The Hybrid Approach Using TF-IDF, XG Boost and Light GBM for Log Event Detection

Syed Suriya Bhanu, Ms. M. Pavithra
PG Student, Vemu Institute of Technology, P. Kothakota
Assistant Professor, Vemu Institute of Technology, P. Kothakota

Abstract: **The continuing growth of large-scale and complex software systems has led to growing interest in examining the possibilities of using the log files that were created during the runtime of the software. These files can be used for various purposes like error prediction, performance evaluation, learning of usage patterns, improving reliability, and so on. With software systems continuously becoming more and more complicated, the distinction of log files that were generated by different components of the software becomes a new task. The classification of log files is important for several reasons like resource optimization, compliance and auditing automation and analysis, or understanding the general system health. By classifying log files, organizations can better understand the health and performance of their systems. They can identify patterns, potential security threats, anomalies, errors, and malicious behaviors and storage can also be optimized. In the log files, each line represents a specific event that has occurred.**

INTRODUCTION

All software is expected to create log files during its run, which can be used to monitor the states it went through, the operations performed, and other fundamental information. Log files are created via print statements that were inserted into the source code via the developers to save necessary information. They have many areas of use such as data mining and analysis. One of the most common types of logs is weblogs. They record which pages were visited by which users, what were their requests, what were the response times and so on. In [1] it is shown how one's supposed to configure a web server to gain useful log files that can be used for analytic purposes. The main advantage of the analyzation is that it can be used by an e-commerce site operator to acquire the global overview of the feedback immediately and an extensive understanding of events occurring on the e-commerce site as well as on social media. The only drawback of this approach is that it does not take information from the physical shopping floor into account. The authors of [2] investigate how clustering algorithms such as K-means and DBSCAN can be used to cluster the dataset and retrieve practical information from web server log files. They found that DBSCAN has a better performance in clustering such files than K-means.

LITERATURE SURVEY

1. Aivalis (2022) – Log File Analysis in E-Tourism
Aivalis explored the use of log file analysis in e-tourism, demonstrating how data from web server logs can provide valuable insights into customer behavior. The study showed that log data analysis could help e-commerce operators understand consumer interactions in real-time. However, the study highlighted a major limitation: the exclusion of offline customer behaviors, which led to an incomplete understanding of consumer preferences.

2. Fawzia et al. (2022) – K-Means and DBSCAN for Web Log Clustering
Fawzia et al. compared K-Means and DBSCAN clustering techniques for web server log file analysis. Their research found that DBSCAN was superior for clustering web log data, as it was more effective in identifying outliers. However, scalability remained a significant challenge, as the DBSCAN algorithm struggled with extremely large datasets, limiting its usability in high-traffic environments.

3. Neelima & Rodda (2016) – Session Identification for User Behavior Analysis
This study focused on identifying user sessions from web log data to analyze consumer behaviors. The proposed approach provided detailed behavioral insights by recognizing distinct user sessions. However, the researchers noted that session identification required significant computational resources, making it difficult to apply in real-time processing scenarios.

4. Kim et al. (2019) – Anomaly Detection for Insider Threats
Kim et al. developed an anomaly detection system that utilized log files to identify insider threats in organizations. Their methodology relied on behavior modeling and anomaly detection algorithms, reducing the workload for security teams. One drawback was that inconsistencies arose when applying the method across different datasets, highlighting the need for more generalized models.

5. Stachlet al. (2020) – Personality Prediction from Smartphone Log Data
This research employed machine learning techniques on smartphone log data to predict personality traits. The results showed that several personality dimensions could be accurately determined using log file patterns. However, the model struggled with certain traits, suggesting the need for additional data sources or more complex feature engineering to improve accuracy.

PROBLEM STATEMENT

As software systems grow more complex, distinguishing logs generated by different components becomes challenging. Effective log file classification is necessary to optimize resource use, detect security threats, and improve overall system performance.

PROPOSED METHOD

In propose work, instead of using the full-sized log files, we change each line to its corresponding event ID and use the resulting smaller file for classification purposes. We use numerous classifying algorithms like Random Forest, K-NN, AdaBoost Classifier, and Decision Tree to assign the files to groups corresponding to their origin types. 80% of the data is used for learning purposes while the remaining 20% is used for testing. We conduct numerous different experiments to verify the effectiveness of our method like evaluating the precision, recall, f-score, and accuracy values and measuring the time it takes to classify the files.

ARCHITECTURE

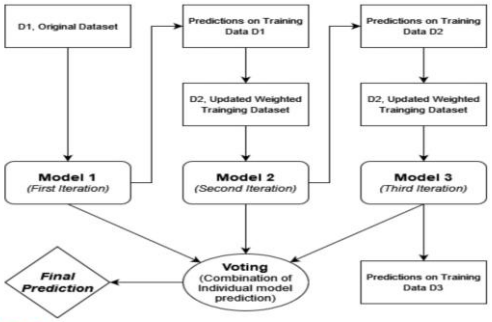

FIGURE 2. 3 steps of an AdaBoost Classifier with 3 estimators.

METHODOLOGY

A. LOG PARSING AND THE PROPOSED ALGORITHM

The entries of a log file are usually unstructured and raw due to the fact that programmers can insert free-text messages into their print statements. Each log entry contains runtime information about events that have happened like restarts, messages being sent or received, error occurrences, and so on. A log message usually starts with a list of information like timestamps, the module name that produced the message, and others. In this paper, we only focus on the message part. Each word in a message can be either a constant or a parameter. A constant token is always the same at each occurrence of a log line corresponding to an event type. Parameter tokens can be different on each occasion. A fragment of our working data can be seen in Figure 1. The corresponding event template of this log message is ‘‘Receiving block src : dest: ’’. The ‘‘’’ symbols indicate the presence of a parameter token, and the parameters of this specific entry can be seen in the parameter list. The template that was just discussed is identified by the unique ID ‘‘ABC123’.

B. DECISION TREE AND RANDOM FOREST

Decision tree learning is a supervised learning method commonly used in data mining, machine learning, and statistics [27]. Let’s assume that all of the input attributes have finite discrete domains, and there is a single target attribute called the ‘‘classification’’. Classes are the elements that can be found in the classification attribute’s domain. All the internal nodes of the decision tree are labeled with an input attribute. The edges coming from a node labeled with an input attribute must be labeled with each of the possible values of the target attribute or the edge has to lead to a ‘‘lower’’ internal node on a different input attribute. Each leaf of a tree

C. K-NEAREST NEIGHBORS (K-NN)

(1) The k-nearest neighbors algorithm is a supervised non parametric learning algorithm that can be used for classification and was proposed in [30] and later broadened in [31]. Training examples are represented as vectors in a multidimensional attribute space with each having their represents a class. corresponding class label. The attribute vectors and the labels are stored in the training phase. Let k be a user defined constant, generally a small positive integer. In the classification stage, the test point that is an unlabeled vector is being classified by a plurality vote of its neighbors. In other words, the class that is the most frequent among the k neighbors of the test point is being assigned to it. If k = 1, the assigned class is simply the class of the vector that is closest to the test point. Various distance metrics like Minkowski distance [32], Manhattan distance [32], Jaccard distance [33], Cosine distance, Cheb ysev distance [34] and Hamming distance [35] can be used with the algorithm.

D. ADABOOST CLASSIFIER

Ada-boost or Adaptive Boosting is an ensemble boosting classifier that was proposed in 1996 by Yoav Freund and Robert Schapire [36]. To increase the accuracy of classification, the base idea is to combine multiple classifiers. AdaBoost is an iterative ensemble method. By combining multiple weakly performing classifiers, AdaBoost Classifier builds a strong classifier with high accuracy. The basic concept behind Adaboost is to set the weights of classifiers and train the data sample in each iteration such that it ensures accurate predictions of unusual observations. As the base classifier, any machine learning that accepts weights on the training set can be used. The algorithm works in the following steps. First, a random subset of the training data is selected.

CONCLUSION

The classification of log files with different origins has become an important assignment in recent years. They are made up from log lines that are usually free-text messages with parameters. Each line belongs to an event type. An event type is a template, where the constant part of the template is the same at any occurrence, while the parameter parts might change. Classifying log files with full-length log messages is a resource and time-consuming task. To combat this, in this paper, we propose a new algorithm designed to modify log files, which are subsequently used as input for classification algorithms. Our algorithm alters the log files to only contain the IDs of the event types instead of the full text. To evaluate the performance of the classification methods that had the modified files as their input, we conducted various experiments like investigating the precision, recall, f score, and accuracy values achieved during the classification. The results yielded that in the case of KNN and AdaBoost Classifier, the original algorithms outperformed the ones using the proposed algorithm with an average of 20% and 10%respectively. In the case of Random Forest and Decision Tree Classifiers, the ones with the input generated by the proposed algorithm have surpassed their original counterpart with an average of 28% and 9%.In terms of speed, the classification takes 52 to 90 times less time, if the altered log files created by our proposed algorithm are used as input for the classifiers.

REFERENCES

[1] C. J. Aivalis, ‘‘Log file analysis,’’ in Handbook of E-Tourism. Springer, 2022, pp. 659–683.

[2] A. Fawzia Omer, H. A. Mohammed, M. A. Awadallah, Z. Khan, S. U. Abrar, and M. D. Shah, ‘‘Big data mining using K-means and DBSCAN clustering techniques,’’ in Big Data Analytics and Computational Intelligence for Cybersecurity. Springer, 2022, pp. 231–246

[3] G. Neelima and S. Rodda, ‘‘Predicting user behavior through sessions using the web log mining,’’ in Proc. Int. Conf. Adv. Human Mach. Interact. (HMI), Mar. 2016, pp. 1–5.

[4] J. Kim, M. Park, H. Kim, S. Cho, and P. Kang, ‘‘Insider threat detection based on user behavior modeling and anomaly detection algorithms,’’ Appl. Sci., vol. 9, no. 19, p. 4018, Sep. 2019.

[5] C. Stachl, Q. Au, R. Schoedel, S. D. Gosling, G. M. Harari, D. Buschek, S. T. Völkel, T. Schuwerk, M. Oldemeier, T. Ullmann, H. Hussmann, B. Bischl, and M. Bühner, ‘‘Predicting personality from patterns of behavior collected with smartphones,’’ Proc. Nat. Acad. Sci. USA, vol. 117, no. 30, pp. 17680–17687, Jul. 2020.

[6] J. Svacina, J. Raffety, C. Woodahl, B. Stone, T. Cerny, M. Bures, D. Shin, K. Frajtak, and P. Tisnovsky, ‘‘On vulnerability and security log analysis: A systematic literature review on recent trends,’’ in Proc. Int. Conf. Res. Adapt. Convergent Syst., 2020, pp. 175–180.

[7] C. N. Kabat, D. L. Defoor, P. Myers, N. Kirby, K. Rasmussen, D. L. Saenz, P. Mavroidis, N. Papanikolaou, and S. Stathakis, ‘‘Evaluation of the elekta agility MLC performance using high-resolution log files,’’ Med. Phys., vol. 46, no. 3, pp. 1397–1407, Mar. 2019.

[8] J. Kimball, R. A. Lima, and C. Pu, ‘‘Finding performance patterns from logs with high confidence,’’ in Proc. 27th Int. Conf., Held Services Conf. Fed., SCF Web Services–ICWS, Honolulu, HI, USA. Springer, Sep. 2020, pp. 164–178.