SigMal: A Static Signal Processing Based Malware Triage

Dhilung Kirat Lakshmanan Nataraj Giovanni Vigna B.S Manjunath

Ezeanaka Kingsley

CISC850
Cyber Analytics

Abstract

- Sigmal as a malware detection framework
- Results of testing Sigmal on samples

Introduction

- Static, dynamic and statistical analyses
- Malwares variants
- N-gram feature extraction

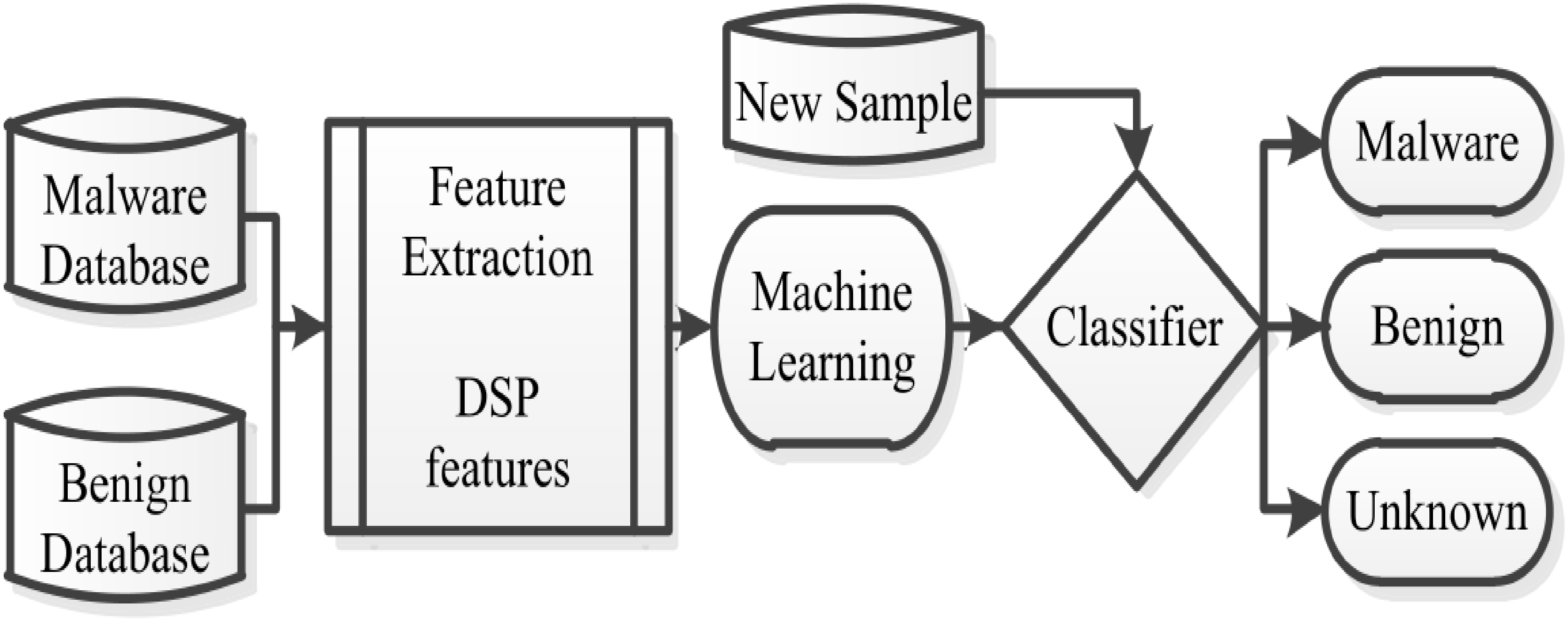
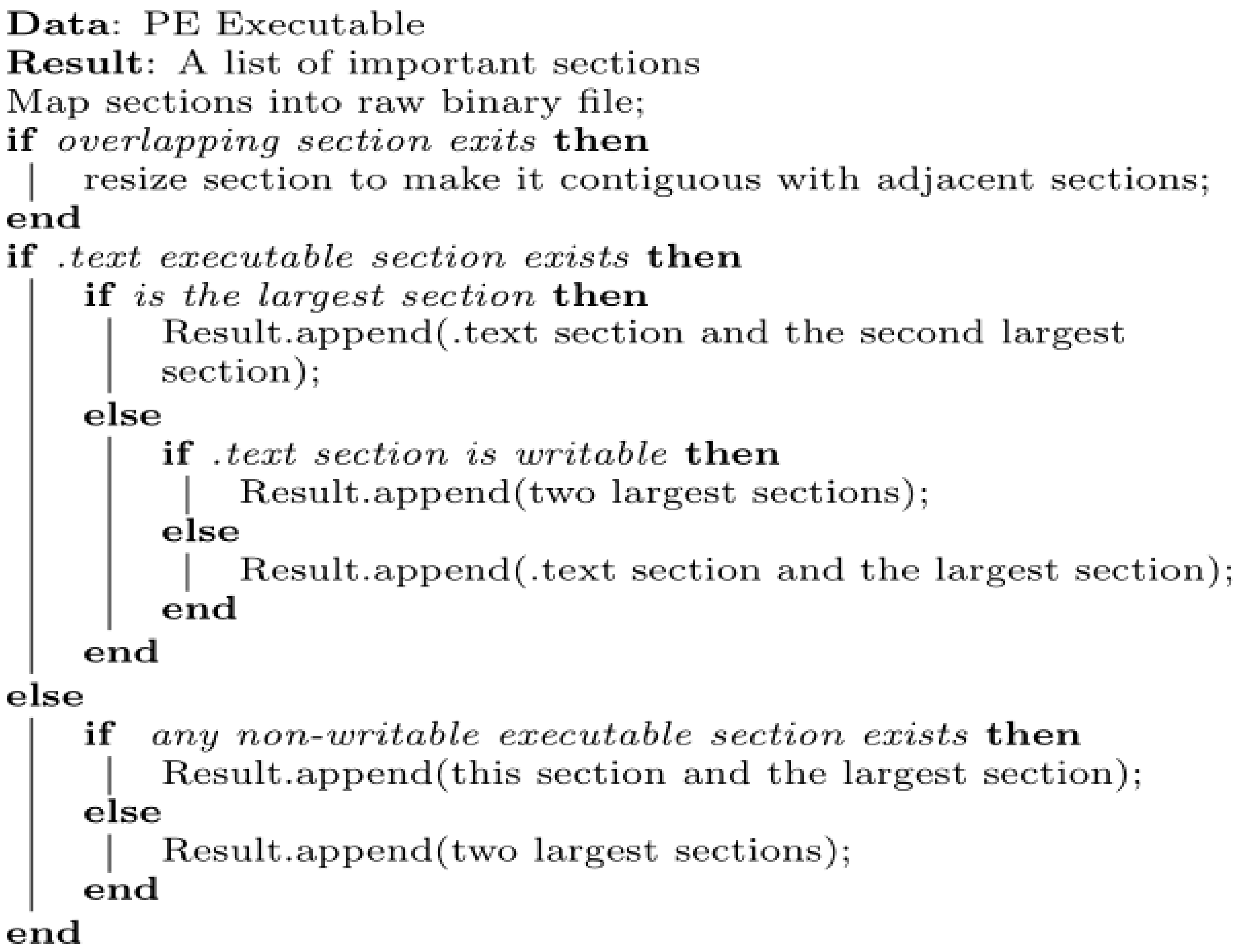


Figure 1: SigMal overview.

Signal processing based features

- Feature extraction, Feature computation
  Section aware feature extraction



```
Data: PE Executable
Result: A list of important sections
Map sections into raw binary file;
if overlapping section exits then
    resize section to make it contiguous with adjacent sections;
end
if .text executable section exists then
    if is the largest section then
        Result.append(.text section and the second largest
        section);
    else
        if .text section is writable then
            Result.append(two largest sections);
        else
            Result.append(.text section and the largest section);
        end
    end
else
    if any non-writable executable section exists then
        Result.append(this section and the largest section);
    else
        Result.append(two largest sections);
    end
end
```

Algorithm 1: Finding important sections.

Comparison

- N-gram based detection

$$J(s_a, s_b) = \frac{s_a \cap s_b}{s_a \cup s_b}$$

- PE structure based detection

- Control flow graph-based detection

$$CFG\ similarity = \frac{number\ of\ matching\ subgraphs}{total\ number\ of\ subgraphs}$$

- Benign, Malicious and real world datasets collected

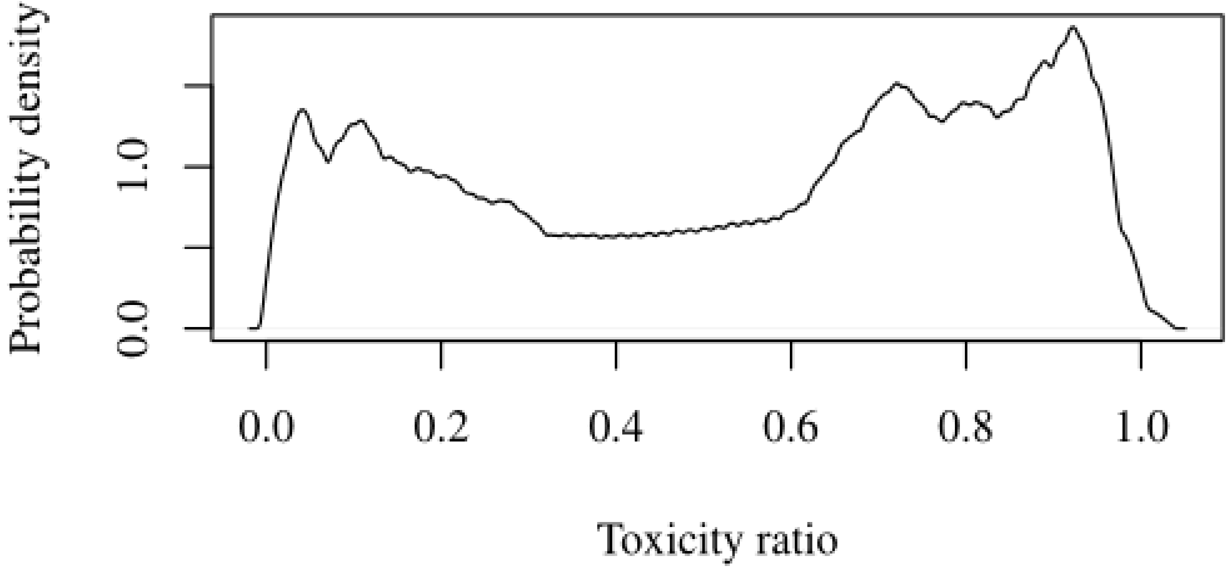


Figure 4: The *toxicity ratio* distribution of 1.2 million malware samples.

Evaluation

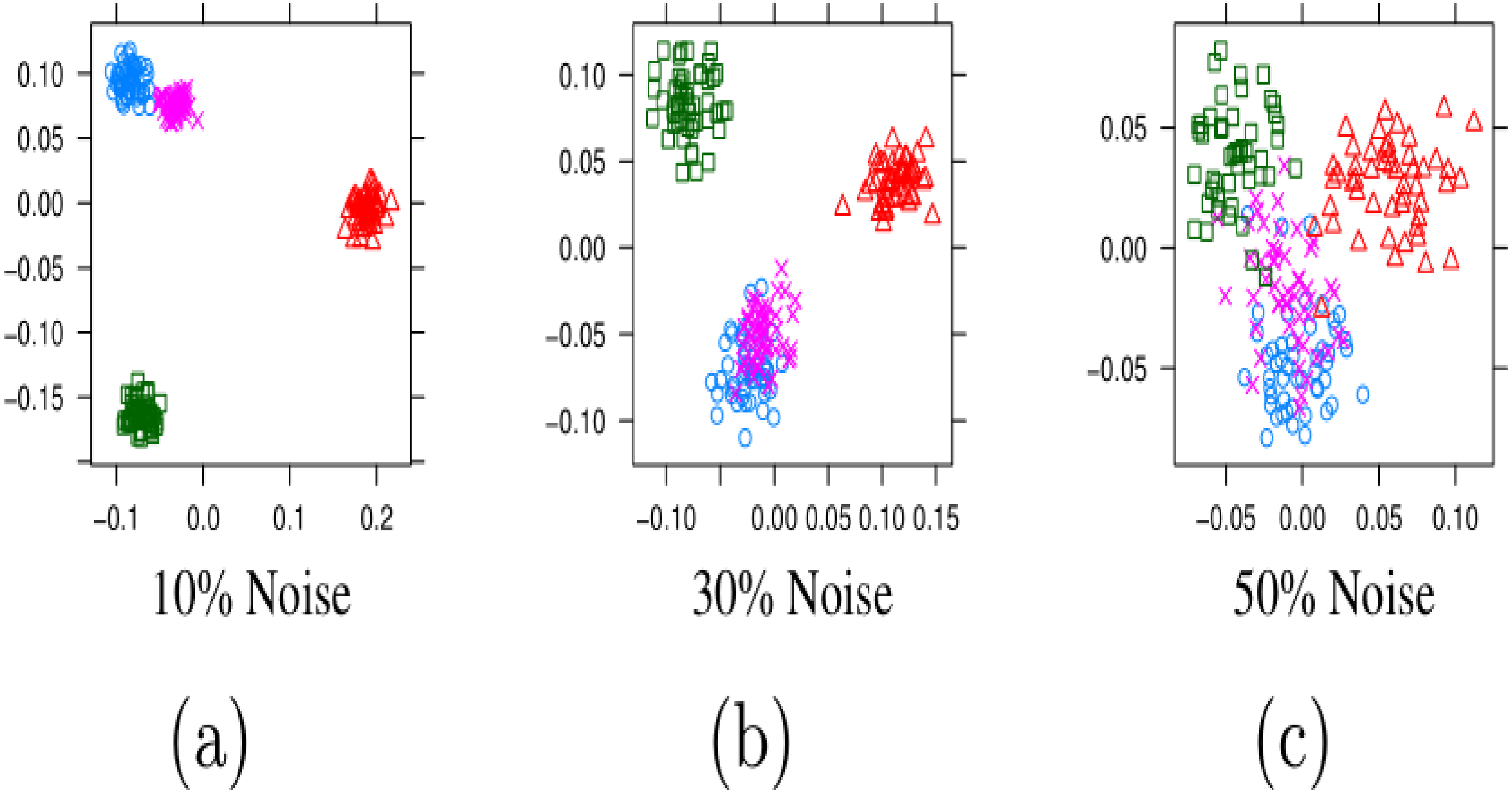


Fig 5: Feature robustness against noise.

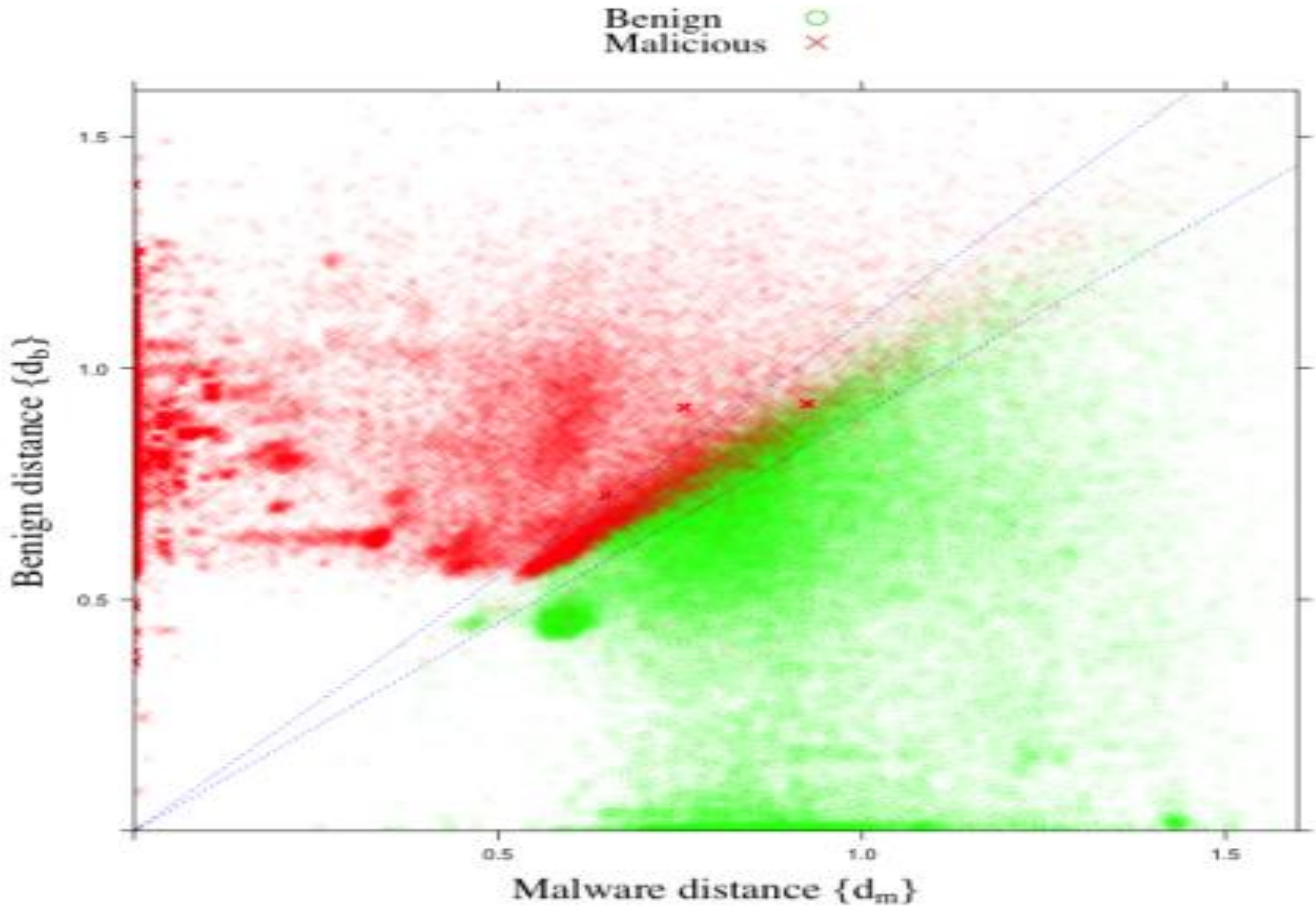


Fig. 6 : Nearest neighbor distribution for a 100 thousand samples

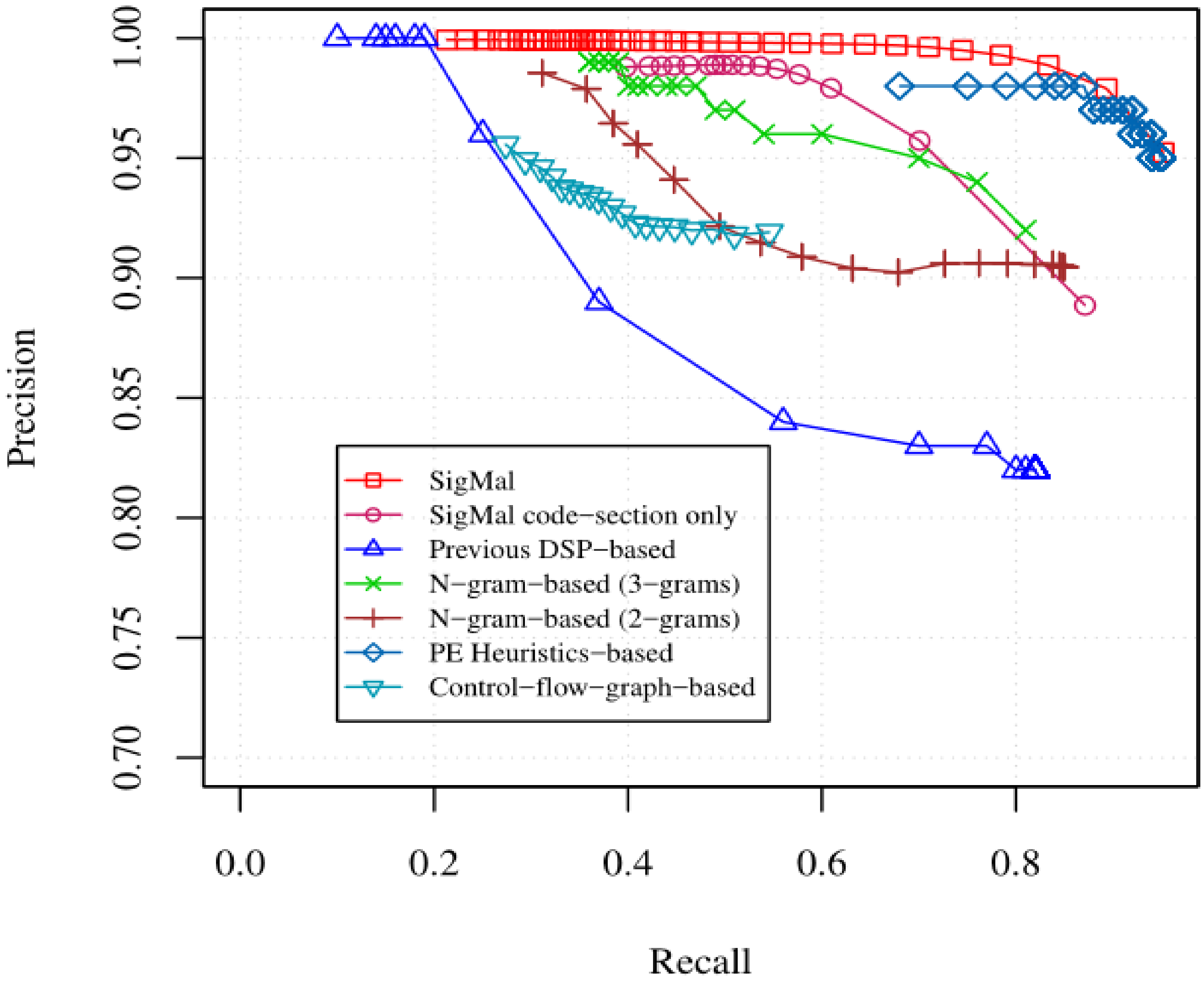
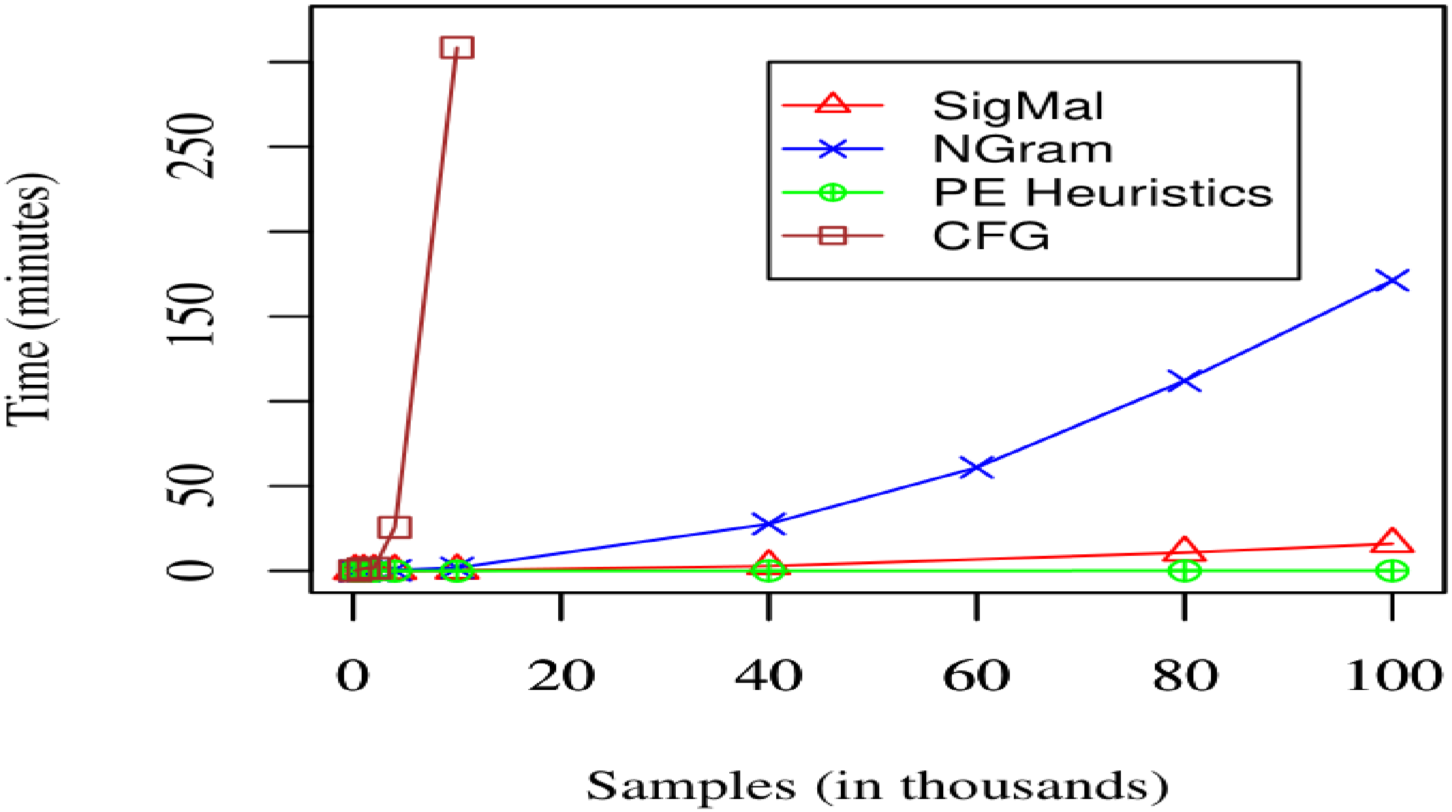


Fig. 7 : Comparison of malware detection algorithms



	SigMal	*N-gram*	*PE-heuristics*	*CFG*
Time	0.0265	0.1965	0.0024	0.1379
Space	3.783	8.000	0.0664	297.745

Fig. 8 : Query performance comparison.

Real world experiments

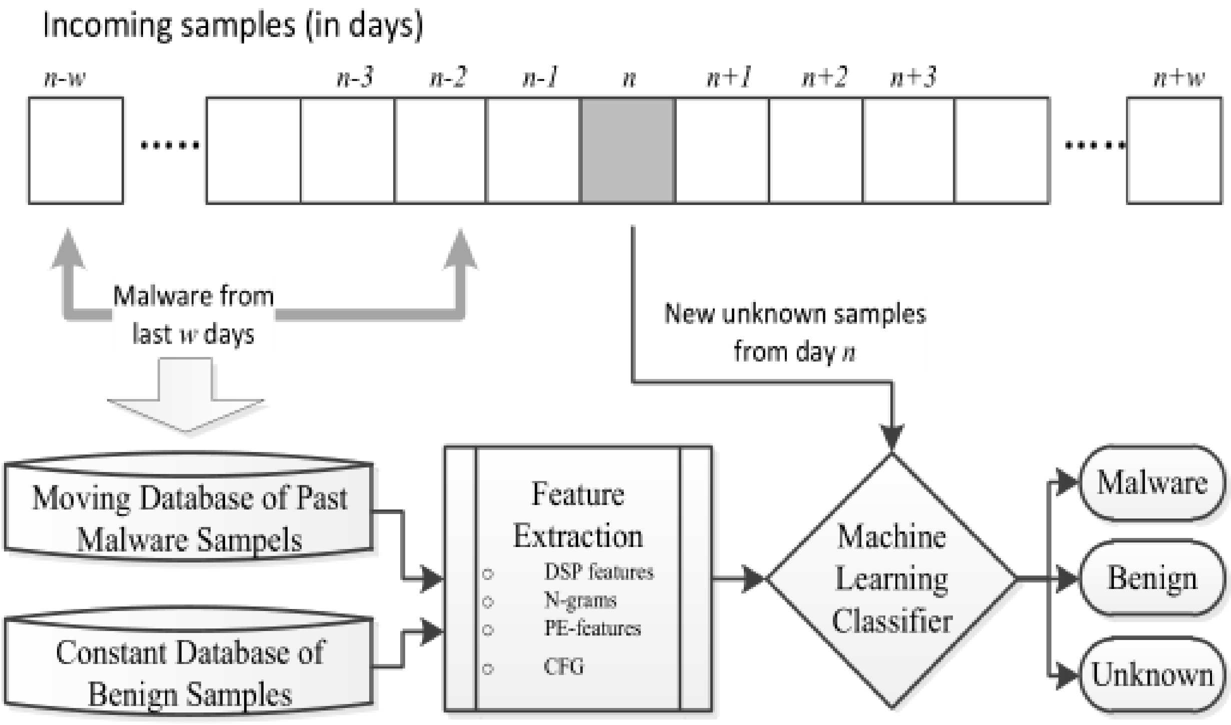

Incoming samples (in days)
n-w
n-3
n-2
n-1
n
n+1
n+2
n+3
n+w
Malware from last w days
New unknown samples from day n
Moving Database of Past Malware Sampels
Constant Database of Benign Samples
Feature Extraction
DSP features
N-grams
PE-features
CFG
Machine Learning Classifier
Malware
Benign
Unknown

Figure 9: Overview of the sliding window experiment on the real world samples.

Results:

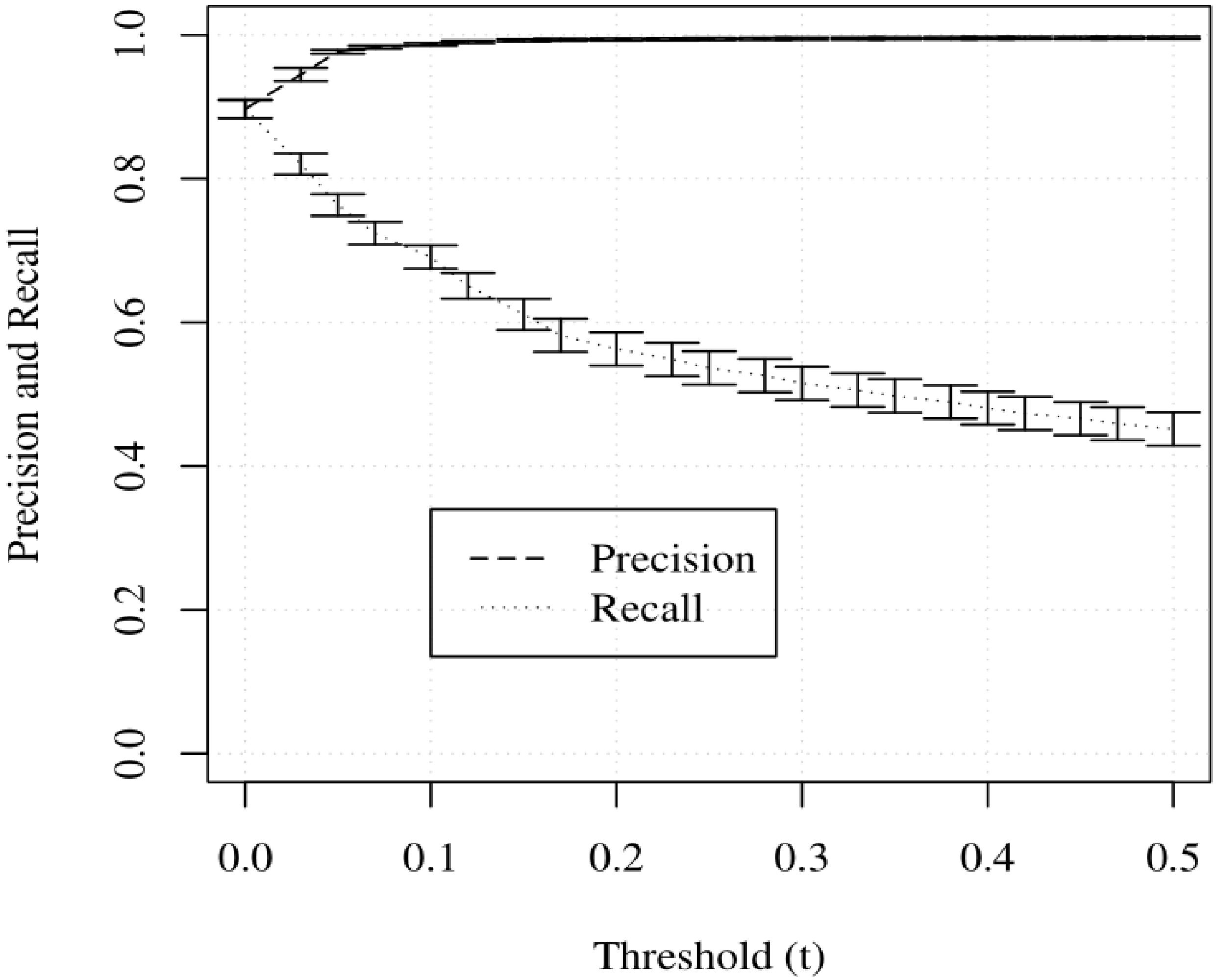


Fig. 10: Precision and recall of the Sigmal detection on the real world samples.

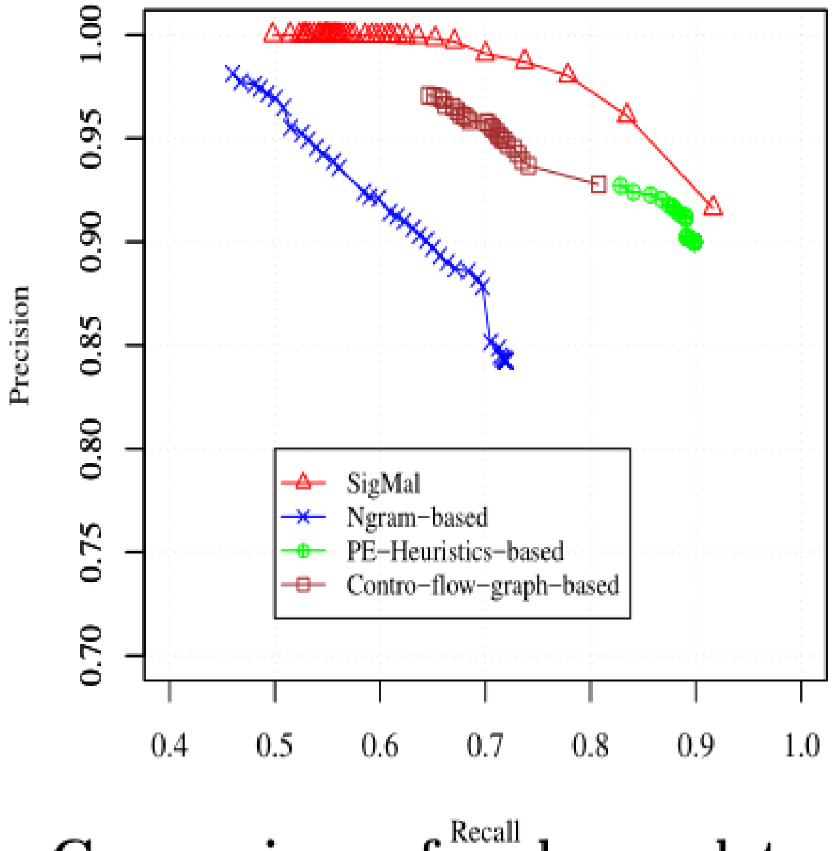


Figure 11: Comparison of malware detection methods with a live malware feed (2012-12-01).

Limitations and Related Work:

- Signal Processing
- Static malware similarity

Conclusion:

- Sigmal detection framework.
- Heuristics based features
- High precision capability.