HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

Unsupervised Anomaly Detection on Software Logs

Author:
Doctoral researcher Jesse Nyyssölä
Department of Computer Science
University of Helsinki

Supervisor:
Prof. Mika Mäntylä
Department of Computer Science
University of Helsinki

Rationale

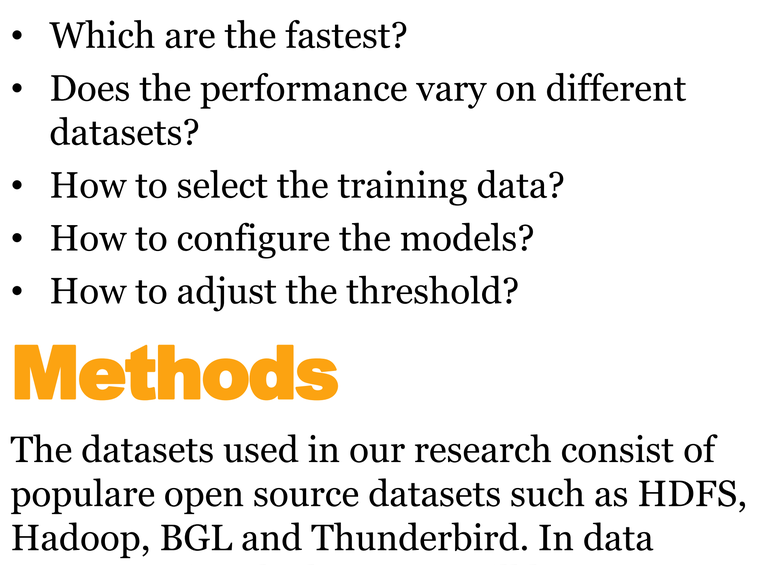
Software system logs are considered as one of the primary sources for determining the cause of failure. **Log event prediction** is among the key strategies of anomaly detection providing means for root cause detection as well as failure identification, tolerance, and recovery. Pinpointing anomalous events can help test and application engineers to identify and fix subtle bugs that result in crashes after a long time. The research is gravitating towards more complex algorithms, despite some evidence suggesting that computationally intensive deep-learning approaches may not offer significant benefits over simpler approaches.

Research interests:

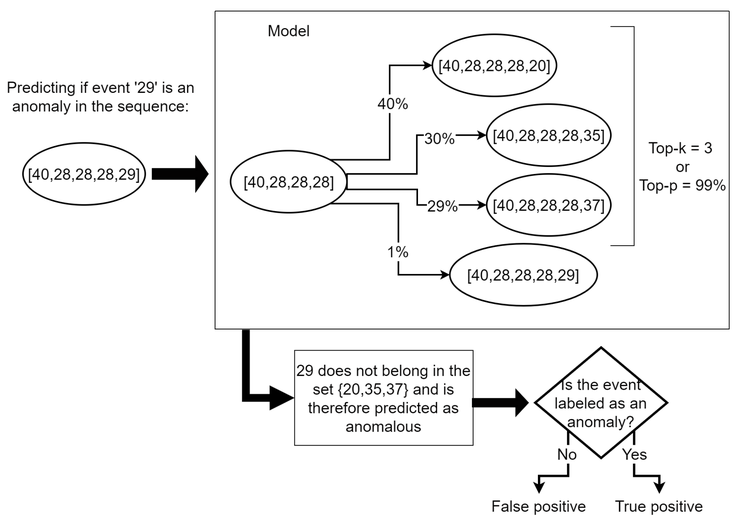
- Which models are the most accurate?
- Which are the fastest?
- Does the performance vary on different datasets?
- How to select the training data?
- How to configure the models?
- How to adjust the threshold?

Methods

The datasets used in our research consist of populare open source datasets such as HDFS, Hadoop, BGL and Thunderbird. In data preprocessing the log event will be transformed to the specified **representation** which can be a list of words, trigrams or an event identifiers. For the anomaly detection itself, my research has focused on two branches: **Next Event Prediction** (NEP) and **Unsupervised Machine Learning** (ML)**.**



With **NEP**, the goal is to predict an event by masking the other events in the selected window. The assumption is that as our model becomes good at predicting normal behaviour, false predictions are likely to be anomalies. The **ML** models score each event or sequence in isolation utilizing approaches such as clustering.

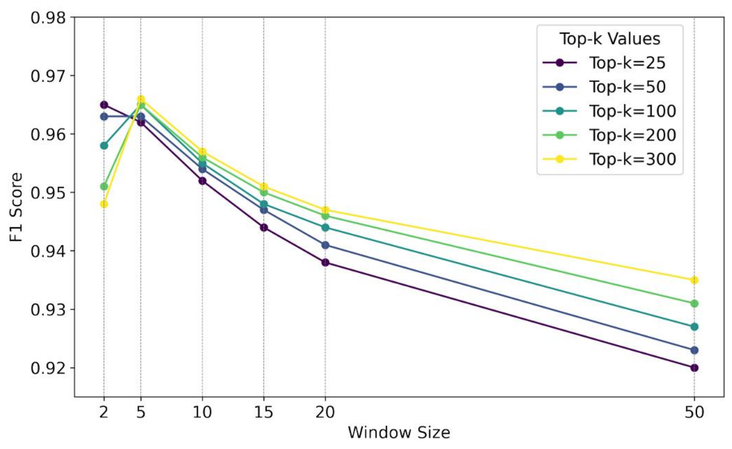


Example of the NEP process with event IDs and window size 5.

To aid in our research, we created a tool for benchmarking anomaly detection algorithms, datasets and data representations: **LogLead**.
https://github.com/EvoTestOps/LogLead

Results

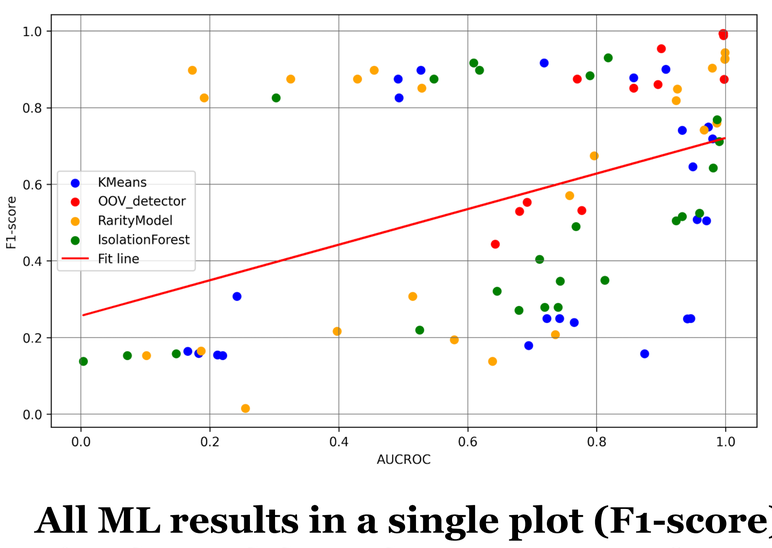
Adjusting the mask position in NEP window can increase the performance.



Small window sizes with NEP yield better results. The set of accepted events (threshold) can be quite large.

	BGL		Hadoop		HDFS		Profilence	
	Default	Baseline	Default	Baseline	Default	Baseline	Default	Baseline
LSTM	0.938	**0.946**	0.817	**0.953**	0.847	**0.955**	0.856	**0.919**
CNN	0.938	**0.943**	0.81	**0.953**	0.845	**0.954**	0.856	**0.913**
N-Gram	**0.935**	0.918	0.819	**0.938**	0.848	**0.954**	**0.851**	0.833

Simple model (N-Gram) can perform as well as deep learning (accuracy). Configuration plays an important role.



All ML results in a single plot (F1-score). Simple models such as OOVD perform well here too. Performance varies greatly between datasets, models, representations and even metric.

Future work

In the future, I intend to look further into the dataset specific differences. Is it in the nature of the data? Can we mitigate that with specific training data selection? What's the effect of non-chronological data leakage?

We have, and are eager to find, more real-world data from industry to work with. It poses new challenges and opportunities.