An Anomaly Prediction of Spark Log Based on Self-Attention GRU Network

Yanyu Gong, Xinjiang Chen, Xiaoli Zhang, Haotian Xu, Xue Zhang and Haifeng Wang*
Linyi University, Linyi, China

Keywords: Spark, Log Anomaly Prediction, Multiple Attention, GRU, Transformer.

Abstract: This paper proposes a GRU network training prediction model to solve the problem of the difficulty of separating time series in Spark framework, so that the abnormal prediction model of Spark system can be realized in big data frameworks. Task is used to separate log data, SwissLog is used to transform it into a vector, and multiple attention mechanisms are used to deepen the repeated log series (X. Li, Li Xiaoyun). To begin with, this paper solves the problem that Spark log data workflows are difficult to separate due to multi-thread output, and then log data cannot be converted into vectors. The robustness of structured data of log sequence conversion is further improved by optimizing and modifying SwissLog prefix tree and replacing it with Jaccard similarity algorithm, which improves anomaly prediction accuracy. To train the normal prediction model, the repeated time series is taken as the incremental dimension, and a GRU network with multiple attention mechanisms is used. As a result, the operation efficiency and model accuracy are improved, and equipment memory requirements are greatly reduced when training a large number of data sets. Based on the results presented in this paper, the GRU model for repetitive log sequences with multi-attention mechanism achieves the highest accuracy of 86.77% in the general public Spark data set from LogHub, and the accuracy and performance are 1.16 percent higher than the latest benchmark model LSTM, indicating that the proposed model can enhance anomaly detection accuracy and robustness effectively.

1 INTRODUCTION

The reliability of big data systems has become increasingly important as large-scale complex systems penetrate all aspects of social life. Reliability can be improved by analyzing tens of thousands of logs generated by big data systems. Detecting system anomalies can be achieved by mining, analyzing, and automatically identifying system logs.

Log anomaly analysis mainly refers to the analysis of system-generated logs to detect possible abnormal behaviors. Log anomaly analysis can help improve the security and stability of the system, and avoid potential risks and losses by finding system problems early. For example, log anomaly analysis on the server can be used to detect abnormal behaviors such as unauthorized access behavior and malware attacks. In the security field, log anomaly analysis is also an important part of intrusion detection system. However, it is difficult for programmers to control the system manually because of the large amount of data and chaotic relations in the logs. With the popularization and application of large-scale complex systems, it is difficult for operators and programmers to quickly find the problems in the system through naked eye monitoring or simple keyword retrieval. However, finding problems through naked eye monitoring or simple keyword search requires a lot of learning costs, and the abnormal problems of a complex system often involve multiple problems. Therefore, how to make the system monitor itself according to the logs generated by the system to realize the observability of the system has become a very important research direction in the current industry development.

Analyzing logs for anomalies refers primarily to detecting possible abnormal behavior in system logs. By detecting system problems early, log anomaly analysis can improve the security and stability of the system. Log anomaly analysis on the server, for instance, can be used to detect unauthorized access behavior and malware attacks. Log anomaly analysis is also an important part of intrusion detection systems in the security field. Due to the large amount of data and chaotic relationships in the logs, it is difficult for programmers to control the system manually. Large-scale complex systems are becoming increasingly popular, making it difficult for

Gong, Y., Chen, X., Zhang, X., Xu, H., Zhang, X. and Wang, H.
An Anomaly Prediction of Spark Log Based on Self-Attention GRU Network.
DOI: 10.5220/0012285100003807

In *Proceedings of the 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (ANIT 2023)*, pages 415-421
ISBN: 978-989-758-677-4

operators and programmers to identify problems quickly with naked eye monitoring or simple keyword searches. However, finding problems through naked eye monitoring or simple keyword searches involves a lot of learning costs, and abnormal problems of complex systems often involve multiple problems. As a result, making the system monitor itself using the logs generated by the system in order to realize the observability of the system has become a very important research direction.

2 RELATED WORK

Currently, log anomaly detection is mostly based on rules. Check the log for violations of the known log patterns and rules, so that abnormal logs can be found. Sakhnini, S, et al. proposed a rule-based anomaly detection algorithm for processing Web server logs (Sakhnini, S., 2019). This rule-based anomaly detection algorithm was tested on two real Web server log files, and compared with other common anomaly detection algorithms based on machine learning and statistics. Its disadvantages include the need to define rules manually and its inability to handle uncertain logs. In order to detect abnormal logs, the number of occurrences and expected values are counted according to the statistical method. Accordingly, Hou, M., proposed a statistical anomaly detection method based on normal distribution models and Z-score statistics (Hou, M., 2019), which detects abnormal behavior by comparing the deviation between the observed and expected values. Behavioral analysis has the disadvantage that it cannot handle dynamically changing log types and distributions of data. It compares and analyzes logs according to the system's normal behavior pattern in order to detect abnormal behavior. As an example, N. Feng et al. propose a method for detecting abnormal events in large-scale logs based on deep reinforcement learning (N. Feng, 2020). Under the condition of ensuring that the correct rate is achieved, this method uses deep reinforcement learning algorithm to make decisions and minimize the false alarm rate. It is suitable for dynamically adjusting systems and services, but it requires manually defining behavior patterns and feature extraction techniques. LogST is a practical log-based anomaly detection method proposed by Mingyang Zhang and others (L. Forgor, 2019). Using the SBERT model, a GRU model for anomaly detection is constructed by considering the word order relationship in log events. The detection accuracy of LogST is stable when there are sufficient normal logs and a small number of normal logs are marked. For multi-task log output like Spark, it is difficult to separate logical time series; Traditional prediction models lack special methods for handling repeated sequences.

On stable log data, log-based anomaly detection has achieved satisfactory results. As a result of these studies, the algorithm's performance has improved greatly, making anomaly detection more accurate.

Research has been limited to optimizing the algorithm and improving the accuracy of the model, ignoring the fact that logs are often output in parallel by multiple threads and workflows in real systems. As a result, logs cannot be used directly for anomaly analysis. This paper focuses on how to separate and convert log data into time series with actual logical relationships for log process analysis.

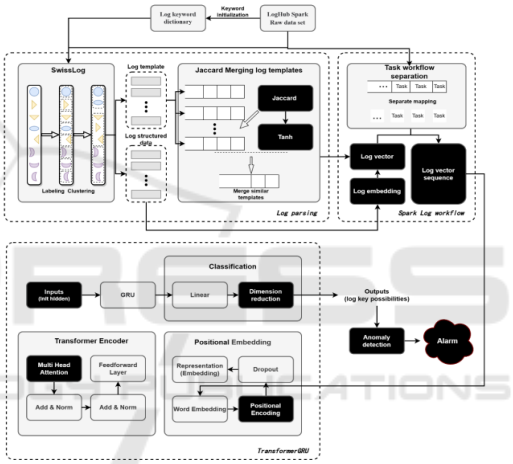

Figure 1. Master design drawing.

3 THIS MODEL

Log anomaly prediction based on machine learning is divided into two parts: log parsing and anomaly detection. The unstructured log is transformed into structured log, the log template is extracted, and finally the log time series vector is resolved. The second part of this paper uses machine learning to extract the characteristics of log time series and build a model to identify abnormal log events. Based on the existing log data, the model can learn the normal log event pattern and identify abnormal events.

According to figure 1, this paper analyzes log data using SwissLog log parsing and sentiment embedding; Secondly, a Spark log is separated from the workflow and transformed into a time series using Task as a benchmark (Z. Liu, 2021). A model is

constructed to identify abnormal log events based on a Transformer-GRU network model.

3.1 Log Parsing Module of SwissLog

SwissLog extracts multiple templates by tagging, lexicography, and clustering historical log data using a novel log parsing method. Instead of event IDs, these templates are saved as natural sentences. Use a sliding window to construct a log sequence called "session" by linking these log statements with the same identifier. The log sequence is then converted into semantic and time information.

Generally, log statements are readable, and most of the words can be found in dictionaries. To parse logs, you can use a dictionary-based method. By using this method, the log statement is divided into a combination of valid and invalid words. In the case of a dictionary, a word divided from the log that exists in the dictionary is a valid word. A valid word in a log can be formed by the collection of valid words. "Launching container container _ 144800611297 _ 0138 _ 01 _ 0000-10 for on host meso-slave-18", the log will be divided into seven words, namely, launching, container, and container _ 144800611297 _ 018. Figure 2 shows the results of searching these tags in the dictionary to get a set of valid words {"launching", "container", "for", "on" and "host"}.

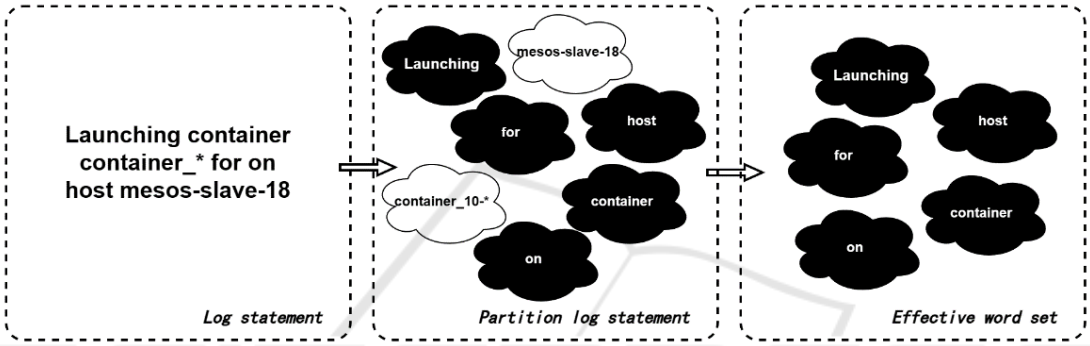

Figure 2. Log Word Segmentation Process.

The occurrence times of words in the valid word set should be recorded. Merging the two logs into one template is only possible if the valid words and their occurrence times are identical in both logs. The public part of the log is the constant part, while the variable part is the variable part of the log of the same template. Using the longest common subsequence algorithm, replace the variable part with *. Let's say there are two log statements in a cluster, A and B. As follows:

Table 1. A and B log words.

Journal	Log statement
A	{"Launching"、"container"、"container_1448006111297_0138_01_000010"、"for"、"on"、"host"、"mesos-slave-18"}
B	{"Launching"、"container"、"container_1448006111297_0138_01_000010"、"for"、"on"、"host"、"mesos-slave-18"}

Since A and B have the longest common subsequence ("launching", "container", "for", "on", and "host", the shielding result is "launching container * of for on host *".The variable contains only valid words. In e4, the variable part is admin, and in e5, the variable part is root. Due to the fact that these two words are valid words, they will be divided into two different templates. SwissLog solves this problem using a prefix tree. However, the prefix tree can only solve the problem that two different templates are divided into different templates because of different suffix effective words, not in actual situations (Binlong Zhang, Fuhong Tang).

The following two templates are displayed:

Table 2. Log template.

Journal	Log template
T1	Block <*> stored as bytes in memory (estimated size <*>, free <*>)
T2	Block <*> stored as values in memory (estimated size <*>, free <*>)

Basically, there is no difference in the actual business level between log templates T1 and T2 when it comes to storing data in memory. Because the effective words are in the middle, log templates T1 and T2 cannot solve this kind of merging problem.

The paper proposes a Jaccard similarity template merging algorithm based on information entropy to solve the above problems (T. Kongsin, 2020). To implement the attention mechanism, the algorithm introduces the activation function tanh,

focuses on the important suffix information, and merges the secondary log templates. For processing, two log templates that reach the similarity threshold are merged into one log template (O. Ertl, 2022).
The Jaccard similarity index measures the similarity between two sets. In Jaccard similarity, two sets are intersected and their union ratio is considered. The definition of Jaccard similarity is as shown in Formula 1 assuming it is a valid word in the corresponding position of two log templates.

$$\text{Jaccard}(\text{T1}_k, \text{T2}_k) = \frac{|\text{T1}_k \cap \text{T2}_k|}{|\text{T1}_k \cup \text{T2}_k|} \quad (1)$$

The formula for calculating the similarity of two log templates is shown in Formula 3, where s is the similarity (0~1) of input templates T1 and T2, and $\text{Jaccard}((t1, t2)_k)$ is used to calculate the similarity of each corresponding position of the log template. In the absence of a template string at that position, return Formula 2 directly and finally use tanh activation will be used to multiply and quadrature (U. Srinivasarao, 2022).

$$S(T1, T2)_k = \frac{Max(Len(T1), Len(T2)) - 1}{Max(Len(T1), Len(T2))} \quad (2)$$

$$S(T1, T2) = \prod_{k=1}^{n} (\tanh(\text{Jaccard}(t1, t2)_k)) \quad (3)$$

When that similarity S(T1,T2) of the two log template of t1 and t2 reaches a threshold value, the two log templates are merged into one log template, whereby t1 and t2 can be used as a unified log template; When the similarity s between pairwise is greater than the threshold, the unified log template is taken from the template with the highest pairwise direct similarity sum.

3.2 Word2vector Spark Processing and Workflow Module

Log Workflow abstracts log data into an event stream, which is processed, filtered, and transformed in time. Many steps are involved in this process, such as collection, storage, cleaning, conversion, and analysis. Machine learning can be applied to log workflows if each log is converted into a vector.

The log anomaly prediction model is based on one workflow because log data generated by each application system is often output in parallel. A log file, however, is often composed of several workflows. Due to this, it is very important to deal with multiple mixed workflows in a way that makes them conform to the characteristics of time series. The Spark workflow model separates Spark mixed log sequence data.

Based on the data set structure of Log Hub Spark in Figure 3, the highlighted part accounts for the majority of all logs. According to the figure, Spark generally reports errors due to running calculation code and data errors in Task code. The following table illustrates how a Task life cycle can often be viewed as a time series:

17/06/09 20:10:45 INFO executor.Executor: Running task 1.0 in stage 0.0 (TID 0)
17/06/09 20:10:45 INFO executor.Executor: Running task 2.0 in stage 0.0 (TID 2)
17/06/09 20:10:45 INFO broadcast.TorrentBroadcast: Started reading broadcast variable 9
17/06/09 20:10:45 INFO storage.MemoryStore: Block broadcast_9_piece0 stored as bytes in memory (estimated size 5.2 KB, free 5.2 KB)
17/06/09 20:10:45 INFO broadcast.TorrentBroadcast: Reading broadcast variable 9 took 160 ms
17/06/09 20:10:46 INFO storage.MemoryStore: Block broadcast_9 stored as values in memory (estimated size 8.8 KB, free 14.0 KB)
...
...
...
17/06/09 20:10:48 INFO executor.Executor: Finished task 2.0 in stage 0.0 (TID 3). 2703 bytes result sent to driver
17/06/09 20:10:48 INFO executor.Executor: Finished task 1.0 in stage 0.0 (TID 1). 2703 bytes result sent to driver

Using Task as a workflow separation standard, however, often includes other redundant logs, including other Tasks coupled with the target Task, which is a time series. It is necessary to set a threshold or degree of completion to terminate the workflow based on Running task 1.0 as the workflow start and Finished task 1.0 as the workflow end. When separating a workflow, if the separated workflow log includes Warnings, Error levels, or abnormal errors, then mark the workflow as abnormal, otherwise mark it as normal.

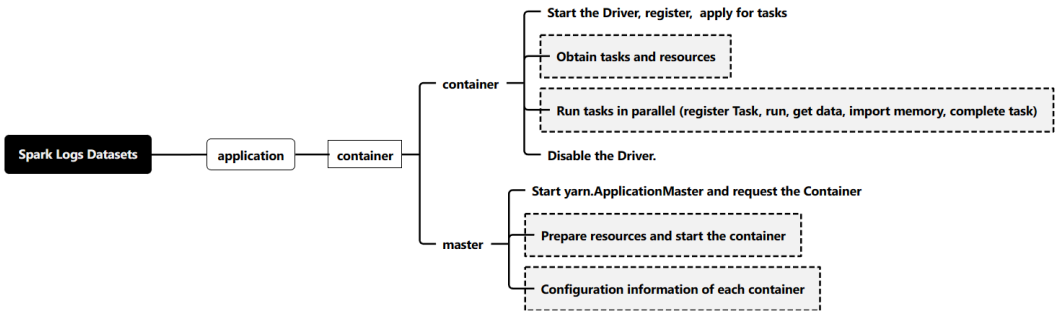

Figure 3. LOGHUB SPARK data set structure.

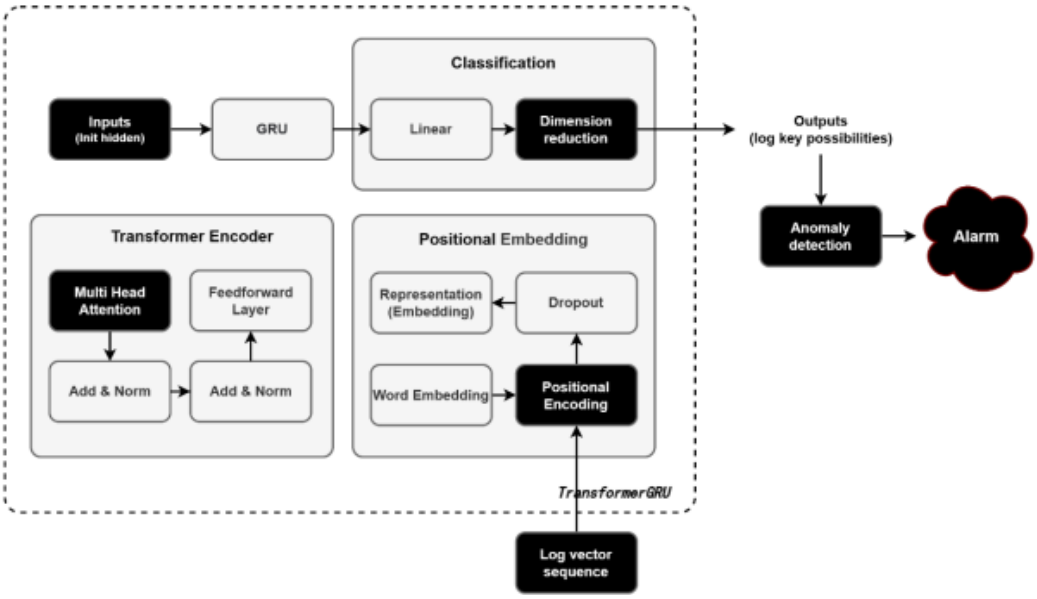

Figure 4. Architecture diagram of anomaly detection model.

3.3 GRU Anomaly Detection Module of Self-Attention Mechanism

By combining the powerful global feature extraction capabilities of Transformer and the powerful sequence feature extraction abilities of circular models, this paper proposes a Transformer-GRU (T-GRU), which combines the GRU and Transformer structures, and applies it to log a nomaly detection (Haoyi Zhou, Shitao Wang). In this paper, it is compared with LSTM, LSTM, and GRU. As a result of the model, good results have been achieved.

As shown in the diagram below, the model architecture is as follows:

SwissLog parsing and workflow time series conversion are followed by mapping the log data into a continuous vector representation, and the output encodes each layer's position information (G. Mengrong, 2022).

Each position is represented by a special encoding vector in Positional Encoding. A sine and a cosine coding vector is made up of sine and cosine functions, and each position is a unique coding vector. As a result, the Transformer model is able to recognize words or symbols in different positions and determine their positional relationship (J. -H. Wang, 2022).

GRU acts as a decoder to decode and output, and the output is linearly transformed to produce the final results based on the encoded information input into Transformer Encoder according to the time window for encoding (S. Dutta, N. Dhingra).

4 EXPERIMENT AND RESULT ANALYSIS

4.1 Experimental Environment and Data Set

Python version 3.9 is used to program the code in this paper, based on PytorchV1.10.1 deep learning framework. A GTX 3090Ti graphics card was used in an experimental environment with Ubuntu22 as the operating system.

Specifically, the Spark data set provided in LogHub is aggregated and collected in a laboratory environment that contains 32 physical computers and comes from the common public data set of LogHub intelligent log analysis. At the machine level, log data is aggregated, and it exceeds 2 GB in size. As the data set has not been manually processed, it contains raw data, including abnormal application records. As shown in Table 1, this specific data set contains the following information:

Table 3. LogHub Spark Data Set.

Dataset	Number of Abnormal workflows	Number of normal workflows
Training set	-	820895
Test set	3745	204969

4.2 Experimental Parameter Setting

This paper embeds the 64-dimensional word2vector vector initialization word, and all weight parameters are initialized uniformly. Adam is selected as the optimizer for the hidden layer with a dimension of 64. Use a learning rate of 0.001, a random inactivation parameter of 0.01, and a batch size of 256. A random initialization of 100 times was performed, and the results of the average of those 100 times were taken as the final results. Multiple dimensions are used to evaluate the effectiveness and robustness of the experimental results, including accuracy, recall, and F1 value.

4.3 Experimental Structure Analysis

In Table 2, experimental results demonstrate that the performance of this method is superior to that of other comparable models, providing evidence for its effectiveness. Using two network models to capture semantic data from multiple directions, this paper successfully reduces the gradient of model training data by analyzing the SwissLog log and separating the log workflow. This also improves the training effect and efficiency, and eliminates the problem of overfitting.

Table 4. Comparison of experimental effects

Model name	Accuracy/%	Recall rate/%	F1 value
LSTM	85.61	78.54	84.51
GRU	84.10	76.91	82.86
TransformerGRU	86.77	81.39	86.01

5 CONCLUSION

Several studies have been conducted on the text classification of traditional texts, but little research has been conducted on the text classification of update logs. Update logs contain a great deal of functional and security information, so they are valuable for future security research and topic annotation. In order to improve SwissLog's parsing logs based on the characteristics of updating logs, we propose a new TransformerGRU network model and develop a new workflow for log serialization preprocessing that is more efficient and accurate. Furthermore, the self-attention model is used to balance different log key data sets with a high degree of quality, which further enhances the classification effect. Experiments have demonstrated that this method is highly accurate and efficient for classifying objects.

ACKNOWLEDGEMENTS

This project is supported by Shan dong Province Science and Technology Small and Medium Enterprises Innovation Ability Enhancement Project of China (No. 2023TSGC0449)

REFERENCES

X. Li, P. Chen, L. Jing, Z. He and G. Yu, "SwissLog: Robust and Unified Deep Learning Based Log Anomaly Detection for Diverse Faults", 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), Coimbra, Portugal, 2020, pp. 92-103.

Li, Xiaoyun and Chen, Pengfei and Jing, Linxiao and He, Zilong and Yu, Guangba, "SwissLog: Robust Anomaly Detection and Localization for Interleaved Unstructured Logs", 2022 IEEE Transactions on Dependable and Secure Computing, Coimbra, Portugal, 2022.

Sakhnini, S., & Alazab, M. (2019). Rule-Based Anomaly Detection Algorithm for Web Server Log Files. IEEE Access, 7, 24478-24486.

Hou, M., Jin, R., & Zhou, S. (2019). A statistical approach for anomaly detection in web log data. Journal of Network and Computer Applications, 131, 52-59.

N. Feng, Y. Zhang, Y. Chen, Detecting Anomalies in Large Scale Logs via Deep Reinforcement Learning. In IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2020,1185-1198.

L. Forgor, W. Brown-Acquaye, J. K. Arthur and S. Owoo, "Security of Data on E-waste equipment to Africa: The Case of Ghana, " 2019 International Conference on Communications, Signal Processing and Networks (ICCSPN), Accra, Ghana, 2019, pp. 1-5.

Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 9992-10002.

Transformer-Encoder-GRU (T-E-GRU) for Chinese Sentiment Analysis on Chinese Comment Text by Binlong Zhang, Wei Zhou, 2021, 211-234.

Fuhong Tang and Kwankamol Nongpong. Chinese sentiment analysis based on lightweight character-level bert. In 2021 13th International Conference on Kno wledge and Smart Technology (KST), pages, 2021 27–32.

T. Kongsin and S. Klongboonjit, "Machine Component Clustering with Mixing Technique of DSM, Jaccard Distance Coefficient and k-Means Algorithm" 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), Bangkok, Thailand, 2020, pp. 251-255.

O. Ertl, "ProbMinHash – A Class of Locality-Sensitive Hash Algorithms for the (Probability) Jaccard Similarity, " in IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 7, 2022, pp.3491-3506.

U. Srinivasarao, R. Karthikeyan, P. K. Sarangi and B. S. Panigrahi, "Enhanced Movie Recommendation and Sentiment Analysis Model Achieved by Similarity Method through Cosine and Jaccard Similarity algorithms, " 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 214-218..

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. arXiv preprint arXiv: 2020,2012.07436.

Shitao Wang, Jiangfeng Li, and Defeng Hu. Bigru-multi-head self-attention network for chinese sentiment classification. In Journal of Physics: Conference Series, volume 1827, 2021, page 012169.

G. Mengrong and L. Hongjian, "Research on temperature prediction of subway transformer based on LSTM, " 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2022, pp. 555-558.

J. -H. Wang, M. Norouzi and S. M. Tsai, "Multimodal Content Veracity Assessment with Bidirectional Transformers and Self-Attention-based Bi-GRU Networks, " 2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM), Naples, Italy, 2022.

S. Dutta and S. Ganapathy, "Multimodal Transformer with Learnable Frontend and Self Attention for Emotion Recognition" ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 6917-6921.

N. Dhingra, F. Ritter and A. Kunz, "BGT-Net: Bidirectional GRU Transformer Network for Scene Graph Generation, " 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 2021, pp. 2150-2159.