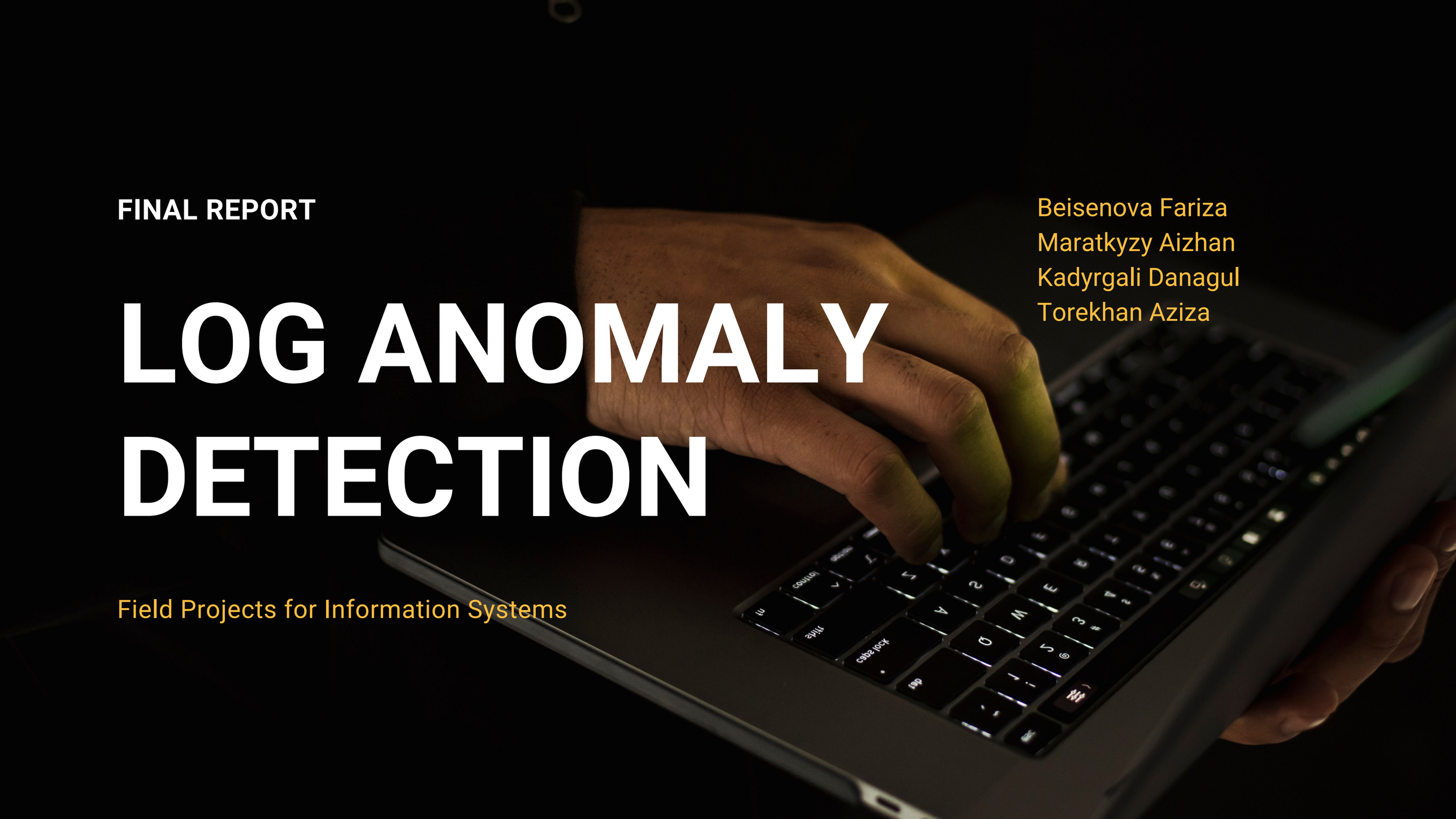
FINAL REPORT

LOG ANOMALY DETECTION

Field Projects for Information Systems

Beisenova Fariza
Maratkyzy Aizhan
Kadyrgali Danagul
Torekhan Aziza



WHAT'S LOG ANOMALY DETECTION

Introduction

01 Anomaly detection in log file deals with finding text which can provide clues to the reasons and the anatomy of failure of a run.

02 In this study, we suggest using a neural network language model developed from "successful" log files to predict anomalies in a "failed" run.

03 Logs are widely used for anomaly detection, recording system runtime information, and errors. Finding text in a log file that can offer hints about the causes and the physical characteristics of a run's failure is known as anomaly detection.

What is Anomaly Detection?

[RED, YELLOW,
GREEN, BLUE, CHAIR]

Dataset

Most of the log files are original text and unstructured files. We take a dataset from the Blue Gene/L (BGL) supercomputer system of Lawrence Livermore National Laboratory (LLNL) as an example. Lists some fragments of the BGL dataset:

—1117838570 2005.06.03 R02-M1-N0-C:J12-U11 2005-06-03-15.42.50.675872 R02-M1-N0-C:J12-U11 RAS KERNEL INFO instruction cache parity error corrected

—1117843015 2005.06.03 R21-M1-N6-C:J08-U11 2005-06-03-16.56.55.309974 R21-M1-N6-C:J08-U11 RAS KERNEL INFO 141 double-hummer alignment exceptions

—1117956980 2005.06.05 R24-M1-NB-C:J15-U11 2005-06-05-00.36.20.945796 R24-M1-NB-C:J15-U11 RAS KERNEL INFO generating core.728

—KERNDTLB 1118538836 2005.06.11 R30-M0-N9-C:J16-U01 2005-06-11-18.13.56.287869 R30-M0-N9-C:J16-U01 RAS KERNEL FATAL data TLB error interrupt

—KERNSTOR 1118765360 2005.06.14 R04-M0-ND-C:J15-U01 2005-06-14-09.09.20.459609 R04-M0-ND-C:J15-U01 RAS KERNEL FATAL data storage interrupt

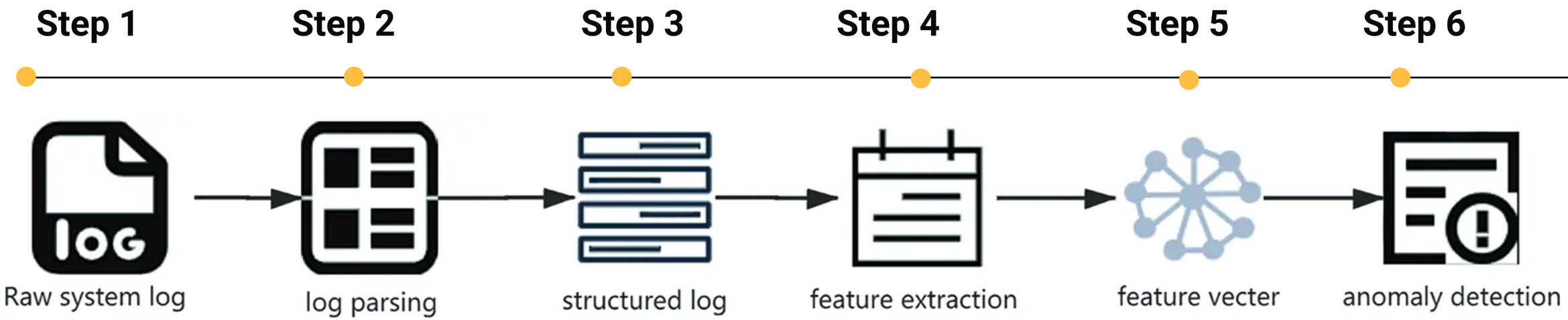
The timestamp is 1118765360, the record date is 2005.06.14, the node is R04-M0-ND-C:J15-U01, the type of the log message is RAS, the location where the message is generated is KERNEL, the corresponding level of the log message is FATAL, and the content is “data storage interrupt”. We remove irrelevant information and keep useful log messages, that is, “data storage interrupt”.

Table 1: Details of BGL log

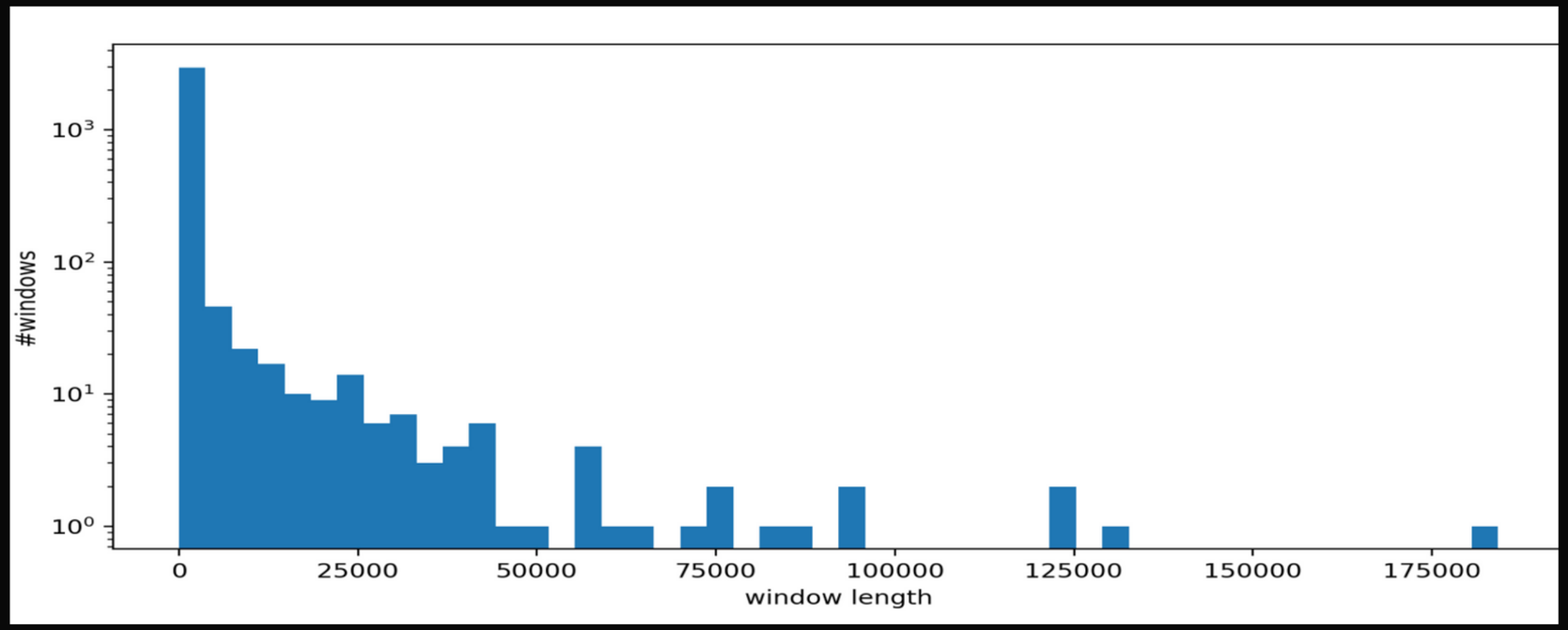
Project	Data
Timestamp	1118765360
Date	2005.06.14
Node	R04-M0-ND-C:J15-U01
Information type	RAS
Generate location	KERNEL
Information level	FATAL
Content	Data storage interrupt

Models (Methods)

Log anomaly detection models typically use unsupervised learning techniques to identify anomalies in log data. One common approach is to use clustering algorithms to group similar log events together and identify outliers that do not fit into any of the clusters. Another approach is to use dimensionality reduction techniques to visualize the log data in a lower-dimensional space, making it easier to identify patterns and anomalies.

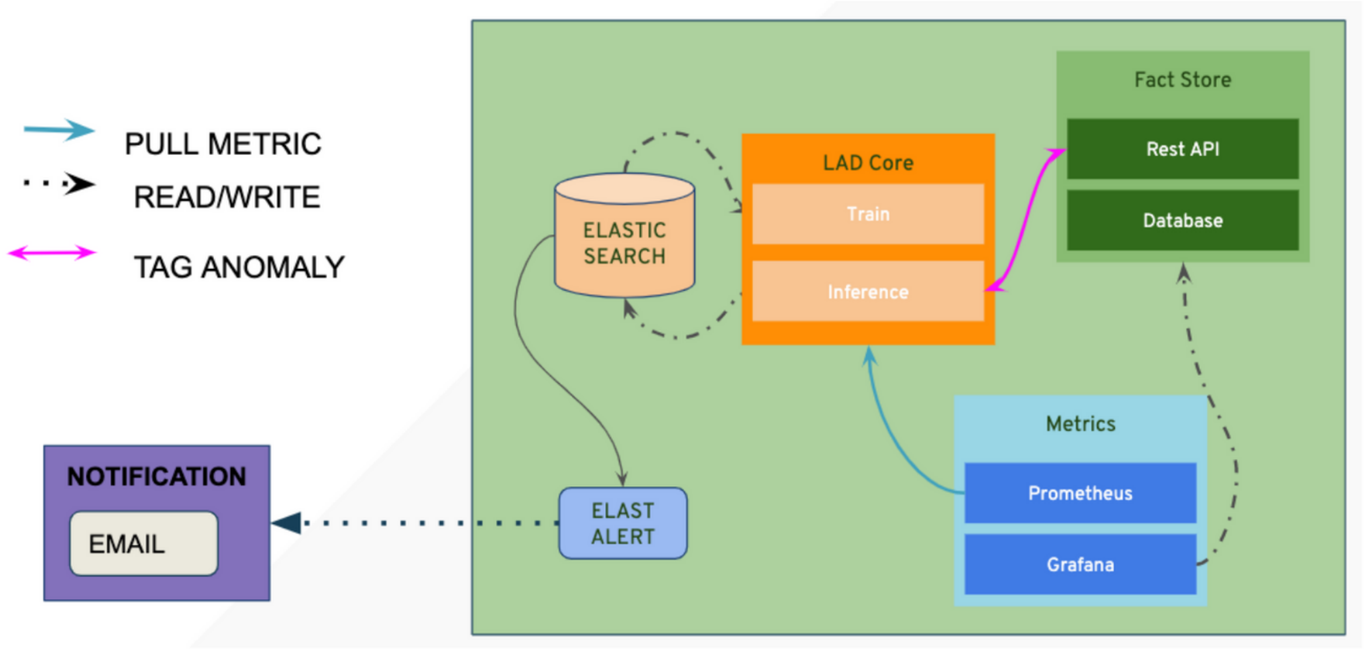


Visualization



Window length distribution BGL

System Architecture Overview



This is a high level overview of the architecture for log anomaly detector which consists of three components.

1. Machine Learning (LAD-Core) Contains custom code to train model and predict if a log line is an anomaly. Also can using W2V (word 2 vec) and SOM (self organizing map) with unsupervised machine learning.
2. Monitoring (Metrics) To monitor this system in production we can utilize grafana and prometheus to visualize the health of this machine learning system.
3. Feedback Loop (Fact-Store) In addition we have a metadata registry for tracking feedback from false_positives in the machine learning system and to providing a method for ML to self correcting false predictions called the “fact-store”.

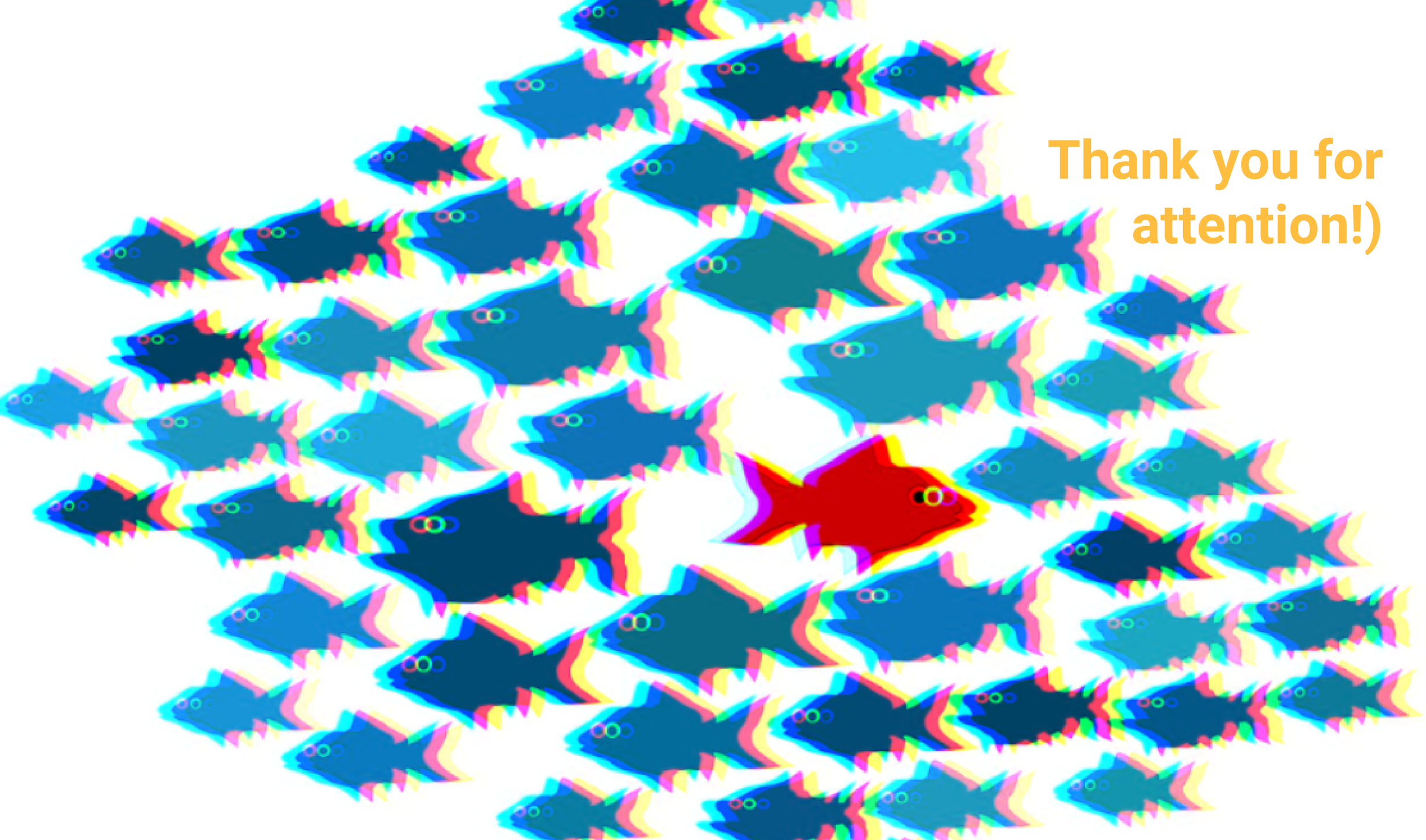
Results

- This research provides a high-level overview of the architecture for log anomaly detection, which consists of three components. These include machine learning (LAD-Core), monitoring (Metrics), and feedback loop (Fact-Store).

- We have also discovered some of the challenges involved in log anomaly detection, such as the need for data preprocessing and feature extraction. However, we can solve these problems by using appropriate machine learning models and continuously improving the model through feedback and retraining.

Conclusion:

In this work we have analyzed models and methods for anomaly detection and proposed an approach for discovering security breaches in log records. Such approach generates rules dynamically and able to learn new types of attacks, while minimizing the need of human actions.



Thank you for attention!)