Literature Review

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Title (year) | Authors | Dataset | Methodology | Findings | Improvements done/needed |
| Hit Anomaly: Hierarchical Transformers for Anomaly Detection in System Log (2020) | Shaohan Huang, Yi Liu, Carol Fung, Rong He, Yining Zhao, Hailong Yang and Zhongzhi Luan | * HDFS * BGL dataset * OpenStack dataset | Hit-Anomaly has three main components: log sequence encoder, parameter values encoder, and attention-based classification. The first step is log parsing which transfers raw messages into structured log templates associated with key parameters. Given a log template sequence, a log sequence encoder transforms it into a fixed-dimension vector (log sequence representation). Parameter value encoder convert parameter values into parameter value representation. After log sequence encoding and parameter value encoding, HitAnomaly leverages the attention-based structure to classify anomalous logs, which can learn to assign various degrees of importance to log sequences or parameters. | In general, the precision of HitAnomaly is fairly stable with respect to different window sizes or the number of layers. Compared to the original HitAnomaly, HitAnomaly without log sequence encoding has a much lower precision. The information on the log template sequence is essential for the anomaly detection model. | Most early research on log-based anomaly detection uses rule-based approaches which uses keywords or regular expression to match anomaly logs. These are limited by specific scenarios and also require domain knowledge. HitAnomaly is a general approach that does not rely on any domain-specific knowledge. One of the future directions is to incorporate the transformer structure into a log-based anomaly prediction task. This, in turn, will be able to predict anomalies before they occur and allows actions to be taken to prevent anomalies from happening and reduce the damage from anomalies. |
| LAnoBERT: System Log Anomaly Detection based on BERT Masked Language Model (2021) | Yukyung Lee, Jina Kim and Pilsung Kang | * HDFS * BGL | The operation mechanism of LAnoBERT proposed here because it is executed through MLM, which is a pre-training method of BERT, MLM. MLM was not only applied in the training phase but also in anomaly detection to detect abnormal logs because of the following reasons. First, there is ample data available for training BERT because the log data are collected in real time. Second, MLM does not require the labeling of tasks and accords with the purpose of anomaly detection where only normal data are used for training. Third, MLM is an appropriate methodology to apply to anomaly detection from the perspective of prompt-based learning. Fourth, the context of abnormal log data can be identified if MLM is performed using only normal log data. | The paper does a comparison study between the performance of LAnoBERT with four selected models - DeepLog, LogRobust, HitAnomaly and LogSy. These results indicate that the log anomaly detection performance is heavily dependent on a parser | Some of the limitations of the LAnoBERT model are proposed as future work. First, the proposed model recorded the performance for the HDFS and BGL datasets, which are typical datasets used in log anomaly detection. The performance of the proposed model should be evaluated through experiments using log data generated from different systems. Second, the proposed model uses the original architecture of BERT and requires an extensive amount of time to learn large-scale log data. Therefore, the training time should be shortened by effectively selecting the log data necessary for training, and a lighter model should be constructed to perform research on real-time anomaly detection. Third, the proposed model constructs a BERT model per dataset to perform anomaly detection. |
| TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data (2022) | Shreshth Tuli, Giuliano Casale, Nicholas R. Jennings Loughborough | * Numenta Anomaly Benchmark (NAB) * HexagonML (UCR) dataset * MIT-BIH Supraventricular Arrhythmia Database (MBA) * Soil Moisture Active Passive (SMAP) dataset * Mars Science Laboratory (MSL) dataset * Secure Water Treatment (SWaT) dataset * Water Distribution (WADI) dataset * Server Machine Dataset (SMD) * Multi-Source Distributed System (MSDS) Dataset | TranAD, a deep transformer network-based anomaly detection and diagnosis model which uses attentionbased sequence encoders to swiftly perform inference with the knowledge of the broader temporal trends in the data is proposed in this paper.  It uses focus score-based self-conditioning to enable robust multi-modal feature extraction and adversarial training to gain stability. Additionally, model-agnostic meta learning (MAML) allows us to train the model using limited data | The transformer based encoder-decoder allows quick model training and high detection performance for a variety of datasets considered in this work.TranADleveragesself-conditioning and adversarial training to amplify errors and gain training stability. Moreover, meta-learning allows it to be able to identify data trends even with limited data. Specifically, TranAD achieves an improvement of 17% and 11% for F1 score on complete and limited training data, respectively | The method with other transformer models like bidirectional neural networks to allow model generalization to diverse temporal trends in data. To explore the direction of applying cost-benefit analysis for each model component based on the deployment setting to avoid expensive computation. |
| AnomalyAdapters: Parameter-Efficient Multi-Anomaly Task Detection (2021) | UĞUR ÜNAL AND HASAN DAĞ, (Member, IEEE) | * Hadoop Distributed File Systems (HDFS) * The firewall dataset | Anomaly Adapters (AAs) which is an extensible multi-anomaly task detection model. It uses pretrained transformers’ variant to encode a log sequence and utilizes adapters to learn a log structure and anomaly types is proposed in this paper. Considering each log as a sentence and system-calls as a language; aim is to gain semantic information through adapters to distinguish anomalies. Using the nature of language models, we aim to use a multi-purpose approach, which is expandable to new sources without loss of information and overuse of parameters. | This paper compared this work with other recent studies in the field and also tested model decisions to get feedback in a readable form. Exploitability is a known issue for black-box models; thus, it also enables threat intelligence actively in the log semantic-based learning which opens a new direction for enhancing solution of anomaly detection problem | To focus on collaborating with algorithms in learning which interprets semantic-based anomaly detection models. By this way, create intelligible decisions may be created, which can be acted efficiently and timely. |
| Machine Learning based Anomaly Detection of Log Files using Ensemble Learning and Self-Attention(2020) | Markus Falt, Stefan Forsstrom, Tingting Zhang | * BGL * Thunderbird * Spirit | Nedelkoski, Bogatinovski, Acker, et al. presents a method called Logsy, for anomaly detection of log data.  Logsy uses additional log data sources to supplement the negative training samples, so that the model will learn the system’s normal behavior better. Inspired by Logsy,this article attempts to explore the possibility of creating generalized models for deviation detection of log data.  This article tries to see if generalized models trained on only additional log data sources can be successfully applied to unseen log data sources. | The work was compared to Logsy because it uses a similar method and was the inspiration for this work.  When training Logsy, data from the target system as well as data from auxiliary systems are used. This makes it possible for Logsy to learn the behavior of the target system well.  MWT Log does not use data from the target system for training, and instead relies completely on labeled log data from auxiliary systems.  MWT Log achieves a slightly higher f1-score for both the BGL and TBIRD data sets compared to Logsy.  MWT Log is also likely to be much slower with both training and predicting. The reason MWT Log is much slower is because of the ensemble learning technique that is used | The performance of the method was only tested on three labeled data sets, more testing is however required to better judge the method’s performance |
| TRANSLOG: A Unified Transformer-based Framework for Log Anomaly Detection(2021) | Hongcheng Guo1, Xingyu Lin4, Jian Yang1, Yi Zhuang2 | * HDFS * BGL * Thunderbird | Our model is first pretrained on the source domain to obtain shared semantic knowledge of log data. Then, we transfer the pretrained model to the target domain via adapter-based tuning.  The proposed method is evaluated on three public datasets including one source domain and two target domains | we conduct the ablation study in four aspects for a penetrating analysis of TRANSLOG, including the effect of the pretrained model, the gap between pretrained log models, the efficiency of adapter-based tuning, and the low resource study. | In this paper, we propose TRANSLOG, a unified transformer-based framework for log anomaly detection, which contains the pretraining stage and the adapter-based tuning stage. Extensive experiments demonstrate that our TRANSLOG, with fewer trainable parameters and lower training costs, outperforms all previous baselines. We foresee the semantic migration between log sources for a unified multiple sources detection. |
| IT Infrastructure Anomaly Detection and Failure Handling: A Systematic Literature Review Focusing on Datasets, Log Preprocessing, Machine and Deep learning approaches and Automated tool (2021) | Deepali Arun Bhanage, Ambika Vishal Pawar and Ketan Kotecha | * Hadoop distributed file system log dataset (HDFS) * Blue Gene/L supercomputer log (BGL) * Hardware replacement log (HPC) * Linux system log * Android framework log * Apache Server error log * Proxifier software log | The failure prediction pipeline in IT infrastructure to avoid failure conditions is divided into 4 parts:   1. Raw log data preprocessing and feature extraction 2. Training the model. 3. Investigate the performance of the trained model 4. Predictions   In the preprocessing phase, a log template is derived from the raw data using a log parsing tool (Drain Parser) from which a semantic vector sequence is derived. This semantic analysis is performed by using BERT tokenizer and KNN clustering algorithms. Relevant features are thus put forward to the deep learning models for training. In the testing phase, a balanced testing dataset with sufficient lead time of failure prediction will be provided to the model and an alert is generated to notify the system admin if there is a prediction of potential failures in the system. | The paper is systematic literature review (SLR) of previous research works that considered log data from each IT service sector, which are given to the proposed architecture. The results were more accurate (above 90% for most of the log data) than the existing literature. The limitations of the existing literature were system specific and cannot be used for various log data. Also, statistical methods were employed in the preprocessing phase rather than NLP methods, which extract the semantic features of the data. Machine and Deep learning models are implemented in the place of rule or method-based approaches. | The manual screening was conducted on available full-text articles to finalize the list of publications for detailed analysis; thus, it is not assured that all the articles from the literature are studied thoroughly. Quality data and historical data can be collected to improve the prediction accuracy and also to identify the pattern of failures to help minimize the data usage. User Interface (UI)-based monitoring consoles can be built to monitor the components in IT infrastructure better. This console can help to visualize the system and provide a detailed view of the overall system. An automated system can suggest corrective action on the identified anomalies and failure conditions. One can explore how these suggestions and techniques can be utilized and executed to avoid failures. |
| Sequential Anomaly Detection for Log Data Using Deep Learning (2021) | Lina Hammargren, Wei Wu | * HDFS dataset * Volvo Group Trucks Technology (Volvo GTT) | The raw log data is filtered out using regex to remove patterns of numerical values, non-alphanumeric characters and empty lines. After cleaning the log files, the log data is converted into integer sequence using sliding window technique to resolve the issue of long-term dependencies with sequence modelling and to decrease the length of the sequences to perform pattern recognition on. This sequence is one-hot encoded and given to the deep neural network models. A total of four neural network model architectures are applied; unidirectional LSTM, bi-directional LSTM, LSTM sequence-to-sequence & sequence-to-sequence transformer model. | Out of the anomaly detection approaches investigated in this project, the uni-LSTM model with a top-n threshold and the Transformer model trained on 1% of the training data perform the best on the Volvo GTT data and 100% of the normal training data perform the best on the HDFS data. The uni-LSTM method also gets the best result for the training on mixed data compares to other methods, but the F1-score is low indicating poor performance. Another observation is the bi-LSTM model performs worse than the uni-LSTM model. The issue is due to the creation of inputs for the bi-LSTM model. If the anomalies exist at the bottom of the log file, then the bi-LSTM model has no chance to indicate them as it reads from both ends. Moving on to the LSTM sequence-to-sequence model, the issue might be that we are not using a top-n threshold here to compute the n most likely log keys in the output. Comparing the results from the sequence-to-sequence LSTM model to the sequence-to-sequence Transformer model, we can see that the Transformer is better at learning the log sequences. This result is supported by the fact that Transformers perform better in neural machine translation tasks. | The preprocessing method regex can be replaced by log parsers such as Spell and Drain. Adding an attention mechanism to the LSTM sequence model would be interesting to see as that does not reconstruct the log line windows well. Other problem definition approaches to the sequence-to-sequence models could be considered, such as predicting more than one log line based on some context of past log lines. |