Anomaly Detection from System Log Files using Deep Learning techniques

Aastha Dhamija

Seattle Pacific University

Abstract

Robust anomaly detection is imperative for building a secure and trustworthy information system and sustain business continuity by maintaining confidentiality, integrity and availability of the systems. Identifying and resolving anomalies in advance can improve quality of service and defer any kind of revenue loss because of system breakdown. Systems logs are an effective way of monitoring system health as it encapsulates the current state, transition and various events happening at regular intervals in the system. Traditionally, engineers would manually examine the system logs to identify any abnormal behavior based of their domain knowledge and existing literature on system's known anomalies. However, with advancing technology landscape, systems are growing in terms of scale and complexities and are also prone to increased adversaries which can obstruct systems' performance in not yet known ways. Hence traditional manual ways of anomaly detection with a defined set of rules become insufficient to deal with huge amount of log data amongst intertwined complex systems with unknown attack vectors. To address these problems, this paper provides a way of including deep learning techniques in various stages of log analysis to make it more robust. It proposes to use Apache Hadoop technology for analyzing large amount of log sets in parallel using MapReduce technique to make it faster and then use recurrent neural network for feature extraction and anomaly detection.

Keywords: Anomaly detection, system logs, log analysis, machine learning, neural networks, Apache Hadoop, MapReduce, Information security

Anomaly Detection from System Log Files using Deep Learning techniques

Systems are getting more and more complex by the day and detecting anomalies in its behavior are essential towards building a secure and trustworthy system. With increasing technological advancements, systems are subjected to unknown adversaries that may exploit them further and cause outbreaks. With evolving scale and strategic dependence of information systems, it is imperative to maintain their consistent performance and monitor any abnormalities to retain their normal state.

Anomaly detection aims to uncover abnormal system behaviors in a timely manner and hence plays an important part in incident management of large complex systems (He S. , Zhu, He, & Lyu, 2016). System logs are widely used to record system state, its transitions and various significant events in timely manner and are used to debug system performance issues and failures, hence they become an excellent source of detecting anomalies in any system. System logs are collected and stored widely across all organizations to monitor their system's health and real time analysis of these logs can predict many shortcomings before they become a blunder and impact confidentiality, integrity and availability of critical systems. Log based anomaly detection has become a prevalent method for maintaining system performance and effective incident management across academic institutions and industry (He S. , Zhu, He, & Lyu, 2016).

Traditionally, developers used to manually inspect these system logs for standalone information systems, compare them to their logs in normal circumstances and highlight anomalies on the basis of their subject knowledge and information gathered over course of time. However, such reliance on manual inspection is not practically feasible with the scale of data that each system produces presently and their interconnectivity and reliance with other systems. Few other challenges are as follows:

- The interconnectivity and large scale of each system makes it impossible for an individual developer or engineer to understand its anomalies as they understand just one part of it and not the whole complex systems. Some results which appear as normal in individual parts might behave completely difference in tandem with the whole system.
- In this age of big data, each system collects and generates gigabytes of data on frequent interval and thus becomes infeasible to manually discern important behavioral information from such large set of irrelevant information.
- Due to build in complexity and scale of information systems, individual anomalies at times become false positives in view of the complete system. Solving and finding root cause of false positives increases the manual inspection effort by large scale and results in frustration amongst developers.
- Traditional searches on the basis of keywords or domain knowledge fails in case of a new unknown anomaly which has no prior dataset or signature in the existing attack log and hence does not come under the radar of defined anomalies. This can significantly impact the overall performance of system if anomalies remain unknown.

As a result, automated log analysis for anomaly detection is much in demand to capture abnormalities in a robust manner. Process of log analysis for anomaly detection involves four main steps: log collection, log parsing, feature extraction and anomaly detection (He P. , Zhu, He, Li, & Lyu, 2016). Further, according to the type of data set involved and machine learning technique used, anomaly detection can be classified into supervised and unsupervised anomaly detection. Supervised anomaly detection requires a preexisting labeled dataset of normal and abnormal behavior for training the model and unsupervised does not require labeled dataset, it

works on the abnormal occurrence which comes across as an outlier in tandem with other points in the dataset (Chandola, Banerjee, & Kumar, 2009).

In this paper, we understand the process of automated log analysis and discover how we can use deep learning technologies in various stages to make anomaly detection process more vigorous. Further this paper explains usage of Apache Hadoop technology for analyzing large amount of log datasets and processing them faster using MapReduce method and then using a Recurrent Neural Network for feature extraction and anomaly detection. Machine learning techniques require rules to diagnose behavior of a system and report anomaly, this paper focuses on inclusion of deep learning techniques over machine learning to capture those unknown vectors, which cannot be defined with any set of rules.

Literature Review

There are several papers explaining the process of anomaly detection using system log files in traditional manner along with advance machine learning and deep learning methods.

Chandola, Banerjee & Kumar (2009) shares a survey of various techniques used for anomaly detection, this paper clearly explains the meaning and types of anomalies and how their varied characteristics makes it difficult to capture them early on in the process. It also covers various application domains in which anomaly detection can be used and how it varies in each of them. He P. , Zhu, He, Li, & Lyu, (2016) explains why traditional manual methods of anomaly detection does not work anymore in modern context of large scale and complex information systems. It also clarifies difference between supervised and unsupervised anomaly detection and various techniques under each category.

Breier & Branisova (2015) explains various types of logging and how various data mining techniques can be used for log analytics and Du, Li, Zheng, & Srikumar (2017) experiments with the usage of neural networks in analyzing system logs for detecting anomalies with minimal labeled data available for model training.

After reviewing various points of view in above stated research work, this paper in particular showcases combination of faster log processing in tandem with neural networks for identifying unknown anomalies with minimal human intervention.

Understanding Log Data

Log data is automatically produced by the system which encapsulates its current state, its transitions over time and a unique identifier at every defined interval (Breier & Branisova, 2015). There are various types of logging such as follows:

- Security logging: It includes information obtaining from security systems and can be used to detect potential adversaries or breaches in the system.
- Access logging: It includes information of all the users who are authorized to use a particular system and also record their incoming and outgoing times.
- Operational logging: It contains information about current state of the system and can point out system errors and malfunctions.
- Compliance logging: It contains information about compliance with defined security requirements.

A combination of above logs is used for the purpose of identifying abnormal behavior of the system. This abnormal behavior is termed as an anomaly and is analyzed further to identify the root cause and provide possible solution for bringing system back to normal conditions.

Understanding Anomaly

Anomalies are defined as patterns in data that do not conform to a well-defined notion of normal behavior (Chandola, Banerjee, & Kumar, 2009). They can be induced in the system for various reasons such as cyber intrusion, terrorist activity or mere break down of another related system, but all of these reasons need to be tackled with, to bring the system back to normal conditions. An anomaly is defined as novelty detection when previously unobserved patterns are identified in the data set. Once identified, these novel patterns are incorporated in the normally identified anomalies.

Types of Anomalies:

- Point Anomalies: If an individual data point can be termed as abnormal in tandem with other points in the data set, then that is termed as a point anomaly.
- Contextual Anomalies: If a data point is abnormal only in a specific context, but not otherwise, then that is termed as a contextual or conditional anomaly.
- Collective Anomalies: It a collection of data points is termed as abnormal in tandem with other points in the data set, then that is termed as a collective anomaly. In this case, individual data points might not be anomalies by themselves but are considered abnormal when all of them occur together.

Anomaly detection framework

(He S. , Zhu, He, & Lyu, 2016) illustrates the overall framework for log-based anomaly detection method and it comprises of four stages – log collection, log parsing, feature extraction and then finally anomaly detection.

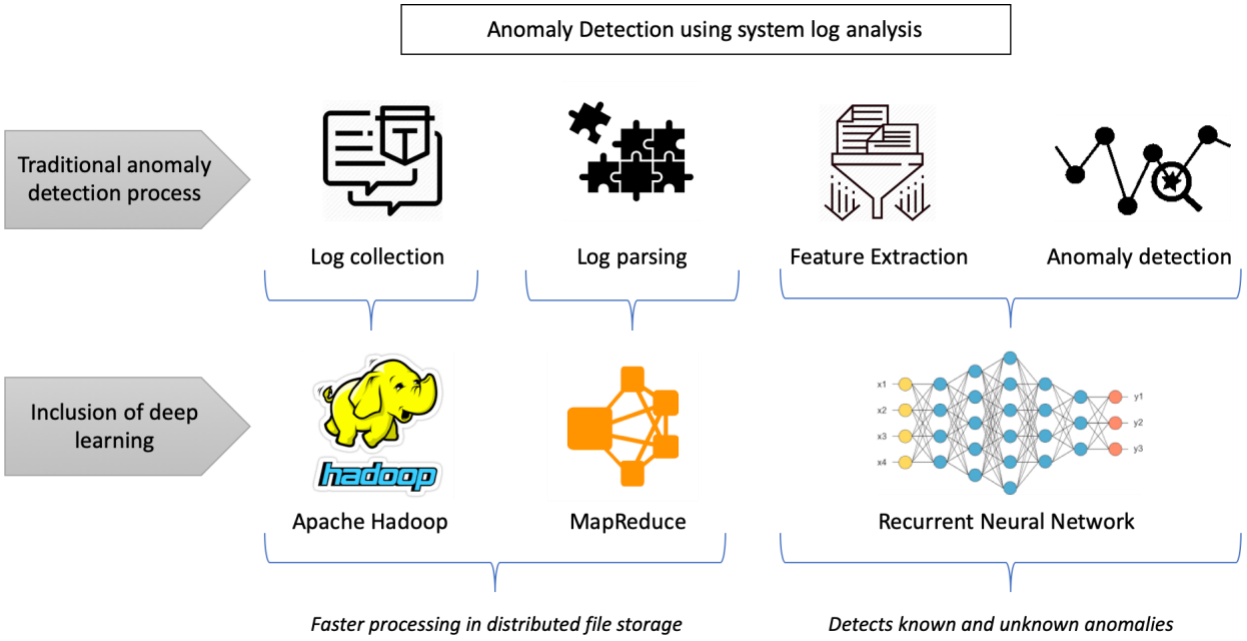

Figure 1: Anomaly Detection using system log analysis

Stage 1: Log Collection

Log data is collected from various sources and as described above; it can be of various types – like security log, compliance log, access log, operational log etc. . Log data is usually timestamped to define the time interval and contains essential information about systems' current state. Each log entry is assigned a unique primary key which act as an identifier however format of every kind of log can be different.

New addition: In modern day scenario, due to large scale data of systems, this log data is stored in big data platform - Apache Hadoop – a distributed file storage system for better and faster performance. Apache Hadoop is a collection of open-source software utilities that facilitate

using a network of many computers to solve problems involving massive amounts of data and computation (Apache Hadoop, n.d.).

Stage 2: Log Parsing

Logs are unstructured and can have varied formats depending on the system or application, hence carving out important information from assorted log data is essential. Purpose of this stage is to have a common event templet which can be used to record essential state information about the system in most effective and consistent manner. (He P. , Zhu, He, Li, & Lyu, 2016) found that each log message can be parsed into a constant event part and a variable parameter part.

New addition: MapReduce tool engine is used in Apache Hadoop platform for faster parallel processing of the log data. MapReduce function further processes the data and generates a list in uniquely identifiable key-value form, which can be used by the user to join all the relevant values with same key (Breier & Branisova, 2015). Through this method, much larger amount of data is processed in the same amount of time.

Stage 3: Feature Extraction and Anomaly detection

Post log parsing, end result is a structured and well segregated log set which is then used for extracting relevant features out of the data set and ultimately feeding it to the machine learning algorithm for training the model and finding relevant patterns for anomaly detection. In case of deep neural networks, it automatically extracts out relevant features for training the model and does not require a manual intervention of feature extraction. It therefore sorts out even those features which are deemed as irrelevant by the human and matches it with multiple other features to find all the hidden and unknown anomalies.

Recurrent neural network is specifically used in system log analysis because of its ability to determine semantic relationship between the words, as most of the log files are in textual format (He P. , Zhu, He, Li, & Lyu, 2016). Further the hidden layer in recurrent neural network acts a memory that stores the current state of the log data and when memory is updated, final assessments are done in consideration of both new and previously held state input. These neural networks can work with minimal amount of labeled training data and hence are applicable for both supervised and unsupervised anomaly detection (Du, Li, Zheng, & Srikumar, 2017). Since it is a learning driven approach, these neural networks can be updated on a regular basis to include more findings which can result in more robust anomaly detection. They can also take user feedback as one of the inputs and improve the abnormality search (Jagreet Kaur Gill, 2018).

Choosing deep learning over machine learning techniques

In terms of the above log analysis process, major difference between machine learning and deep learning techniques is the feature extraction part. For machine learning techniques, developers or engineers are required to manually extract features from the log data set whereas deep learning automatically extracts those features without human intervention.

- This reduces developer or engineers' burden to manually find relevant features and at the same time reduce error by human intervention
- At times, few features on surface level looks irrelevant to human and hence can be ignored in the feature selection process. However, with increased complexity and interconnected of disparate systems, judging the exact impact of a particular feature on systems' current state is very difficult, such features are picked up in deep learning techniques for in depth analysis.

- Machine learning techniques work well with supervised detection as it needs larger amounts of data to train the model and is efficient in capturing defined anomalies as per the past data, however deep learning techniques work well in deciphering all kinds of supervised, unsupervised or semi supervised data sets and since it does not rely on limited number of extracted features, it does a better job in understanding underneath complexities and identifying unknown or novel anomalies.

Hence, deep learning does nonlinear transformation by analyzing all the features from all angles and pick out patterns which can be missed easily with predefined features, rule-based analysis and known anomalies.

Conclusion:

In this paper, traditional anomaly detection using system log analysis has been modified with distributed file storage and neural network techniques to handle large scale and complex information systems. This paper clarifies the challenges with traditional manual anomaly detection, justifies why system logs become a good source of examining systems' health and explains various stages of log analysis and how can they can be improved with distributed file system and deep learning techniques to make anomaly detection fast and robust. It further elaborates on advantage of using neural networks over machine learning techniques to identify both known and unknown anomalies.

Future research scope:

Right now, this paper is based on theoretical knowledge from various research papers. To prove the accuracy and estimate the increase in anomaly detection percentage, we should put this

paper into practice and run a public database with this methodology to check its effectiveness. Further, various other types of neural networks can be researched and tried in the same methodology to find the best fit.

References

Breier, J., & Branisova, J. (2015). Anomaly Detection from Log Files Using Data Mining Techniques. *ICISA*.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. *ACM Comput.*, 58.

Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. *CCS'17*.

He, P., Zhu, J., He, S., Li, J., & Lyu, M. R. (2016). Evaluation Study on Log Parsing and Its Use in Log Mining. *46th Annual IEEE/IFIP*.

He, S., Zhu, J., He, P., & Lyu, M. R. (2016). Experience Report: System Log Analysis for Anomaly Detection. *IEEE* .

Jagreet Kaur Gill. (2018, October 21). Automatic Log Analysis using Deep Learning.

Navdeep Singh Gill. (2018, December 15). Log Analytics, Log Mining and Anomaly Detectoin with Deep Learning.

Apache Hadoop. (n.d.). Retrieved from Apache Hadoop: http://hadoop.apache.org/