Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/92/20241730

BERT-based log exception detection algorithm LADB

Ruochun Zhao[1,2,*], Meng Zhang[1,3]

[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Malaysia

[2]zrc2664308120@gmail.com
[3]zm264818@163.com
*corresponding author

Abstract. With the development of information systems, they have become increasingly large and complex and have generated a large amount of log information. These log information records the system's health, but the number of log information is huge, and the traditional exception detection algorithm in the case of very large amounts of data is difficult to efficiently and accurately detect anomalies due to poor generalization performance. A BERT-based log anomaly detection algorithm, LADB, is proposed, essentially a semi-supervised, multi-classification algorithm. (1) LADB uses the Transformer encoder as the base component for problems such as feature degradation and gradient explosion. (2) In order to make better use of the bidirectional context, and in view of BERT's excellence in the NLP field, the Masking Log Key Prediction (MLKP) self-monitoring task was designed, drawing on the idea of BERT's Masking Language Model. (3) In order to solve the problem of difficult and slow processing of high-dimensional data, the Deep SVDD algorithm is used for minimum superspheres capacity (VHM) self-supervision training task. Experiments have shown that LADB's combined performance is superior to the four representative log anomaly detection algorithms.

Keywords: Log Anomaly Detection, Transformer, BERT, Masked Logits Key Prediction, Minimum hypersphere capacity.

1. Introduction

Log information is an important resource in system operation and records system status and specific events. Log analysis is critical to anomaly detection, system diagnosis, and security, helping maintenance personnel reproduce and correct errors and improve system reliability[1]. However, the large amount of log data generated daily by the system, such as a 1000-employee enterprise with over 100GB of log traffic per day and a peak of 22,000 events per second, shows the critical importance of effectively analyzing logs.

This paper aims to compare the various algorithms available for log anomaly detection, based on neural network optimization (pre-selection of BERT), to propose a neural network-based log anomaly detection algorithm and design to implement a streaming log anomaly detection algorithm.

DOI: 10.54254/2755-2721/92/20241730

2. Background

As information systems become more complex and complex, a large amount of log information is generated. These log messages record the health of the system, and log exception detection is the detection of logs to determine if the system is functioning properly. Log anomaly detection significantly reduces the effort of the operational maintenance staff while providing high accuracy. Common techniques include machine learning, deep neural networks[2].

2.1. Transformer

The Transformer is a deep learning model architecture specifically designed to work with sequence data. It dynamically focuses on all positions in the sequence through self-attention mechanisms, capturing long-distance dependencies, and is more efficient in handling complex patterns than traditional models[3] The Transformer consists of an encoder and a decoder that converts the input sequence into a context-sensitive representation, and the decoder generates the target sequence. Its multi-head attention mechanism and location coding design make it[3] an excellent performer in natural language processing tasks, widely used in machine translation, text generation, etc[4].

The Transformer is divided into two parts:Decoder and Encoder. Both parts can be repeated N times, and the paper defaults to N=6, which is 6 Encoders and 6 Decoders. The Transformer was originally used for machine translation and is structured as shown in Figure 1.

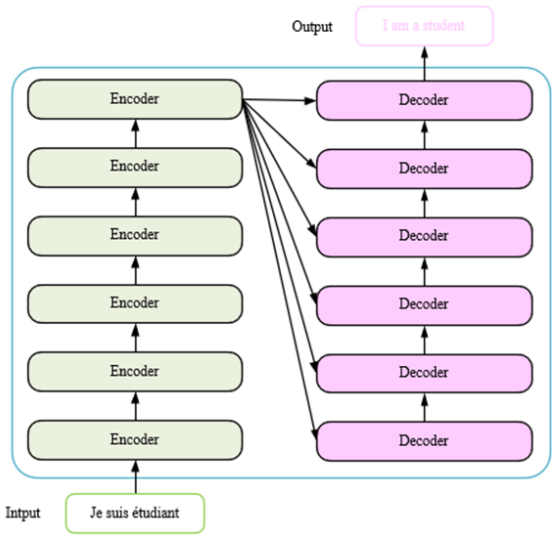

Figure 1. Transformer general structure.

2.2. BERT

BERT(Bidirectional Encoder Representations from Transformers)is an advanced pre-trained language algorithm. BERT uses a multi-layer bidirectional Transformer codec (6 encodings, 6 decoders), a masked language model, and a next sentence prediction to take advantage of bidirectional contextures[5]. A two-way context is a context that is subject to both the left and right context.

3. BERT-based log exception detection algorithm LADB

3.1. Algorithm Principles

LADB is a deep learning neural network based on BERT log anomaly detection. The main purpose of LADB is to learn the context information in the log sequence. LADB designed two self-monitoring tasks, Masking Log Key Prediction (MLKP) and Hypersphere Minimizing Capacity (VHM), to analyze log sequences in both directions.

DOI: 10.54254/2755-2721/92/20241730

The MLKP mimics the MLM, randomly replaces a fixed percentage of the log key in the log sequence with the [MASK] field, and then predicts the masked log key using the probability distribution. The VHM task inserts a [DIST] field at the beginning of the log sequence and uses the training results of [DIST] to represent the log sequence in potential space. The VHM goal is to calculate the minimum volume of a supersphere containing a normal log sequence vector. Intuitively, the normal log sequence vectors are concentrated in the center of the supersphere, while the abnormal log sequence is far from the center.

The main structure of LADB is a Transformer encoder that relies entirely on the attention mechanism. The input to the Transformer encoder is the sum of the log sequence embedding vector and the position embedding vector. The output of the Transformer encoder is then sent to a fully connected layer and a SoftMax, and after cycling six encoders, a probability distribution for each log key in the log sequence is generated as a prediction of the masked log key.

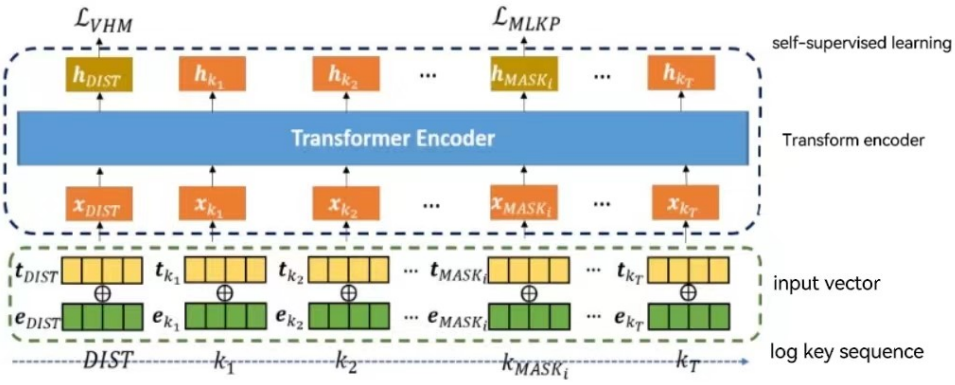

Figure 2. LADB schematic.

3.2. Algorithm Steps

The complete flow of the algorithm is shown in Figure 3. First log preprocessing, feature extraction to get the log sequence, mask the log sequence and then compute the embedded vector, calculate the position encoding vector, Add the embedded vector and the position vector to get the input vector, then feed the LADB model to train the best training model, supersphere center, and then enter the model prediction after processing the closest window series of the sequence to be detected. The prediction requires the use of the supersphere center to eventually generate a probability sequence, and the algorithm Logsy[6] is used to determine the anomaly in the same way.

DOI: 10.54254/2755-2721/92/20241730

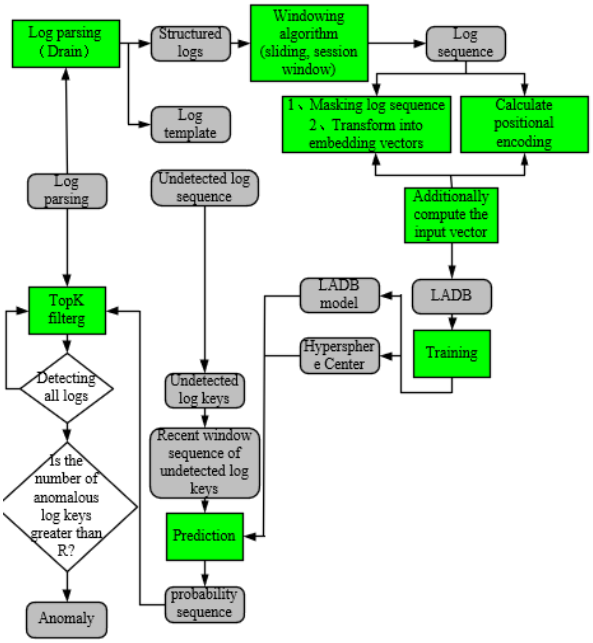

Figure 3. Flowchart of BERT-based log anomaly detection.

4. Experiment and Analysis

4.1. Algorithm Steps

The objective of the experiment is two, one is to compare the comprehensive performance of the other four representative classical algorithms to verify whether the algorithm is comprehensive high performance. Second, verify the comprehensive performance of self-supervision tasks.

4.2. Experimental Data Sets

This article uses a smaller HDFS log dataset for offline training. HDFS data is a Hadoop system log collected from the Amazon EC2 platform and is flagged.

4.3. Evaluation Indicators

In this section, both experiments use the evaluation index precision, recall, F1 to measure the overall performance of the algorithm, using memory consumption. The run time is an evaluation metric to evaluate the efficiency of the algorithm.

4.4. Analysis of Experimental Results

(1) Integrated performance verification of algorithms

First, without ablation experiment, normal training, visualization of the model training effect, to confirm that the model has achieved the best results in the current data set. The loss approximation of

DOI: 10.54254/2755-2721/92/20241730

0.2 in the figure is ideal, with neither overfitting nor underfitting. Its loss variation curve is shown in Figure 4.

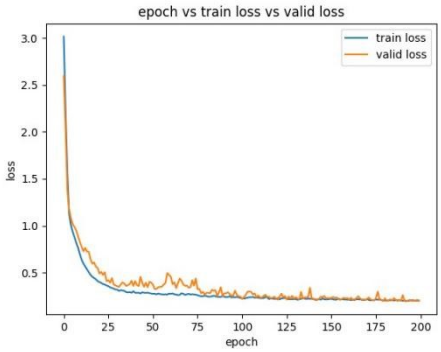

Figure 4. LADB training loss curve.

As shown in Table 1, the F1 of the LADB model proposed in this paper is nearly 80%, recall is nearly 70%, and precision reaches 92.6%. The comprehensive model is better than Logsy and other algorithms.

Table 1. HDFS Comparison of experimental results1.

model	F1	precision	recall	loss	TopK
PCA	0.312	0.967	0.186	-	-
AutoEncoder	0.082	0.188	0.053	0.005	-
DeepLog	0.494	0.950	0.333	0.0673	5
Logsy	0.667	0.967	0.509	0.2118	5
LADB	0.796	0.926	0.698	0.2583	5

As can be seen from Table 2, LADB's memory is smaller with PCA similar to the space complexity is very good, although the training time is longer but can be received, in general the comprehensive performance of the LADB model is better, and reached the expected experimental goals.

Table 2. HDFS Comparison of experimental results2.

model	Memory consumption	Time consuming
PCA	637.4Mb	10min 17s
AutoEncoder	2315.7Mb	12min 5s
DeepLog	2363.3Mb	12min 39s
Logsy	1763.0 Mb	12min 43s
LADB	787.6Mb	13min 57s

(2) Self-monitoring task performance

The results in Table 3 show that training with only the masked log key prediction task (MKLP) achieves good log anomaly detection, demonstrating its effectiveness. Without VHM training, LADB still outperforms DeepLog in F1 scores, indicating the Transformer encoder's advantage over LSTM due to its multi-head attention and masked log key prediction. However, VHM alone performs poorly, suggesting that distance is insufficient for effective anomaly detection. Training LADB with both self-supervised tasks improves performance, especially on the HDFS dataset, where the F1 score (79.6) is higher than using MKLP alone (75.8), as shorter log sequences enhance prediction accuracy and vector aggregation. Thus, using both tasks generally leads to better performance.

DOI: 10.54254/2755-2721/92/20241730

Table3. Results of ablation experiments.

Target function	F1	precision	recall
MLKP	75.8	84.6	68.7
VHM	5.08	2.74	34.5
MLKP+VHM	79.6	92.6	69.8

5. Summary

This paper proposes a BERT-based log anomaly detection algorithm LADB, which is essentially a semi-supervised multi-classification algorithm. To take advantage of the bidirectional context, the Masking Log Key Prediction (MLKP) self-monitoring task is designed. In order to solve the problem of difficult and slow operation of high-dimensional data, a minimum supersphere capacity (VHM) self-monitoring task based on DeepSVDD was designed. To solve the problem of gradient explosion and gradient disappearance, the Transformer encoder was used as the base component of LADB. Finally, the overall performance of LADB was proven to be superior.

References

[1] Oliner A, Ganapathi A, Xu W. Advances and challenges in log analysis[J]. Communications of the ACM, 2012, 55(2): 55-61.
[2] RAPIDS.cy. BERT: Neural network, that's the tech to free your staff from bad regex[EB].
[3] A. D. V., M. J. G., Pau R., et al. Logo detection with no priors[J]. IEEE Access, 2021, 9: 106998-107011.
[4] He P, Zhu J, Zheng Z, et al. Drain: An online log parsing approach with fixed depth tree[C]//2017 IEEE international conference on web services (ICWS). IEEE, 2017: 33-40.
[5] Du M, Li F. Spell: Streaming parsing of system event logs[C]//2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, 2016: 859-864.
[6] STUDIAWAN H, FERDOUS S, PA YNE C. A survey on forensic investigation of operating system logs[J]. Digital Investigation, 2019, 29: 1-20.