A Survey on Anomaly Detection Methods for System Log Data

Devika Ajith

Abstract: ***System logs are often a collection of unrelated print statements which records certain events that occur while the system is running. Log file analysis often can be crucial for finding system faults which can otherwise be quite difficult to detect. Traditional log analysis involves analyzing line by line until a discrepancy is spotted. This process is tedious, time consuming and is prone to human errors. With the advent of machine learning, several new methods have been devised which can make anomaly detection much easier. Further, in the past decade, deep learning has evolved so much that new techniques and algorithms spring every now and then. This paper examines several existing techniques that can be used for system log analysis.***

Keywords: Anomaly Detection, Deep Learning, Log File Analysis, Machine Learning

1. Introduction

Application and systems generate huge amounts of log data. With the rise of embedded systems anywhere and everywhere and the massive amount of data that is being generated, it's inevitable that one way or the other a system will run into fault. More than often the developer will be clueless and will be forced to look at system log data at the time of fault. Analyzing millions of lines manually can turn out to be an impossible task.

Researchers have been trying out various methods of anomaly detection since time immemorial. V. Chandola [2008] et al has given a detailed study of various anomaly detection models. However deep learning has progressed much over the past decades and numerous new methods have evolved which makes anomaly detection much easier for text data. From Splunk to Loggly, Druva to Deeplog, log data analysis field has seen much boom over the past years. This paper gives an overview of several existing techniques that can be used for anomaly detection of log data.

1.1 Anomalies or Outliers

One popular definition of anomalies or outliers, as they are often called is one that appears to deviate markedly from other members of the sample in which it occurs [1]. An exact definition of an outlier often depends on hidden assumptions regarding the data structure and the applied detection method. Hawkins defines an outlier as an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism. Johnson (1992) defines an outlier as an observation in a data set which appears to be inconsistent with the remainder of that set of data [2].

To begin with, anomalies can be classified into point anomalies, contextual anomalies and collective anomalies. If an individual data instance can be considered as anomalous with respect to the rest of data, then the instance is termed a point anomaly. This is the simplest type of anomaly and is the focus of majority of research on anomaly detection. If a data instance is anomalous in a specific context, but not otherwise, then it is termed a contextual anomaly (also referred to as conditional anomaly). On the other hand, if a collection of related data instances is anomalous with respect to the entire data set, it is termed a collective anomaly [5].

1.2 Existing anomaly detection methods

Anomaly detection techniques can be classified into three broad categories: supervised, semi supervised, and unsupervised anomaly detection models. Supervised anomaly detection assumes the availability of a training data set that has labeled instances for normal as well as anomaly classes. This is analogous to building a predictive model. But obtaining accurate representative models for anomaly classes is challenging. Another type is semi-supervised anomaly detection which assumes that the training data has labeled instances only for the normal class. Since they do not require labels for the anomaly class, they are more widely applicable than supervised models. The third category is unsupervised anomaly detection model which does not require any training data at all [5]. Two main goals of system log anomaly detection are log size reduction and root cause analysis.

1.2.1 Log size reduction

When log files are ten to thousands of lines long, even a minor reduction in the size is so important in further analysis since it helps in eliminating unwanted noise. The very first step in this process is to recognize the log format and identify the important parameters with the help of a domain expert. The next step is to extract information by removing redundant and expected data using regular expressions to separate fields. The objective of this step is to obtain information by noise elimination. This process converts unstructured data to structured format which can be further used as input to anomaly detection algorithm or manual analysis.

1.2.2 Anomaly detection techniques

The following gives various techniques used in system log data anomaly detection.

A. Statistical Anomaly Detection Models

In Statistical methods, the log dataset is organized in terms of its overall statistic distribution and the data points which stand out or do not conform to this distribution are removed or examined [8]. One possible method is to count the relative frequency of words/events and to examine those with less frequency since it's assumed that anomalies would be sudden and infrequent. Another approach would be to set certain threshold for various parameters above or below of which would denote an anomaly. However such a method is heavily biased. Additionally it can be used only for detecting point anomalies.

B. Nearest Neighbor Based Techniques

This concept is based on the assumption that normal data instances locate in dense neighborhoods, while anomalies lie far from their closest neighbors [9]. Euclidean, Hamming (for equal length), Mahalanobis distances or K-Nearest neighbor technique can be used for anomaly detection. Though this method works well in certain situations, it fails when applied to datasets with an unpredictable distribution with both sparse and dense regions [8].

C. Clustering Based Anomaly Detection Methods

Clustering methods fundamentally rely on the assumption that normal data instances belong to a cluster in the data, while anomalies either do not belong to any cluster or lie far away from their closest cluster centroid [5]. One such approach is to maintain micro clusters which will be continuously updated and deleted as the log size grows. K-Means clustering, D-Stream, DBSCAN, Modified and hybrid clustering approach, FCM, DENstream, BIRCH with k-means and BIRCH with CLARANS, CURE with k-Means and CLARANS, HDDstream[10] are some popular techniques used for anomaly detection.

D. Support Vector Machines

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side [13]. Often, SVMutilizes the sequential nature of log file. One major disadvantage of this technique is that as the number of features grow exponentially, it will be difficult to balance message information with sequence length [14].

E. Bayesian Networks

A Bayesian network represents the causal probabilistic relationship among a set of random variables, their conditional dependences, and it provides a compact representation of a joint probability distribution. It consists of two major parts: a directed acyclic graph and a set of conditional probability distributions [11]. This approach is like a classification problem, where a trained Bayesian network on training dataset aggregates information from different variables and provides an estimate on the expectancy of that event to belong to normal/abnormal class for unseen test dataset. The biggest disadvantage of this technique is that they rely on the availability of accurate labels for various classes, which is, most often not possible [12].

F. Neural Network Based Models

Recurrent Neural Network (RNN) and auto encoder-decoder are two popular techniques used presently. There has been a proliferation of Neural Network based anomaly detection methods in the past decade. At present, several commercially available tools make use of Artificial Neural Network (ANN) advantages. An RNN is a class of ANN where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feed forward neural networks, RNNs can usetheir internal state (memory) to process sequences of inputs [15].

Yang T. et al proposes word vector based and char vector based RNN models to reduce the effort needed to analyze the log file by highlighting the most probably useful text in the failed log file, which can assist in debugging the causes of the failure [16]. Further, F1 score of Char-LSTM (Long Short Term Memory), char-RNN, char-GRU (Gated Recurrent Unit) methods are calculated and compared.

Lu S. et al, argues that a Convolutional Neural Network-based approach has better accuracy(reaches to 99%) compared to other approaches using LSTM and Multilayer Perceptron (MLP) on detecting anomaly in Hadoop Distributed File System (HDFS) logs. This approach uses a deep neural network which consists of logkey2vec embeddings, three 1D convolutional layers, dropout layer, and max-pooling [18].

Wang et.al, uses feature extraction algorithms such as Word2vec and Term Frequency-Inverse Document Frequency to obtain log information and applies LSTM for anomaly detection [19]. Their results indicate that LSTM can capture contextual semantic information effectively in log anomaly detection and will be a promising tool for log analysis

Zhou et al [20] demonstrate Robus Principal Component Analysis inspired novel extensions to deep auto encoders which not only maintain a deep auto encoders' ability to discover high quality, non-linear features but can also eliminate outliers and noise without access to any clean training data.

However the most matured technique among all these is Deeplog, which uses LSTM based deep neural network model to model system log as a natural language sequence [21]. Deeplog automatically learns log patterns from normal execution and detect anomalies when log patterns deviate from the model trained from log data under normal execution.

2. Concluding Remarks

Log file analysis can be a tedious task since bugs are unpredictable and labeled data of abnormal scenarios is not readily available. However in the recent years significant developments have been made in the direction of root cause analysis through the use of LSTM based RNN models and machine learning methods.

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

[1] Grubbs, F. E.: 1969, ‘Procedures for detecting outlying observations in samples’. Technometrics 11, 1–21.

[2] Ben-Gal, I. (n.d.). Outlier Detection. Data Mining and Knowledge Discovery Handbook, 131–146. doi:10.1007/0-387-25465-x_7

[3] Hawkins D., Identification of Outliers, Chapman and Hall, 1980.

[4] Johnson R., Applied Multivariate Statistical Analysis. Prentice Hall, 1992.

[5] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection. ACM Computing Surveys, 41(3), 1–58.doi:10.1145/1541880.1541882

[6] Berkay K., Unsupervised anomaly detection in unstructured log data for root cause analysis, Tampere University of Technology

[7] A. Katal, M. Wazid, and R. H. Goudar. "Big data: issues, challenges, tools and good practices." in

[8] Martí, Luis, Nayat Sanchez-Pi, José Manuel Molina, and Ana Cristina Bicharra Garcia. "Anomaly detection based on sensor data in petroleum industry applications." Sensors 15, no. 2 (2015): 2774-2797. Contemporary Computing (IC3), 2013 Sixth International Conference on, pp. 404- 409. IEEE, 2013.

[9] Grover A., Anomaly Detection for Application Log Data, San Jose State University, Spring 2018 [9] Wu Xuanfan, Metrics, Techniques and Tools of Anomaly Detection: A Survey, http://www.cse.wustl.edu/~jain/cse567-17/ftp/mttad/index.html

[10] S. Anitha, Matilda M., A Survey on Cluster Based Outlier Detection Techniques in Data Stream, , International Journal of Data Mining Techniques and Applications Volume 5, Issue 1, June 2016, Page No.96-101 ISSN: 2278-2419

[11] Murphy K. (1998): A Brief Introduction to Graphical Models and Bayesian Networks

[12] Babbar Sakshi, Chawla Sanjay, On Bayesian Network and Outlier Detection, School of Information Technologies, University of Sydney, Sydney NSW 2006, Australia

[13] Amarappa S, Sathyanarayana S V,Data classification using Support vector Machine (SVM), a simplified approach, International Journal of Electronics and Computer Science Engineering, ISSN 2277-1956/V3N4-435-445

[14] EW Fulp, GA Fink, JN Haack Predicting Computer System Failures Using Support Vector Machines, WASL, 2008 - usenix.org

[15] https://en.wikipedia.org/wiki/Recurrent_neural_network

[16] Yang T, Agarwal V., Log File Anomaly Detection, NVIDIA

[17] Malhotra P., Ramakrishnan A., Anand G., Lovekesh V., Agarwal P., Shroff G., LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection, ICML 2016 Anomaly Detection Workshop, New York, NY, USA, 2016, arXiv:1607.00148v2 [cs.AI] 11 Jul 2016

[18] Lu S., Wei X., Li Y., Wang L., Detecting Anomaly in Big Data System Logs Using Convolutional Neural Networks, 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)

[19] Wang M., Xu L., Guo L., Anomaly detection of system logs based on natural language processing and deep learning, 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), DOI https://doi.org/10.1109/ICFSP.2018.8552075

[20] Zhou C., Paffenroth R.C., Anomaly detection with robust deep autoencoders, KDD’17 Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[21] Du M., Li F., Zheng G., Srikumar V., DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning, CCS’17, October 30-November 3, 2017, Dallas, TX, USA

Author Profile

Devika Ajith received her B Tech degree in Electronics and Communication from Kerala University in 2017. She is currently working in Tata Consultany Services as Assistant System Engineer. Her research areas include outlier detection using deep learning, digital signal processing and communications.