EasyChair Preprint

№ 9842

GenGLAD: a Generated Graph Based Log Anomaly Detection Framework

Haolei Wang, Yong Chen, Chao Zhang, Jian Li, Chun Gan, Yinxian Zhang and Xiao Chen

March 7, 2023

GenGLAD: A Generated Graph Based Log Anomaly Detection Framework

Haolei Wang, Yong Chen, Chao Zhang, Jian Li, Chun Gan, Yinxian Zhang, and Xiao Chen(✉)

State Grid Zhoushan Electric Power Supply Company of Zhejiang Power Corporation, Zhoushan 316021, China
151491628@qq.com, 181494974@qq.com, 948451226@qq.com, 88018211@163.com, 66864790@qq.com, 970163182@qq.com, 18768070482@163.com

Abstract. Information systems record the current states and the access records in logs, so logs become the data basis for detecting anomalies of system security. To realize log anomaly detection, frameworks based on text, sequence, and graph are applied. However, the existing frameworks could not extract the complex associations in logs, which leads to low accuracy. To meet the requirements of the hyperautomation framework for log analysis, this paper proposes GenGLAD, a generated graph based log anomaly detection framework. The generated graph is used to express the log associations, and the node embedding of the generated graph is obtained based on random walk and word2vec. Finally, we use clustering to realize unsupervised anomaly detection. Experiments verify the detection effect of GenGLAD. Compared with the existing detection frameworks, GenGLAD achieves the highest accuracy and improves the comprehensive detection effect.

Keywords: Log anomaly detection · Graph learning · Hyperautomation.

1 Introduction

Information technology [14, 31] and computer capability [30, 36, 37] promoted the machine learning [11, 34, 39] and intelligent development [26, 27] of various industries. Many enterprises and institutions rely on open environments to carry out business. To deal with the automatic and diversified network attacks [24, 28, 35] in open environments, a series of network security devices and systems are deployed on the network boundary to ensure the security of the business system within the boundary. The security system [6, 8, 15] prompts the administrator for network attacks, SQL injection, and other abnormal or malicious behaviors in the network environment through access logs and accompanying alarm labels.

With the proposal of the hyperautomation framework [2], automatic analysis of logs has become a major demand in the industry. However, traditional log analysis frameworks rely on manual analysis. Through simple statistics of the log content, the key information such as the source IP or source user and the

attack duration in the log is extracted, and the authenticity of each logline is determined by combining the prior knowledge such as the sensitivity of the source IP.

These frameworks face a series of problems in real-world scenes, including 1) the volume of logs is huge [5, 29, 46]. There are many kinds of logs in the existing system, but most kinds of log files contain a large number of lines [43], which exceeds the capacity limit of manual analysis; 2) Most of the logs are low-level and useless [22], which are not triggered by real malicious behaviors. Malicious behaviors [32,33] are hidden in a large number of messy logs and invalid alarms, resulting in great security vulnerabilities [7,9]. Therefore, how to detect anomaly logs that represent malicious behaviors is the key to maintaining network security.

To meet the need for hyperautomation, a series of representation learning technologies based on machine learning or deep learning, such as TCN [1] and LSTM [17], have been proposed and applied to the detection of various logs [3,41] or optimization for security systems [38]. On this basis, the detection methods based on graph models such as log2vec [18] are proposed. The complex associations in original logs are characterized by the graph structure. After obtaining appropriate representations of loglines or network entities, the detection of anomaly logs could be realized through a relatively simple classification method.

Therefore, we propose GenGLAD, a log anomaly detection framework based on the generated graph. We use the generated graph model to characterize the original log associations, optimizing the existing generated graph construction method. The initial attribute of the generated graph node is determined by setting the key features of the log, and the detection of anomaly logs is realized based on clustering. Through experiments on public simulation datasets, the availability and high accuracy of GenGLAD are proved.

The main contributions of this paper are as follows:

- The construction method of the generated graph is optimized, and the number of edges is reduced. as a result, the speed of model training is improved.
- GenGLAD, a novel generated graph based anomaly detection framework for logs is proposed, which can effectively detect the anomaly logs out of a large number of logs.
- The detection effect of GenGLAD exceeds that of the popularly used methods.

The rest of this paper is organized as follows: Sections 2 introduce the related work. Section 3 describes the framework of GenGLAD, while Section 4 shows the experiments. Finally, Section 5 summarizes the work and discusses future work.

2 Related Works

Our research belongs to the field of log anomaly detection. The existing methods could be divided into three categories: text-based methods, sequence-based methods, and graph-based methods.

2.1 Text Based Methods

Logs are semi-structured text data. Therefore, researchers have realized the anomaly detection of logs by migrating methods in *Natural Language Processing* (NLP), word embedding, and other fields. The most typical idea of the text-based methods is to analyze the keywords contained in the log and the related word frequency of the recorded access behaviors [12], considering the significantly different text features as anomalies. However, such methods could only directly use text features, lacking the ability to characterize the high-level features and deep associations contained in logs, so the detection accuracy is not as high as that of other kinds of methods.

2.2 Sequence Based Methods

Logs are real-time records of systems, so it is naturally a kind of time-series data. Deeplog [3] regards the normal logs as a sequence with a certain pattern, and learns the normal log sequence based on LSTM, to analyze the abnormal possibility when a new logline is recorded. On this basis, the technology migration of the GRU classifier and full connection layer further improves the detection index [42].

Transformer framework, which shows unparalleled sequence learning ability has also been applied in the field of log detection [44], and realizes feasible log detection. In addition, the generative adversarial network is also directly applied to the log detection scenario [4,13], and the method migration of attention mechanism realizes the effective detection of anomalies in logs. However, sequence based methods could only analyze the associations between loglines from the perspective of time series, and could not extract and analyze the complex associations between days, resulting in a decrease in accuracy.

2.3 Graph Based Methods

Thanks to the representation ability of graph structure, graph based frameworks are used to model original logs. Graph anomaly detection algorithms are migrated into the field of log anomaly detection. Log2vec [18] defines the node construction rules and edge link rules of the log generation graph, detecting anomalies with high accuracy through the direct definition of log associations. The use of a heterogeneous graph further improves the graph embedding effect of the algorithm.

On the other hand, the provenance graph based methods derived from network attack detection are also migrated into the field of log anomaly detection [10]. This kind of method constructs a provenance graph that completely describes the behaviors of each IP or user, and analyzes the behaviors based on the critical paths in the graph. The graph based methods could extract the complex association between logs, and effectively improve the accuracy of existing detection methods. However, existing graph construction methods would lead to an excessive number of nodes and edges. The accuracy of the relevant graph anomaly detection methods also needs to be further improved.

3 GenGLAD: Detection Framework

GenGLAD includes two main steps, as shown in figure 1. The first step is to construct a generated graph based on raw logs, and the second step is to detect anomaly nodes based on the generated graph, to detect the anomaly logs.

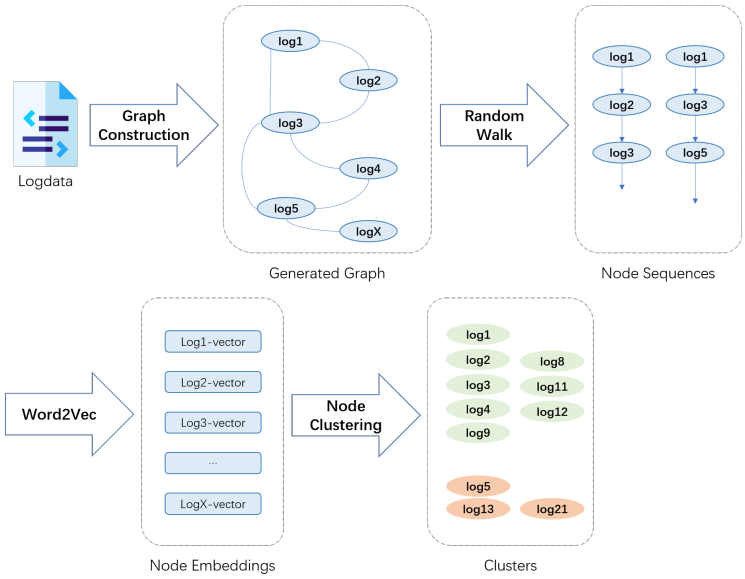

Fig. 1. The framework of GenGLAD. There are two main steps: 1) Construct the generated graph based on original logs; 2) Perform node anomaly detection, including node sequences calculation by random walk, embedding vectors calculated by word2vec, and unsupervised anomaly detection based on clustering.

3.1 Generated Grapg Construction

We use each node in the graph to represent a logline. Therefore, the edges between nodes represent the association between loglines. Through the construction method, we characterize the associations between logs on the graph.

Nodes Each node corresponds to a logline raw logs, so if a node is detected as an anomaly node, its corresponding logline is an anomaly log. Based on this node definition, we can assign the attributes and label to each node.

The attributes of each node are the string of the corresponding logline, that is, the feathers of the access behavior recorded in the log. The most important attributes include 1) Source entity of the behavior, such as an IP address or user; 2) Destination entity of the behavior; 3) Type of the behavior; 4) Time when the behavior occurred.

The labels include normal and abnormal. GenGLAD is an unsupervised detection framework, so the initial tags are set to normal.

Edges In anomaly detection scenarios, if behavior is more closely related to a known anomaly behavior, it is more likely to be abnormal [40]. Therefore, the edges in the graph should express the associations between loglines corresponding to the nodes. The existing method defines 10 connection rules between nodes to achieve this [18].

However, the existing rules are relatively complex and are not suitable for the single log style in real-world scenes. Most anomaly behaviors in real-world scenes represent potential attacks, so the destination is the core parameter. In addition, anomaly behaviors usually show concentration in time dimension [25].

Therefore, we improve the connection rules so that they could be applied to the anomaly detection tasks of most kinds of logs.

The connection rules we define are as follows. To make the expression more concise, the key information of the log line corresponding to a node is called the information of the node, such as time and source entity.

- *Rule1: Connections within one day.* Within the same day, all nodes are connected in chronological order, and nodes with the same source entity or action type are connected.
- *Rule2: Connections between days.* Daily node sequences are connected in chronological order, and sequences with at least one same source entity or action type are connected.
- *Rule3: Connections based on destination.* Nodes with the same destination entity are connected.

The simplified rules strengthen the portability of the method, and reduce the number of edges in the generated graph. Thereby reducing the time consumption of model training.

3.2 Node Anomaly Detection

Labels of unknown nodes could be obtained through the node anomaly detection framework on the generated graph, to infer the anomaly logs existing in the original data. We use the random walk algorithm on the graph to obtain the node sequence [45], and migrate the word2vec algorithm from the NLP domain to realize graph embedding [19, 20]. Finally, we can obtain the abnormal node set by clustering the embedding vectors.

Random Walk Most of the existing random walk methods [45] could be applied to heterogeneous graphs, heterogeneous information networks, and knowledge graphs. The generated graph is a static isomorphic graph, so the transition probability of random walk needs to be adjusted. Specifically, we adjust the transfer probability to:

$$P(t|v) = \begin{cases} \frac{1}{\boldsymbol{N}(v)}, (t,v) \in E \\ 0, \quad otherwise \end{cases} \tag{1}$$

where $\boldsymbol{N}(v)$ denotes specific neighbor nodes of node v.

It is proven that the random walk sequence generated by focusing on only one kind of association could achieve the best effect in generating graph anomaly detection, the best practice walk sequence length is also provided [18].

Embedding based on Word2vec The word2vec algorithm, which is migrated into the graph learning field, is a coding method that embeds nodes into vectors and makes the embedding vectors obtained by nodes with similar attributes as close as possible [21]. It aims to maximize the probability of the neighbors conditioned on a node. For node n_v, in node list $n_{v-c}, \ldots, n_{v+c}$, The objective function to be maximized is:

$$\sum_{v=1}^{V} logP(n_{v-c}, \ldots, n_{v+c}|n_v) \tag{2}$$

We regard logging as independent and identically distributed events, so the probability in formula 2 could be converted into the product of a series of probabilities. In addition, softmax function is used to define function P. Therefore, the objective function could be calculated as:

$$\frac{e^{V_{n_v}^T V'_{n_{v+j}}}}{\sum_{i=1}^{V} e^{V_{n_v}^T V'_{n_i}}} \tag{3}$$

where V_{n_i} represents the input vector of node n_i, and V'_{n_i} represents the output vector of node n_i. In the process of i increasing from 1 to V, formula 3 calculates the embedding results of all nodes.

Anomaly Detection Method Graph embedding based on random walk and word2vec makes the distribution of embedding vectors easy to distinguish. Therefore, the unsupervised clustering method could be used to detect anomaly nodes. Based on the satisfactory embedding results, we adopt a simple distance based clustering method. Specifically, we add all nodes to the initial set N_0, and let sets $N_1, N_2, \ldots$ be empty. Then, check whether each node meets condition 4 and condition 5.

$$\forall n_1 \in N_i, \forall n_2 \in N - N_i, dis(n_1, n_2) \geq d_0 \tag{4}$$

$$\forall n_1 \in N_i, \exists n_2 \in N_i \ s.t. \ dis(n_1, n_2) \leq d_0 \tag{5}$$

Where N represents the set of all nodes, $dis()$ is the distance between embedding vectors of two nodes, and d_0 is a distance threshold. If a node does not meet the requirements of the conditions, move the node to an existing or new set to make it meet the requirements. Condition 4 makes the distance between nodes in different clusters relatively far, and condition 5 makes nodes in the same cluster have close embedding vectors so that the nodes in each cluster would have the same label.

4 Experiments and Results

4.1 Experimental Setup

Datasets To verify the detection effect of GenGLAD, we use CERT [16], an open synthetic dataset for testing. CERT contains many different log files, describing more than 100 million behaviors of 4000 users. It covers the logs of device interaction, e-mail, file system, and so on. It covers the logs of device interaction, e-mail, file system, and other aspects. We select the device login and logout logs in version r4.2 to simulate the logs with insufficient data and features in real-world scenes. Specifically, we selected device interaction logs, recording the users' login actions on PCs for a span of 45 days. The main fields include device ID, user ID, action type, and time.

Baselines We adopted three representative log anomaly detection methods as baselines: one-class *Support Vector Machine* (SVM), *Gaussian Mixture Model* (GMM), and Deeplog.

- *SVM [23].* SVM trains a non-probabilistic binary linear classifier, which represents the logs as points in the space, and obtains the classification plane through the training process. Then SVM maps a new log to the same space and predicts its label based on which side of the classification plane it falls on.
- *GMM [47].* GMM is one of the most widely used statistical methods. It uses maximum likelihood estimation to estimate the mean and variance of Gaussian distribution. Several Gaussian distributions are combined to represent the feature vectors and find out the anomalies.
- *Deeplog [3].* Deeplog regards the logs as a sequence, calculates the type of each log through an analysis algorithm, and determines whether the newly generated log is an anomaly log based on a sequence learning framework, such as LSTM.

Parameters Selection We use 40,000 device interaction logs, containing 1,154 anomaly ones. In the process of random walk, we choose a walk length of 60 and only focus on the edges generated by the same rule in each walking path. During node embedding, the dimension is set as 100 and the window length is 10. We set the threshold based on the average distance in the process of clustering.

Metrics We use common metrics in the field of anomaly detection to measure the detection effect, including: $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, $recall = \frac{TP}{TP+FN}$, $precision = \frac{TP}{TP+FP}$. And F1 score for comprehensive evaluation.

$$F1 = 2 \cdot \frac{recall \cdot precision}{recall + precision} \tag{6}$$

Where TP represents true positive, FP represents false positive, TN represents true negative, and FN represents false negative. We also use the *Receiver Operating Characteristic* (ROC) curve and *Area Under the Curve* (AUC) to evaluate the detection effect of GenGLAD. The closer the AUC value is to 1, the better the detection effect is.

4.2 Results

Figure 2 shows the ROC curve of GenGLAD.

It is shown that the AUC value exceeds 0.948, indicating that GenGLAD has obtained effective detection results. However, in the anomaly detection scenario,

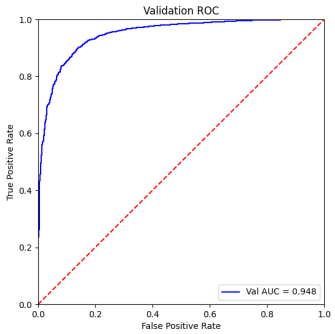

Fig. 2. ROC curve of the detection result of GenGLAD with AUC 0.948.

the sample proportion of datasets is not balanced, and the detection effect could not be comprehensively evaluated only by the ROC curve. Therefore, it is necessary to compare various metrics in detail. Table 1 shows the detection results of GenGLAD and baselines on the CERT dataset.

Table 1. Detection Results

Framework	Accuracy	Recall	Precision	F1
GenGLAD	**0.9273**	**0.6646**	0.7973	**0.7249**
SVM	0.6032	0.4474	**0.8095**	0.5763
GMM	0.5780	0.4168	0.1109	0.1752
Deeplog	0.9039	0.6310	0.6742	0.6519

It is shown that GenGLAD obtains the highest F1 score, which indicates that it has the best comprehensive detection effect. GenGLAD is superior to all baselines in accuracy and recall. However, for precision, SVM has better performance. This might be due to the more strict classification of SVM as a linear classifier, which makes it get a low false positive rate with low accuracy. As shown in table 1, the accuracy of GMM is the lowest among all methods.

It is also shown that both GenGLAD and Deeplog have achieved relatively high accuracy, which means that deep learning helps to improve the effect of log anomaly detection. Among the baseline methods, GMM could only predict the distribution of normal logs, and SVM could only provide a linear classification surface, so these two methods could not achieve high detection accuracy. On the other hand, Deeplog regards logs as time series data, which could capture the correlation in the time dimension. In contrast, GenGLAD captures more correlations between logs, so it achieves the highest accuracy rate, although this results in higher time complexity.

5 Conclusion

In this paper, we proposed GenGLAD, a novel framework for log anomaly detection. To realize the automatic analysis and detection of logs, we first constructed a graph based on logs, then performed a random walk on the generated graph, using word2vec to obtain the node embedding vector, and finally detected anomaly logs based on clustering. We realized the automatic processing of logs and made GenGLAD could be integrated into a hyperautomation system. Through experiments, we proved that the detection effect of GenGLAD is better than the popularly used frameworks. The accuracy reached 0.927, and the AUC value was 0.948. Compared with Deeplog which is widely used, GenGLAD achieved an improvement of about 11% in F1 score. In further research, we plan to adjust relevant parameters to further improve the detection effect. In addition, we consider using deep neural network based methods like GCN to embed nodes in an attempt to obtain better embedding vectors, so that the clustering results could accurately reflect the distribution of anomaly nodes.

Acknowledgements. This work was supported by State Grid Zhoushan Electric Power Supply Company of Zhejiang Power Corporation under grant No. B311ZS220002 (Research on hyperautomation for information comprehensive inspection).

References

1. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. CoRR **abs/1803.01271** (2018)
2. Bornet, P., Barkin, I., Wirtz, J.: Intelligent Automation: Welcome to the World of Hyperautomation - Learn How to Harness Artificial Intelligence to Boost Business & Make Our World More Human. WorldScientific (2021)
3. Du, M., Li, F., Zheng, G., Srikumar, V.: Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In: ACM SIGSAC Conf. on Comp. and Comm. Security. pp. 1285–1298 (2017)
4. Duan, X., Ying, S., Yuan, W., Cheng, H., Yin, X.: A generative adversarial networks for log anomaly detection. Comput. Syst. Sci. Eng. **37**(1), 135–148 (2021)
5. Gai, K., Du, Z., et al.: Efficiency-aware workload optimizations of heterogeneous cloud computing for capacity planning in financial industry. In: IEEE 2nd CSCloud (2015)
6. Gai, K., Qiu, M., Elnagdy, S.: A novel secure big data cyber incident analytics framework for cloud-based cybersecurity insurance. In: IEEE BigDataSecurity conf. (2016)
7. Gai, K., Zhang, Y., et al.: Blockchain-enabled service optimizations in supply chain digital twin. IEEE TSC (2022)
8. Gai, K., et al.: Electronic health record error prevention approach using ontology in big data. In: IEEE 17th HPCC (2015)
9. Gao, X., Qiu, M.: Energy-based learning for preventing backdoor attack. In: KSEM (3). pp. 706–721 (2022)

10. Han, X., Pasquier, T.F.J., Bates, A., Mickens, J., Seltzer, M.I.: Unicorn: Runtime provenance-based detector for advanced persistent threats. In: 27th Network and Dist. System Security Symp., NDSS 2020 (2020)
11. Hu, F., Lakdawala, S., et al.: Low-power, intelligent sensor hardware interface for medical data preprocessing. IEEE Trans. on Infor. Tech. in Biome. **13**(4), 656–663 (2009)
12. Kent, A.: Cyber security data sources for dynamic network research. Dynamic Networks and Cyber-Security pp. 37–65 (05 2016)
13. Kulyadi, S.P., Mohandas, P., et al.: Anomaly detection using generative adversarial networks on firewall log message data. In: 13th IEEE Conf. on Electro., Compu. and Arti. Intell. ECAI. pp. 1–6 (2021)
14. Li, J., Ming, Z., et al.: Resource allocation robustness in multi-core embedded systems with inaccurate information. J. of Sys. Arch. **57**(9), 840–849 (2011)
15. Li, Y., Gai, K., et al.: Intercrossed access controls for secure financial services on multimedia big data in cloud systems. ACM TMCCA (2016)
16. Lindauer, B.: Insider threat test dataset (2020), https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247
17. Lindemann, B., Maschler, B., Sahlab, N., Weyrich, M.: A survey on anomaly detection for technical systems using LSTM networks. Comput. Ind. **131**, 103498 (2021)
18. Liu, F., Wen, Y., et al.: Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise. In: ACM SIGSAC Conf. on Comp. and Comm. Security. pp. 1777–1794 (2019)
19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings (2013)
20. Mikolov, T., Sutskever, I., et al.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Infor. Proc. Sys. vol. 26. Curran Associates, Inc. (2013)
21. Moon, G.E., Newman-Griffis, D., et al.: Parallel data-local training for optimizing word2vec embeddings for word and graph embeddings. In: IEEE/ACM Workshop on Machine Lea. in High Perf. Computing Env., MLHPC@SC, 2019. pp. 44–55 (2019)
22. Nehinbe, D.J.: A review of technical issues on ids and alerts. Global Journal of Computer Science and Technology (2018)
23. Nguyen, T., Liao, T., Vu, T.: Anomaly detection using one-class SVM for logs of juniper router devices. In: Indu. Net. and Intel. Sys. - 5th EAI Int"l Conference, INISCOM 2019. vol. 293, pp. 302–312. Springer (2019)
24. Niu, J., Gao, Y., et al.: Selecting proper wireless network interfaces for user experience enhancement with guaranteed probability. JPDC **72**(12), 1565–1575 (2012)
25. Pawlicki, M., Kozik, R., Choras, M.: A survey on neural networks for (cyber-) security and (cyber-) security of neural networks. Neurocomputing **500**, 1075–1087 (2022)
26. Qiu, H., Dong, T., et al.: Adversarial attacks against network intrusion detection in IoT systems. IEEE IoT J. **8**(13), 10327–10335 (2020)
27. Qiu, H., Zheng, Q., et al.: Topological graph convolutional network-based urban traffic flow and density prediction. IEEE Trans. on ITS (2020)
28. Qiu, M., Chen, Z., et al.: Energy-aware data allocation with hybrid memory for mobile cloud systems. IEEE Sys. J. **11**(2), 813–822 (2014)

29. Qiu, M., Gai, K., Xiong, Z.: Privacy-preserving wireless communications using bipartite matching in social big data. FGCS **87**, 772–781 (2018)
30. Qiu, M., Jia, Z., et al.: Voltage assignment with guaranteed probability satisfying timing constraint for real-time multiproceesor DSP. J. of Signal Proc. Sys. (2007)
31. Qiu, M., Li, H., Sha, E.: Heterogeneous real-time embedded software optimization considering hardware platform. In: ACM sym. on Applied Comp. pp. 1637–1641 (2009)
32. Qiu, M., Qiu, H.: Review on image processing based adversarial example defenses in computer vision. In: IEEE 6th Intl Conf. BigDataSecurity. pp. 94–99 (2020)
33. Qiu, M., Qiu, H., et al.: Secure data sharing through untrusted clouds with blockchain-enabled key management. In: 3rd SmartBlock conf. pp. 11–16 (2020)
34. Qiu, M., Sha, E., et al.: Energy minimization with loop fusion and multi-functional-unit scheduling for multidimensional DSP. JPDC **68**(4), 443–455 (2008)
35. Qiu, M., Xue, C., Shao, Z., et al.: Efficient algorithm of energy minimization for heterogeneous wireless sensor network. In: IEEE EUC Conf. pp. 25–34 (2006)
36. Qiu, M., Xue, C., et al.: Energy minimization with soft real-time and DVS for uniprocessor and multiprocessor embedded systems. In: IEEE DATE Conf. pp. 1–6 (2007)
37. Qiu, M., Yang, L., Shao, Z., Sha, E.: Dynamic and leakage energy minimization with soft real-time loop scheduling and voltage assignment. IEEE TVLSI **18**(3), 501–504 (2009)
38. Qiu, M., Zhang, L., Ming, Z., Chen, Z., Qin, X., Yang, L.T.: Security-aware optimization for ubiquitous computing systems with SEAT graph approach. J. Comput. Syst. Sci. **79**(5), 518–529 (2013)
39. Shao, Z., Wang, M., et al.: Real-time dynamic voltage loop scheduling for multi-core embedded systems. IEEE Trans. on Circuits and Systems II **54**(5), 445–449 (2007)
40. Wang, S., Balarezo, J.F., Kandeepan, S., Al-Hourani, A., Chavez, K.G., Rubinstein, B.: Machine learning in network anomaly detection: A survey. IEEE Access **9**, 152379–152396 (2021)
41. Wang, Z., Tian, J., Fang, H., Chen, L., Qin, J.: Lightlog: A lightweight temporal convolutional network for log anomaly detection on the edge. Comput. Networks **203**, 108616 (2022)
42. Xie, Y., Ji, L., Cheng, X.: An attention-based GRU network for anomaly detection from system logs. IEICE Trans. Inf. Syst. **103-D**(8), 1916–1919 (2020)
43. Zeng, L., Xiao, Y., Chen, H., Sun, B., Han, W.: Computer operating system logging and security issues: a survey. Secur. Commun. Networks **9**(17), 4804–4821 (2016)
44. Zhang, C., Wang, X., Zhang, H., Zhang, H., Han, P.: Log sequence anomaly detection based on local information extraction and globally sparse transformer model. IEEE Trans. Netw. Serv. Manag. **18**(4), 4119–4133 (2021)
45. Zhang, H., Duan, D., Zhang, Q.: Rwrel: A fast training framework for random walk-based knowledge graph embedding. In: ACAI 2021: 4th International Conference on Algorithms, Computing and Artificial Intelligence. pp. 67:1–67:6. ACM (2021)
46. Zhang, L., Qiu, M., Tseng, W., Sha, E.: Variable partitioning and scheduling for mpsoc with virtually shared scratch pad memory. J. of Signal Proc. Sys. **58**(2), 247–265 (2018)
47. Zhou, F., Qu, H.: A gmm-based anomaly IP detection model from security logs. In: Smart Computing and Communication - 5th International Conference, SmartCom 2020. Lecture Notes in Computer Science, vol. 12608, pp. 97–105. Springer