Root Cause Analysis in a GPON Network Alarm Manager

Data mining; Pattern matching; Alarm floods; Similarity analysis

White paper

Version 1.0, September 2023

Introduction

In the realm of modern technology, intricate systems are pivotal for the smooth operation of diverse sectors, from telecommunications to transportation and more. However, complexity also brings the risk of system failures. Uncovering the reasons behind these failures is crucial, and Root Cause Analysis (RCA) is key in this context.

RCA is a systematic approach to identifying the core reasons behind a system problem. By revealing these root causes, organizations can take effective actions to address issues and prevent recurrence [1][2][3]. Yet, traditional RCA involves manual investigation, which can be time-consuming, resource-intensive, and subjective.

For complex systems like fiber optic networks or large-scale industries, traditional RCA methods fall short due to the exponential increase in variables [4][5][6]. Handling vast data, spotting patterns, and deriving insights becomes challenging [4][7]. Delays in RCA can lead to prolonged downtime and financial losses.

This paper delves into data mining techniques to enhance RCA in complex settings. By developing new algorithms, we automate identifying AFs, characterizing alarm clusters, and highlighting representative sequences.

Data mining, part of AI and data analysis, offers a solution. It employs algorithms to explore data, uncover patterns, and reveal correlations. Applying data mining to RCA automates alarm data analysis, identifies recurring patterns, and prioritizes critical issues [2][8][9].

RCA is vital for system reliability [1][2][3]. As traditional methods have limitations, innovative approaches are essential. Through data mining, this paper empowers organizations to efficiently analyze alarm data, enhance RCA, and ensure continuous service. Leveraging data mining can create more resilient systems that proactively tackle issues and maintain uninterrupted operations.

Alarm Manager

Before exploring data mining techniques, it's crucial to introduce the central role of the alarm manager in this process. In this case study, the RCA tool in use is Alarm Manager, developed by Altice Labs. The Alarm Manager is partly responsible for generating and storing alarm instances tied to equipment failures for customers [10][11][12]. Understanding its functionality and data storage is vital for grasping project details.

Alarm management involves monitoring, analyzing, and responding to alarms from various systems. The aim is effective and reliable alarm handling to prompt appropriate actions during abnormal events [10][11].

Altice Labs' AGORA serves as a comprehensive Fault Management tool. It handles alarm reception, storage, treatment, and correlation. This central hub manages alarm parameters, cycles, and processing across diverse collection channels. Drawing on telecom insights, it streamlines fault monitoring, detection, and ticket initiation, boosting operational efficiency and adaptability [10][12].

AGORA offers a user interface that provides a more intuitive and practical real-time visualization of active alarms in the network. Additionally, it enables users to generate reports on the historical record of occurred alarms, enhancing accessibility and ease of use.

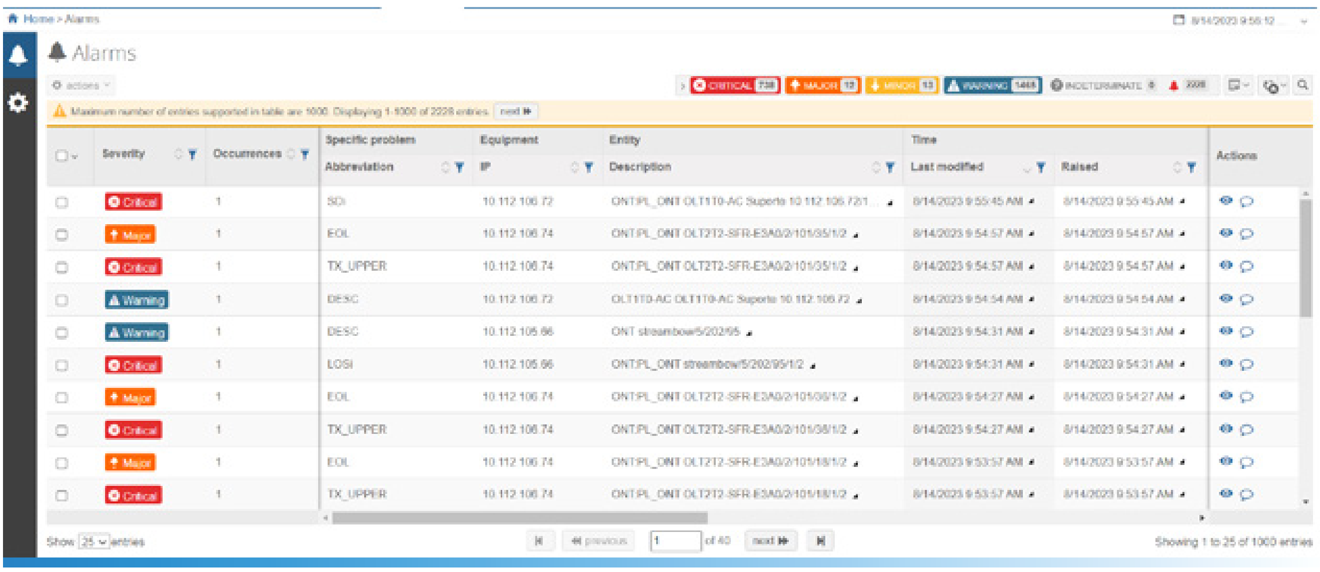

Figure 1 – Agora Alarm Manager

Through Alarm Manager, organizations can quickly respond to changing needs, enhancing operations, and agility.

Data Mining

With the rapid advancements in alarm monitoring and control systems, the volume and complexity of generated data have increased. This data escalation poses challenges for network operators who must quickly process and interpret it [4][13]. Data mining techniques have emerged as a solution to this issue.

Alarm data mining involves extracting valuable insights and patterns from alarm data across diverse industries such as manufacturing, telecommunications, and energy [9][14]. These alarm systems are pivotal for alerting operators about abnormal conditions. By analyzing large alarm datasets, alarm data mining seeks to reveal patterns, trends, anomalies, and relationships. This approach aims to enhance the comprehension of alarm behavior, optimize alarm settings, and bolster the efficiency of alarm management systems [15]. Through the utilization of data mining, organizations can gather insights into system behavior, streamline operations, and improve decision-making related to alarm management [8][16][17].

The objectives of alarm data mining include:

Alarm optimization: Identifying redundant or nuisance alarms, reducing false positives, and prioritizing critical alarms. This ensures operators can focus on vital tasks

Fault detection and diagnosis: Detecting abnormal conditions and diagnosing the root causes of alarms, facilitating quick issue resolution.

Performance enhancement: Analyzing alarm response times, resolution rates, and operator actions to enhance system reliability and operator efficiency.

Predictive analytics: Utilizing historical alarm data to build predictive models that anticipate potential failures or abnormal events, enabling proactive maintenance actions.

Pursuing these goals empowers organizations to optimize their alarm management systems, streamline operations, and make informed decisions. By unveiling hidden patterns and correlations, operators can effectively manage alarms in intricate network environments [8].

Data mining employs diverse techniques like statistical analysis, data visualization, machine learning, pattern similarity, and pattern mining algorithms [5][9][14][18]. In this study, the primary focus is on pattern matching AF sequences, using the Smith-Waterman Algorithm [4][8][19] for comprehensive analysis. This approach promises to enhance the understanding of AFs, contributing to more effective root cause analysis and system optimization.

Pattern Matching

Pattern matching entails the recognition and examination of repetitive patterns or alarm sequences within a system or network. Alarmistic data mining aims to unveil valuable connections and correlations from extensive alarm data produced by intricate systems [4][7][16] like telecommunication networks, industrial operations, or fiber optic networks.

In the context of alarmistic data mining, pattern matching encompasses the quest for particular alarm sequences that might signify shared problems or origins of issues within the system[7][9]. This identification of recurrent patterns empowers network operators and analysts to glean crucial insights into system behavior and to pinpoint possible anomalies or irregular states.

Literature

In existing literature, methods for pattern matching have been extensively employed in the analysis of AFs similarities. These techniques are designed to align and match alarm sequences, revealing hidden correlations and patterns within the data [4][6][8][16]. Among these methods, the Dynamic Time Warping (DTW) algorithm is notable [13][17]. DTW enables flexible alignment of sequences, accommodating time variations and distortions, rendering it suitable for time-series data like alarm sequences [13][17].

Another widely recognized sequence alignment algorithm applied in AF analysis is the Smith-Waterman algorithm. Initially developed for bioinformatics [19], it has been adapted to handle the temporal aspect of AFs [4][8]. This algorithm computes the similarity score between two alarm sequences, considering both the sequence order and their temporal relationships. Its effectiveness in identifying similar alarm patterns and subsequences has established it as a benchmark in the field [4][8][19].

Besides DTW and Smith-Waterman, other techniques such as the Basic Local Alignment Search Tool (BLAST) algorithm [18]and match-based accelerated alignment algorithms [5][6][16] have also been used for AF similarity analysis. These methods contribute to further identifying recurring alarm patterns and uncovering latent patterns in large-scale alarm datasets.

Pattern matching methodologies have proven invaluable in identifying and eliminating redundancy within alarm data, leading to more efficient alarm rationalization. By aligning alarm sequences, operators can gain insights into the temporal patterns of alarms, thus enhancing their understanding of network dynamics [4][7][15]. This helps in pinpointing root causes and streamlines the troubleshooting process for network issues.

Similarity of Alarm Floods

The AF pattern matching issue centers on gauging similarity between segments and categorizing floods accordingly. An AF is a sequence of alarms within a time frame, and the goal is to unearth shared patterns among them for differentiation [4][7][9][14].

Through a similarity index, we quantify the likeness between sequences. If floods share substantial alarms or exhibit akin patterns, they're considered alike and grouped as such [4][9]. These common segments act as distinguishing features to classify incoming floods.

Essentially, the problem involves identifying flood similarities, grouping based on these, and using shared segments to classify new floods. This process detects and categorizes floods, bolstering analysis and decision-making [7].

The next chapter explores data processing methods for efficiency, alongside a possible algorithmic approach for Pattern Matching of AF Sequences, inspired by the Smith-Waterman Algorithm.

Modified Smith-Waterman Algorithm

The Smith-Waterman algorithm identifies pairs of segments with the highest similarity within two sequences. Initially designed for molecular sequences, its applications have expanded to analyzing industrial alarm sequences [4][8].

This algorithm focuses on local alignment, pinpointing where the highest similarity occurs between sequences. Yet, its limitation lies in disregarding the temporal order of alarms, potentially leading to misalignment and inaccurate results.

To overcome this, a modified version tailored for alarmistic data mining can consider temporal order during alignment [4][8]. By incorporating timestamps, alarms can align based on temporal proximity, ensuring accurate matching of similar alarms.

This modified approach includes pre-processing to incorporate time information into sequence representation. It accommodates variations in reporting order and identifies alarm pairs with significant temporal differences, enriching the understanding of alarm patterns and relationships.

Data from Altice Labs' alarm manager, capturing fiber optic network alarms, is used. A 7-day dataset with over 5.5 million alarms is employed to develop and assess the algorithm's real-world effectiveness. Pre-processing techniques are crucial to efficiently manage this substantial data volume.

Pre-Processing

In data analysis and machine learning, pre-processing is vital for accurate results. It converts raw data into a suitable format for algorithms [4][15][20][21]. This includes feature selection, addressing chattering and arranging data into temporal sequences.

Feature selection identifies key features that impact the target variable, reducing complexity and enhancing model performance. This study emphasizes alarm correlations, leveraging features like the location, timing, and nature of the problem for a multidimensional analysis, thereby enhancing precise issue identification.

Chattering, caused by errors or noise, disrupts model accuracy. Techniques like smoothing mitigate this. Effective chattering detection optimizes real-time RCA [20][21][22], crucial for applications requiring prompt responses.

Around 2.7 million alarms (49%) were successfully removed by the chattering algorithm in the analyzed dataset, highlighting its repetitive nature.

Organizing data into time-based segments enhances predictive power. Guided by the ISA18.2 standard [23], alarms are grouped into AFs based on a set threshold of alarm occurrences per time interval. This pre-processing technique enables focused analysis of closely related alarm sequences, enhancing understanding of underlying issues.

Proposed algorithm

This innovative approach extends the principles of the Smith-Waterman Algorithm, incorporating temporal information. Time integration provides a nuanced interpretation of alarm sequences [4][8]. For instance, when two alarms occur closely, this method treats them interchangeably, accommodating variations in reporting order.

Furthermore, it distinguishes between alarms with significant temporal gaps, indicating less correlation. This approach offers richer insights into alarm patterns and relationships, enhancing complex system alarm management accuracy.

To gain a better understanding of the algorithm, let's consider a scenario where we have a set of alarms, denoted by K, and define a time-stamped AF as follows:

$$A = < a_1, a_2, \cdots, a_M >$$

$$a_m = (e_m, t_m), \qquad m = 1, 2, \cdots, M$$

Here, e_m represents the alarm type from the set K, and t_m represents the corresponding timestamp of the alarm occurrence for each alarm am.

To explain the new method of calculating the similarity score for pairs of time-stamped alarms, it is necessary to calculate the temporal distance between each alarm in relation to their time stamps. This is achieved by introducing the concept of a time distance vector for each AF:

$$d_m = [d_m^1, d_m^2, \cdots, d_m^K]^T$$

$$d_m^k = \begin{cases} \min\limits_{1 \le i \le M} \{|t_m - t_i| \ : \ e_i = k\}, & \textit{if the set is not empty} \\ \infty, & \textit{otherwise} \end{cases}$$

Each entry (d_m^k) in the time distance vector (d_m) provides valuable information about the time gap between the m-th alarm and the nearest alarm of type k on the time axis. If there are no alarms of type k present in the alarm sequence (i.e., none of the alarms have the alarm type k), the corresponding time gap is considered infinite ∞.

It's important to note that the m-th alarm itself has the alarm type e_m, so the corresponding entry (d_{em}^m) in the time distance vector is naturally zero since the time gap between an alarm and itself is zero.

Distances are then transformed into time weights by applying a monotonically decreasing function, for example, a scaled Gaussian function.

$$f(x) = e^{-\frac{x^2}{2\sigma^2}}$$

By adjusting σ, sensitivity to temporal proximity can be fine-tuned. Higher σ values capture broader time gaps, while lower values focus on immediate relationships.

Time weight vector (W_m) captures influence based on temporal proximity:

$$W_m = [w_m^1, w_m^2, \cdots, w_m^K]^T$$
$$W_m = [f(d_m^1), f(d_m^2), \cdots, f(d_m^K)]^T$$

Now, let's define the similarity score for a pair of time-stamped alarms $s((e_a, t_a), (e_b, t_b))$. In the original Smith-Waterman algorithm, the similarity score function only had two possible values: a match score of 1, indicating that the alarms are similar, or a mismatch penalty denoted as μ, indicating a dissimilarity between the alarms.

The modified algorithm introduces a flexible approach for calculating similarity scores between alarms (e_a, t_a) and (e_b, t_b). A weighted linear combination of match and mismatch components determines the score, with weights based on time weight vectors (W_a, W_b).

$$s\big((e_a, t_a), (e_b, t_b)\big) = \max_{1 \le k \le K}\left[w_a^k \times w_b^k\right](1 - \mu) + \mu$$

The similarity index constructs the H matrix, similar to the original Smith-Waterman algorithm. Elements represent positive similarity indexes, aiding in segment matching. The value of $H_{p+1,q+1}$ can be recursively computed using the following equation, considering a uniform penalty δ , when a gap is inserted into the sequence:

$$H_{\mathrm{p+1,q+1}} = max\left\{H_{\mathrm{p,q}} + s\big((e_a, t_a), (e_b, t_b)\big), H_{\mathrm{p,q+1}} + \delta, H_{\mathrm{p+1,q}} + \delta, 0\right\}$$

After obtaining the complete matrix, the similarity index is represented by the maximum value in the matrix, and the optimal local alignment is obtained through a backward path search starting from that element.

Normalization ensures fair comparisons between sequences of different lengths. The normalized similarity score ranges from 0 to 1, reflecting pattern similarity rather than sequence length.

This approach refines sequence matching, incorporating temporal nuances for improved alarm analysis in complex systems.

Post-processing

After calculating similarity indexes among AFs, post-processing, and analysis methods are crucial for refining, enhancing, and transforming data output [6][8][16]. To efficiently manage AF analysis, hierarchical clustering is employed. This technique brings together similar AFs, enabling deeper insights.

Hierarchical clustering gathers data points into clusters based on similarity, which facilitates pattern identification [24][25]. The process involves transforming similarity values into a distance matrix, essential for clustering. Each AF initially forms an individual cluster. Clusters with the smallest distance are continuously merged, halting when no clusters are within a defined maximum distance.

The Complete-linkage method determines the distance between merged clusters based on the largest distance within the merged cluster [25]. Clusters maintain high similarity, ensuring that coherent patterns emerge.

A visual representation of the clustering process is depicted in **Figure 2** (on the next page).

Archetype AFs are selected to represent clusters, reducing computational load. The medoid method, selecting an actual AF with the smallest distance to others in the cluster [26][27][28], is chosen for complex cases.

Using this approach, 25 distinct clusters are obtained. Each archetype signifies various problem types with distinct root causes. This showcases the model's ability to uncover diverse issues, aiding efficient problem diagnosis and resolution for network operators.

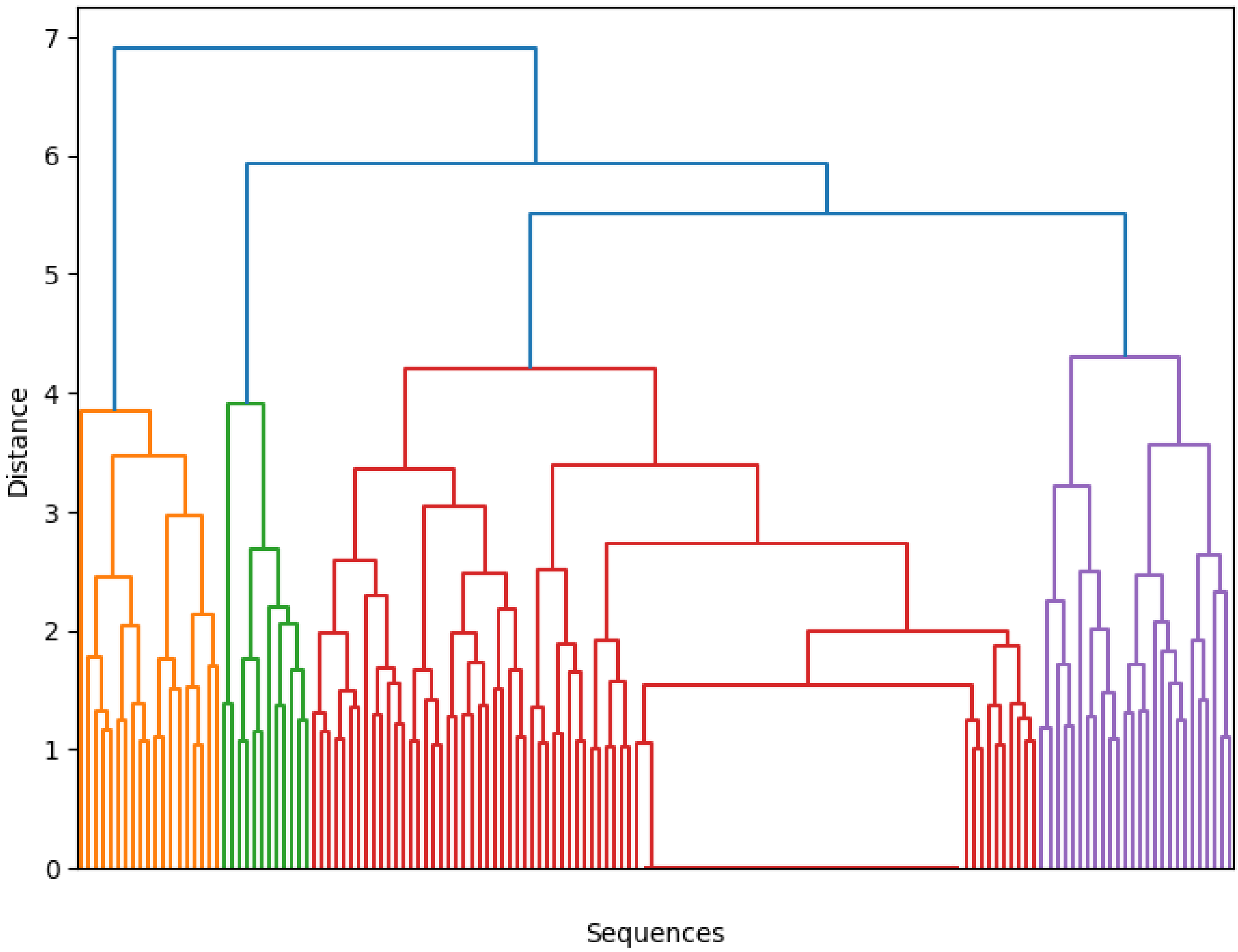

Figure 2 – Hierarchical clustering representation

Test model

Following the selection of archetype AFs from training clusters, the subsequent phase involves analyzing AFs within a validation dataset. Instead of comparing each AF with all others, they are matched against the training-generated archetypes using the proposed similarity algorithm.

After computing similarities, we observed that 542 AFs within the test dataset exhibited a similarity index of 0.6 or higher with at least one cluster archetype. This accounts for 71.3% of AFs tied to known clusters. This outcome highlights that a substantial portion of the new dataset encompasses alarm sequences already represented by archetypes. As a result, these sequences can be recognized as familiar AFs, allowing network operators to concentrate resources on deciphering the root causes of the remaining unfamiliar patterns. By identifying known AFs, operators can efficiently allocate resources, prioritize analysis of novel alarm patterns, and expedite root cause identification.

Conclusion

Seeking the root cause of each alarm or alarm sequence by involving personnel familiar with the platform would prove to be highly time-consuming and resource-intensive. This proposed algorithm brings the advantage of automatically identifying alarm patterns and grouping them according to their similarities, streamlining the process of root cause analysis. As a result, a substantial number of newly reported alarms can be promptly linked to a pre-existing known root cause. Consequently, for novel and previously uncharted cases, a thorough investigation can be carried out to unveil their underlying origins.

This research leverages the modified Smith-Waterman Algorithm to enhance the precision and efficiency of pattern matching within AF sequences. The algorithm's proficiency in local alignments as well as its sensitivity to gaps and discrepancies makes it particularly capable of capturing intricate patterns and subtle variations in alarm occurrences.

The study not only delves into theoretical exploration but also practically applies the algorithm. This extends to adapting the algorithm to address the specific characteristics of alarm data, including timing and contextual information. This ensures its relevance and efficacy within the domain of AF analysis.

By using the Smith-Waterman Algorithm and its tailored adaptations, this study contributes to the advancement of pattern matching methodologies for AF sequences. The outcomes from this research have the potential to enrich alarm management strategies, elevate fault detection and diagnosis capabilities, and ultimately optimize system performance and dependability.

In the realm of prospects, several pivotal tasks can amplify the capabilities of the developed system. An immediate focus lies in creating an online model for AFs similarity, which can efficiently correlate novel AFs with pre-trained clusters in real-time or near-real-time. This dynamic model empowers network operators to efficiently scrutinize unfamiliar alarm sequences while the system handles routine correlations.

Further refining the distance calculation between AFs emerges as a priority, reducing the processing time for new AFs training. This refinement fosters heightened scalability, enabling the system to accommodate substantial data volumes while ensuring agility and robustness.

Incorporating multi-threading stands as a strategic move to expedite processing, elevating the overall analysis speed. This acceleration translates to quicker outcomes, bolstering the efficacy of RCA-related decisions.

Lastly, rigorous real-world testing and validation across diverse datasets and network landscapes are imperative. This process validates the model's versatility and efficiency across varied operational contexts. By perpetually refining the model through new data and insights, organizations can uphold a proactive RCA approach, thus augmenting system reliability and adaptability.

References

[1] K. B. Percarpio, B. V. Watts, and W. B. Weeks, “The Effectiveness of Root Cause Analysis: What Does the Literature Tell Us?,” *The Joint Commission Journal on Quality and Patient Safety*, vol. 34, no. 7, pp. 391–398, 2008, doi: https://doi.org/10.1016/S1553-7250(08)34049-5.

[2] S. O. Al-Mamory and H. Zhang, “Intrusion detection alarms reduction using root cause analysis and clustering,” *Comput Commun*, vol. 32, no. 2, pp. 419–430, 2009, doi: https://doi.org/10.1016/j.comcom.2008.11.012.

[3] J. Wang et al., “Root-cause analysis of occurring alarms in thermal power plants based on Bayesian networks,” *International Journal of Electrical Power & Energy Systems*, vol. 103, pp. 67–74, 2018, doi: https://doi.org/10.1016/j.ijepes.2018.05.029.

[4] B. Zhou, W. Hu, K. Brown, and T. Chen, “Generalized Pattern Matching of Industrial Alarm Flood Sequences via Word Processing and Sequence Alignment,” *IEEE Transactions on Industrial Electronics*, vol. 68, pp. 10171–10179, Aug. 2021, doi: 10.1109/TIE.2020.3026287.

[5] W. Hu, J. Wang, and T. Chen, “Fast Sequence Alignment for Comparing Industrial Alarm Floods**This work was partially supported by an NSERC CRD project, and the National Natural Science Foundation of China under grant No. 61061130559.,” *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 647–652, 2015, doi: https://doi.org/10.1016/j.ifacol.2015.09.041.

[6] F. Zhang, X.-Z. Qiao, and Z.-Y. Liu, “A parallel Smith-Waterman algorithm based on divide and conquer,” in *Fifth International Conference on Algorithms and Architectures for Parallel Processing, 2002. Proceedings*., 2002, pp. 162–169. doi: 10.1109/ICAPP.2002.1173568.

[7] W. Hu, J. Wang, and T. Chen, “A local alignment approach to similarity analysis of industrial alarm flood sequences,” *Control Eng Pract*, vol. 55, pp. 13–25, 2016, doi: https://doi.org/10.1016/j.conengprac.2016.05.021.

[8] Y. Cheng, I. Izadi, and T. Chen, “Pattern matching of alarm flood sequences by a modified Smith–Waterman algorithm,” *Chemical Engineering Research and Design*, vol. 91, no. 6, pp. 1085–1094, 2013, doi: https://doi.org/10.1016/j.cherd.2012.11.001.

[9] S. Lai, F. Yang, T. Chen, and L. Cao, “Accelerated multiple alarm flood sequence alignment for abnormality pattern mining,” *J Process Control*, vol. 82, pp. 44–57, 2019, doi: https://doi.org/10.1016/j.jprocont.2019.06.004.

[10] R. Costa, N. Cachulo, and P. Cortez, “An Intelligent Alarm Management System for Large-Scale Telecommunication Companies,” Aug. 2009, pp. 386–399. doi: 10.1007/978-3-642-04686-5_32.

[11] J. Zupan and D. Medhi, “An alarm management approach in the management of multi-layered networks,” in *Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IPOM 2003) (IEEE Cat. No.03EX764)*, 2003, pp. 77–84. doi: 10.1109/IPOM.2003.1251227.

[12] P. Thorat and N. Kumar Dubey, “SDN-based Machine Learning Powered Alarm Manager for Mitigating the Traffic Spikes at the IoT Gateways,” in ***2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)***, 2020, pp. 1–6. doi: 10.1109/CONECCT50063.2020.9198356.

[13] D. J. Berndt and J. Clifford, “Using Dynamic Time Warping to Find Patterns in Time Series,” in ***Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining***, in AAAIWS’94. AAAI Press, 1994, pp. 359–370.

[14] M. Lucke, M. Chioua, C. Grimholt, M. Hollender, and N. F. Thornhill, “Advances in alarm data analysis with a practical application to online alarm flood classification,” ***J Process Control***, vol. 79, pp. 56–71, 2019, doi: https://doi.org/10.1016/j.jprocont.2019.04.010.

[15] S. Lai and T. Chen, “Methodology and Application of Pattern Mining in Multiple Alarm Flood Sequences**This work was supported by an NSERC CRD project.,” ***IFAC-PapersOnLine***, vol. 48, no. 8, pp. 657–662, 2015, doi: https://doi.org/10.1016/j.ifacol.2015.09.043.

[16] S. Lai, F. Yang, and T. Chen, “Online pattern matching and prediction of incoming alarm floods,” ***J Process Control***, vol. 56, pp. 69–78, 2017, doi: https://doi.org/10.1016/j.jprocont.2017.01.003.

[17] S. Salvador and P. Chan, “Toward Accurate Dynamic Time Warping in Linear Time and Space,” in ***Intelligent Data Analysis***, Aug. 2004, pp. 70–80.

[18] Y. Xu, W. Tan, and T. Li, “An alarm flood pattern matching algorithm based on modified BLAST with leveshtein distance,” in ***2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)***, 2016, pp. 1–6. doi: 10.1109/ICARCV.2016.7838817.

[19] T. F. Smith and M. S. Waterman, “Identification of common molecular subsequences,” J Mol Biol, vol. 147, no. 1, pp. 195–197, 1981, doi: https://doi.org/10.1016/0022-2836(81)90087-5.

[20] J. Wang and T. Chen, “An online method for detection and reduction of chattering alarms due to oscillation,” ***Comput Chem Eng***, vol. 54, pp. 140–150, 2013, doi: https://doi.org/10.1016/j.compchemeng.2013.03.025.

[21] E. Naghoosi, I. Izadi, and T. Chen, “Estimation of alarm chattering,” ***J Process Control***, vol. 21, no. 9, pp. 1243–1249, 2011, doi: https://doi.org/10.1016/j.jprocont.2011.07.015.

[22] J. Wang and T. Chen, “An online method to remove chattering and repeating alarms based on alarm durations and intervals,” ***Comput Chem Eng***, vol. 67, pp. 43–52, 2014, doi: https://doi.org/10.1016/j.compchemeng.2014.03.018.

[23] B. Hollifield, “Understanding and Applying the ANSI/ ISA 18.2 Alarm Management Standard,” 2010. [Online]. Available: http://www.iec.ch/cgi-bin/procgi.pl/www/iecwww.p?wwwlang=e&wwwprog=pro-det.p&progdb=db1&He=IEC&Pu=62682&Pa=&Se=&Am=&Fr=&TR=&Ed=1.0.

[24] F. Nielsen, “Hierarchical Clustering,” in ***Introduction to HPC with MPI for Data Science***, Cham: Springer International Publishing, 2016, pp. 195–211. doi: 10.1007/978-3-319-21903-5_8.

[25] A. Jarman, “Hierarchical Cluster Analysis: Comparison of Single linkage,Complete linkage, Average linkage and Centroid Linkage Method.” Aug. 2020. doi: 10.13140/RG.2.2.11388.90240.

[26] V. R. Wingate, F. O. Akinyemi, and C. I. Speranza, “Archetypes of remnant West African forest patches, their main characteristics and geographical distribution,” Applied Geography, vol. 158, p. 103024, 2023, doi: https://doi.org/10.1016/j.apgeog.2023.103024.

[27] D. Sisodia, “Clustering Techniques: A Brief Survey of Different Clustering Algorithms,” Aug. 2012.

[28] H. Menéndez, F. Barril Otero, and D. Camacho, “Medoid-based clustering using ant colony optimization,” ***Swarm Intelligence***, vol. 10, Aug. 2016, doi: 10.1007/s11721-016-0122-5.

Acronyms

AF	Alarm Flood
BLAST	Basic Local Alignment Search Tool
DTW	Dynamic Time Warping
RCA	Root Cause Analysis

Authors

Lucas Leite Baptista

Junior Tester

Altice Labs, Portugal

lucas-l-baptista@alticelabs.com

https://www.linkedin.com/in/lucas-baptista

Contacts

Address
Rua Eng. José Ferreira Pinto Basto
3810 - 106 Aveiro (PORTUGAL)

Phone
+351 234 403 200

Media
contact@alticelabs.com
www.alticelabs.com