2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
Amity University, Noida, India. June 4-5, 2020

A Survey on Log Anomaly Detection using Deep Learning

Rakesh Bahadur Yadav
Department of Computer Science and Engineering
Defence Institute of Advanced Technology
Pune , India
rakesh0510@gmail.com

P Santosh Kumar
Department of Computer Science and Engineering
Defence Institute of Advanced Technology
Pune , India
p.santoshk@gmail.com

Sunita Vikrant Dhavale
Department of Computer Science and Engineering
Defence Institute of Advanced Technology
Pune , India
sunitadhavale@diat.ac.in

***Abstract*—Logs generated from the security systems, network devices, servers, and various software applications are one of the ways to record the operational happening of the equipment or software. These logs are assets for extracting meaningful information related to system behavior. Increasing usage of computer devices and the evolution of software systems can be considered as one of triggering acts for the concentration on the analysis of logs. Also, considering the massive volume of unstructured data, it raises the requirement for automatic analysis of these logs. The log analysis is helpful for understanding system behavior, malfunctioning detection, security scanning, and failure prediction. Machine learning(ML) and Deep Learning (DL) methods have been proved potent tools for data classification problems and have been applied to various fields of research. The purpose of this survey is to review recent research on log anomaly detection using Deep Neural Networks. Survey also presents the brief of log parsing approaches, types of datasets used for log analysis, and various concepts proposed for Log Anomaly detection.**

Keywords—Log Anomaly, Deep Learning, LSTM, CNN, Autoencoder, Log Parsing

I. INTRODUCTION

Anomaly detection in system logs has become critical for large enterprises as the systems and applications are getting more sophisticated and generating large event logs. The security systems are subject to more bugs and vulnerability, which may be exploited to launch an attack by the attacker. Researchers are working for efficient analysis of log data and exploring the possibility of timely identification of threats in logs before its getting activated. The log anomaly detection based on the traditional method is no longer useful since the attacks are becoming more sophisticated. So far, data mining and machine learning approaches such as Decision Tree (DT), Support Vector Machine (SVM), and Principal Component Analysis (PCA) have been used for extracting more relevant features. These approaches give better accuracy at the same time, reduce complexity as well. However, analyzing the concealed relationships in extracted features is still tricky by these approaches. More sophisticated methods like deep learning approach overcome this limitation. In the last few years, log anomaly detection using deep learning approaches with NLP techniques have achieved better accuracy by harnessing semantic relationships in logs. Mengying Wang et al. [1], Min Du et al. [2], Wang et al. [3], Meng et al. [4] have achieved higher accuracy in log anomaly detection by using Long Short Term Memory (LSTM) for anomaly detection. Siyang et al. [5] have used the Convolution Neural Network (CNN) based deep learning model for achieving an accuracy of 99 percent. Amir et al. [6] have used autoencoder [7] for feature extraction and further DL models for the identification of anomaly. Zhang et al. [8] and Brown et al. [9] have used attention mechanisms with deep learning models to give more consideration for a particular sequence of data. The lack of a detailed study for the amount of research work carried out on log datasets and deep learning approaches applied to these datasets for log anomaly detection is decelerating further research in this area. To the best of our knowledge, there is no paper available which studies the various explored methods for log anomaly detection using DL. In this paper, we survey and present a comparative analysis of various available datasets, log parsing methods, and deep learning models used for log anomaly detection.

II. BACKGROUND

A. Feature Extraction

1) TF-IDF: Term frequency-inverse document frequency (TF-IDF) [10] is a extensively used method for feature extraction. It is a metric that reflects how important the word is to a document in the corpus. TF-IDF gives little importance to words that are very common to the entire corpus e.g., “a”, “the”, and “of”, and provide more importance to words that occur more frequently in a particular document but not very common in the entire corpus.

2) Word Embedding: Mikolov et al. [11] have introduced the word2vec model to compute and generate a dense vector for word representation, which captures the semantics of words. Word2vec presented two novel models for the transformation of a word to word embedding. The first

978-1-7281-7016-9/20/$31.00 ©2020 IEEE

model, Continuous Bag Of Words(CBOW) predicts the central word with reference to surrounding words within the provided window size. The second model, skip-gram, predicts the surrounding words with reference to the central word. The foremost advantage of the word embedding is that the different words which are used similar context will remain in proximity in the vector space. Furthermore, we can apply basic algebraic operations on word embeddings to get interpretable results. For example, the difference of vectors between "man" and "woman" is similar to the difference of vectors between "king" and "queen". Meaning that if we add the vectors of "king" and "woman" and subtract vector of "man" from that, then we will get the vector of "queen". The word embeddings have demonstrated to be very useful in many NLP applications, such as sentiment analysis, parsing, and anomaly detection.

B. Machine Learning

1) Decision Tree: Decision Trees (DT) is a useful supervised learning technique in many areas, such as data mining, information extraction, and machine learning for classification and prediction. A DT can be depicted in a simple treelike graphical representation, and the outcome decision can be explained easily. A leaf node specifies the class of the instances. The instances are classified by sorting them down the tree from the root to some leaf node.

2) Support Vector Machine: In Support Vector Machine (SVM), different classes of instances are separated by drawing a hyperplane in high-dimension space. When there is no apparent classifier between the classes, then SVM works by moving data into a relatively high dimension space where the hyperplane can classify observation. For the transformation of data from low dimensional to higher dimensional space SVM use kernel functions which systematically find support vector classifier in high dimensional space.

3) PCA: PCA is a widely used method for dimension reduction. PCA enables the transformation of high dimensional data into a low dimensional data without losing important information. The basic idea of PCA is to remove redundant data and highly correlated features while retaining significant features.

C. Deep Learning

1) RNN: Recurrent Neural Network (RNN) is an artificial neural network that can capture sequential or temporal information. It has memory to store previous output, which is further used as input like a loop for making predictions. The RNNs can look back only a few steps as it has limited memory.

2) LSTM: Long Short-Term Memory (LSTM) [12] networks are a type of RNNs that can look back on long-term temporal dependencies over sequences. LSTMs have shown promising results in various Natural Language Processing (NLP) tasks such as sentiment analysis, text classification, and log anomaly detection.

3) Bi-LSTM: Bidirectional LSTM (Bi-LSTM) is an expansion of LSTM that divides the hidden layer of neurons of a regular LSTM into two opposite directions, i.e., forward and backward. Bi-LSTM can capture the knowledge of log sequence from both of the input directions.

4) Autoencoder: An autoencoder [7] is an unsupervised neural network technique. It efficiently compresses and encodes the data and, again, from encoded data, reconstructs the output, which is close to the original input. Autoencoders are used for dimensional reduction similar to PCA; moreover, it can also learn non-linear transformation with a non-linear activation function.

III. CHALLENGES IN LOG ANALYSIS

Logs are generated from the collection of an ordered sequence of statements captured as a piece of evidence. The execution of a set of instructions performs a task. The sources of logs are spread across every entity of Information technology infrastructure, i.e., network devices, security devices, servers, storage, etc. This diversified nature of log creates many challenges for their processing.

A. Unstructured data

The logs are mainly in an unstructured or semi-structured format, which varies for different devices, Operating system, Software version, OEM. There is no defined formal structure and syntax for log files. Centralized collection of all this data and processing encounters another big challenge.

B. Instability

Xu Zhang et al. [8] acknowledged the problem of log instability. They identified a few reasons for it, such as the evolution of logging statements by modification of source code and processing of noise in log data. There is no fixed rule for a set of a distinct set of logs or several logs to be generated for any task.

C. Log burst

The volume of log data is increasing many folds due to the increasing sense of security and intelligence in devices and software. The centralized collection of all logs for storage, correlation, and processing are making this a big data problem. As a result, the generated log burst is another problem for log analysis and extracting of information from raw log data.

D. Availability of public dataset

The logs are unstructured most of the time, and its contents are susceptible, hence considering security concerns, it cannot be disclosed publicly for everyone. This nature limits the possibility of the availability of public datasets for research works.

The above discussion also supports the fact that it is infeasible for humans to comprehend and perform anomaly detection from log data.

IV. LOG ANOMALY DATASETS

The type of log data source profoundly influences the procedure for log anomaly detection. Log data source gathered from independent Devices software or servers, supercomputer logs, distributed system logs, Security Information and Event Management (SIEM) systems collect the variety of logs from the different systems in one place. Jieming Zhu et al. [13] worked with sixteen log datasets of different categories of systems and assessed the performance of 13 log parsers. The datasets of logs can be categorized based on the source of logs and volume of log entries as supercomputer logs, distributed system logs, and software system logs etc. Summary of log datasets and their brief feature are illustrated in Table -I

High Performance Cluster (HPC) logs are from the supercomputer setup of 49 nodes at Los Alamos National Laboratory and each node configured with 6,152 cores and 128GB of memory.

Amir Farzad et al. [6] Uses four Datasets, i.e., BGL, Thunderbird, openstack, and IMDB [22] for experiments. They used the IMDB dataset to prove the generalization of their proposed model for the text classification activities. The IMDB data set consists of 50,000 movie review sentences, with equal numbers that are positive and negative.

Xu Zhang et al. [8] proposed a model to overcome the issue of instability in log data For log anomaly detection. They worked with the HDFS dataset and also created an unstable testing dataset using the original HDFS dataset by inserting some unstable events and sequences of logs. They also use Service X Dataset from Microsoft, which is a realworld industrial log data.

Xiaojuan Wanget et al. [3] work is based on one-month log data of 1.09 GB with 18727517 log entries obtained from the router device of NetEngine40E series installed in the real network.

V. METHODOLOGY

Candace Suh-Lee et al. [23] proposed the idea to extend techniques of text processing and Natural Language Processing (NLP) for unstructured log data. The work identifies some properties of log data concerning text data as

- log entries repeat large portion messages
- less number of Natural language words used
- vocabulary size for log data is larges as it contains error codes, status codes, numbers in different formats.

Hence, the text processing techniques may be useful only with specific processing steps and methods.Preprocessing and manual labeling are always a point of discussion and challenge for researchers because it is a costly activity for voluminous log data. Preprocessing of log data, feature extraction, and applying techniques on processed data are the core steps involved for anomaly detection from a massive volume of log data. Understating the log format and applying domain knowledge is always a valuable addition to anomaly detection activity, which improves accuracy.

Log Parsing: Text log message can be viewed in two parts, i.e., a fixed portion of the message and changing sections. The fixed part text remains the same in messages while changing parts contains different value as per parameter. This observation leads to a concept of event and template for log messages. Shilin He et al. [24] worked with system log analysis and some log parsing tools. The parsing methods categorization is as clustering-based (e.g., LKE [25], LogSig [26]) and heuristic-based (e.g., iPLoM [27], SLCT [28]). Pinjia et al. [17] propose a parallel log parser(POP) and compare the accuracy of this with other log parsers. Log parsing gives structured data from unstructured log data.

Next, to get the structured data with preprocessing, the data should be processed as per the required of the model we are going to use, i.e., Machine Learning or deep learning. TF-IDF [10] Word embedding techniques are useful for processing textual data. TF-IDF is a statistical analysis technique based on the frequency of words that appear on the document. On the other side, word embeddings are successful in capturing semantic information, which is more relevant in case of log anomaly detection. Text log contains words from natural languages, and word embedding is an NLP technique for feature extraction, which transforms words in vectors of a real number.

T.F. Yen et al. [29] preprocessed the using domain knowledge and normalized it in terms of Timestamp, IP Address to host mapping, and static IP Address assigned, etc. as the log is from DHCP servers and policy, host and traffic based features are extracted.

This domain knowledge application to log data increases the accuracy, but the methods can not be generalized. M Du et al. [2], Weibin Meng et al. [4], and Xu Zhang et al. [8] have created the log event templates and log sequences using log parsers like SPELL[30], FT-Tree[31] and Drain[32] respectively. Further, a variety of approaches are available based on the problem statement. Approaches for anomaly detection are statistical, Machine Learning, and Deep learning based on the model technique used. The anomaly can be analyzed in the sequence of logs and also with the semantics of logs.

VI. DEEP LEARNING AND LOG ANOMALY DETECTION

TABLE I. SUMMARY OF LOG ANOMALY DATASETS

Dataset Name	Log Source	No of Message	Log Source Type	References
HDFS	Hadoop Distributed File System	11175629	Distributed System	[14] , [15] , [2], [8] , [16], [17] , [5], [4]
Spark	Spark Job	33236604	Distributed System	[18]
ZooKeeper	Zookeeper Service	74380	Distributed System	[17]
OpenStack	OpenStack Software	207820	Distributed System	[2] , [6], [19]
BGL	Blue Gene/L super computer	4747963	Super Computer	[20], [6], [17] , [4]
HPC	High Performance Cluster	433489	Super Computer	[21]
Thunderbird	Thunderbird Supercomputer	211212192	Super Computer	[20], [6] , [1]
Proxifier	Proxifier Software	21329	Standalone Software	[17]
Router Logs	NetEngine40E Router	18727517	Router	[3]

TABLE II. SUMMARY OF CHALLENGES ADDRESSED AND METHODS USED BY VARIOUS AUTHORS ON LOG ANOMALY DETECTION

Year	Citation	Challenges Addressed	DL Model	Data Set	NLP /Other Method	Pre-Processing / Parsing
2019	Xiaojuan Wang [3]	Extracted Time period Anomaly and causes for surge of Log , also represented semantics.	LSTM	Netengine40E Router Log	Directed Graph	Parsed on Behavior type
2019	Xu Zhang [8]	Instability of log data , worked for contextual information of log sequences.	Bi-LSTM with Attention	HDFS, Other security system of Microsoft	FastText[36]	Drain[32]
2019	Weibin Meng[4]	To detect sequential and quantitative anomaly simultaneously, worked for semantic relation of logs lost due when only log templates are used.	LSTM	BGL, HDFS	Template2Vec[4]	FT-Tree[31]
2019	Amir Farzad [6]	LSTM and BiLSTM models with autoencoders are used for Log message classification and anomaly detection.	Auto-LSTM, Auto-BLSTM,Auto-GRU	BGL, IMDB, Openstack, Thunderbird	Word Frequency	--
2018	Siyang Lu [5]	Compared CNN performance for log anomaly detection with LSTM and Multilayer Perceptron (MLP)	CNN based model	HDFS	LogKey2Vec	Logs-Key Sequences and session key
2018	Andy Brown[9]	Effect of attention for sequence modeling	LSTM with 5 attention mechanism	LANL, cybersecurity dataset	--	Language Modeling and Tokenization
2018	Mengying Wang [1]	Use of NLP techniques like word2vec and TFIDF for log anomaly detection	LSTM	Thunderbird	Word2Vec , TF-IDF	Data Cleaning of logs
2017	Min Du [2]	”Log Key” and ”Parameter value” anomaly detection and diagnosis using work flows.	LSTM	HDFS, Openstack	Log Key, Parameter value and Workflow	Spell[30]

T.F. Yen et al. [29] worked with SIEMs log data collected with approx 1.4 billion logs per day for detection of suspicious activity specific to enterprise settings and user profile behavior. This work faced the challenges of scalability, data noise, and no availability of ground truth. The proposed method requires the creation of a feature vector for each internet host using the history data. They utilize the unsupervised clustering with data specific features for the identification of potential security threats. Lack of ground truth is observed by the need of experts for manual labeling. The method is rule based, and processing of history logs required expert domain knowledge.

Min Du et al. [2] presented an architecture for the detection of an anomaly in log data, which does not require any domain related prior familiarity. The proposed approach identifies log key and parameter value anomaly from logs and also crates a workflow which is used for diagnosis purpose. A neural network based algorithm LSTM is applied to predict the likelihood of the next log key. Additionally, a similar type of LSTM neural network detects an anomaly in the log parameter sequence. The algorithm is also using manual feedback about false positive for improving future accuracy. The LSTM is used to support the fact that the log sequence is a natural language sequence and can be processing similarly.

Siyang Lu et al. [5] worked with the Log key and CNN model with the embedding of Logkey as input to the model. Two level parsing is applied on raw log data to get the log key and vector for log key sequences as per the execution order. They have presented a comparison of the CNN model with the LSTM and Multilayer Perceptron Model and found that CNN model accuracy is better than other models used.

Amir Farzad et al. [6] proposed deep learning based model for log message anomaly detection and presented a comparison between these proposed models for better efficiency using BGL, Thunderbird, Openstack, and IMDB datasets. IMDB dataset is used to prove the generalization of their proposed model for other text classification problems. In the architecture, word frequency is used to present textual log messages in numeric form. This positive and negative labeled data is passed to two different autoencoders for training to get an improved relationship with original data, and this output is used as input for the Deep Learning algorithm.

Mengying Wang et al. [1] also work to explore the prospect of using the Natural Language Processing algorithms for anomaly identification from log messages. In the experiments, word2vec and TF-IDF feature extraction algorithms are used, and activity is completed with LSTM deep learning algorithm for classification. They have concluded that the word2vec is performing better than TF-IDF for log message anomaly detection tasks.

W Meng et al. 2019 [4] designed an attention-based LSTM model for the detection of both types of anomaly i.e., sequential and quantitative simultaneously. It is using FT-Tree for parsing of logs and also proposed novel word representation method template2vec based on synonym and antonym to extract the semantic for anomaly detection effectively. This approach overcomes the problem of missing the valuable information from logs where only log template index is considered, and semantics relation of logs could not be revealed in [2]

Xiaojuan Wang et al. [3] have worked on Router logs collected form NetEngine40E and analyze the for the type of behavior, attribute, and status. The proposed model is a neural network LSTM to predict the surge in logs by analyzing the number of logs in the time period. Also, attribute syntax forest is used for performing semantic analysis on the attribute information. The work has been extended with training unsupervised machine learning models i.e. Isolation Forest [33], OneClassSVM [34], and density-based algorithm LocalOutlierFactor [35] using attribute information and value for finding the logs which are the cause of log surge.

Xu Zhang et al. [8] have proposed "Robust Log" one of recent work in log anomaly detection. They developed a BiLSTM classification model using vector representation of each log event considering its semantic information called the semantic vector of the log. Word vectorization using fastText [36] and TF-IDF based aggregation is performed for the creation of vector from the log event. The robust-log claims to perform well in case of unstable logs events also which is tested by creating a synthetic HDFS log dataset.

Table-II presents a summary of the challenges addressed by researchers along with Deep Learning models, Dataset and method used in various research work for log analysis.

VII. CONCLUSION

Applying deep learning for log analysis is a rapidly growing practice for effectively extracting knowledge from the unstructured textual log messages. This work is an addition to the consolidation work done on the domain of log anomaly using Deep Learning. In this paper, we surveyed various deep learning algorithms implemented for log anomaly detection. We also summarized different NLP feature extraction techniques used to capture semantic and context information of log messages. Word embedding has proved significant results for capturing semantic information from log messages. The unsupervised deep learning models like autoencoder provided satisfactory results, while the huge task of manual labeling of log messages can be avoided. We presented a comparative study of challenges addressed, and DL methods applied. Our work is also summarizing the standard datasets, which are useful for log analysis related study. We observed that the DL algorithms are performing much better as compared to traditional data mining and Machine Learning methods. Many DL methods have been explored, but still, there is much scope for improvement in results by optimizing hyperparameter and using other DL models.

REFERENCES

[1] M. Wang, L. Xu, and L. Guo, "Anomaly detection of system logs based on natural language processing and deep learning," in *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*, pp. 140–144, IEEE, 2018.

[2] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1285–1298, 2017.

[3] X. Wang, D. Wang, Y. Zhang, L. Jin, and M. Song, "Unsupervised learning for log data analysis based on behavior and attribute features," in *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*, pp. 510–518, 2019.

[4] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun, *et al.*, "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization*, vol. 7, pp. 4739–4745, 2019.

[5] S. Lu, X. Wei, Y. Li, and L. Wang, "Detecting anomaly in big data system logs using convolutional neural network," in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th IntlConf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pp.151–158, IEEE, 2018.

[6] A. Farzad and T. A. Gulliver, “Log message anomaly detection and classification using auto-b/lstm and auto-gru,” *arXiv preprint arXiv:1911.08744*, 2019.

[7] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[8] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li, *et al.*, “Robust log-based anomaly detection on unstable log data,” in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 807–817, 2019.

[9] A. Brown, A. Tuor, B. Hutchinson, and N. Nichols, “Recurrent neural network attention mechanisms for interpretable system log anomaly detection,” in *Proceedings of the First Workshop on Machine Learning for Computing Systems*, pp. 1–8, 2018.

[10] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[12] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] J. Zhu, S. He, J. Liu, P. He, Q. Xie, Z. Zheng, and M. R. Lyu, “Tools and benchmarks for automated log parsing,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 121–130, IEEE, 2019.

[14] W. Xu, L. Huang, A. Fox, D. Patterson, and M. Jordan, “Largescale system problem detection by mining console logs,” *Proceedings of SOSP’09*, 2009.

[15] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, “Detecting large-scale system problems by mining console logs,” in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pp. 117–132, 2009.

[16] S. Bursic, A. D’Amelio, and V. Cuculo, “Anomaly detection from log files using unsupervised deep learning,” 09 2019.

[17] P. He, J. Zhu, S. He, J. Li, and M. R. Lyu, “Towards automated log parsing for large-scale log data analysis,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 931–944, 2017.

[18] S. Lu, B. Rao, X. Wei, B. Tak, L. Wang, and L. Wang, “Log-based abnormal task detection and root cause analysis for spark,” in *2017 IEEE International Conference on Web Services (ICWS)*, pp. 389–396, IEEE, 2017.

[19] B. Debnath, M. Solaimani, M. A. G. Gulzar, N. Arora, C. Lumezanu, J. Xu, B. Zong, H. Zhang, G. Jiang, and L. Khan, “Loglens: A real-time log analysis system,” in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1052–1062, IEEE, 2018.

[20] A. Oliner and J. Stearley, “What supercomputers say: A study of five system logs,” in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN’07)*, pp. 575–584, IEEE, 2007.

[21] M. C. Dani, H. Doreau, and S. Alt, “K-means application for anomaly detection and log classification in hpc,” in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 201–210, Springer, 2017.

[22] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th annual meeting of the association for computational linguistics:*

[23] *Human language technologies-volume 1*, pp. 142–150, Association for Computational Linguistics, 2011.

[24] C. Suh-Lee, J.-Y. Jo, and Y. Kim, “Text mining for security threat detection discovering hidden information in unstructured log messages,” in *2016 IEEE Conference on Communications and Network Security (CNS)*, pp. 252–260, IEEE, 2016.

[25] S. He, J. Zhu, P. He, and M. R. Lyu, “Experience report: System log analysis for anomaly detection,” in *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 207–218, IEEE, 2016.

[26] Q. Fu, J.-G. Lou, Y. Wang, and J. Li, “Execution anomaly detection in distributed systems through unstructured log analysis,” in *2009 ninth IEEE international conference on data mining*, pp. 149–158, IEEE, 2009.

[27] L. Tang, T. Li, and C.-S. Perng, “Logsig: Generating system events from raw textual logs,” in *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 785–794, 2011.

[28] A. A. Makanju, A. N. Zincir-Heywood, and E. E. Milios, “Clustering event logs using iterative partitioning,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1255–1264, 2009.

[29] R. Vaarandi, “A data clustering algorithm for mining patterns from event logs,” in *Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IPOM 2003)(IEEE Cat. No. 03EX764)*, pp. 119–126, IEEE, 2003.

[30] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda, “Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks,” in *Proceedings of the 29th Annual Computer Security Applications Conference*, pp. 199–208, 2013.

[31] M. Du and F. Li, “Spell: Streaming parsing of system event logs,” in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 859–864, IEEE, 2016.

[32] S. Zhang, W. Meng, J. Bu, S. Yang, Y. Liu, D. Pei, J. Xu, Y. Chen, H. Dong, X. Qu, *et al.*, “Syslog processing for switch failure diagnosis and prediction in datacenter networks,” in *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, IEEE, 2017.

[33] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, “Drain: An online log parsing approach with fixed depth tree,” in *2017 IEEE International Conference on Web Services (ICWS)*, pp. 33–40, IEEE, 2017.

[34] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, IEEE, 2008.

[35] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C.¨ Williamson, “Estimating the support of a high-dimensional distribution,” *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[36] L. Xu, Y.-R. Yeh, Y.-J. Lee, and J. Li, “A hierarchical framework using approximated local outlier factor for efficient anomaly detection,” *Procedia Computer Science*, vol. 19, pp. 1174–1181, 2013.

[37] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jegou, and T. Mikolov,´ “Fasttext. zip: Compressing text classification models,” *arXiv preprint arXiv:1612.03651*, 2016.