Semi-supervised Configuration and Optimization of Anomaly Detection Algorithms on Log Data

Master's Thesis Computational Science and Engineering

Carried out at the Faculty of Informatics TU Vienna
Supervised by Prof. Dr. A. Rauber, DDr. F. Skopik, Dr. M. Wurzenberger & Dr. M. Landauer

Author

LinkedIn

AIT

Viktor Beck MSc

- **Bachelors:** Physics
- **Key Area Informatics**
  - Machine Learning (A. Rauber)
  - Data-oriented Programming Paradigms (A. Hanbury)
  - Algorithmics (G. Raidl)
  - Efficient Programs (M. Ertl)
- **Key Area Electronics**
  - Intro. to FEM in Solid Mechanics (D. Pahr)
  - Intro. to Semiconductor Physics and Devices (M. Waltl)
  - FEM for Multi-Physics (F. Toth)
- **Current Occupation:** Junior Scientist at Austrian Institute of Technology AIT
- **Favorite Lectures:** Machine Learning, Algorithmics

Introduction

Cyber threats are evolving rapidly, making anomaly detection (AD) in system log data increasingly important for detection of known and unknown attacks [1]. The configuration of AD algorithms heavily depends on the data at hand. It often involves a complex feature selection process and the determination of parameters such as thresholds or window sizes. In many cases, configuration requires manual intervention by domain experts. This work therefore introduces the Configuration-Engine (CE), which employs a semi-supervised approach to automate the configuration process or optimize existing configurations. The CE utilizes statistical methods to identify log line properties to recognize meaningful tokens for AD methods to monitor. It categorizes variables by their characteristics and behavior over time, then specifies which log parts a detector should observe, and sets appropriate configuration parameters.

Note, that this poster is a condensed version of the paper "Semi-supervised Configuration and Optimization of Anomaly Detection Algorithms on Log Data" that was created with the contents of my thesis.

Configuration Methods

Since each detector requires different input parameters it has to be assessed individually which parameter values are suitable. The first step in finding the right parameters is to assess which variables to choose. This configuration step can be automated by mapping the variable properties to the corresponding detection method and thereby classifying the variables into certain feature sets that suit the corresponding detector. The mapping process can be generalized for all detectors:

1. Choose detector.
2. Define data characteristic from detection method.
3. Expand characteristic to a measure of stability.

A simple example incorporating these steps is provided by the NewMatchPathValueDetector (1) of the AMiner [2] which triggers an alert whenever a new and unknown value of a specified variable is found in a log line [2]. Consequently, we do not want to pass certain variables to this detector. Imagine a feature that has a different value in every occurrence. For this detector any learning would be irrelevant with this kind of values and it would trigger false-positives for every occurrence. The corresponding characteristic is therefore based on "unique occurrence" (3) (see Fig. 1). The suitable measure of stability hence is the asymptotic decrease of new unique occurrences of variables over time (4). In other words, the occurrence of new and unknown values has to stabilize after some time or some number of events.

Configuration Method: Stability

The stability of a variable depends on the considered characteristic. We call a variable "stable" regarding that characteristic if the corresponding curve approaches a constant value within the training period. Fig. 1 exemplarily shows the behavior of different values regarding the number of unique occurrences against the number of occurrences.

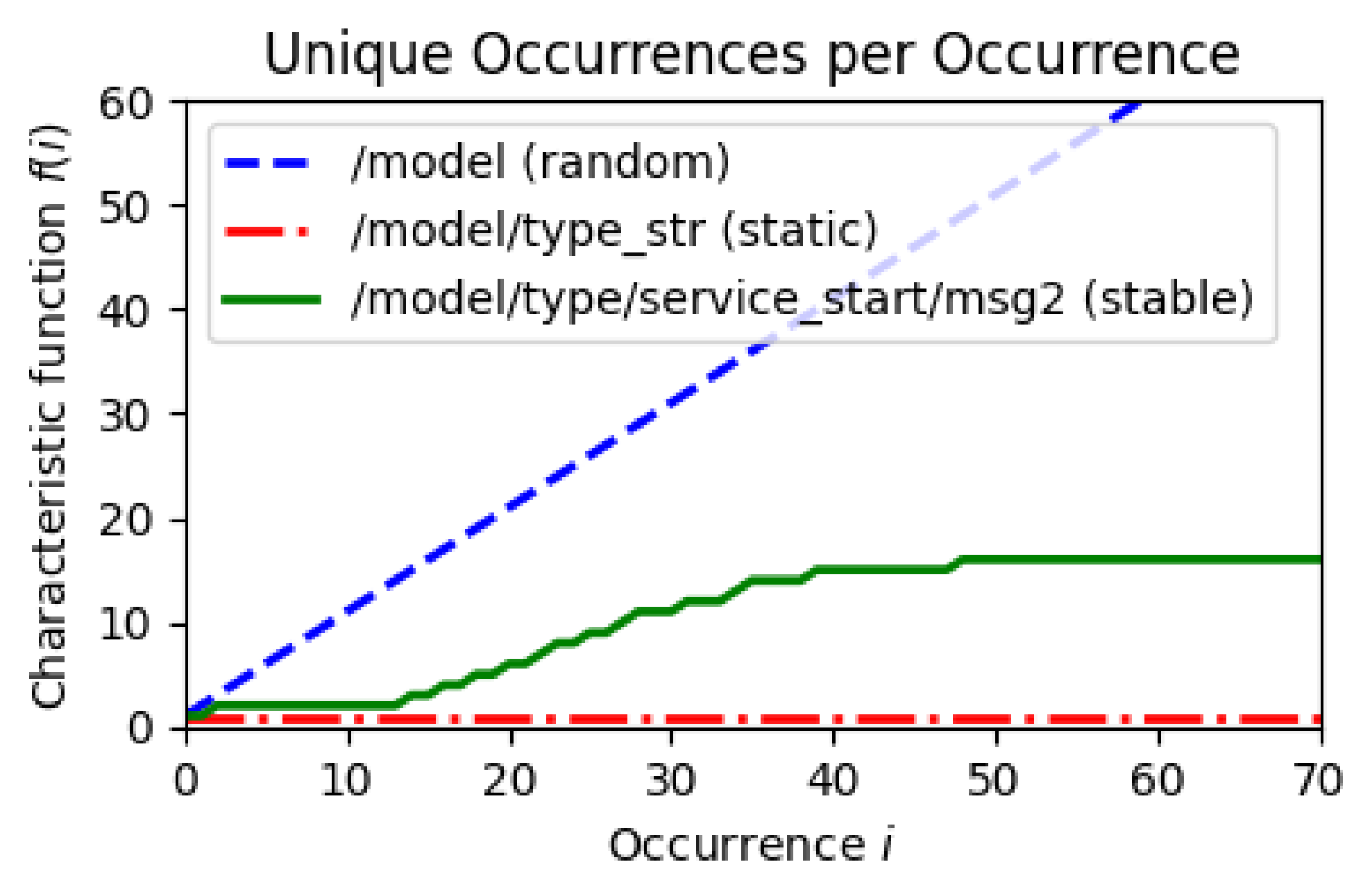

Figure: Unique occurrences per occurrence of a static (red), stable (green) and random variable (blue).

To check whether a variable is "stable", a threshold curve is defined that acts as an upper limit for the curve $f(i)$, representing the data characteristic we are interested in. $i \in \mathbb{Z}$ is the number of occurrences of a variable x. This threshold curve is applied to the derivative $f'(i)$ representing the change in $f(i)$ per occurrence. For a stable variable, this curve should therefore approach 0 within the period of the training data. For the detectors covered in this work, we are actually not interested in the magnitude of change but whether a change has occurred or not. Consequently, $f'(i) \in \mathbb{R}$ should be an element of the binary space $\{0,1\}$ and we define the boolean conversion (denoted by the subscripted b), which later allows us to define relative thresholds in the range $[0,1]$:

$$f_b(i) = \begin{cases} 1 & \text{if } f(i) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Thus, the function $f'_b(i)$ is 1 if a change in $f(i)$ occurred at occurrence i or 0 for no change.

Stability is based on the assessment of the mean values of the segments $s_m(i)$ with $m = 0, 1, ..., n_s - 1$. n_s being the number of segments. To be precise, the m-th segment of the function $f'_b(i)$ is:

$$s_m(i) = \begin{cases} f'_b(i) & \text{if } \quad \frac{m}{n_s}|x| \leq i < \frac{m+1}{n_s}|x| \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The set of stable variables is then defined as:

$$V_{stable} = \left\{ x \in \mathcal{V} \mid \frac{1}{|s_m|} \int_{\mathbb{R}} s_m(i)\, di \leq \theta_m \quad \forall m \right\} \tag{3}$$

$|s_m|$ is the length of each segment m. $|s_m|$ is not uniform if it is not divisible by n_s. Therefore, we define quotient $q = |x| : n_s$, remainder $r = |x| \mod n_s$ and:

$$|s_m| = \begin{cases} q+1 & \text{if } m+1 \leq r, \\ q & \text{if } m+1 > r. \end{cases} \tag{4}$$

If each of the segment means $s_m(i)$ is below the thresholds θ_m, the corresponding variable is classified as "stable". The thresholds θ_m represent a discrete threshold curve that serves as an upper boundary for the change in each segment of $f(i)$. $f'_b(i) \in \{0,1\}$ represents this relation as the "relative change per segment". This is also convenient for the selection of the thresholds as we can define them within range $[0,1]$.

For the characteristic described before ("unique occurrence"), we take

$$f(i) = |\{x_0, x_1, x_2, \ldots, x_i\}|. \tag{5}$$

Experimental Results

We compare the performance of the configurations of the CE against the baseline consisting of configurations of three different experts in the field of AD. Furthermore, we optimize each configuration and show the performance:

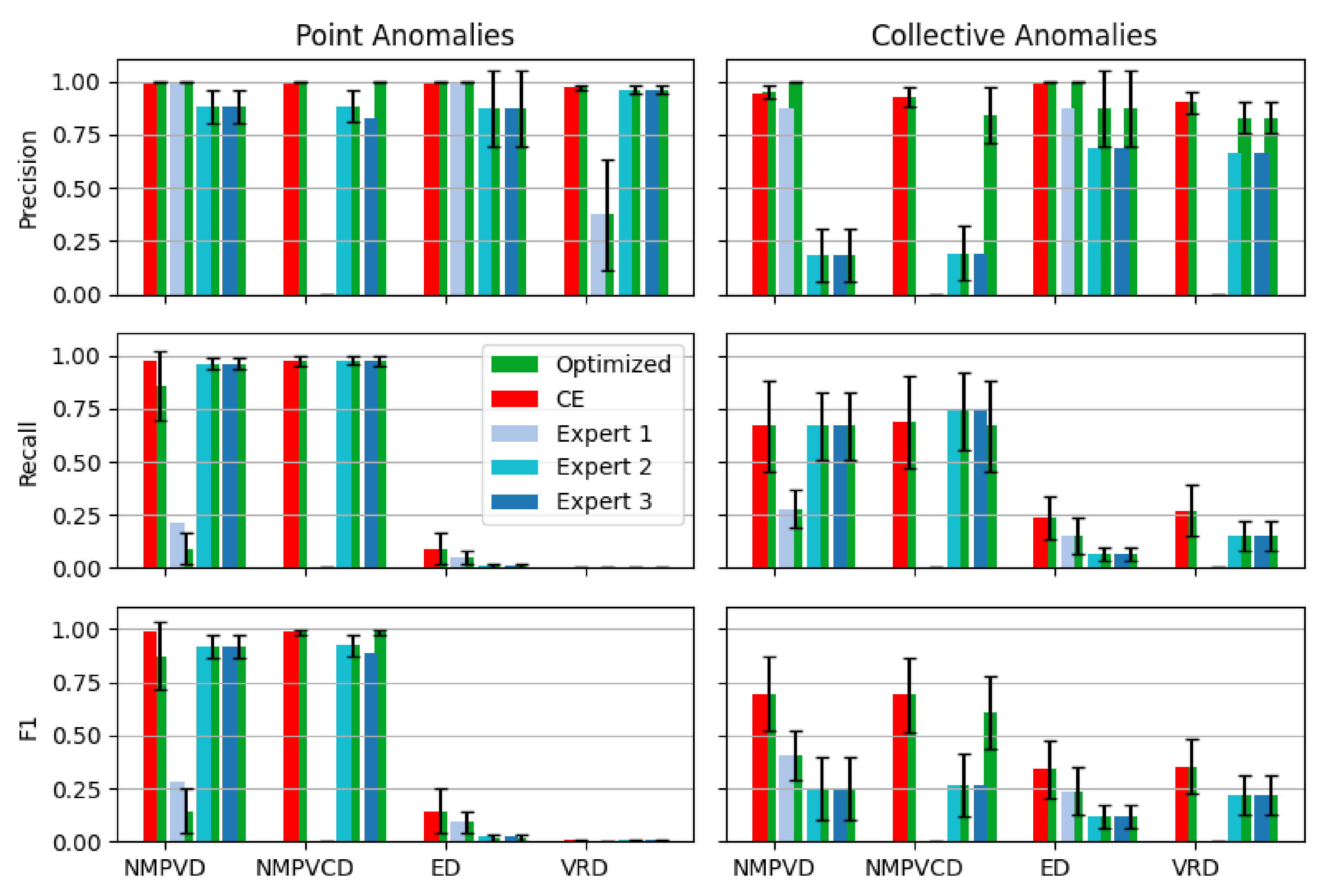

Figure: Performance of four different detectors for Apache log datasets (optimized and non-optimized).

Conclusion & Outlook

The CE was evaluated using four different detectors. Evaluations on different Apache Access datasets containing attack traces showed that the CE achieved an average precision of over 0.94, while maintaining high recall, competing with the performance of expert-crafted configurations. The optimization approach was able to strongly improve the precision of both the CE's and the experts' configurations. Furthermore, the CE's configurations were significantly dissimilar to each other when generated on audit data, highlighting the importance of automated configuration.

References

[1] *CrowdStrike 2024 Global Threat Report.* Retrieved from https://www.crowdstrike.com/resources/reports/crowdstrike-2024-global-threat-report/, accessed 24-June-2024. 2024.

[2] Max Landauer, Markus Wurzenberger, Florian Skopik, Wolfgang Hotwagner, and Georg Höld. "AMiner: A Modular Log Data Analysis Pipeline for Anomaly-Based Intrusion Detection". In: *Digital Threats* 4.1 (Mar. 2023). ISSN: 2692-1626.