ScienceDirect

Expert Systems with Applications

Volume 171, 1 June 2021, 114556

A graph-based approach to detect unexplained sequences in a log

Marcello Cinque, Raffaele Della Corte, Vincenzo Moscato, Giancarlo Sperlí [1]

Show more

Share Cite

https://doi.org/10.1016/j.eswa.2020.114556

Get rights and content

Highlights

- A graph mining approach has been designed for recognizing anomalous sequences.
- It supports both real time and batch processing for large scale data analysis.
- A probabilistic penalty graph has been used for modeling log temporal sequences.
- The approach's effectiveness has been evaluated for different system configurations.

Abstract

In this paper we challenge the issue of detecting anomalous events in computer systems log files, through a novel graph mining approach. The basic idea is to model log temporal sequences as a particular graph and event detection as a particular path finding problem. Thus, anomalous sequences correspond to log parts that can not be “explained” by any path in the graph. We propose a novel Iterative Partitioning Log Mining technique to parse any kind of logs and to model their temporal sequence as a probabilistic penalty graph. The approach has been implemented in a framework supporting both real time and batch processing realized on the top of the Apache Spark analytics engine for large-scale data processing. Experimental results show the advantages of the proposed framework in terms of effectiveness for different system configurations.

Introduction

Event logs are text files containing information about computer systems behavior. Events in the logs are emitted ubiquitously by the software components running in a system, for tracing and troubleshooting purposes. Their use for the detection of anomalies that might be caused by application failures or misuse is known since the early days of computers (Oliner, Ganapathi, & Xu, 2012).

One of the most challenging task is known as knowledge elicitation from logs. It presents several well-recognized issues, including vast amount of data, absence of ground truth, frequent false alerts, and rigid constraints on the number of events that analysts can investigate on a daily basis (Oprea, Li, Norris, & Bowers, 2018). In the common practice, heterogeneous logs are usually transformed into a uniform format and are then analyzed by experts to develop sets of regular expressions – embedding the evidence of failure or misuse – which are verified on the runtime logs to match misuse signatures. For example, second generation Security Information and Event Management (SIEM) (Kavanagh et al., 2016, Cardenas et al., 2013) representing the state-of-the-art in security analytics, still lack built-in support when facing unstructured data. Solutions like AlienVault USM,[2] IBM QRadar,[3] LogRhythm,[4] Splunk Enterprise Security (ES)[5] rely on internal representation formats to import and consolidate any data source with built-in adapters; however they require analysts to configure his/her own custom adapter to cope with unstructured data. Once the log is imported, analysts are expected to write ad hoc filters to extract specific fields or to search for entries matching given patterns.

A commonly-used method involves the search of “bad words” in the log, such as “fatal”, “corrupt” or “error”, since they might denote a potential problem. Then, up-to-dated log

collections are usually kept using monitoring tools like Splunk or Logstash.

This practice has some known limitations, as it is known that logs are fraught with unexplained sequences, e.g., missing and noisy events in an event flow. For instance, in Pecchia and Russo (2012) authors discovered that around **50%** of application failures are not detectable by looking for “bad words” in the logs. On the other side, logs gathered in normative conditions might occasionally encompass rare keywords or entire events that do not represent problems or attack, but indicate the occurrence of extraordinary conditions tolerated by the application. This brings to unexplained sequences in logs, that are usually difficult to classify as anomalous or correct behavior.

In this paper we propose a novel graph mining approach to deal with unexplained sequences in event logs. The idea is to formalize the data mining challenge as a particular graph problem, i.e., the problem of finding any path in the graph capable of “explaining” the current temporal sequence of log events. A novel technique, based on Iterative Partitioning Log Mining, has been defined to parse logs and to model their temporal sequence as graphs handled by a NoSql database. In particular, to better handle noisy logs, we adopt probabilistic penalty graphs as in Molinaro et al. (2014). Differently from the current practice on the application of graphs to log mining, our approach does not require well-structured logs, and it does not require any background knowledge or system expert view. Hence, it can be applied to unstructured and heterogeneous log files with no human intervention. The approach has been implemented in a framework supporting both real time and batch processing, according to the Lambda Big Data architectural pattern, implemented on the Apache Spark analytics engine for large-scale data processing.

The framework has been applied on logs emitted by a real-world critical information system for the Air Traffic Control (ATC) domain. The system installation has been made available by a top-leading industrial company in electronic and information technologies for both aerospace and defense. Large volumes of highly-unstructured proprietary logs are generated by the system; in normative operations logs encompass around 6,500 lines per minute, exceeding 10,000 lines in peak minutes. The knowledge base is built through offline experiments, beforehand, while test logs are gathered by emulating both normative and misuse scenarios. Test logs are used to asses the effectiveness of the method. Results indicate that, with a proper tuning, the approach can be successfully used to spot unexplained sequences related to misuse conditions and correctly classify them with high precision and recall.

The rest of the paper is organized as follows. Section 2 positions our work with respect to the state of the art. The proposed approach is then presented in Section 3. The ATC case study, along with the experimental campaign conducted to collect normative and misuse data, is described in Section 4. The results obtained by applying the approach on real logs are presented in Section 5. Finally, Section 6 ends the paper with final remarks.

Section snippets

Related work

We position our research with respect to existing work in both event-logs-based and graph-based anomaly detection. …

Framework

In this section we present the main characteristics of the proposed framework for recognizing anomalous sequences within a log through a novel *graph mining* approach. The idea is to formalize our data mining challenge as a particular graph problem, i.e., the problem of finding any path in the graph capable of “explaining” the current temporal sequence of log events.

Our framework supports both real time and batch processing in according to the *Lambda* Big Data architectural pattern (Marz & Warren, …

Case study

This section provides an overview of the setup used to conduct the experiments, encompassing both the reference system and the collected training/test logs. …

Evaluation results

In this section, we, firstly, evaluate the scalability of the proposed *log parsing and clustering* phase. Then, we discuss the results obtained through our anomaly detection approach by varying time windows length, edge penalties computation and threshold values.

For the efficiency analysis of the log parsing module, we have considered the *BLG* dataset (Oliner & Stearley, 2007), that is composed by more than 4,7 millions of logs and 376 events related to BlueGene/L supercomputer at the Lawrence …

Conclusion

In this paper we challenged the log event detection issue through a novel graph mining approach. Our intuition was to model log temporal sequences as a particular graph and event detection as a path finding problem.

We exploited the probabilistic penalty graphs that some of the authors have proposed in a recent work. Summarizing, a novel Iterative Partitioning Log Mining technique has been defined to parse any kind of logs and to model their temporal behaviour. Event detection was then obtained …

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. …

Recommended articles

References (41)

B. Fazzinga *et al.*

An ensemble-based approach to the security-oriented classification of low-level log traces

Expert Systems with Applications (2020)

S. Khan *et al.*
Eliciting and utilising knowledge for security event log analysis: An association rule mining and automated planning approach
Expert Systems with Applications (2018)

S. Khan *et al.*
Discovering and utilising expert knowledge from security event logs
Journal of Information Security and Applications (2019)

M. Lopez-Martin *et al.*
Application of deep reinforcement learning to intrusion detection for supervised problems
Expert Systems with Applications (2020)

A. Pecchia *et al.*
Discovering process models for the analysis of application failures under uncertainty of event logs
Knowledge-Based Systems (2020)

V. Persico *et al.*
Benchmarking big data architectures for social networks data processing using public cloud platforms
Future Generation Computer Systems (2018)

J. Roldán *et al.*
Integrating complex event processing and machine learning: An intelligent architecture for detecting iot security attacks
Expert Systems with Applications (2020)

W. Wang *et al.*
Botmark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors
Information Sciences (2020)

L. Akoglu *et al.*
Graph based anomaly detection and description: A survey
Data Mining and Knowledge Discovery (2015)

Albanese, M., Moscato, V., Picariello, A., Subrahmanian, V., & Udrea, O. (2007). Detecting

stochastically scheduled...

View more references

Cited by (9)

LogETA: Time-aware cross-system log-based anomaly detection with inter-class boundary optimization

2024, Future Generation Computer Systems

> *Citation Excerpt :*
>
> ...The methods mentioned above are all unsupervised. Nevertheless, supervised log detection methods [32–40] using labeled anomalous logs in the training process achieve higher accuracy on many datasets. Most supervised methods are classification-based approaches....

Show abstract

An anomalous sound detection methodology for predictive maintenance

2022, Expert Systems with Applications

> *Citation Excerpt :*
>
> ...Traditional approaches adopt different kinds of supervised machine learning models to accomplish classification or regression tasks on labeled data (Bala & Chana, 2015; Nunes, 2021; Yin, Zhang, Wang, & Xiong, 2020). Alternatively, unsupervised models have been used to distinguish between normal and abnormal situations with and without any a-priori knowledge (Cinque, Della Corte, Moscato, & Sperlí, 2021; Thudumu, Branch, Jin, & Singh, 2020; Wu, Zhao, Sun, Yan, & Chen, 2020). Surely, more recently, the most diffused machine learning techniques are represented by deep learning approaches that have been successfully exploited in different and heterogeneous contexts, such as medical, surveillance, finance and predictive maintenance applications as demonstrated by several recent surveys (Ballings, Van den Poel, Hespeels, & Gryp, 2015; Dalzochio et al., 2020; Fernando, Gammulle, Denman, Sridharan, & Fookes, 2020; Jalayer, Orsenigo, & Vercellis, 2021; Li, Li, Zhang, Liu, & Wang, 2018; von Birgelen, Buratti, Mager, & Niggemann, 2018)....

Show abstract

Threat classification model for security information event management

focusing on model efficiency

2022, Computers and Security

> *Citation Excerpt :*
>
> ...We classify previous studies for SIEM into the following categories: 1) large-scale event data management (El Arass et al. 2019; R. Andrade et al., 2018; Cinque et al., 2021), 2) signature-based threat detection (B.D. Bryant et al., 2020; Eswaran et al., 2021), and 3) machine learning (ML)-based threat detection (Kim., 2014; Lee et al., 2019; Naseer et al., al.,2018; A. Kim et al., 2020). Large-scale event data management for SIEM includes large event data processing (El Arass et al., 2019; R. Andrade et al., 2018) and unstructured event data processing (Cinque et al., 2021). El Arass et al. (2019) proposed Smart-SIEM consisting of Elasticsearch, Logstash, and Kibana (ELK)....

Show abstract

An Unsupervised Graph-Based Approach for Detecting Relevant Topics: A Case Study on the Italian Twitter Cohort during the Russia–Ukraine Conflict ↗

2023, Information (Switzerland)

Unsupervised Anomaly Detection in Predictive Maintenance using Sound Data ↗

2023, CEUR Workshop Proceedings

Landscape of Automated Log Analysis: A Systematic Literature Review and Mapping Study ↗

2022, IEEE Access

View all citing articles on Scopus ↗

---

1 ORCID: 0000-0003-4033-3777.

View full text