A Year of Automated Anomaly Detection in a Datacenter

Rufaida Ahmed Joseph Porter Abubaker Abdelmutalab Robert Ricci

School of Computing

University of Utah

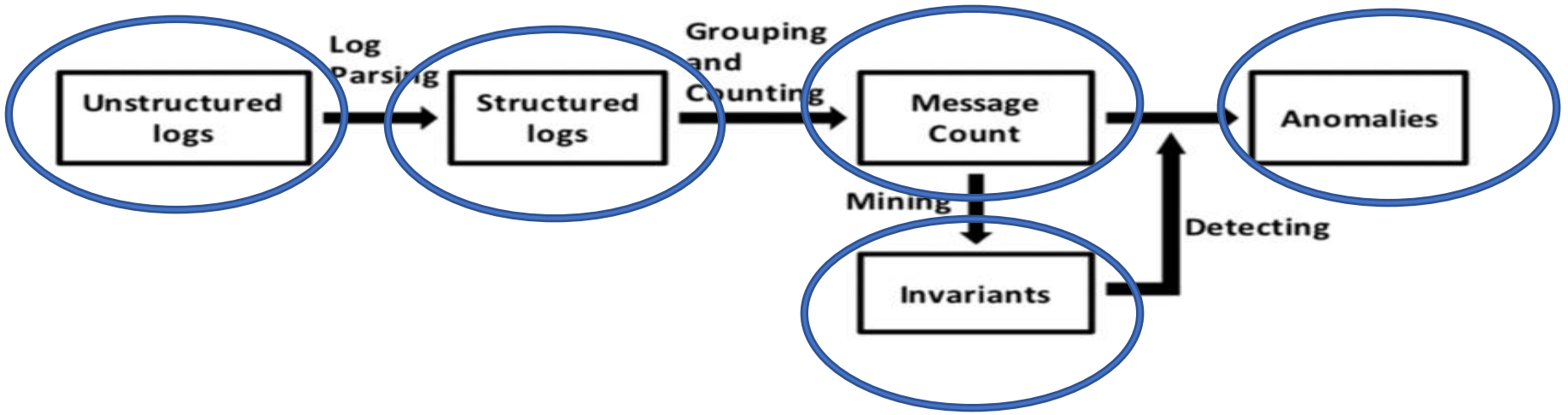
PROBLEM STATEMENT

- What is an anomaly?
- Why is it hard to detect?
- Understanding “normal” and anomalous behavior is not always straightforward.
- Manual inspection is tedious.
- We are dealing with millions of log entries.

Anomaly Detection By *Invariant Mining*

- Discover underlying linear characteristics of program workflows.
- Now, we have a definition of a correct behavior.
- We can automatically detect system anomalies.
- Can easily recalculate invariants upon changes or updates.
- Produces interpretable models.

Lou, J. G., Fu, Q., Yang, S., Xu, Y., & Li, J. (2010, June). Mining Invariants from Console Logs for System Problem Detection. In *USENIX Annual Technical Conference* (pp. 1-14).



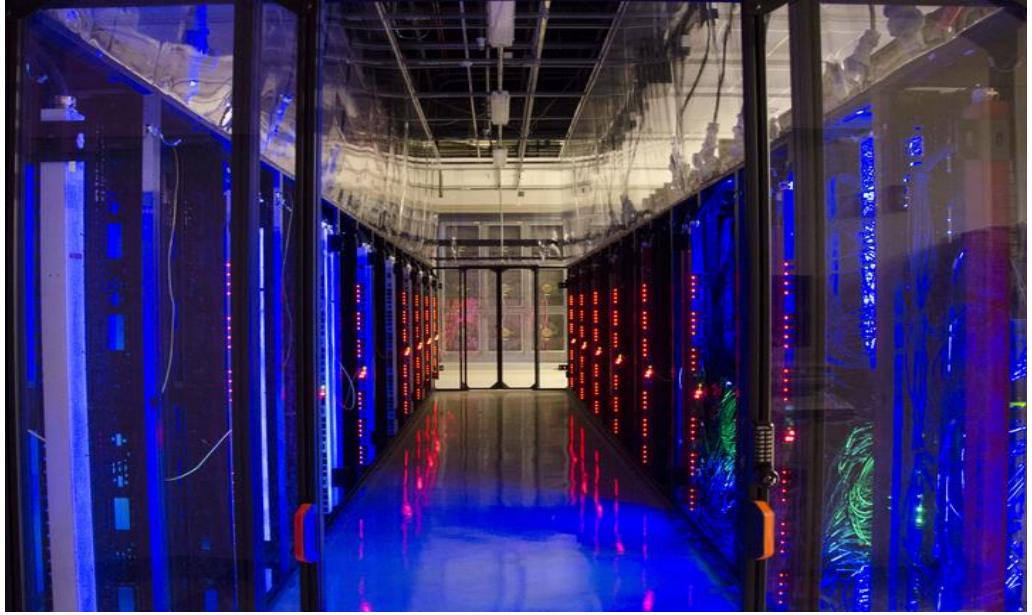
We try to answer these questions:

- Does invariant mining successfully create discriminators capable of distinguishing "normal" behavior from anomalous behavior?
- Do the invariants found provide information that is interpretable by system admins?
- Do the set of invariants change over time?

RELATED WORK

- Several statistical and machine-learning models have been proposed to analyze systems[Bates et al., 1983], [A. Brown, 2018].
- We look at change over time from a full year of data.
- After considering most of the proposed machine learning methods, we used invariant mining.
- Our accuracy results are validated with expert human administrators.

Preparing The Dataset



- Cloudlab: is a facility used by thousands of researchers and educators in computer science.
- It provisions resources at a “bare metal” level.
- Log files are coming from CloudLab.(bare metal provisioning process)
- Data is collected, processed and stored using the ELK.
- Data is parsed and cleaned using 48 unique log patterns.

Resulting Dataset

- Four logfiles that are related to the process of *provisioning* and *booting* nodes.
- The resulting dataset contains over 15 million log entries for 583 nodes and forms 51,375 sessions.

Methodology

- Split logs into sessions.
- Pass log entries to invariant miner.
- It produces invariants such as:

$$c(A) - c(B) = 0$$

- Slightly more complicated invariant:

$$c(A) - 2c(B) = 0$$

Comparison Across Time

- Data from year 2019 was divided into four quarters
- Trained the invariant miner with each quarter's data independently.
- These results were used to study how usable the invariants are.
- We compare invariants from each quarter and analyze the reasons behind the difference in invariants.
- The logs data is divided into session which contain all log entries for a particular server in a single day.

Findings

- Invariant miner output:

$$a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_m x_m = 0$$

```
(11, 29): [ 1.0, -1.0]
(17, 18): [ 1.0, -1.0]
(1,  65): [-4.0,  1.0]
```

Findings

Dataset	Jan-Mar		Apr-Jun		Jul-Sep		Oct-Dec	
	sessions	anomalies	sessions	anomalies	sessions	anomalies	sessions	anomalies
Training dataset	5220	3.8%	5713	4.4%	6154	2.7%	8600	4.7%
Test dataset	5220	3.1%	5713	5.0%	6154	1.9%	8601	4.8%

TABLE I

NUMBER OF SESSIONS AND PERCENTAGE OF ANOMALIES FOUND PER QUARTER IN OUR DATASET.

Usefulness and Interpretability

- For an invariant to be considered useful:
  1. They must be *non-trivial* in the sense that it is *possible* to violate them. (six distinct invariants)
  2. An invariant must be *sensible*. We evaluate this by looking at the expected ratio produced by the miner. (15 invariants were filtered out)
  3. Invariants must be *interpretable*, meaning that administrators are able to understand. (harder to evaluate quantitatively, so we examine it qualitatively.)
- We found that while some invariants were “useful”, not all were.

Accuracy of Anomaly Detection

- We consider a “normal” label as negative result and an anomalous label as positive results.

Precision	Recall	F1-score	False Positive	False Negative
0.7087	0.7300	0.7192	30%	27%

Interesting Findings

1. We found that the administrators made a distinction between behavior that indicated a problem with the system and unusual user behavior:
   - For future work, distinguish known-benign classes of anomalies from those that might require intervention.
2. The sessions that were mislabeled by invariants tended to fit very specific patterns:
   - For future work, relatively simple heuristics could be used to greatly improve the accuracy rates.

Evolution of Invariants Over Time

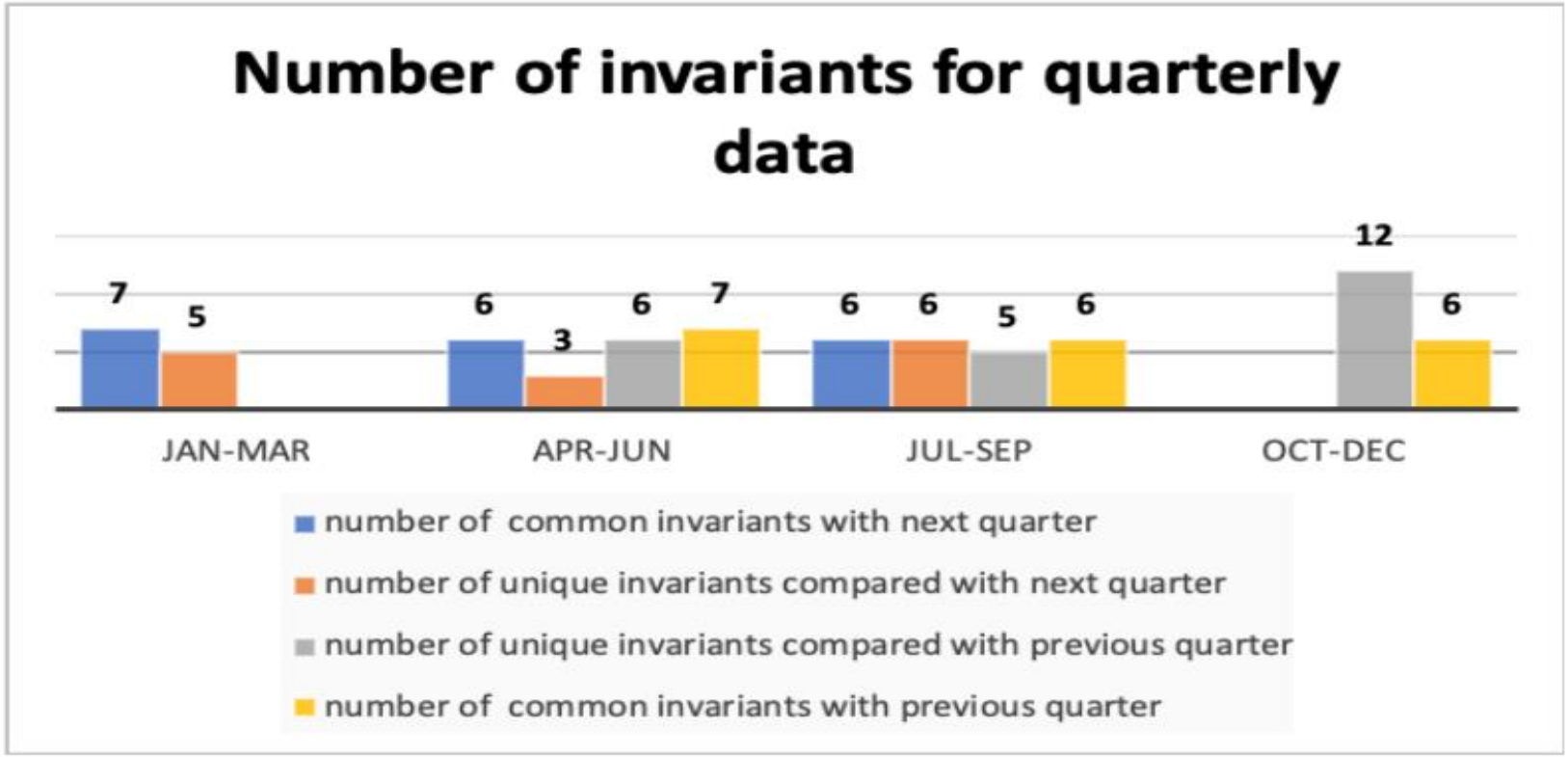


Fig. 2. Comparison of number of invariants throughout one year. Note that the first quarter has no preceding quarter, and the last has no succeeding one.

Conclusion and Future Work

- **Does invariant mining successfully create discriminators capable of distinguishing "normal" behavior from anomalous behavior?**
  - ❑ Yes, and it is fairly accurate on our real-world dataset, agreeing with the "anomaly" labels assigned by system administrators more than 70% of the time.
- **Do the invariants found provide information that is interpretable by system admins?**
  - ❑ We found five invariants that we deemed highly interpretable by these criteria
- **Do the set of invariants change over time?**
  - ❑ Anomaly rates vary substantially between quarters (from 1.9% to 5%), and that the set of invariants that describes these anomalies varies.