Clustering Techniques for Effective Log File Analysis in Cybersecurity

Madhuri Kanojiya[1], Lokesh Chouhan[2]

[1,2]National Institute of Technology,
Hamirpur, Himachal Pradesh, 177005, India

Abstract— **Tracking prior actions within a system is crucial for cybersecurity analysis, and log files provide the data needed for this purpose. Analyzing log files manually is challenging due to their diverse and unstructured nature, so specific algorithms are necessary. Clustering techniques are employed to address this task effectively, and the choice of approach varies based on the specific domain and desired outcomes. This paper groups clustering approaches according to their operational characteristics, domain relevance, and other technical considerations.**

Index Terms— **Anomaly Detection, Clustering, Cybersecurity, Logs, Signature extraction.**

1. INTRODUCTION

LOG is a file that records events for the system that consists of information regarding software and operating systems. They are often used to show events related to a certain application (backup, Web-servers, firewalls). Windows generate all kinds of log files depending upon its various services. The point of a log file is to track the working behind the applications [1]. Moreover, in recovery of the system to convolute nature, rectified imprecise transactions, avoid analytical information loss, mend data logs can be used [11-13]. A commanding issue with forensic log dissolution is that it craves for time and resource exhausting with terrain awareness about the conformity at hand, therefore a shift has been made from forensic to proactive analysis to enable real-time fault detection so that timely responses or just reduction of a casualty by incidents and cyber-attacks [14-19]. Disclose such indicators initial enough and initialing appropriate measures to halt faults altogether [2].

Due to advancements in technology and the generation of immense volumes of data, it is hardly possible for human(s) [20-22]. The Machine learning approaches that rules out the lines and recognizing impressions comes in handy as they are much effective to operate on the system in a condensed form as the manual analysis is impossible for large scale enterprises.

Questions to be considered are as follows:

- Applications of algorithms in cybersecurity.
- Evaluation of approaches.
- Characteristics of the extant clustering approaches.
- What kind of data they work on.

2. SURVEY HISTORY

Log data posses some aspects that have to be considered while designing a clustering algorithm. Therefore seeing the properties of log data, requisite to be aggregated and considering the utilization sides significant in cybersecurity.

(i) THE ESSENCE OF LOG DATA

Log data prevail in diverse profiles but some generic standards could be advised. Mainly log file generally consists of single or multiple string lines compromises of some data placed sequential order. The indicated order is generally a timestamp that gives information about the time of certain activity, this(timestamp) is attached to the log messages. These messages could be orientated highly structurally, partially, or maybe unstructured way. Additionally, these logs can information about the processes that generated them, but this doesn't concern much as the nature of natural language and log files are very distinct. The essence of events are of striving nature and on the other hand procedures falls under loops and major of the log lines are spawn by print statements [2].

We can therefore aggregate system logs into two divergent ways by utilizing these properties. First, clustering classification of log messages gives perception into the elementary program logic and display contrarily concealed interdependence of components and events. Second, grouping separate lines(log) by the resemblance of their messages return an audit of all events that take place in the structure[2].

ii) STAGNANT CLUSTERING

It is referred to as static clustering is also known as stagnant clustering if clustering is done based on individual lines. Now during clustering based on a certain attribute, it is always possible that there exist multiple different possible valid clustering and the efficiency is eventually a subjective decision which in turn based upon the nature of the application utilizing it.

For example, if we consider an example of issues related to clustering in users logging in and out.

```
1:::User  A logs in with state 1
2:::User  A  logs out  with state -1
3:::User  B logs in with state 1
4:::User  C logs in with state 1
5:::User  D logs in with state -1
6:::User  C  logs out with state 1
```

Now clustering can be done upon User name[{1,2},{3},{4,6},{5}] or according to state variable [{1,3,4,6},{2,5}].

Hence it can be deduced that the quality of clustering is a subjective decision and should be taken after proper study of requirements of the system since these clustering would form a basic foundation for further analysis.

iii) DYNAMIC CLUSTERING

By the above example, it can be observed that log files are the best fit for the dynamic clustering, clustering according to appearances to patterns. Since pattern recognization depends upon repeating patterns, the log lines need to be allocated in classes by static clustering.

Considering the above example again:

```
1::User  A logs in with state 1- - - - P
2::User  A performs  action X- - - - -Q
3::User  A logs out  with state 1- - - - R
4::User  B logs in with state -1- - - - -P
5::User  B performs action Y- - - - - - Q
6::User  C logs in with state 1- - - - - P
7::User  C performs action  Z- - - - - - Q
8::User  C logs out with state 1- - - - - -R
```

9::User B logs out with state 1- - - - - R

Now we categorized log lines in three classes:
P for "User *boots in"
Q for "User * implement action *"
R for "User* boots out"

Therefore the result of dynamic clustering could be interpreted by the behavior P, Q, R. In real systems, interlined processes exist in log files and make it complicate pattern elicitation processing [2].

The categorization is not on a class basis, it could be developed by grouping them according to time slots and then analyze the relation between them according to the system requirements [2].

3. APPLICATIONS IN SECURITY DOMAIN

Analysts can make it handy to use the property of log files containing everlasting documentation in figuring out the anomalies in the system.

The ample amount of log data makes it easier for analysts to make a study on how accurately the companies are working their business in aspect of their IT preservation. .

Benefits from logs:
Tracks employees actions
Remedy problems
Investigate faulty system behavior
Cyber attacks detection

Clustering can reduce the efforts of manually analyzing log files and can be used for the disclosure of anomalous behavior in the system by observing the outline of log file significance [3].

Clustering can be helpful in certain types of anomaly detection:

- *FREQUENCY ANOMALIES*:-Those log lines that appeared a different rate in a certain interval of time.
- *OUTLIERS*:-The log lines that are different from previously existed templates or different from other clusters.
- *CORRELATION ANOMALIES*:-Sometimes certain events are expected to occur in pairs(before connecting to the server that request of the connection must have passed through a firewall), but fail to do so falls in correlation anomaly.
- *INTER-ARRIVAL ANOMALIES*:-These occur when the time deviates of occurring of log events.
- *SEQUENCE ANOMALIES*:-Are caused when we miss or add certain log events out of order.

4. ANOMALY DETECTION

Outliers are the only category of anomalies observable by static cluster algorithms as it is based on single log line existence, whereas dynamic clustering procedure deals with the other types of anomalies [2]. Therefore it can be solved by different approaches depending upon static or dynamic clustering.

(i) STATIC OUTLIER DETECTION

The aggregation of log lines in the previously formed clusters shed some light upon the type or detection of the anomalies. If we find some different patterns of the log lines as we have seen in clusters, this could be a case of something fishy or should be reported, these are considered to be outliers. A key search on the current logs can be used for detecting these kinds of anomalies [4].

(ii) DYNAMIC ANOMALY DETECTION

The static based methods only able to detect outlier whereas for failure patterns dynamic based methods are required. The main goal of dynamic methods is the revival of an explicit and expressive label [2].In some cases, we opt to find the root cause of the failure by analyzing the log events of clusters and identifying dependencies which lead to overall failure. Analysts can keep track of the occurrence of log events by filling an event count matrix and thus uses it as a principal factor for reasoning and revealing unconventional points [6].

Other properties such as the execution sequence of logs, correlating the execution times of newly occurring log succession with the learned expression, deviation from interval times as latency anomalies can be used to detect anomaly other than the usual approaches.

5. PROBLEM DOMAINS & TECHNIQUES

The core aspect to understand in clustering is about extraction and generation of signatures, the anomalies are the by-product of these whether it is static or dynamic. The static outlier detection is more of a by-product of extraction and generation whereas in dynamic detection it has higher relevance and therefore dealt with various strategies.

Different approaches were considered to tackle these including a token-based approach or character-based approach. Token-based approaches are generally cost-efficient and take less time computationally as of its good alignment towards heuristics but as a saying goes "there is no free lunch", some of its attributes make character-based approaches to be of more potential [7].

6. APPROACH SELECTION

As log clusters possess different characteristics and properties therefore it is necessary to use tools according to our needs and group these approaches into meaningful classes. Now for grouping, we need to consider several factors as there is a range of relevant attributes to be taken care of.

Now if we have to select an already existing approach for clustering, we need to check each one under various cases to check its compatibility and other the performance as compared to other approaches, it would be time-consuming and thus we need to find some other way for selection. A model has been proposed by [7] that uses the weighted sum method for ease of decision making.

Before deducing the algorithm for clustering it before analyzes the attributes or properties or constraints that need to consider before using an algorithm so that it would not fail in complex systems.

- *OBJECTIVES*: It's important to recognize the objective of the problem domain as the algorithm deals with a definitive problem and that it can deliver maximum efficiency. There is no hardcoded rule to follow it all depends upon the problem specifics.

▪*LOG DATA*: For performing operations such as elicitation of log footprints, some clustering algorithms require advanced techniques, but trivial thing is that the log lines comply with certain norms or embody significant attributes like process IDs [2]. The more thoroughly one insights into the data the more information could be retrieved and be used for generating algorithms.

- *SYSTEM REQUIREMENTS*: The generation of log lines is determined by the system designer which further depends upon system requirements. The pace at which log lines will generate also depends upon system scope, in some system time is the factor, in some efficiency(not in real-time). All these factors help to generate a more suitable algorithm approach [2]. Meta information on the progress can be used for the origin of supplementary attributes that may be of purpose [2].

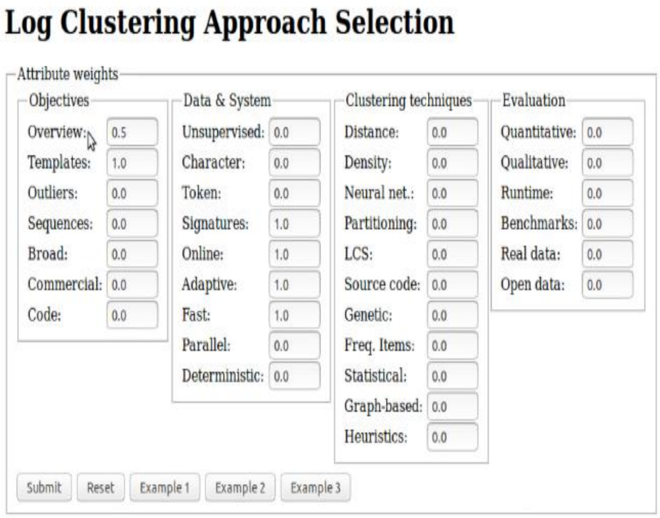
7. WORKING MODEL

After we analyzed the corresponding attributes, weights are assigned to them accordingly to their relevance. These weights are used to compute the overall score of the approach used, if the approach satisfies the attribute we multiply it with(1.0), if only partially satisfied multiply the weight by(0.5), if not then by(0).

The weighted traits are then summed up row-wise in the tables, i.e., the score of each approach is the sum of the weighted trait. Attributes that are weighted higher have a larger influence on the score, attributes that are weighted 0.0 do not influence the scores, and attributes with scores < 0.0 negatively affect the score [2].

The rank of the approach can be computed using these scores, for better understanding normalize them in[0,1].

Considering an example of a ranking model as a web-based application.



Log Clustering Approach Selection

Attribute weights

Objectives
Overview: 0.5
Templates: 1.0
Outliers: 0.0
Sequences: 0.0
Broad: 0.0
Commercial: 0.0
Code: 0.0

Data & System
Unsupervised: 0.0
Character: 0.0
Token: 0.0
Signatures: 1.0
Online: 1.0
Adaptive: 1.0
Fast: 1.0
Parallel: 0.0
Deterministic: 0.0

Clustering techniques
Distance: 0.0
Density: 0.0
Neural net.: 0.0
Partitioning: 0.0
LCS: 0.0
Source code: 0.0
Genetic: 0.0
Freq. Items: 0.0
Statistical: 0.0
Graph-based: 0.0
Heuristics: 0.0

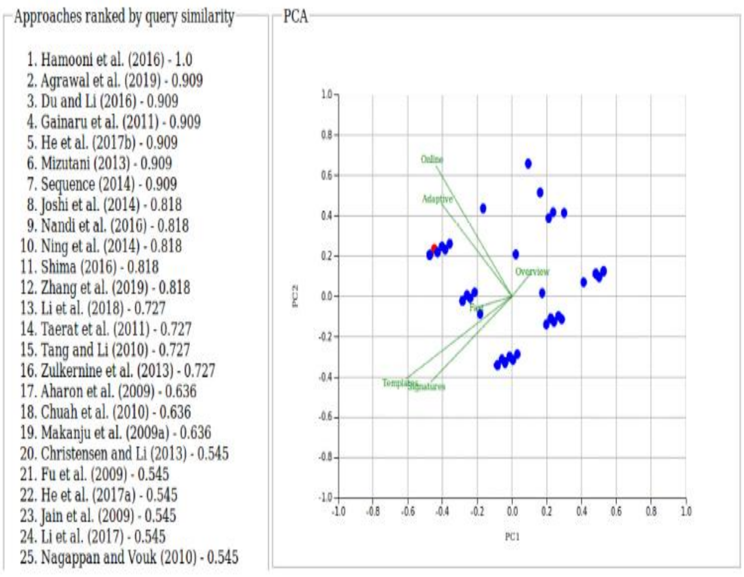
Evaluation
Quantitative: 0.0
Qualitative: 0.0
Runtime: 0.0
Benchmarks: 0.0
Real data: 0.0
Open data: 0.0

The above fig shows the main interface providing information regarding the weighted attributes, rank score of approaches.

Now for analyses plotting a graph representing various information about approaches, features, weights. The principal component analysis (PCA) is very useful as it provides a visual representation of a comparison of approaches at a single point and helps to form clusters based upon the closeness of the distance between the points.

The blue lines are representing the approaches and green 7lines denote feature axes (weight equal to 0). Clicking the "Submit" button reconsiders the ranking and procreate a new plot for the currently entered weights, the “Reset” button nullify all weights to 0.0, and the “Example” buttons act according to predefined scenarios [2].

By following the axes lines, one can get the idea of fulfillment or lacking attributes satisfied by the approach and thus gives a better outlook.



Approaches ranked by query similarity

1. Hamooni et al. (2016) - 1.0
2. Agrawal et al. (2019) - 0.909
3. Du and Li (2016) - 0.909
4. Gainaru et al. (2011) - 0.909
5. He et al. (2017b) - 0.909
6. Mizutani (2013) - 0.909
7. Sequence (2014) - 0.909
8. Joshi et al. (2014) - 0.818
9. Nandi et al. (2016) - 0.818
10. Ning et al. (2014) - 0.818
11. Shima (2016) - 0.818
12. Zhang et al. (2019) - 0.818
13. Li et al. (2018) - 0.727
14. Taerat et al. (2011) - 0.727
15. Tang and Li (2010) - 0.727
16. Zulkernine et al. (2013) - 0.727
17. Aharon et al. (2009) - 0.636
18. Chuah et al. (2010) - 0.636
19. Makanju et al. (2009a) - 0.636
20. Christensen and Li (2013) - 0.545
21. Fu et al. (2009) - 0.545
22. He et al. (2017a) - 0.545
23. Jain et al. (2009) - 0.545
24. Li et al. (2017) - 0.545
25. Nagappan and Vouk (2010) - 0.545

Therefore from the above diagram, we can conclude that (Hamooni et al.2016) proved to be the most efficient approach by being on top of the list. By studying the radical of the plot we can deduce that it can only be used for signature extraction and not for online or dynamic log processing.

8. FUTURE WORK

For future work, we can focus upon building log profiles with the help of tools and to do that we can prioritize more on the rule to create algorithms. To find out an affiliation between event attributes within a single event cluster and detection of temporal impression is also considered to be a compelling task in the future.

9. RESEARCH WORK

Log-positioned problem description has become a very popular area in novel years. Based on data elicit from logs, these fed as work to engage machine learning and data mining approaches to evaluate logs for anomaly detection and problem diagnosis. They are also extensively used for the maintenance and diagnosis function for various software systems as they provide generous information. The essence of anomaly detection is to find the system’s peculiar behaviors and use this as an assessment to engineers for in addition to interpretation.

10. CONCLUSION

Analyzing log data is very crucial for applications to give better performance. Several approaches have been used for different specifications of the system requirements but it takes time to select a particular algorithm as to study exhibit characteristics of the problem. In this paper, we have studied a survey that group certain reflect upon clustering approaches according to the attributes.

We examined different division of clustering mode which compromises neural networks, partitioning, pattern tracing, source code scrutiny and other like character -based, token-based, density-based and distance -based techniques. The occurrence usually follows either one or many dominant intention: Overview and filtering, parsing and signature eradication, stagnant outlier detection, and sequences and dynamic anomaly detection.

By using previously mentioned we presented the befitting tool for log clustering, the tool ranks the approaches based upon their performance on the problem domain.

REFERENCES

1. G. Brady, "What is a Log File?," How-To Geek, July 2018. [Online]. Available: https://www.howtogeek.com/359463/what-is-a-log-file/.
2. F. Skopik, M. Wurzenberger, A. Rauber, and M. Landauer, "System log clustering approaches for cybersecurity applications: A survey," January 2020, p. 17.
3. S. Security, "3 Major Benefits of Log File," Cygilant, November 2016. [Online]. Available: https://blog.cygilant.com/blog/3-major-benefits-of-log-file/.
4. J. Zhang, M. Debnath, "LogMine: Fast pattern recognition for log analytics," in 2016, pp. 1573-1582.
5. L. Zhu, L. He, "An online log parsing approach," in International Conference on Web Services, 2017, pp. 33-40.
6. R. Kastner, S. Wurzenberger, and M. Landauer, plying high performance bioinformatics tools for outlier detection in log data," in Proceedings of the 3rd International Conference on Cybernetics, 2017, pp. 1-10.
7. J. S. Marler and A. Arora, "Survey of multi-objective optimization methods for engineering," in Structural and Multidisciplinary Optimization, 2004, pp. 368-395.
8. Z. Lou and Z. Qingwei, "Log clustering based problem identification for online," 2014.
9. K. Bonsor, "Workplace Surveillance," HowStuffWorks. [Online]. Available: https://computer.howstuffworks.com/workplace-surveillance4.htm.
10. T. Taerat, B. Baler, "Deterministic lossless log message clustering tool," in 2011, pp. 3-4.
11. Lokesh Chouhan and H.S. Lalventhangi, "Adaptive Energy Detection Based Solution for Fronthaul Problem in C-RAN," *Wireless Personal Communications (WPC)*, Springer Publication, vol 103, 2018, pp. 2743–2755, ISSN: 0929-6212, 14 Sep, 2018. DOI: 10.1007/s11277-018-5960-6.
12. Lokesh Chouhan and Aditya Trivedi, "Performance Study of a CSMA based Multiuser MAC Protocol for Cognitive Radio Networks," *Wireless Networks (WINE)*, Springer Publication, vol 22, no 1, 2016, pp. 33-47, ISSN: 1022-0038. DOI: 10.1007/s11276-015-0947-7.
13. Jayanti Rastogi, Lokesh Chouhan, and Aditya Trivedi, "Multichannel CSMA Based MAC Scheme for Unsaturated Cognitive Radio Networks," *Wireless Personal Communications (WPC)*, Springer Publication, vol 85, no 3, 2015, pp. 1279-1294, ISSN: 0929-6212. DOI: 10.1007/s11277-015-2840-1.
14. Lokesh Chouhan and Aditya Trivedi, "MAC Layer Protocols for Cognitive Radio Network," *Self Organization and Green Applications in Cognitive Radio Networks*. IGI Global, 2013, pp. 154-189. Web. 5 Jul. 2013. ISBN13: 9781466628120, DOI: 10.4018/978-1-4666-2812-0.
15. Rajni Dubey, Sanjeev Sharma, and Lokesh Chouhan, "Security for Cognitive Radio Networks," *Cognitive Radio and Interference Management: Technology and Strategy*. IGI Global, 2013, pp. 238-256. Web. 18 Nov. 2011. ISBN13: 9781466620056, DOI: 10.4018/978-1-4666-2005-6.
16. Wali Ullah Farooqui and Lokesh Chouhan, "Coordinated Multi-Robot Navigation Using Sectorization of Environment," *Conference on IT in Business, Industry and Government (CSIBIG) 2014*, IEEE, CSI Indore Chapter, pp. 1-6, 08-09 March 2014. ISBN: 978-1-4799-3064-7. DOI: 10.1109/CSIBIG.2014.7057009.
17. Purushottam, Aditya Trivedi, and Lokesh Chouhan, "Channel Allocation and Resource Optimization in Cognitive Radio Cloud Network," *Conference on Advances in Mobile Communications, Networking and Computing*, organized by ICEIT, New Delhi, 27 – 28 September, 2013.
18. Lokesh Chouhan and Aditya Trivedi, "Analysis of MAC Schemes for Cognitive Radio Network: Perfect and Imperfect Learning Modelling," *Proceedings of 10th IEEE International Conference on Wireless and Optical Communications Networks (WOCN-2013)*, July 26-28, 2013, Bhopal, India, pp. 1-6. ISBN: 151-7703. DOI: 10.1109/WOCN.2013.6616247.
19. Lokesh Chouhan and Aditya Trivedi, "Priority based MAC scheme for cognitive radio network: A queuing theory modeling," *Proceedings of 9th IEEE International Conference on Wireless and Optical Communications Networks (WOCN-2012)*, September 20-22, 2012, Indore, India, pp. 1-5. ISBN: 151-7703. DOI: 10.1109/WOCN.2012.6331886.
20. Rajni Dubey, Sanjeev Sharma, and Lokesh Chouhan, "Secure and Trusted algorithm for Cognitive Radio Network," *Proceedings of 9th IEEE International Conference on Wireless and Optical Communications Networks (WOCN-2012)*, September 20-22, 2012, Indore, India, pp. 1-7. ISBN: 151-7703. DOI: 10.1109/WOCN.2012.6331887.
21. Lokesh Chouhan and Aditya Trivedi, "Cognitive radio networks: Implementation and application issues in India," *Seminar on Next Generation Network - Implementation and Implication*, Telecom Regulatory Authority of India (TRAI), Govt. of India, New Delhi, 25-26th August, 2011. Web. PDF link.
22. Lokesh Chouhan and Sanjeev Sharma, "Implementation of RSA shared key algorithms to secure Mobile Ad Hoc Networks," *Proceeding of IEEE International Conference on recent trends in soft computing and information technology*, Bhopal, pp. 416-421, 7-8 Jan 2010