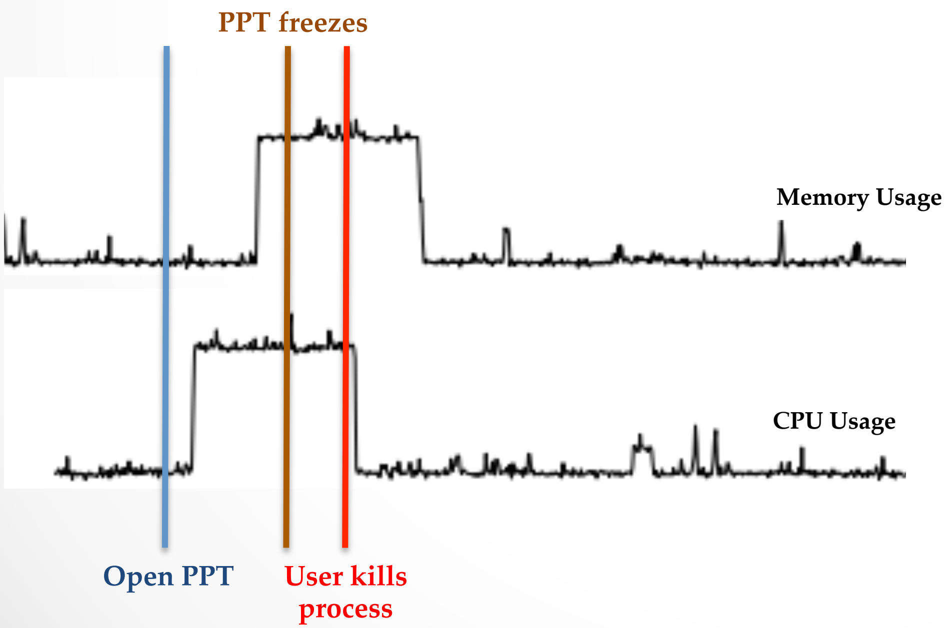
Ricardo Reimao

Idea: Identifying Patterns in Series and Events



Problem!

- How to correlate events with temporal series?
- How to identify anomalous behavior?
- How to predict incident causes?



Series 1: CPU Usage
Series 2: Memory Usage
Event Series: Windows logs

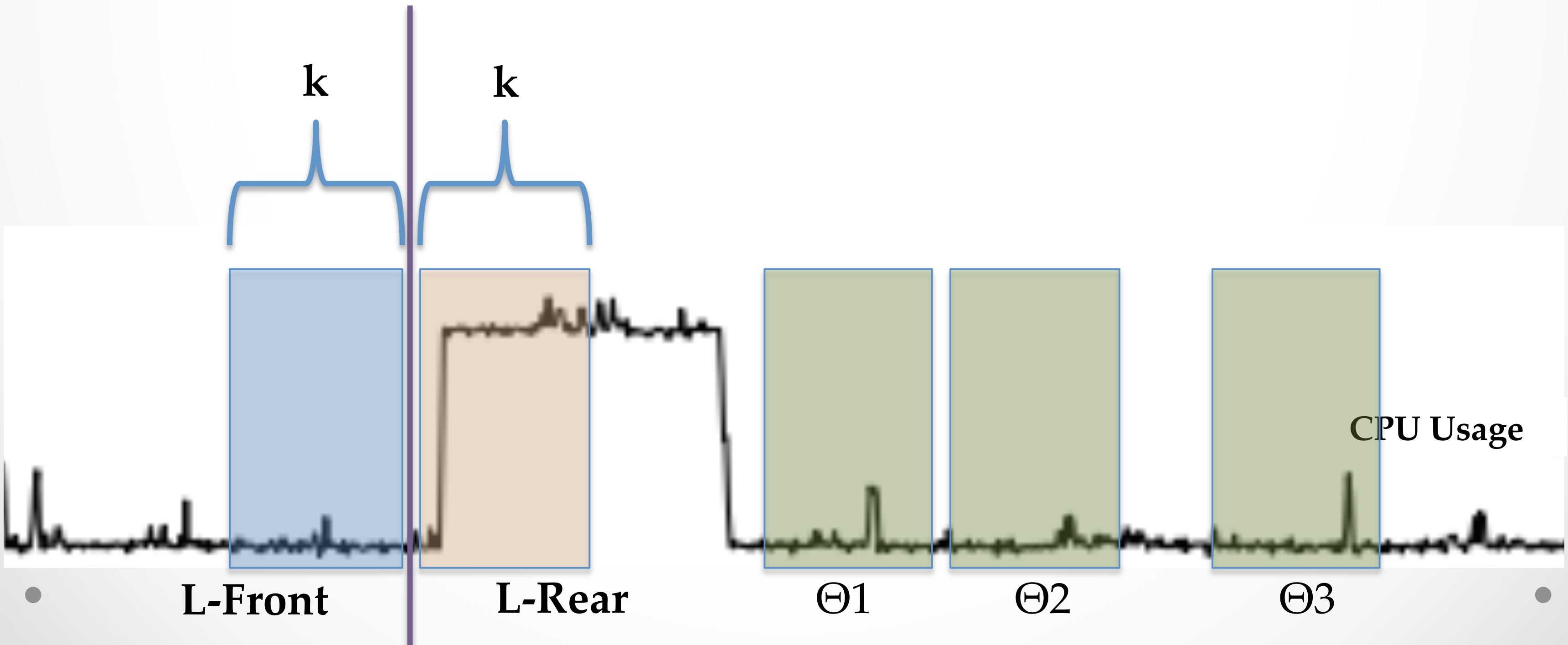
Formalizing the Problem

Three Main Questions

- Existence Dependency
  - "Is there a correlation between the event sequence and the time series?"
  - "Does opening powerpoint affect my CPU usage?"
- Temporal Order of Dependency
  - "Does X influences in Y? Or Y influences in X?"
  - "The powerpoint freezes because the memory usage is high? Or the memory usage is high because the powerpoint is frozen?"
- Monotonic Effect of Dependency
  - "Does the event impact negatively or positively on the measure?"
  - "When I open powerpoint, does the memory usage increases or decreases?"

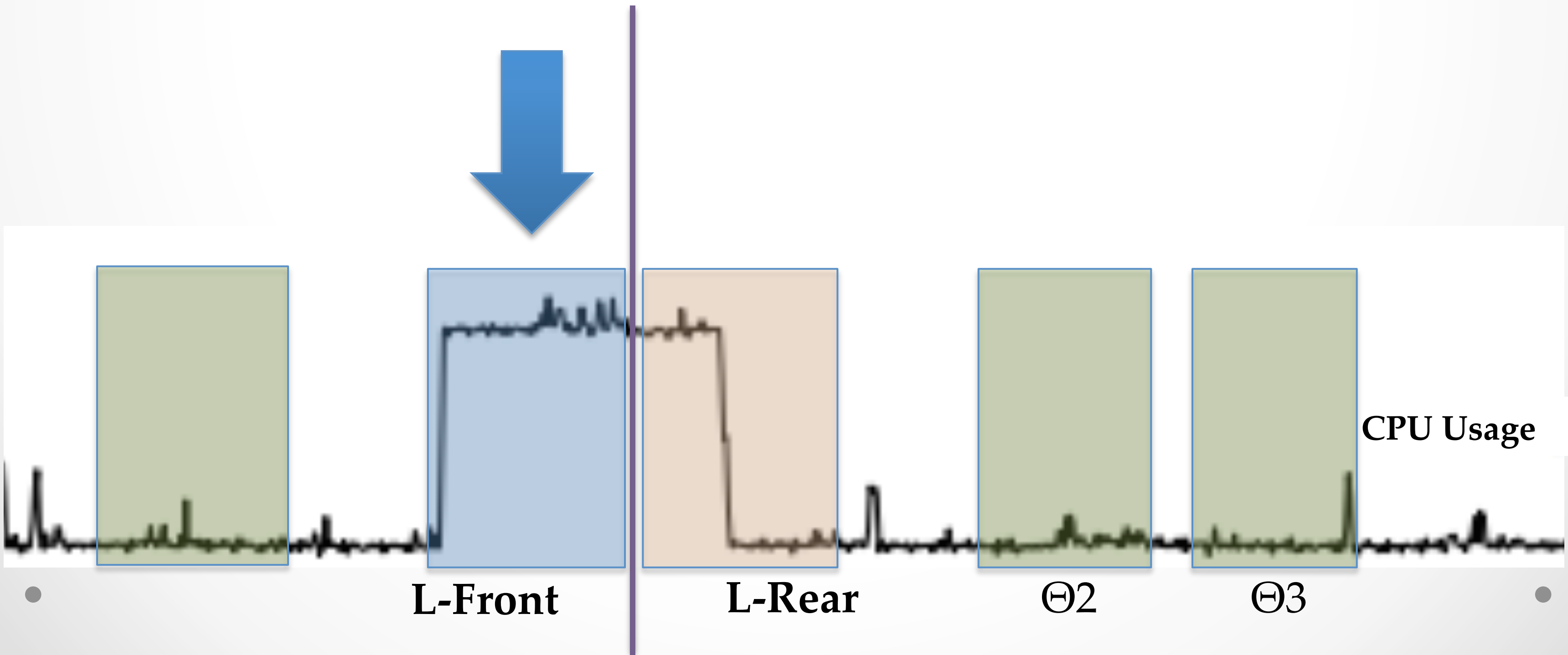
Subset definitions

- L-Front: The sub-series BEFORE the event
- L-Rear: The sub-series AFTER the event
- Θ : A set of random sub-series
- k: Size of the sub-sets



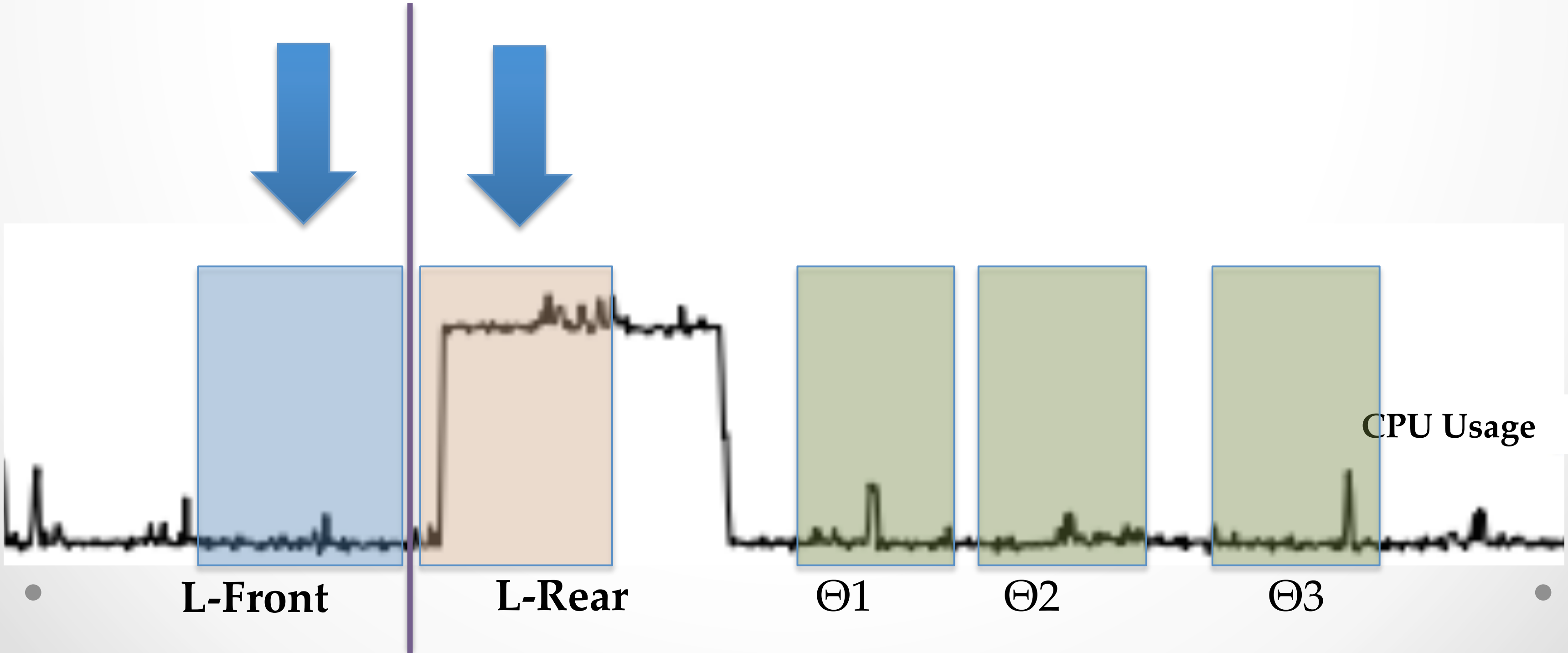
Definition 1

- "An event sequence E and a time Series S are correlated and E often occurs **after** changes of S (S > E) if and only if the probabilistic distribution L-Front is statistically different from the randomly sampled Θ.



Definition 2

- "An event sequence E and a time series S are correlated and E often occurs **before** the changes of S (E > S), if and only if the probabilistic distribution of L-Rear is statistically different from the randomly sampled sub-series Θ and the probabilistic distribution of L-Front is not statistically different from Θ."



Definition 3

- An event sequence E and a time series S are correlated (E ~ S) if there is a relationship such as E > S or S > E

Definition 4

- If E > S (or S > E) and the event occurrences of E are related to significant value increases of S, we denote the correlation as E +> S.
  If S decreases, we denote the correlation as: E -> S

Challenge:
How to test if L-Rear are statistically similar to Θ?

Approach:
Two Sample Problem

What is Two Sample Problem?

- Multivariate two-sample hypothesis-testing problem
- Objective: Identify if two samples are from the same distribution
- In our context:
  - Check if L-Rear and Θ are from the same distribution
  - Check if L-Front and Θ are from the same distribution

- Two Hypothesis:
  - $H_0 : S = \Theta$ (The series and Θ are from the same distribution, or in other words, S and Θ are statistically equal)
  - $H_1 : S \neq \Theta$ (The series and Θ are from different distributions, or in other words, S and Θ are statistically different)

How to check? Nearest Neighbor!

- Why?
- Verify the distance between an item and items in a database
- Process:
  - Generate the subset of L-Front/L-Rear
  - Generate the subset of Θ
  - Concatenate L-Front/L-Rear and Θ (this becomes the DB)
  - Whenever a new item A (event + L-Front + L-Rear) is tested:
    - Use k-NN to check which item is more similar to A
    - If the closest item is an item of Θ, then there's no correlation
    - Else, the item may be correlation

Monotonicity Check

- To check the monotonic effect, a new artifact is introduced: t_{score}

$$t_{score} = \frac{\mu_{\Gamma front} - \mu_{\Gamma rear}}{\sqrt{\frac{(n_1-1)\sigma^2_{\Gamma front} + (n_2-1)\sigma^2_{\Gamma rear}}{n_1+n_2-2}\left(\frac{1}{n_1} - \frac{1}{n_2}\right)}}$$

- Idea: Measure "how big is the impact" of E in S.
- If t_{score} is higher than a threshold, then: E +> S
- If t_{score} is lower than a threshold, then: E -> S

Algorithm

Inputs/Outputs

- Input:
  - Event vector $E = (e_1, e_2, \ldots, e_n)$
  - Time Series $S = (s_1, s_2, \ldots, s_m)$
  - Subseries length k
- Output:
  - Correlation flag C
  - Correlation direction D
  - Effect type t
- Important: ‘k’ (subseries length) and n (number of knn neighbours to evaluate) have high impact on performance!

General Idea

- Test L-Front and Θ
- Test L-Rear and Θ
- If correlation is found:
  - Verify t_{score} to identify direction
  - Return

Algorithm 1: The Overall Algorithm

Input: Event $E = (e_1, e_2, ..., e_n)$, and Time Series $S = (s_1, s_2, ..., s_m)$, and the sub-series length k.

Output: The correlation flag C, the direction D, and the effect type T

Initialize Γ^{front} and Γ^{rear};
Initialize Θ;
Initialize $R = false$, $D = NULL$, $T = NULL$;
Normalize each $\ell_k^{front}(S, e_i)$ and $\ell_k^{rear}(S, e_i)$.;
Test Γ^{front} and Θ using Nearest Neighbors Method. The result is denoted as D_f.;
Test Γ^{rear} and Θ using Nearest Neighbors Method. The result is denoted as D_r.;
if $(D_r == true \&\& D_f == false)$ **then**
    $R = true.$;
    Calculate t_{score} using Equation (8).;
    if $(t_{score} > \alpha)$ **then**
        $T = E \xrightarrow{-} S.$;
    else if $(t_{score} < -\alpha)$ **then**
        $T = E \xrightarrow{+} S.$;
else if $(D_r == false \&\& D_f == true) \,||\, (D_r == true \&\& D_f == true)$ **then**
    $R = true.$;
    Calculate t_{score} using Equation (8).;
    if $(t_{score} > \alpha)$ **then**
        $T = S \xrightarrow{-} E.$;
    else if $(t_{score} < -\alpha)$ **then**
        $T = S \xrightarrow{+} E.$;
Out put R, D and T;
Algorithm End.

Empirical Evaluation

Previous Works

- Pearson Correlation
  - One of the most used methods for measuring correlation between <u>two time series</u>
  - Cannot be directly used to correlate event and series data
    - Need to transform event data into a serie
- J-measure Correlation
  - One of the most used methods for measuring correlation between <u>event data</u>
  - Cannot be directly used to correlate event and series data
    - Need to transform series into event data

Tests in a Controlled Environment

Table 2: Results of the data from controlled environment

Name	Proposed Method			Pearson Correlation			J-Measure		
	CPU	Memory	Disk	CPU	Memory	Disk	CPU	Memory	Disk
CPU Intensive Program	$\xrightarrow{+}$	NC	NC	$\overset{+}{\sim}$	NC [1]	NC	NC	$\sim$	$\sim$
Memory Intensive Program	$\xrightarrow{+}$	$\xrightarrow{+}$	NC	NC	$\overset{+}{\sim}$	NC	NC	$\sim$	$\sim$
Disk Intensive Program	NC	NC	$\xrightarrow{+}$	NC	NC	$\overset{+}{\sim}$	NC	$\sim$	$\sim$
Query Alert	$\xleftarrow{+}$	$\xleftarrow{+}$	NC	$\overset{+}{\sim}$	NC	NC	NC	$\sim$	$\sim$

- Person did not capture some correlations
- Person does not give you the direction of the correlation
- J-Measure did not identify correlation in one whole series

Tests in Real-World Environments

Evaluation Metric:

$$F_1 = \frac{2 * TruePositive}{2 * TruePositive + FalseNegative + FalsePositive}$$

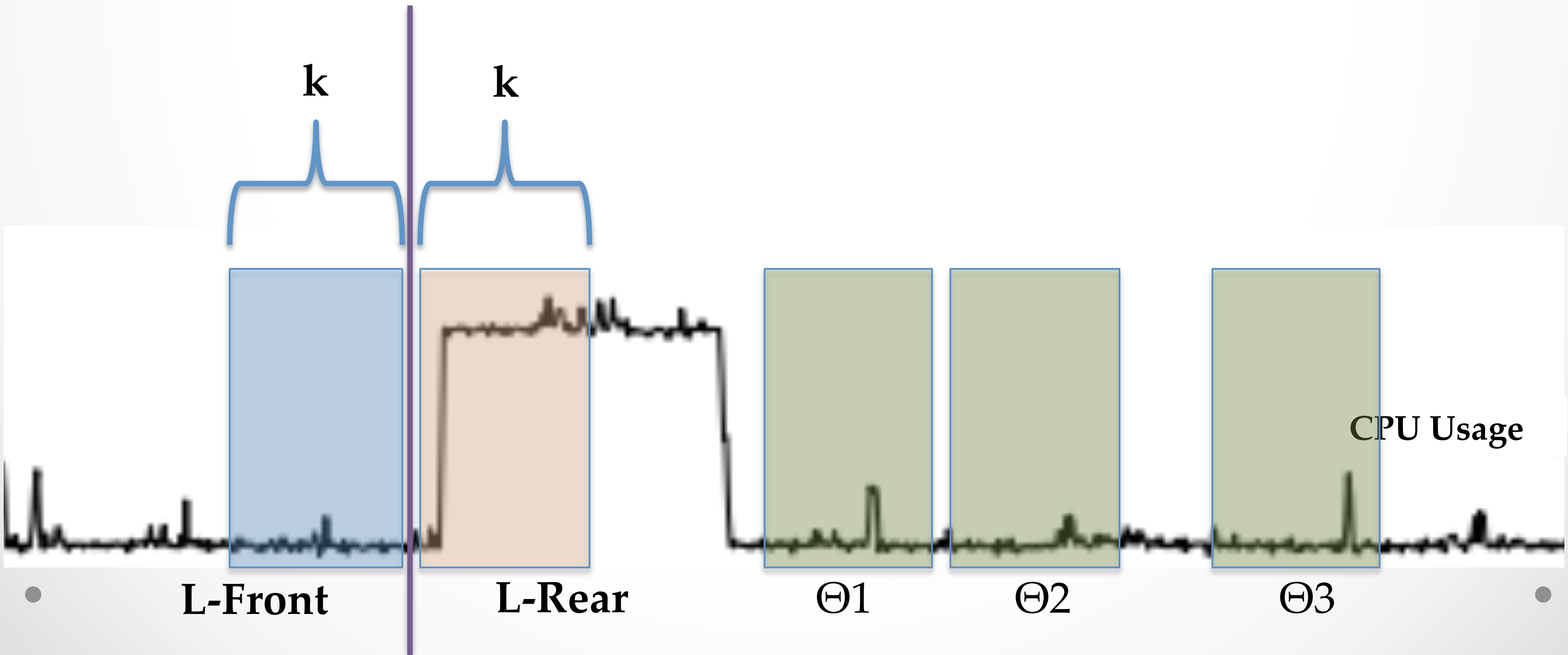
Table 3: Result in real data set

Data Set	Methods	Existence F_1 Score	Temporal Order F_1 Score	Effect Type F_1 Score
System Monitoring Data	Correlation Mining (L1)	0.7916	0.8020	0.8016
	Correlation Mining (L2)	**0.8205**	0.7612	**0.8780**
	Correlation Mining (DTW)	0.7962	**0.8021**	0.8210
	Pearson Correlation	0.6974	N/A [2]	0.6732
	J-Measure	0.6148	N/A	N/A
Custom Support Data	Correlation Mining (L1)	0.7915	0.7659	0.7204
	Correlation Mining (L2)	0.8423	0.7870	0.8334
	Correlation Mining (DTW)	**0.8631**	**0.8205**	**0.8532**
	Pearson Correlation	0.6030	N/A	0.6501
	J-Measure	0.7398	N/A	N/A

Summary

Concept Summary

- L-Front: The sub-series BEFORE the event
- L-Rear: The sub-series AFTER the event
- Θ : A set of random sub-series
- k: Size of the sub-sets



Process

Identify L-Front/L-Rear → Generate Random Θ → Compare L-Front/L-Rear to Θ → Identify Correlation (F/R=Θ?) → Identify Direction (F=Θ? R=Θ?) → Identify Monotonicity (Tscore)

Pros | Cons

Correlate time series and event data

Identify not only correlation, but also direction and monotonicity

Can be applied against multiple time series

More effective then previous works (Pearson and J-Measure)

Utilizes a slow-search method: Nearest Neighbors

Does not consider the event combination problem

Questions?

Ricardo Reimao