Anylog: Anomaly Detection of Heterogeneous Logs using Unsupervised Techniques

Ayesha Aziz
Dept. of Computer Science
FAST- NUCES
Islamabad, Pakistan
i212231@nu.edu.pk

***Abstract*—Anomaly detection is a critical mining task for computer system security and reliability. In almost every computer system, logs are the primary source for anomaly detection methods. General information, errors, warnings, and debugging information are all contained in these logs. Previously, the focus in this domain has been on machine learning methods on predefined log patterns that are unpredictable. Techniques such as Deeplog and LogAnomaly are used because they are incapable of detecting anomalies in log structures. One significant limitation is that we must specify the log patterns that will guide the models. There are numerous issues with testing large-scale systems that are rapidly changing states. When automated testing fails to identify problems, manual log analysis becomes critical. Can we do this in real-time for heterogeneous logs? The primary goal of this survey is to create a system that does not rely on predefined log patterns to detect anomalies when using Unsupervised techniques. Heterogeneous logs are logs of any type, including Apache, Nginx, hdfs, OpenStack, Hadoop, and any other service logs. Following a prediction from the model, an alarm will be triggered, and an email notification will be sent to the system administrator. The false alarm rate will be reduced as a result of this.**

***Index Terms*—Anomaly detection, Classification, Heterogeneous Log messages, Unsupervised learning**

I. Preliminary Introduction

In the existing techniques, supervised and unsupervised log data have been used to detect anomalies in systems. In the domain of anomaly detection Isolation Forest [2], Template2Vec [13], Density-Based Clustering [3], K-Means [8][11], KNN [15] have used unsupervised log data. There had been many shortcomings such as accuracy issues, false alarms, costly in their different working techniques, and models but some resolved as the time-to-time research and progressed. The authors of [4] [13] have focused on predefined log templates and patterns that are unpredictable because there is huge diversity in the logs domain. Many tools have been used to detect anomalies like a deeplog [4], LogAnomaly [13] and many other approaches. The main limitation of these techniques is that they require log patterns for their models to be trained and these patterns are predictable in many conditions. Many problems exist in the testing of large-scale systems that are rapidly changing their states using different approaches like data mining techniques or clustering-based approaches. If we look at the previous work, in this domain RNN, LSTM, KNN based models were used which were not too efficient as they are slow to train and do not have a long-term dependency. So to overcome all these issues and problems we aim to achieve better results using transformer-based models. As BERT model comes under the transformer base models and it already knows the context of the language. It is already trained in Wikipedia and books corpus dataset [1]. Transformer architecture includes encoder and decoder. After the encode layer it has a decoder layer and specifically, BERT uses only the encoder part of it. It includes multiple layers of encoders top on the stack. The following hierarchy Figure I.1 shows the main limitations of some related work.

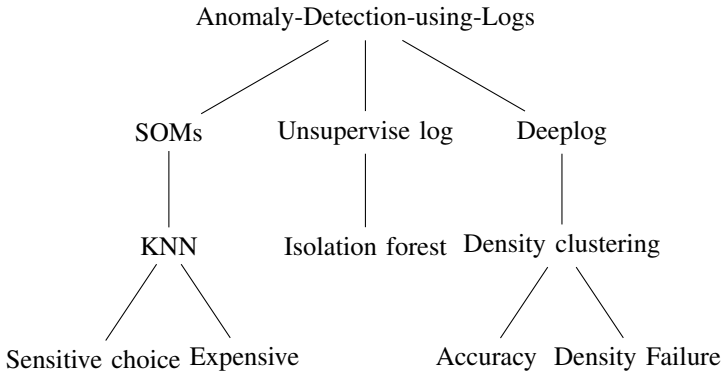

Fig. I.1: Hierarchy of Proposed Techniques

A. Supervised Techniques

KNN technique was used in [15]. In this technique, the efficiency of neighbors and automatic selection of neighbors were proposed. In this Minhash and MVP-tree was used to bucket up a similar group of logs then that bucket was sent to MVP-tree and processed next. This technique showed accurate and precise results but it was computationally very expensive and could not find the similarity of functions for comparing instances[14]. The result of this technique was 89.91% accurate. Logs are the major component of information of system each system have its format of logs. So to narrow down the information provided in logs log-parser is used to get the bottleneck of the log. The main purpose of using log parsing is to structure raw logs by extracting a group of event

TABLE I: Related Work

Authors	Year	Tools/Techniques	Strength	Gap	Result
Wang et al. [15]	2020	KNN	• Automatic selection of k neighbors • Reduce the effort of distance calculation	• Computationally-expensive • Sensitive to the choice of the similarity function for comparing instances	$Accuracy = 83.4\%$
Du et al. [3]	2017	Density Based Clustering	• Root cause analysis • Automatically learn log patterns • Online Updates the anomalies	• Accuracy • False alarm • Fails in case of varying density clusters.	$Accuracy = 87.9\%$
Hajamydeen et al. [6]	2016	K-Mean	• Two-step strategy clustering • Reduce the volume of events	• Equal number of observation • Specifying the value of K • Accuracy issues	$Accuracy = 87.2\%$
Farzad et al. [4]	2020	Isolation Forest Auto-encoder	• Positive sample prediction • Better results from other models	• Predict only Positive Logs • Accuracy issue	$Accuracy = 78.8\%$
Meng et al. [13]	2019	Template2Vec	• Extract the semantic information • Avoid false alarms	• Wrong template mapping • Cannot detect sequential and quantitative anomalies simultaneously	$Accuracy = 85.6\%$

TABLE II: Previous Survey Comparison

Tools / Techniques	Liu et al. [12]	Ippoliti et al. [10]	Lima et al. [11]	Xu et al. [11]	Ding et al. [2]	Guo et al. [5]
K-means	✓	x	✓	✓	x	✓
KNN	✓	x	✓	✓	x	x
SOM	✓	✓	✓	x	✓	x
Isolation Forest	x	x	✓	✓	x	✓
Template2Vec	x	✓	✓	x	✓	x
Density Based Clustering	x	✓	x	✓	✓	x

templates. There are certain rules on which log parsing works, this becomes very difficult in the case of each unstructured log. For certain logs, we cannot apply a log parser because the log parser looks for the specific templates which are defined in the log parser but what if the log pattern is completely new? In some cases like the supervised approach, it is easy to apply log parsing as many online tools are also available like drain so it becomes easy to extract the format of logs. LogMine is a tool that recognizes all the patterns present in the logs [7]. But in the case of an unsupervised approach, it's not that easy because we don't know the exact format of logs.

In the previous most of the work uses the log parsing technique

before they ingest log information to machine learning model but when comes to a large amount of logs data and in real-time it's not easy to use this approach. In real-world problems we have millions of logs are being ingested in real-time and log parsing for those logs will be very costly so we decided to bypass the log parser step and ingest data to the machine learning model.

B. Unsupervised Techniques

K-means technique was used in a Heterogeneous log-based framework [6]. To cluster the log events two-step strategy was used and thresholding filters are used in order to reduce the size of events analysis. On the basis of identifying anomalies, the logs are gathered and analyzed. In this technique, the value of **K** was specified but if a different structure of data came in then the value of **K** was not able to satisfy the data. Isolation Forest technique was proposed in Unsupervised log message anomaly detection [4]. For the feature extraction and training of unlabeled data, the Auto-encoder networks are used and for the anomaly detection, it was used. For only prediction of positive logs[13], isolation forest is used and ignores false-negative logs. The data set used for this proposed method is BGL, OpenStack, and Thunderbird log message. It only predicts positive logs and has an accuracy of 78.8% which is critical as it has not been able to detect anomalies from each log. Template2Vec technique was proposed in LogAnomaly [13]. They have used Sequential and Quantitative Unstructured Logs. In this technique, some templates for logs were statically pre-defined and then compared to logs to check which log belongs to the comparing template. But there were two issues with this technique that it was mapping the templates wrongly with logs and can't detect sequential and quantitative anomalies simultaneously. The result was 85.6% accurate. Local outlier factor algorithm is created for outlier detection, for the high dimensional data, it is used as an unsupervised approach [16]. This algorithm is also used to detect the novelty of the data. It works by finding the local density of its neighbors. If a data has a lower density than its neighbors, then it is considered an outlier. The only difference of its with other algorithms is that it only uses the fit_predict() method along with training of data and it doesn't use predict() a method that fails in our case as we are only predicting the anomaly in our system. There are different types of outliers such as point outlier, collective outlier, sequence outlier, and trajectory outlier. In some cases, it is not possible to detect the outlier because of the density issue of large-scale data. Density based clusterin technique was proposed in Deeplog [3]. It uses Long Short-Term Memory (LSTM) in order to utilize a deep neural network model. It also uses the clustering-based technique to cluster the logs to their relevant group. Although this technique was very easy to process as compared to other techniques used to detect anomalies using logs it failed in case of varying density of clusters and gave a false alarm. Accuracy of this technique 87.91%. In the case of high-dimensional data, it becomes difficult to handle in real-time at a single model. Theone approach is to reduce the information by maintaining the most relevant and important topological relationships of the data elements into a two-dimensional data plane or pick only the most related features. The main advantage of using SOM is that even if the data is very noisy, it can handle this efficiently [9]. For this, no precedent knowledge and no assumptions are required to make for this class membership of data. The multiple server logs are handling each server log with a separate map using which the results will be much confident. In the Self-Organizing-Map the concept of BMU(Best Matching Unit) is used, the upper weighted vector is chosen which is understood as Best Matching Unit (BMU). The neighbors across the BMU are reduced when the new neighborhood of BMU is calculated[10]. The higher weight becomes more like the sample vector and the neighbors also become more like the sample vector. Simply if the BMU is higher, the higher chances are that their weights will be changed and more it will learn. While far away from BMU, the less it will learn.

Some papers related to the domain of anomaly detection are listed in Table I but each one of them uses a different technique. The gaps of different techniques are discussed below in Table I. The main gaps are the accuracy issue due to the high false-positive rate, prediction of only positive logs, and wrong template mapping. And even sometimes it cannot detect the sequential and quantitative anomalies simultaneously. The techniques that are used in different papers are also listed in Table II.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Zhiguo Ding and Minrui Fei. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*, 46(20):12–17, 2013.

[3] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1285–1298, 2017.

[4] Amir Farzad and T Aaron Gulliver. Unsupervised log message anomaly detection. *ICT Express*, 6(3):229–237, 2020.

[5] Yicheng Guo, Yujin Wen, Congwei Jiang, Yixin Lian, and Yi Wan. Detecting log anomalies with multi-head attention (lama). *arXiv preprint arXiv:2101.02392*, 2021.

[6] Asif Iqbal Hajamydeen, Nur Izura Udzir, Ramlan Mahmod, and ABDUL AZIM ABDUL GHANI. An unsupervised heterogeneous log-based framework for anomaly detection. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24(3):1117–1134, 2016.

[7] Hossein Hamooni, Biplob Debnath, Jianwu Xu, Hui Zhang, Guofei Jiang, and Abdullah Mueen. Logmine: Fast pattern recognition for log analytics. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1573–1582, 2016.

[8] LI Han. Using a dynamic k-means algorithm to detect anomaly activities. In *2011 Seventh International Conference on Computational Intelligence and Security*, pages 1049–1052. IEEE, 2011.

[9] Albert J Hoglund, Kimmo Hatonen, and Antti S Sorvari. A computer host-based user anomaly detection system using the self-organizing map. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 5, pages 411–416. IEEE, 2000.

[10] Dennis Ippoliti and Xiaobo Zhou. A-ghsom: An adaptive growing hierarchical self organizing map for network anomaly detection. *Journal of Parallel and Distributed Computing*, 72(12):1576–1590, 2012.

[11] Moisés F Lima, Bruno B Zarpelao, Lucas DH Sampaio, Joel JPC Rodrigues, Taufik Abrao, and Mario Lemes Proença. Anomaly detection using baseline and k-means clustering. In *SoftCOM 2010, 18th International Conference on Software, Telecommunications and Computer Networks*, pages 305–309. IEEE, 2010.

[12] Ao Liu and Bin Sun. The improved model for anomaly detection based on clustering and dividing of flow. In *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, pages 23–30. IEEE, 2019.

[13] Weibin Meng, Ying Liu, Yichen Zhu, Shenglin Zhang, Dan Pei, Yuqing Liu, Yihao Chen, Ruizhi Zhang, Shimin Tao, Pei Sun, et al. Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. In *IJCAI*, pages 4739–4745, 2019.

[14] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[15] Bingming Wang, Shi Ying, and Zhe Yang. A log-based anomaly detection method with efficient neighbor searching and automatic k neighbor selection. *Scientific Programming*, 2020, 2020.

[16] Lin Xu, Yi-Ren Yeh, Yuh-Jye Lee, and Jing Li. A hierarchical framework using approximated local outlier factor for efficient anomaly detection. *Procedia Computer Science*, 19:1174–1181, 2013.