Developing an Automated Anomaly Detection System for Network Logs

LITERATURE SURVEY

submitted by

Dachepalli Leela Kesava Trinadh

21BAI1579

Nidumolu Sairam Gopal

21BRS1459

Rishi Patri

21BPS1396

of

B. Tech. Computer Science and Engineering

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

October 2024

S.No	Title	Methodology	Takeaways
1	Adanomaly: Adaptive Anomaly Detection for System Logs with Adversarial Learning	BiGAN Model: Uses Bidirectional GANs for feature extraction, enhancing detection accuracy. Ensemble Learning: Combines multiple classifiers to address class imbalance and reduce hyperparameter reliance. Log Parsing: Applies the Drain method to convert raw log data into accurate templates. Feature Extraction: Extracts features from log sequences with BiGAN and balances data for classifier training.	Adanomaly Framework: Introduces a novel log-based anomaly detection method using BiGAN and ensemble learning to improve accuracy and manage class imbalance. Feature Extraction: BiGAN derives features through reconstruction and discriminative losses, enhancing detection precision. Experimental Results: Shows superior recall and accuracy compared to six baseline methods across three public datasets.
2	Anomaly Detection on Servers Using Log Analysis	Log Parsing: Uses the Drain algorithm to structure raw log data. Feature Extraction: • Event Count/TF-IDF: Compiles counts and applies TF-IDF. • Sliding Window Counts: Creates matrices from event sequences. • Final Matrix: Merges sliding window matrices with TF-IDF vectors. Anomaly Detection Model: Implements a CNN model trained on labeled log data.	Deep Learning Model: CNN achieved up to 99% accuracy in anomaly detection. Log Parsing: Used the Drain algorithm for structuring raw log data. Feature Extraction: Employed TF-IDF and Sliding Window Event Counts. Experimental Results: High performance with low error rates and high F1-scores.
3	LogST: Log Semi-supervised Anomaly Detection Based on Sentence-BERT	Log Parsing: Converts raw logs into structured templates using the Drain method. Semantic Embedding: Uses Sentence-BERT (SBERT) for semantic representations of log events. Clustering: Applies HDBSCAN for clustering log sequences based on semantics. Anomaly Detection: Implements a GRU neural network with semi-supervised learning for anomaly detection.	LogST: Semi-supervised log anomaly detection using SBERT and GRU. Improved Accuracy: Outperforms traditional methods through semantic relationships. Stability with Few Labels: Effective with limited labeled normal logs. Experimental Validation: Shows significant improvements on the HDFS dataset.
4	Machine Learning to Detect Anomalies in Web Log Analysis	Two-Level ML Algorithm: Decision tree for classification and HMMs for anomaly detection. Feature Extraction: Extracts HTTP status codes, URL length, and parameter counts from logs. Data Labeling: Automatically labels logs to identify attacks versus normal behavior. Performance Evaluation: Uses accuracy, precision, FPR, and TPR, comparing to Logistic Regression and SVM.	Anomaly Detection System: Uses a two-level algorithm with a Decision Tree and HMM for web log detection. High Accuracy: Achieved 93.54% accuracy and 4.09% false positive rate, outperforming Logistic Regression and SVM. Real-World Data: Tested on data from a real industrial environment. Future Improvements: Plans to add a retraining module for adapting to new attack patterns.

5	Log-based Anomaly Detection Without Log Parsing	Data Splitting: 80% training and 20% testing, with unseen log messages in the test set. Sliding Window: 20-message length with a step size of 1 for log sequences. Comparison of Methods: NeuralLog compared to SVM, LR, IM, LogRobust, and Log2Vec. Evaluation Metrics: Uses Precision, Recall, and F1-score across datasets (HDFS, BGL, Thunderbird, Spirit).	NeuralLog Approach: Transforms raw logs into semantic vectors using BERT, bypassing parsing. Results: Achieves F1-scores over 0.95 on four datasets, outperforming existing methods. Contributions: Highlights limitations of log parsing and NeuralLog's effectiveness with OOV words.
6	A Study on Log Anomaly Detection using Deep Learning Techniques	Feature Extraction: Uses TF-IDF and Word2vec to convert log data into dense vectors. Machine Learning: Employs algorithms like SVM, Decision Tree, and PCA for anomaly detection. Deep Learning: Utilizes LSTM, RNN, Autoencoder, and Bi-LSTM for advanced anomaly detection.	Importance of Anomaly Detection: Essential for maintaining reliability and performance in large-scale networked systems. Deep Learning Techniques: Utilizes models like RNN, LSTM & autoencoders for effective log anomaly detection. Challenges: Addresses complications from unstructured data, log instability, and log bursts.
7	A Comprehensive Review of Anomaly Detection in Web Logs	Rule-based Models: Detect known anomalies but rely on administrator expertise. Statistical Models: Use Regex for anomaly detection based on query parameters. Supervised & Unsupervised ML Models: Supervised methods use labeled data; unsupervised methods employ clustering for unknown anomalies. Deep Hybrid Models (DHM): Combine log sequence encoding and machine learning for semantic anomaly detection.	Focus on Web Logs: Reviews techniques for detecting anomalies in HTTP logs. Categorization: Classifies methods into rule-based, statistical, supervised, unsupervised, and deep hybrid models. Challenges: Discusses issues like high-dimensional data and adaptive thresholds. Applications: Highlights use in cybersecurity, including IDS and FDS.
8			
9			
10			