Spectrum-based Anomaly Detection For Real-time Systems

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

R. Torres (MASc), C. Moreno (PhD), S. Fischmeister (PhD, PEng)
Department of Electrical and Computer Engineering

UNIVERSITY OF WATERLOO

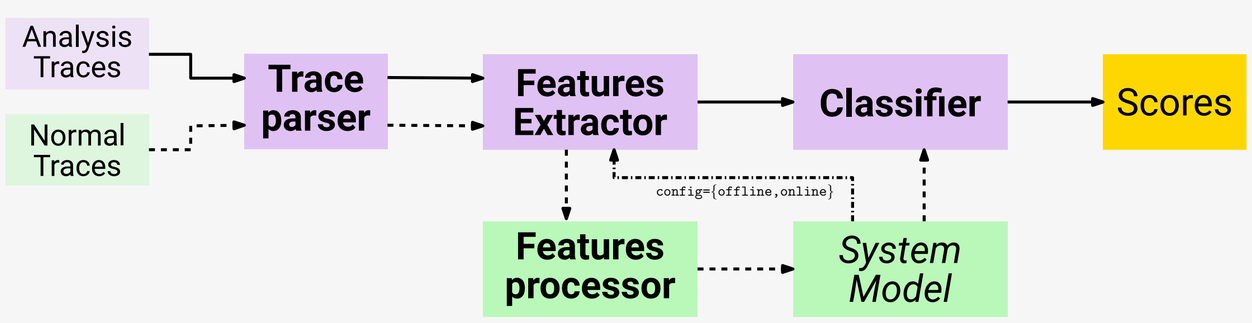
Motivation

Safety-critical systems require continuous monitoring during operation to detect deviations from specified behavior.

- Testing usually starts during implementation and must be completed before deployment. However, testing cannot check for every possible scenario, and thus, additional monitoring is recommended. » **Anomaly Detection**
- Systems usually incorporate some form of event logging with time stamps. Most RTOS, supports process and kernel tracing capabilities. Although mostly used for debugging purposes, it can also be employed for anomaly detection. » **Trace based AD**
- Most real-time systems are implemented using a set of periodic tasks, each releasing a job on every period. Therefore, a timely stream of recurrent events is expected. » **DSP based AD**

Framework Overview

We use a set of normal (good) traces to extract periodic features that constitute a model of the system during safe operation. During analysis it is used to determine whether or not other traces are anomalous.



Trace Parsing

In the example shown, a two tasks system produces the schedule of Figure 2. This schedule repeats after t=28. The operating system generates the trace of Table 1. Each trace entry is issued after a task changes state according to the model of Figure 3.

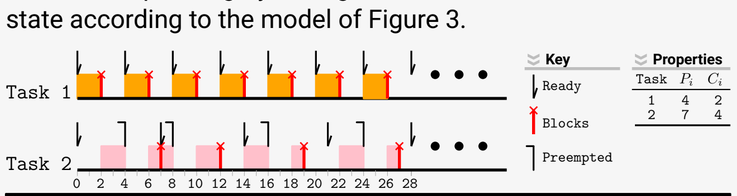


Figure 2: The recurrent schedule of a two tasks system. Each task is periodic and has a fixed capacity.

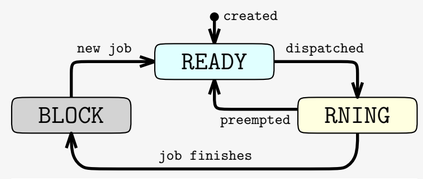


Figure 3: Task transition diagram.

Each row in a Trace ($\mathbb{T}$) is a Trace Entry (E_i). Each trace entry is a tuple ($E_i := \langle idx, t, P \rangle$) consisting of an index, a time stamp and a fixed number of parameter values.

Table1. Trace of system in Figure 1.

Idx	t	THREAD	PID
1	0	READY	1
2	0	READY	2
3	0	RNING	1
4	2	BLOCK	1
5	2	RNING	2
⋮			
36	24	READY	1
37	24	READY	2
38	24	RNING	1
39	26	BLOCK	1
40	26	RNING	2
41	27	BLOCK	2

A Channel ($\mathbb{C}_{i,j}$) is an ordered list of trace entries. It is extracted from $\mathbb{T}$ using parameter p_j in parameter set $\mathcal{P}_i$ as matching criteria.

Table2. Channel $\mathbb{C}_{1,1} : \mathbb{P}_{1,1} = \{READY\}$

Idx	t	THREAD	PID
1	0	READY	1
2	0	READY	2
6	4	READY	1
⋮			
33	21	READY	2
36	24	READY	1
37	24	READY	2

Table2. Channel $\mathbb{C}_{2,2} : \mathbb{P}_{2,2} = \{2\}$.

Idx	t	THREAD	PID
2	0	READY	2
5	2	RNING	2
7	4	READY	2
⋮			
37	24	READY	2
40	26	RNING	2
41	27	BLOCK	2

A Signal ($\mathcal{X}_i$) is the result of intersecting channels along the parameters set class.

Table3. Signal $\mathcal{X}_2 := \{\{\mathbb{C}_1\} \bigcap \{\mathbb{C}_5\}\}$.

Idx	t	THREAD	PID
2	0	READY	2
7	4	READY	2
12	7	READY	2
15	8	READY	2
23	14	READY	2
26	16	READY	2
33	21	READY	2
37	24	READY	2

Real-time stamp sequences ($\mathcal{R}_i$) are extracted from relevant signals. Finally inter-arrival time sequences ($\mathcal{D}_i$) are derived from $\mathcal{R}_i$ applying the mapping function $f : \mathcal{R}_i \to \mathcal{D}_i$. Since $\mathcal{D}_i$ represents unique events within the system, it is the baseline signal for anomaly detection.

$$\mathcal{R}_i \to \mathcal{R}_i[k] - \mathcal{R}_i[k-1] \to \mathcal{D}_i$$

$$\mathcal{R}_1 = \{0, 4, 8, 12, 16, 20, 24, ...\} \xrightarrow{f} \mathcal{D}_1 = \{4, 4, 4, 4, 4, 4, ...\}$$
$$\mathcal{R}_2 = \{0, 4, 7, 8, 14, 16, 21, 24, ...\} \xrightarrow{f} \mathcal{D}_2 = \{4, 3, 1, 6, 2, 5, 3, ...\}$$

Features Extraction

We expect that if $\mathcal{D}_i$ comes from a real-time task, it should be a recurrent sequence, and thus, have a distinctive shape in the Frequency Domain. In real world applications $\mathcal{D}_i$ is expected to be pseudo-recurrent, but still having relevant frequency domain features.

- For a modelling trace we first compute the Periodogram $\hat{\mathcal{P}}_{per}(e^{j\omega}) = \frac{1}{N}|\mathcal{F}(\mathcal{D}_i)|^2$ of its signals. Find its representative peaks and estimate its sequence length using the fundamental frequency component. Figure 4 shows the plot of a sequence for $\mathcal{D}_2$ and its periodogram (excluding the DC component) with some highlighted features.

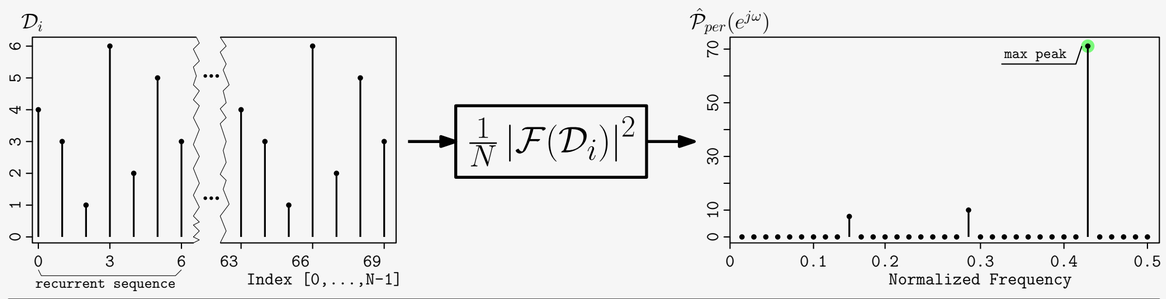


Figure 4: A seventy point sequence for $\mathcal{D}_2$ and its corresponding periodogram. Please note that $\mathcal{F}$ is the unnormalized Discrete Fourier Transform.

- If the length of $\mathcal{D}_i$ is an integer multiple of its recurrent sequence, the amplitude of its frequency domain rises to maxima. Figure 5 shows the periodograms for two versions of $\mathcal{D}_2$ that illustrates this.

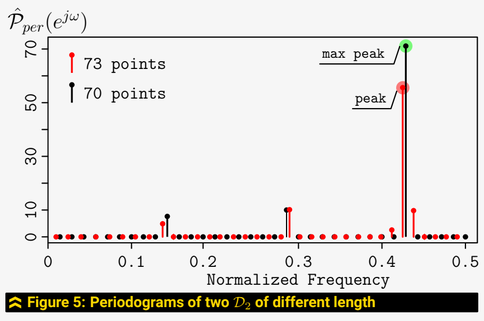


Figure 5: Periodograms of two $\mathcal{D}_2$ of different length

- The final step is the metrics and threshold estimation, using the modified periodogram of $\mathcal{D}_i$ by the Welch method. Figure 6 shows the Welch periodogram of an actual signal, superimposed on its raw periodogram.

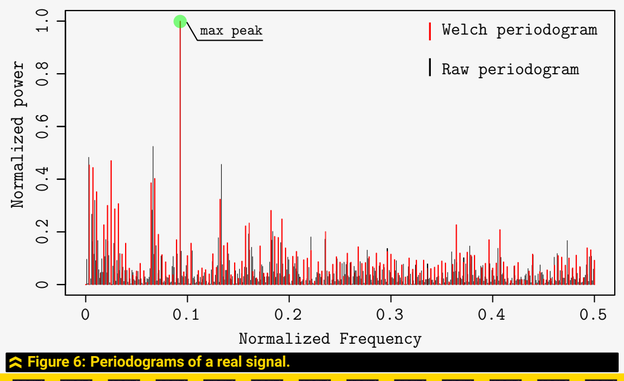


Figure 6: Periodograms of a real signal.

From a set of "normal" traces coming from the same distribution we build a model of the system. Analysis traces can be tested for statistical significance against this model.

Anomaly detection

When a system operates in an anomalous mode we expect a change in the sequence of inter-arrival times of some signals. Although hard to detect in the time domain, differences can be easily seen in the frequency domain. Figure 7 shows the frequency domain of a normal and an anomalous trace.

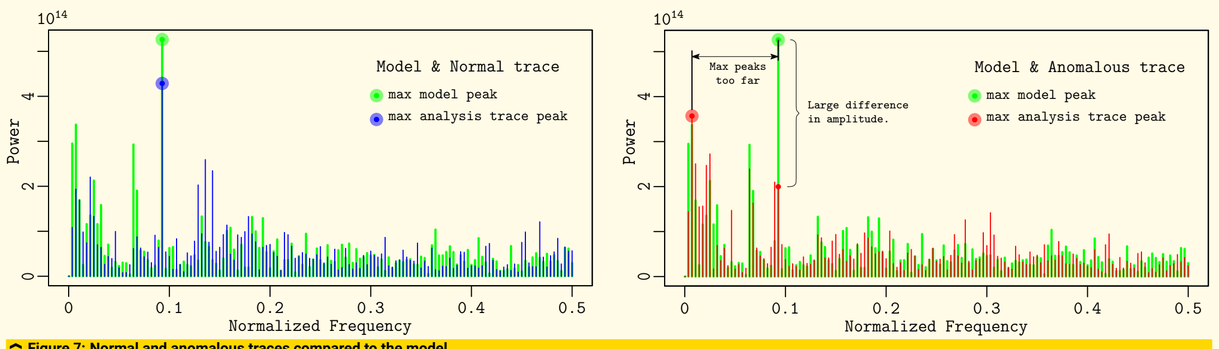


Figure 7: Normal and anomalous traces compared to the model.

Traces can be processed in two modes. The offline mode analyzes the trace as a whole, as shown in Figure 7. In online analysis the trace is an endless stream. For this mode we take a window of data and find its metrics like in offline mode, then the window is shifted. We allow data overlapping between analyses. Figure 8 shows the result of processing two traces using this method.

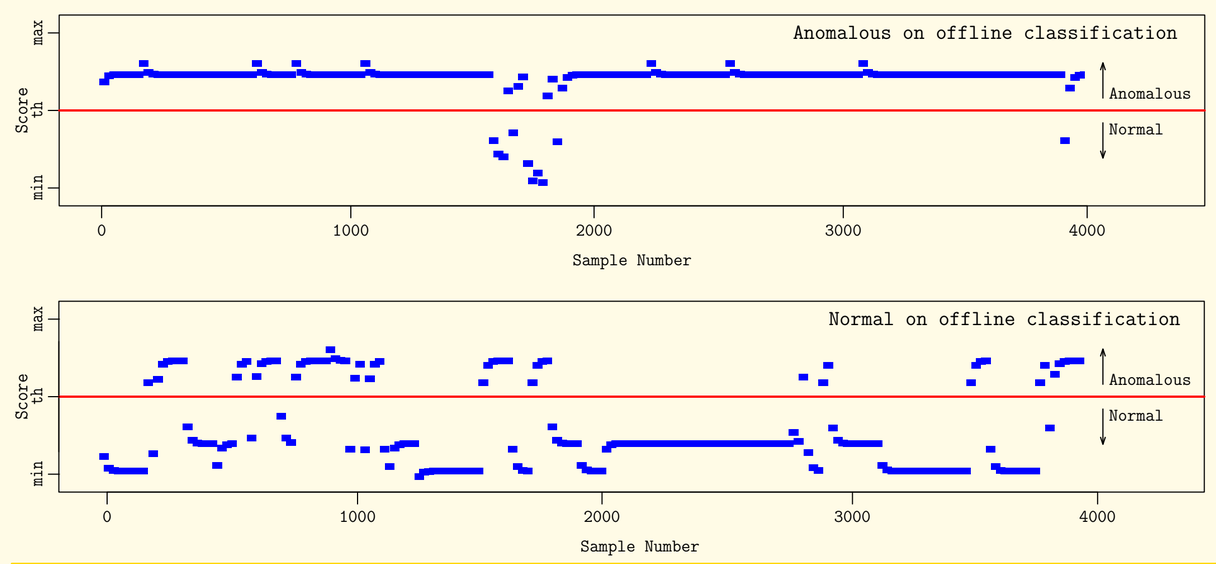


Figure 8: Two traces processed using online method.

Classification

The features extractor output is a set of parameters that are used to calculate scores for each trace under analysis. Some features can be graphically represented, but the purpose of this module is to determine whether or not the trace is anomalous.

For a set of training traces the classifier first computes estimates for each metric (i.e.: MaxPeak, PeakFreq) of each signal. Then it averages the values:

$$\overline{m}_{j,k} = \frac{\sum_{i=1}^{n_t} m^i_{j,k}}{n_t}$$

When analyzing a trace, the classifier computes the weighted average of squares of normalized differences of the metric to the model mean:

$$\hat{m}^i = \sum_{j=1}^{M} \sum_{k=1}^{|\mathbb{P}_j|} \left(m^i_{j,k} - \bar{m}_{j,k} \right)^2 \times w_{j,k}$$

Where $w_{j,k}$ are normalized weights for each metric.: $w_{j,k} = \frac{\overline{m}_{j,k}/_i(m^i_{j,k})}{\sum_j \sum_k \overline{m}_{j,k}/_i(m^i_{j,k})}$

An overall score for the trace is calculated based on the three metrics:

$$Sc^i = \frac{\widehat{FS}^i + \widehat{P}^i + \widehat{F}^i}{3}$$

And a classification is given:

$$C^i = \begin{cases} \texttt{false} & if\ Sc^i \leq \texttt{scoreMax} \times r \\ \texttt{true} & otherwise \end{cases}$$

Results

For the offline version we have tested the same set of traces used in the previous version of the tool. Figure 9 shows the result of analayzing a set of traces using the offline method. For online processing, results must be plotted like in Figure 8.

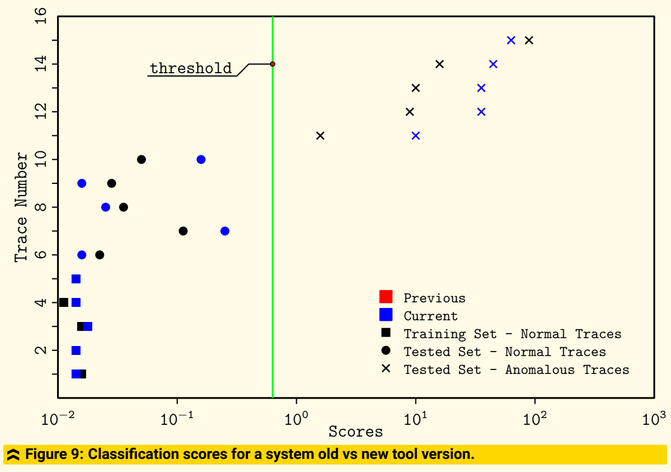


Figure 9: Classification scores for a system old vs new tool version.

Conclusions

- SipTA is based on a modular design that can be modified to suit different application requirements. The parsing engine will be the common subjet to modifications, while other modules can remain unaltered. » **Widely applicable**
- We have extended the tool so it can be used to process streams. This is a significant improvement that enables the use of the SipTA for online anomaly detection. » **Concurrent supervision**
- The current improvement of the tools shows an increase on the accuracy of scores estimation. However, a better method for scores calculations is needed; especifically for online processing.
- SipTA still assumes that there exists a set of normal traces from which a model can be extracted. In many cases, there is knowledge about the system that can be used to improve the detection rate and reduce errors. » **Further research**

Contact: {rtorresl, sfischme, cmoreno}@uwaterloo.ca