IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, May 2023

Multi Perspective Robust Deep Analysis Cyber Anomaly Detection on HDFS Log using Transformers

Yogaraja G S R[1], Deepak M[2], Gowtham K A[3], Kiran S[4], Laxmipathi R[5]
Assistant Professor, Department of Information Science and Engineering[1]
Students, Department of Information Science and Engineering[2,3,4]
S. J. C. Institute of Technology, Chickballapur, Karnataka, India

Abstract: *Log analysis is one of the main techniques engineers use to troubleshoot faults of large- scale software systems. During the past decades, many log analysis approaches have been proposed to detect system anomalies reflected by logs. They usually take log event counts or sequential log events as inputs and utilize machine learning algorithms including deep learning models to detect system anomalies. These anomalies are often identified as violations of quantitative relational patterns or sequential patterns of log events in log sequences. While these systems provide users rich services, they also bring new security and reliability challenges. One of the challenges is locating system faults and discovering potential issues. While anomaly detection has been widely studied in the context of network data, operational data presents several new challenges, including the volatility and sparseness of data, and the need to perform fast detection (complicating application of schemes that require offline processing or large/stable data sets to converge). This paper proposes a log sequence anomaly detection method based on neural network training and feature extraction. This method uses BERT (Bidirectional Encoder Representations from Transformers) to extract the semantic features and statistical features of the log sequence.*

Keywords: HDFS

I. INTRODUCTION

In systems management and monitoring, log files are critical in identifying significant or unusual events. Nearly all software contains log files that document a history of actions performed and the results of those actions by thesystem.2 To mitigate risks, businesses regularly analyze log files for deviations from the norm: anomalies, errors, suspicious, or unauthorized activity. This is because log analysis provides valuable insight into what has happened across the infrastructure. In fact, log analysis has the potential to detect issues before or as they happen, thus reducing cost, time wasted, and unnecessary delays. Modern software systems have become increasingly large and complicated. While these systems Commercial and open-source programs quite often generate event logs related to their operation.

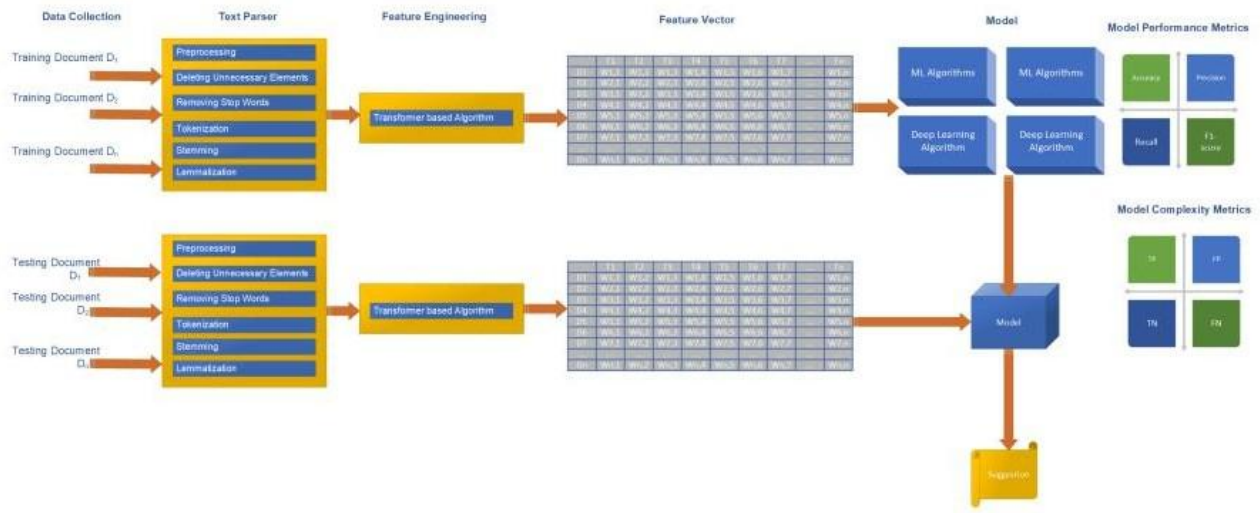
In practice they differ significantly upon application, moreover the interpretation of registered events is not clear or even ambiguous. In Windows these events are stored either in a general application log or in dedicated logs.

In Unix usually these events are stored in various files according to syslog logging scheme. Manual analysis of logs is impossible due to a large amount of generated events, the more that sometimes they are mixed within different log files. Provide users rich services, they also bring new security and reliability challenges. One of the challenges is locating system faults and discovering potential issues. Log analysis is one of the main techniques engineers use to troubleshoot faults and capture potential risks. When a fault occurs, checking system logs helps detect and locate the fault efficiently. However,with the increase in scale and complexity, manual identification of abnormal logs from massive log data has become infeasible. For example, Google systems generate millions of new log entries every month, meaning tens of terabytes of log data daily. For such a large amount of data, the cost of manually inspectinglogs is unacceptable in practice. Another reason is that a large scale modern software system, such as an online service system, may comprise hundreds or thousands of machines and software components. Its implementation and maintenance usually rely on the collaboration of dozens or even hundreds of engineers. It is impractical for a single engineer to have all the knowledge

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, May 2023

of the entire system and distinguish various abnormal logs generated by various software components. Therefore, automated log anomaly detection methods is vital. Event detection has traditionally focused on anomaly detection in industrial tasks such as parts assembly. With increasing access to data, robotics has turned todata driven methods. With increasing computation, sensor, and actuator resources, continuous multi-modal signals are used along with better robot and environmental modeling for better event detection. Recently, there has been interest in not just identifying anomalies but also incrementally classifying them and integrating recovery mechanisms into their systems. The goal is to close loops between high level event detectors and low-level robot controllers that can optimally adjust or recover from anomalous, events. To get better knowledge on application log usefulness we have developed several tools facilitating tracing log morphology and mining their practical significance. In a large extent thisis based on regular expressions and some data exploration algorithms. General event logs (generated by the operating system) provide information on system operation and behavior (logins of the users, system start-ups, closings, crashes, restarts, security problems, problems with I/O drivers, warnings, various information or notices, etc.). The scope of generated events depends upon thesystem load, configuration, etc. Recently operating systems include more and more application logs (it is visible for subsequent versions of Windows) which are generated by applications via operating system and stored in a general application log or directly stored in individual application logs. The registered events relate to the execution of the application. Having analyzed a wide spectrum of application logs we have observed a big dispersion in formats and information contents. In a large extent this results from the fact that they are defined by the application designers. log semantic features of the log sequence, and do not utilize the statistical information contained in the log sequence.

II. ARCHITECTURE



2.1 Existing Algorithm

- Word2Vec Algorithm for Converting Textual Data to Numerical Data.
- Hard and Soft Clustering Algorithm for Anomaly Detection.

2.2 Proposed Algorithm

- Bidirectional Encoder Representation from Transformers Algorithm

2.3 Advantages of Proposed Algorithm

- Bidirectional Encoder Representation from Transformers Advantages.
- Own patterns for generating texts using an encoder-decoder approach.
- Generatecontextualized word embeddings/vectors.
- Performs a number of different NLP tasks.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, May 2023

III. PROJECT MODULES

Module 1 : Preprocessing of Log Data

Preprocessing of log keys refers to preprocessing log keys that are unstructured data and can be divided into two types depending on the use of a log parser. Parsing- based log anomaly detection involves generating log data in a standardized template format using a log parser. Parsing-free log anomaly detection involves simply preprocessing without using a log parser. We started pre-processing our raw data: 1) Switch event description to lower case, Delete URLs, Remove users names, 4) Shorten prolonged words (yeeessss to yes...), 5) Keep stop words, 6) Remove punctuation marks, unknown uni- codes, and additional delimiting characters, 7) Remove hashtags (#) and correct their texts (e.g, notracism to not racism), 8) Eliminate tweets of length less than 2, and 9) Remove emoticons

Module 2 : Feature Extraction

The main purpose of this step is to extract valuable features from log events that could be fed into anomaly detection models. The input of feature extraction is log events generated in the log parsing step, and the output is an event count matrix. In order to extract features, we firstly need to separate log data into various groups, where each group represents a log sequence. To do so, windowing is applied to divide a log dataset into finite chunks.

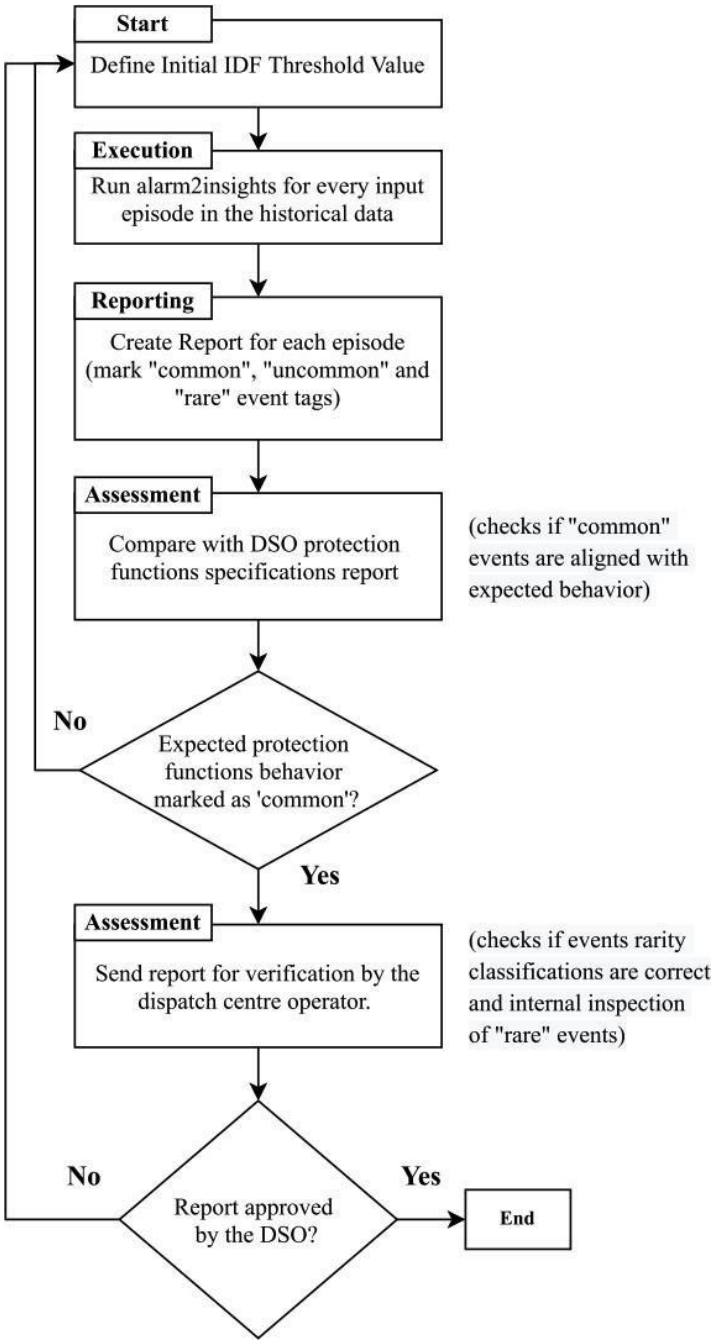
After parsing logs into separate events, we need to further encode them into numerical feature vectors, whereby machine learning models can be applied. To do so, we first slice the raw logs into a set of log sequences by using different grouping techniques, including fixed windows, sliding windows, and session windows. Then, for each log sequence, we generate a feature vector (event count vector), which represents the occurrence number of each event. All feature vectors together can form a feature matrix, that is, an event count matrix.

Module 3 : Anomaly Detection

We assess the contextual information derived from BERT. We fine-tune them using our datasets to get its contextual representations and then, ensemble models with several ensemble learning techniques: aggregation and stacking, aiming to improve performance and robustness, and to get better classification. It considers both the left and right sides of a word to determine its context. BERT is capable of multitask-learning and, performing different NLP tasks simultaneously. BERT is the first bidirectional and deep system for unsupervised learning of anomaly detection tasks from event log.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, May 2023

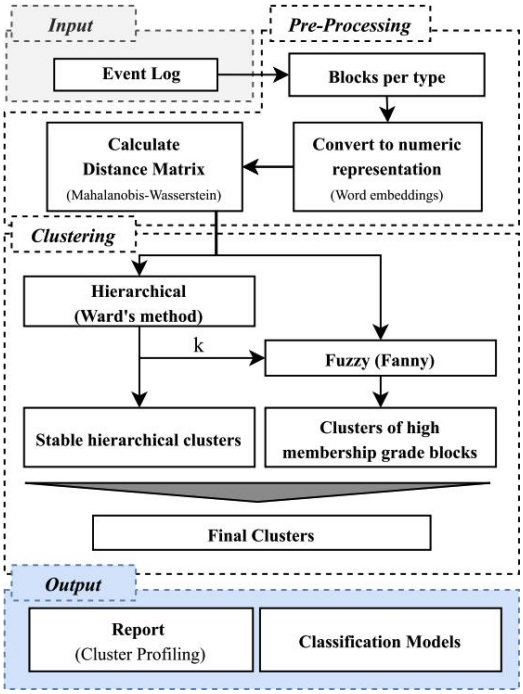
A. Methodology Used



The current circuit breaker opening alarm (i.e., in the same SS). The combination of these two types of behaviors enables the detection of overlooked problems in specific SS (i.e., by comparing with the global typical behaviors) or even the identification of SS with abnormal dynamics in protection pickup events, as presented in Section V. For MV line panels, a rule-based classification was used to aggregate historical data per fault-type, which contributed greatly to improve the definition of typical relay pickups behavior during each fault.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, May 2023

B. Process diagram for Event Profiler clustering phase.



IV. CONCLUSION

In this paper, we have proposed a neural network based log anomaly detection method, which can detect system anomalies from logs with high accuracy and stable performance by combining the structure and the important features. We presented an accurate, robust, fast, and versatile measure for skill and anomaly identification. Finding the process enhancement areas is a fundamental prerequisite for any process enhancement procedure that highly affects its outcome.

REFERENCES

[1]. M. Kezunovic, P. Pinson, Z. Obradovic, S. Grijalva, T. Hong, and R. J. Bessa, ‘‘Big data analytics for future electricity grids, ’’Electr. Power Syst. Res., vol. 189, p. 106788, Dec. 2020.

[2]. L. Wei, W. Guo, F. Wen, G. Ledwich, Z. Liao, and J. Xin, ‘‘An online intelligent alarm- processing system for digital substations, ’’IEEE Trans. Power Del., vol. 26, no. 3, pp. 1615–1624, Jul. 2011.

[3]. S. Pandey, A. K. Srivastava, and B. G. Amidan, ‘‘A real time event detection, classification and localization using synchro phasor data, ’’IEEE Trans. Power Syst., vol. 35, no. 6, pp. 4421–4431, Nov. 2020.

[4]. N. Baranovic, P. Andersson, I. Ivankovic, K. Zubrinic-Kostovic, D. Peharda, and J. E. Larsson, ‘‘Experiences from intelligent alarm processing and decision support tools in smart grid transmission control centers, ’’in Proc. CIGRE Session, Paris, France, Aug. 2016, pp. 1–10.

[5]. C. Feng, L. Wang, R. Ye, J. Gu, L. Xie, Y. Wang, Q. Feng, and C. Cui, ‘‘Research and application of alarm optimization mechanism in power grid operation monitoring, ’’in Proc. IEEE 5th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC), Chongqing, China, Mar. 2021, pp. 2352–2356.

[6]. D. B. Tesch, D. C. Yu, L.-M. Fu, and K. Vairavan, ‘‘A knowledge-based alarm processor for an energy management system, ’’IEEE Trans. Power Syst., vol. 5, no. 1, pp. 268–275, Feb. 1990.

[7]. G. Sun, X. Ding, Z. Wei, P. Shen, Y. Zhao, Q. Huang, L. Zhang, and H. Zang, ‘‘Intelligent classification method for grid-monitoring alarm messages based on information theory, ’’ Energies, vol. 12, no. 14, p. 2814, Jul. 2019.

[8]. D. S. Kirschen and B. F. Wollenberg, ‘‘Intelligent alarm processing in power systems, ’’Proc. IEEE, vol. 80, no. 5, pp. 663–672, May 1992.