# LogStamp: Automatic Online Log Parsing Based on Sequence Labeling


Shimin Tao, Weibin Meng

Huawei

Yimeng Chen

Huawei

Yichen Zhu

University of Toronto

Ying Liu

Tsinghua University

Chunning Du

Beijing University of Posts and Telecommunications

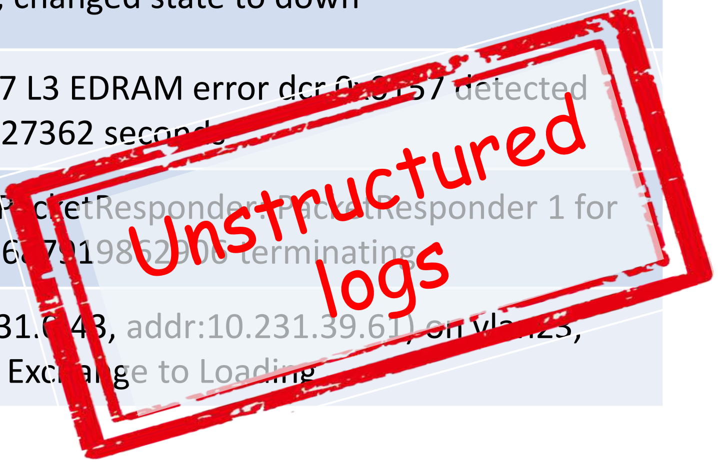
Tao Han, Yongpeng Zhao, Xiangguang Wang, Hao Yang

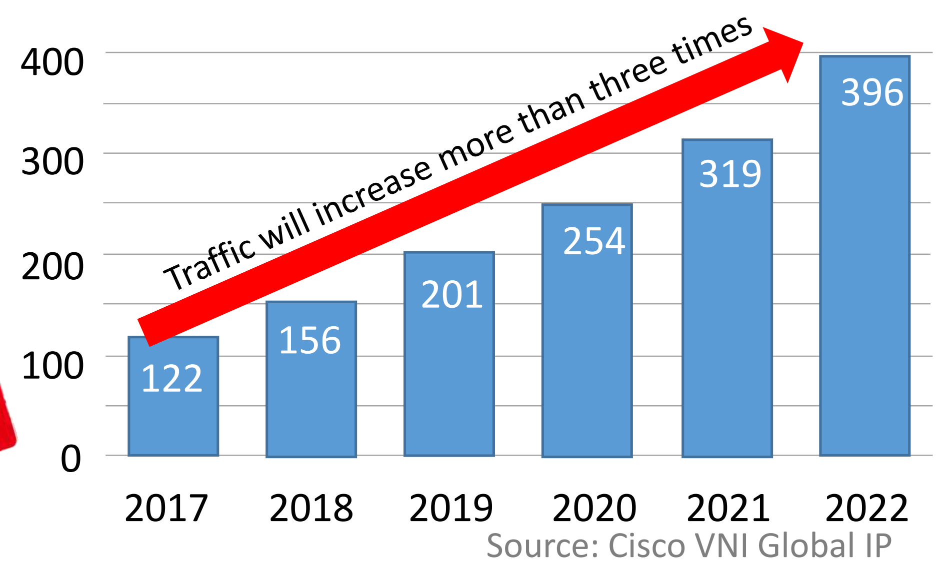
Huawei

# Background

- Internet provides various types of services.
- Logs, which are usually designed by code developers in system source code, records valuable service runtime information, and thus is critical for service management.
- As the volume of unstructured log increase rapidly, it became hard to manually perform log parsing task.

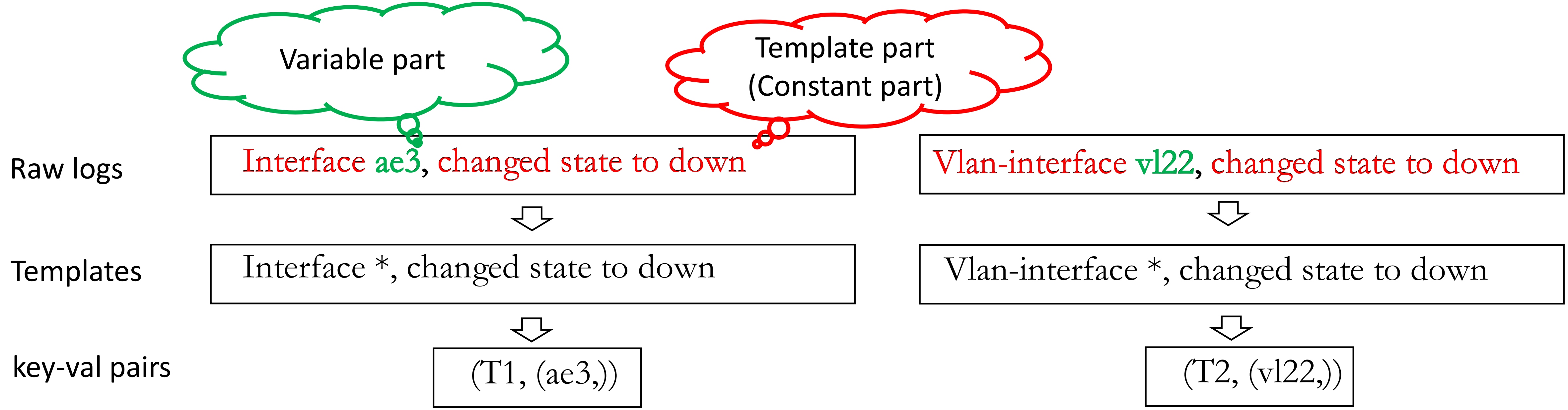
| Types | Timestamps | Detailed messages |
|---|---|---|
| Switch | Jul 10 19:03:03 | Interface te-1/1/59, changed state to down |
| Supercomputer | Jun 4 6:45:50 | RAS KERNEL INFO 87 L3 EDRAM error dcr 0x0157 detected and corrected over 27362 seconds |
| HDFS | Jun 8 13:42:26 | INFO dfs.DataNode$PacketResponder: PacketResponder 1 for block blk_-1608999687919862906 terminating |
| Router | Jul 11 11:05:07 | Neighbour(rid:10.231.0.43, addr:10.231.39.61) on vlan23, changed state from Exchange to Loading |



# Background

- **Log analysis** aims at automatically monitoring real-time performance of Web Service Systems. There are plenty of log-based analysis tasks, for example, anomaly detection, fault diagnosis and failure prediction.
- **Log parsing** is a prior step of log analysis. Its goal is to distinguish between constant part and variable part in log texts. Then, logs can be presented in the format of (key: value) pairs, where the key is the template key number, value is the variable set.

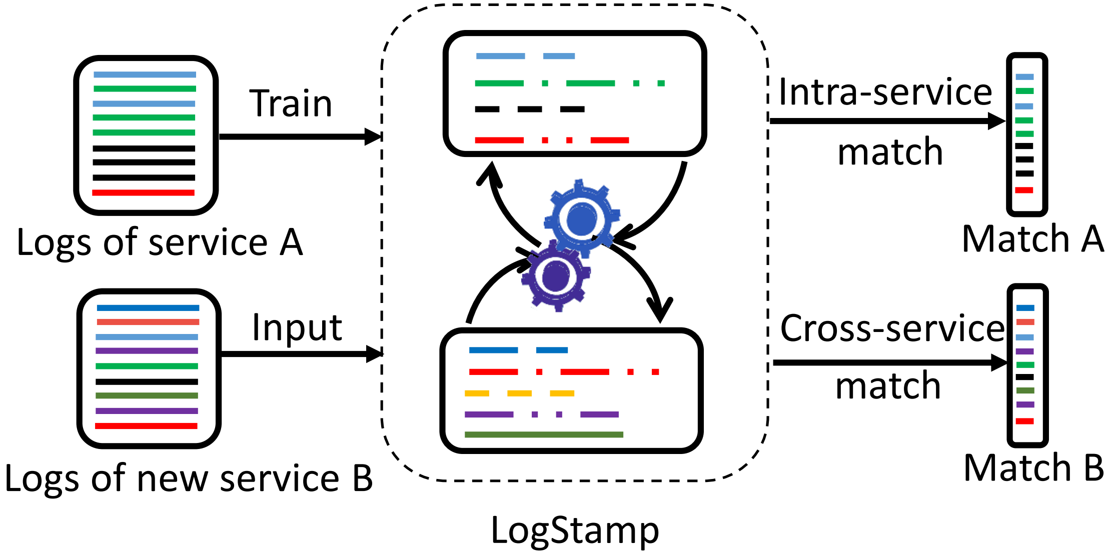


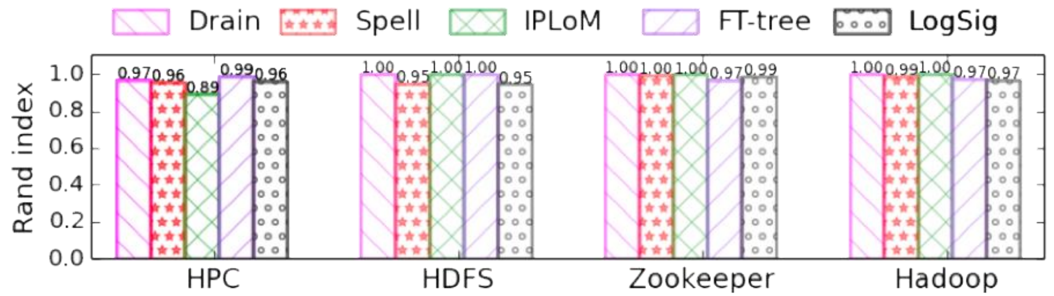
# Background

**Two key challenges of current log parsing approaches:**

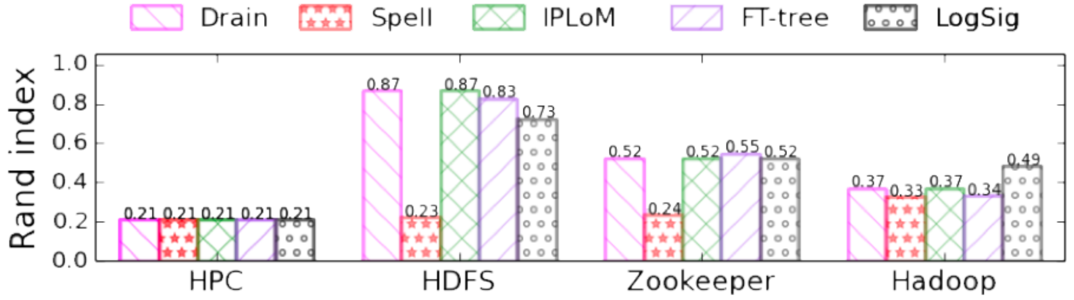
I. **Intra-service Adaptiveness**: Software/firmware updates introduce new types of logs. Most of the existing approaches do not support online analysis or cannot handle new logs without re-training their model.

II. **Cross-service Adaptiveness**: Multiple rules/models have to be defined or trained for different services. A model trained for service A is not able to parse logs of service B.




traditional template extraction results when all the logs are used for training




traditional template extraction results when only 10% of logs are used for training

# Proposal

- **Observation**: Operators usually distinguish variables based on features of words.
- We define the log parsing problem as a **sequence labeling problem**, i.e. we train a model to label each word in log texts to determine whether it is a part of template or a part of variable.

Character and number mixtures are usually variables

Historical logs:
$L_1$. Interface ae3, changed state to down
$L_2$. Vlan-interface vl22, changed state to down
$L_3$. Interface ae3, changed state to up
$L_4$. Interface ae1, changed state to down
Real-time logs:
$L_5$. Interface ae1, changed state to up
$L_6$. Vlan-interface vl22, changed state to up

letters are usually template words

# Proposal

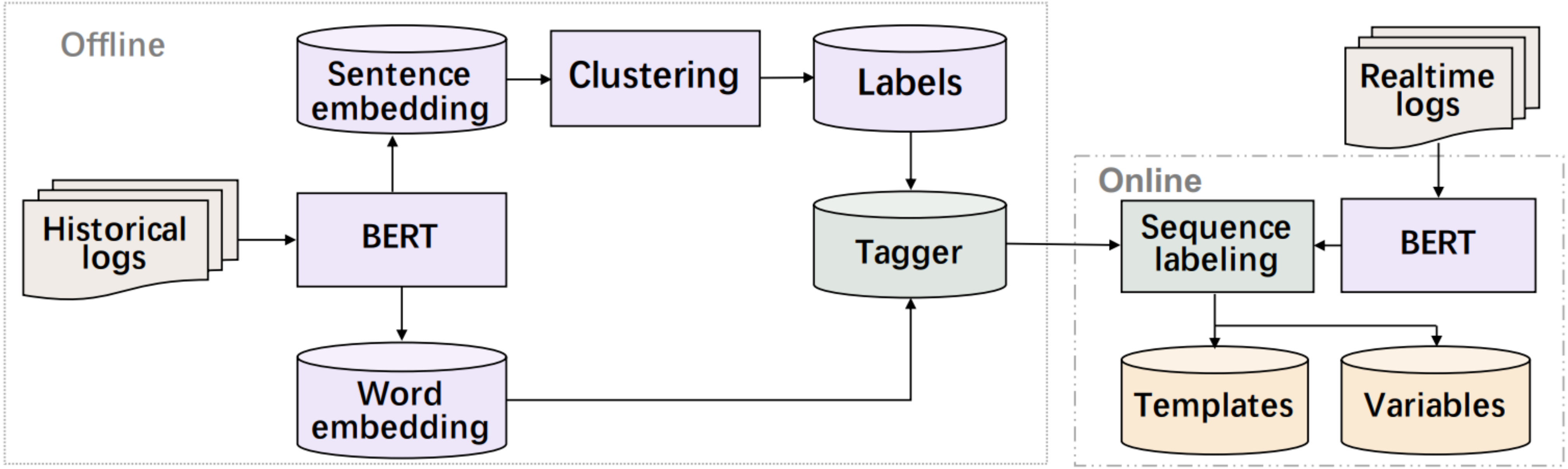



**Figure 2: The workflow of LogStamp**

**Offline learning:**
- Prepare pseudo labels through BERT-based Sentence embedding and Clustering algorithm.
- Train word classifier (Tagger) with BERT-based word embedding and generated pseudo labels.

**Online parsing:**
- Label log text words with a trained Tagger
- Match with exists templates and record a text in (key, val) format.

# Experimental Results

## Experimental Settings

- **Datasets.** We conduct experiments over five public log datasets, namely BGL, HDFS, ZooKeeper, Proxifier and Hadoop. Manually sampled and labeled log templates are served as ground truth label for our evaluation.
- **Baseline.** FT-Tree, Drain, Spell, LogSig, LogParse, MoLFI, and IPLoM.
- **Model.** We experiment three versions of BERT, i.e. BERT-base, BERT-small and BERT-tiny. For tagger, we compare GCN, CNN, LSTM and RNN.
- **Evaluation Metrics.** We use **RandIndex** to quantitatively evaluate our proposal.

# Experimental Results

**Table 2: Offline accuracy of LogStamp with different BERT versions**

| Methods | HDFS | Proxifier | Zookeeper | BGL | Hadoop |
|---|---|---|---|---|---|
| BERT-tiny | 0.9999 | 0.9356 | 0.9998 | 0.9950 | 0.9988 |
| BERT-base | 0.9999 | 0.9836 | 0.9998 | 0.9994 | 0.9987 |
| BERT-small | 0.9999 | 0.9840 | 0.9998 | 0.9979 | 0.9988 |

**Table 3: Online accuracy of LogStamp with different BERT versions**

| Methods | HDFS | Proxifier | Zookeeper | BGL | Hadoop |
|---|---|---|---|---|---|
| BERT-tiny | 0.8888 | 0.9042 | 0.9906 | 0.9788 | 0.9762 |
| BERT-base | 0.8798 | 0.9141 | 0.9760 | 0.9816 | 0.9637 |
| BERT-small | 0.9147 | 0.8820 | 0.9851 | 0.9586 | 0.9752 |

**Table 4: Online Accuracy of LogStamp with different taggers**

| Methods | HDFS | Proxifier | Zookeeper | BGL | Hadoop |
|---|---|---|---|---|---|
| GCN | 0.8888 | 0.9042 | 0.9906 | 0.9788 | 0.9762 |
| RNN | 0.9822 | 0.9180 | 0.9790 | 0.9978 | 0.9962 |
| LSTM | 0.9949 | 0.9998 | 0.9998 | 0.9996 | 0.9974 |
| CNN | 0.9921 | 0.9164 | 0.9998 | 0.9996 | 0.9974 |

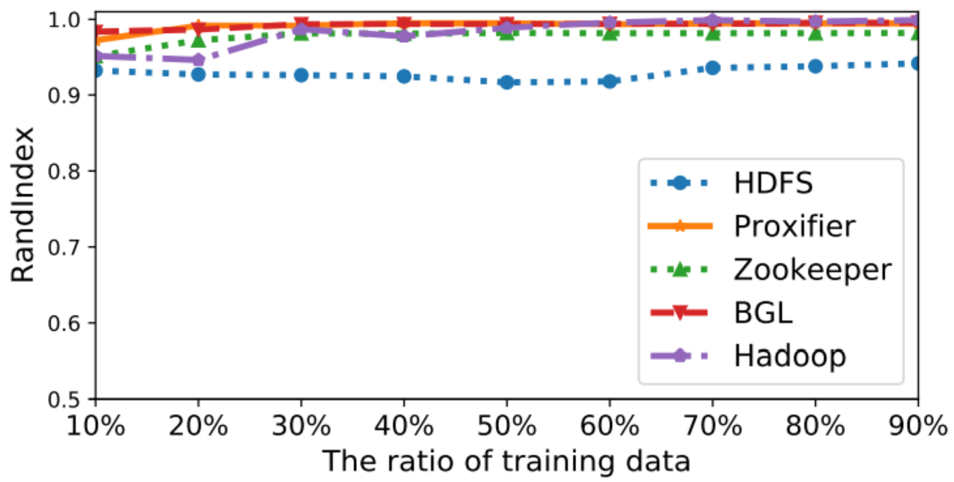



**Figure 5: The log parsing accuracy of LogStamp as the ratio of training data changes**

# Discussion and Conclusion

- Thanks to the powerful BERT!
- We assume that compared to natural language texts which are used to pre-train BERT, log texts (even if from different Services) are usually contain a much plainer semantic information and syntactic structure.
- We then propose a LogStamp framework. We treat the log parsing as a sequence labelling task and employ a pre-trained language model to perform the task. Experimental results on public log dataset illustrate the accuracy of our approach on log parsing task, while it also demonstrate its ability to deal with the problem of intra- and cross-service adaptiveness.

# References

[1] Jieming Zhu, Shilin He, Jinyang Liu, Pinjia He, Qi Xie, Zibin Zheng, and Michael R Lyu. Tools and benchmarks for automated log parsing. In Proceedings of the 41st International Conference on Software Engineering(ICSE), pages 121–130, 2019.

[2] Weibin Meng, Ying Liu, Federico Zaiter, Shenglin Zhang, Yihao Chen, Yuzhe Zhang, Yichen Zhu, En Wang, Ruizhi Zhang, Shimin Tao, et al. Logparse: Making log parsing adaptive through word classification. In 2020 29th International Conference on Computer Communications and Networks (ICCCN), pages 1–9. IEEE, 2020.

[3] Qingwei Lin, Hongyu Zhang, Jian-Guang Lou, Yu Zhang, and Xuewei Chen. Log clustering based problem identification for online service systems. In Proceedings of the 38th International Conference on Software Engineering Companion (ICSE), pages 102–111. ACM, 2016.

[4] Min Du and Feifei Li. Spell: Streaming parsing of system event logs. In 2016 IEEE 16th International Conference on Data Mining (ICDM), pages 859–864. IEEE, 2016.

[5] Shenglin Zhang, Weibin Meng, Jiahao Bu, Sen Yang, Ying Liu, Dan Pei, Jun Xu, Yu Chen, Hui Dong, Xianping Qu, et al. Syslog processing for switch failure diag nosis and prediction in datacenter networks. In 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS), pages 1–10. IEEE, 2017.

[6] Liang Tang, Tao Li, and Chang-Shing Perng. Logsig: Generating system events from raw textual logs. In Proceedings of the 20th ACM international conference on Information and knowledge management, pages 785–794. ACM, 2011.

[7] Pinjia He, Jieming Zhu, Zibin Zheng, and Michael R Lyu. Drain: An online log parsing approach with fixed depth tree. In 2017 IEEE International Conference on Web Services (ICWS), pages 33–40. IEEE, 2017.

[8] Wei Xu, Ling Huang, Armando Fox, David Patterson, and Michael Jordan. Largescale system problem detection by mining console logs. Proceedings of SOSP'09, 2009.

[9] Shilin He, Jieming Zhu, et al. Experience report: System log analysis for anomaly detection. In 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), pages 207–218. IEEE, 2016.

[10] Salma Messaoudi et al. A search-based approach for accurate identification of log message formats. In Proceedings of the 26th Conference on Program Comprehension, pages 167–177. ACM, 2018.