POLITECNICO DI TORINO
Repository ISTITUZIONALE

Mining Patterns in Mobile Network Logs

Original
Mining Patterns in Mobile Network Logs / Zargarian, Golnaz; Vassio, L; Munafo, Maurizio; Mellia, M. - STAMPA. - (2019), pp. 1-6. (Intervento presentato al convegno 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) tenutosi a Arlington, VA (USA) nel Aprile 2019).

Availability:
This version is available at: 11583/2800192 since: 2020-03-16T13:15:01Z

Publisher:
IEEE

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

16 April 2024

Mining Patterns in Mobile Network Logs

Golnazsadat Zargarian
Politecnico di Torino, Italy
golnazsadat.zargarian@studenti.polito.it

Luca Vassio
Politecnico di Torino, Italy
luca.vassio@polito.it

Maurizio M. Munafò
Politecnico di Torino, Italy
maurizio.munafo@polito.it

Marco Mellia
Politecnico di Torino, Italy
marco.mellia@polito.it

***Abstract*—Alarm logs are a valuable source of information and play a crucial role in network management. Network devices such as backbone routers or 3G/4G base stations generate verbose and detailed logs that network managers process to detect problems and identify their root causes. Manual analysis of such logs is extremely time-consuming because of the extensive amount of data. Therefore, finding suitable automatic methods to process logs is an important problem in the network analysis area.**

In this paper, we target the automatic extraction of *situations*, i.e., sequences of events occurring close in time and space which identify common and recurring patterns. We adopt an unsupervised machine learning approach to automatically mine logs and provide information and correlations in network failures. We face a real use case processing more than 2 million alarms generated by 2 months of TIM Network Operations Center in Northern Italy. Most of the features are categorical and call for specific methodologies to process them. We choose rule mining of frequent items. We focus on event logs and apply rule mining methods to extract temporal-spatial correlations and co-occurrences, i.e., situations. To ease the analyst work, we highlight the most important rules and offer visualization techniques in both spatial and temporal dimensions. Results have been verified to be helpful to recognize common situations and identify possible future anomalies.

***Index Terms*—alarm logs, anomaly detection and prediction, monitoring and measurements for management, telecommunication network, predictive and real-time analytics.**

I. Introduction

Studying alarm logs is increasingly becoming a vital factor for improving the performance complex systems such as computer networks. In general, servers and network devices such as routers, 3G/4G Base Transceiver Stations (BTSes), and Mobile Switching Centers (MSCs) generate logs of events containing alarms, warnings, or simple notifications. For a mobile operator with tens of thousands of devices from different vendors and technologies, processing such logs is at the same time both vital to managing the network, and terribly costly and time-consuming.

Clearly, the volume and variety in the data challenge the automatic extraction of useful information. Operators have so far developed custom expert systems that mimic the manual process the network expert follows – see related work for a detailed summary. These solutions are complex and custom and have to be devised for different contexts. Statistical and data science approaches are thus becoming appealing in the area of the log network analysis, with the goal to automatically extract knowledge from the raw data.

In this paper, we leverage unsupervised machine learning techniques to mine data logs and automatically provide meaningful information about possible network problems. We consider a real use case - with data collected from the network operations center of the telecommunications company TIM[1] for their entire 3G/4G mobile network in Northern Italy. Data includes all the alarms raised during 2 months in 2017, involving more than 65 000 devices, and reporting more than 1 million events per month in consolidated logs. Each entry has several fields which range from timestamps to vendor identification, from alarm type to software version. Our goal is to detect common patterns in the data, considering both the temporal and spatial dimensions. We call *situations* these common patterns.

Since most of features in the logs are categorical, it is hard to define a distance measure for applying clustering algorithms. We thus opt for association rule mining on frequent items [1]. Originally developed for market basket analysis, the objective is to extract actionable knowledge from the vast features of transactional databases. In a nutshell, they are designed to extract common patterns that emerge from the data.

Ingenuity must be adopted to apply rule mining in our context. Our adopted work-flow is shown in Fig. 1. The first step is to define the time and spatial granularity that we are interested to study. Events generated by devices which are close each other, and occur close in time are indeed a possible symptom of a major situation. Unfortunately, the logs contain only a coarse timestamping, with precision limited to the minute granularity. This limits thus the application of sequence pattern mining, since a lot of events are artificially co-occurring in our case.

Here, we define a transaction by aggregating all events generated by devices in the same automatically found spatial region, and occurring within a given time interval. Transactions are thus like customers receipts that list which goods (alarm) the customers (network devices) bought (event raised).

Given then a set of transactions, we look for common events. Here, we rely on association rule mining. Those have been proposed in similar context in the past (such as [2], [3]). In our case, the much larger volume of data and the lack of precise timestamping calls for ingenuity to guide the algorithm in finding interesting rules out of the noisy and bloated logs.

We apply the methodology considering four separated datasets, considering alarms raised in the Turin or Milan area, and in May or September 2017. We automatically highlight

[1] https://www.tim.it/

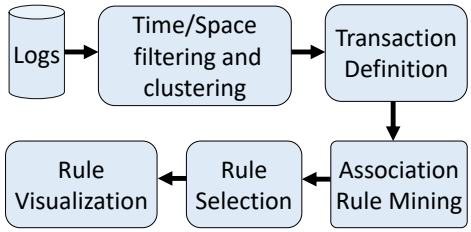

Fig. 1: Followed work-flow for mining patterns in the analyzed TIM network.

interesting rules based on their frequency and peculiarity. Some of these occur in all datasets, highlighting recurrent and common situations. These are presented to the network analyst using proper visualization techniques that let the expert gauge both the spatial and temporal dimensions, along with all details. This actionable information can then be used to properly manage the situation.

In the following, Sec. II reports the main related works in the literature. In Sec. III we discuss the characteristics of the datasets. Sec. IV is devoted to the methodology, in particular pattern mining, association rules and our definition of matrix of transaction and itemsets. Sec. V and VI reports the results by considering rules for each device and each device type, respectively. Finally, Sec. VII concludes the paper.

II. Related Literature

The first works that try to find correlations from alarm logs are [4], [5]. Both works simply consider logs generated by network devices and design a semi-automatic approach using knowledge-based systems where rules must be manually defined by experts in advance. Unfortunately, when there are too many pieces of uncleared relations, the big picture of the network remains vague, with no ability to automatically identify new patterns.

Authors of Telecommunication Alarm Sequence Analyzer (TASA) [6], [7] considered GSM networks, as they were in 1996. They propose an iterative process, containing data collection, pattern discovery, and post-processing. They look for sequences of events, trying to find the first one that led to the cascade of events. In practice, TASA provides an overwhelming amount of rules, presenting all possible combination of alarms. In our work, we face much richer and heterogeneous alarm logs that call for more fine-grained means to extract correlation. Also, we miss accurate timestamping which makes sequencing analysis useless. At last, authors in [8] and [9] also focus on sequential pattern mining. They assume events follow an exact sequential pattern in time. In our work, we cannot rely on fine-grained timing information, and we thus need to take alternative directions.

Among the first to introduce rule mining in the networking field, authors of [2] consider logs generated by servers in a data center. Their focus is on the algorithm scalability, with little interest in the actual insight provided to the system administrator. Here we use well-established algorithms and focus on the overall system design and information exploitation, including visualization modules. Similar in spirit is the work in [3], where both text mining and rule mining are proposed to digest and summarize router logs, which are much more structured than entries generated by mobile network devices, like the one we face here.

Authors in [10] investigated failure detection by using a clustering technique based on text mining. The proposed model constructs clusters by grouping together events on the basis of their message characters and detects anomalies by tracking events which do not belong to any existing cluster. These methods are orthogonal to ours and suffer from the difficulties of processing unstructured text as log events typically are.

Other works focus on the identification of the root cause of a problem. In [11] the authors estimate the likelihood of a node producing an impacting outage. In [12] the authors build a graph to model event correlation and apply causal inference approaches to find the root cause of a sequence of events. They use manually generated templates to normalize log entries, extract a sequence of events and build the graph from which they extract root nodes of all directed acyclic graphs. Our approach is different: we process events without normalizing them, and we only assume a course time dependency among entries.

Authors of [13] propose a spatio-temporal factorization method, which automatically learns underlying network events from unstructured logs. They regard network log data as a tensor of location, time and textual information, and extract template text and relationships that are likely to co-occur. Although such approach can be more complete and detailed with respect to our work, we do not analyze the text of the alarms, nor model the events through a matrix factorization, ending up with a faster and more scalable methodology.

III. Datasets

We rely on datasets collected by TIM network operations center which controls the state of the mobile 3G/4G network and manage anomalous situations to ensure operations.

Collected data includes alarms for Northern Italy during the months of May and September 2017, where network devices produce thousands of alarms daily. Our goal is to reduce the burden of human operators by presenting together alarms that occur close in time and space, i.e., that form a situation.

Each alarm has hundreds of fields. We focus on the most important features, according to the domain experts, to consider only relevant information and exclude fields that are mostly empty or containing automated messages with little information. This reduces the set of fields to 27 features. After a phase of data characterization and analysis, in cooperation with the field experts, we select the most relevant ones:

- **Network Equipment ID (NeID)**: uniquely identifies the network device which has generated the alarm. It derives from the concatenation of three fields:
  - **Technology**: specifies network device technology (GSM, UMTS, or LTE) and its frequency band.

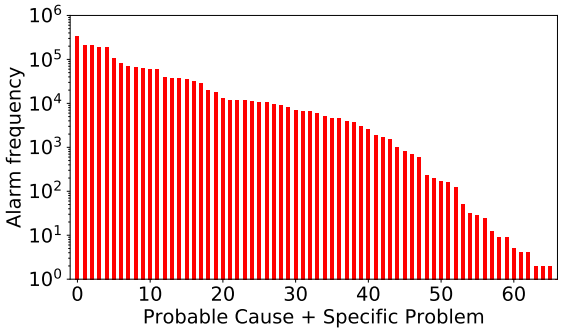

Fig. 2: Ranked alarm frequency of different Probable Causes and Specific Problems generated in Northern Italy in May 2017.

  - **Equipment**: specifies what kind of equipment NeID is. Its possible values are Base Station Controller (BSC), Radio Network Controller (RNC), or Base Transceiver Station (BTS).
  - **Site**: is the site where the device is located. Different devices might belong to the same site. The site includes the abbreviation for the province of the site.

  For instance, GBSCTO033 represents a GSM (G) BSC, located in site 033 in Turin, and 1BTSGE121 represents a LTE 1800 (1) BTS located in site 121 in Genoa.
- **Timestamp**: first instant this alarm is fired. The alarm remains active until it is solved. Due to the log aggregation process, timestamp has a minute granularity.
- **Probable Cause**: text string with the primary cause of the alarm. The set of possible *Probable Causes* is different from vendor to vendor. In the months of the analyses we observed tens of different Probable Causes.
- **Specific Problem**: text with secondary cause of the alarm. We observe hundreds of *Specific Problem* strings. We consider this field in case the Probable Cause field is too generic (see later).
- **Site Coordinates**: the longitude and latitude of the network device site.

Since there are quite different types of vendors, systems, and software releases, the available data is very heterogeneous in format. Furthermore, the temporal granularity of the alarms, which are aggregated at the precision of minutes, creates a large number of simultaneous events.

The whole dataset contains 2 033 678 events. We observe 10 773 distinct devices that belong to 4 130 different sites.

Since we target the identification of common and recurrent situations, we opt to use the Probable Cause as the main features describing the event. Indeed, the Specific Problem categories result too fine-grained to let common pattern emerge.

Observing the frequencies of the Probable Causes, the two most frequent turns out to be *Indeterminate* and *Unavailable*, that clearly carry very little information. To overcome this limit, we opt to detail these two events by using the Specific Problem field. In summary, we obtain 64 different alarm types, defined as a mix of Probable Causes and Specific Problems using domain knowledge.

Figure 2 depicts final ranked frequencies of the alarms in Northern Italy. Notice the log scale on the y-axis. The distribution is very skewed, with some alarms that are very frequent, with up to 300 000 occurrences, but also with a long tail of rare events.

IV. Pattern Discovery

We analyze the data by means of association rule mining, popularly used for *Market Basket Analysis* [1] by large retailers to understand customers purchases. The main objective is to extract actionable knowledge and co-occurrences from features of transactional databases.

A. Frequent pattern mining and association rules

Mining frequent itemsets to extract common patterns is one of the backbones of research in data mining area [14], [15]. Consider the set I of all possible items. A *transaction* $i \subseteq I$ is a subset of items for an event. Transaction database T is the set of all transactions the system has processed within a given time period. Considering the transaction database T, an *itemset* is any subset of any transaction $i \in T$. The *support* of an itemset is the fraction of all transactions containing that particular itemset. An itemset is called *frequent* if its support is greater than or equal to a threshold s. The order of an itemset i is its number of elements, i.e., $|i|$. For a given support value s, the frequent itemset with the highest order is said to be *closed*. Frequent closed itemsets are called *patterns*.

Given a database of transactions T, we want to determine all patterns P that are present in at least a fraction s of the transactions. Looking for all itemsets is an NP-hard problem [16]. In practice, there are algorithms that can efficiently compute patterns.

Association rules are strongly linked to frequent patterns. Association rules are widely used to identify frequent patterns themselves, associations and correlations among itemsets, usually enriched with measures of interestingness [17]. We follow the methodology introduced in [1] for mining association rules in large datasets.

A rule is defined as an implication of the form $x \Rightarrow y$, where $x, y \subseteq I$. Every rule is composed of two different itemsets, x, and y, where x is called antecedent and y consequent. An indication of how often the rule has been found to be true is the *confidence*:

$$\mathit{confidence}(x \Rightarrow y) = \frac{\mathit{support}(x \cup y)}{\mathit{support}(x)}$$

Lift interprets the relevance of a rule and it is defined as:

$$\mathit{lift}(x \Rightarrow y) = \frac{\mathit{confidence}(x \Rightarrow y)}{\mathit{support}(y)} = \frac{\mathit{support}(x \cup y)}{\mathit{support}(x) \cdot \mathit{support}(y)}$$

If the events in x and y are independent and identically distributed (i.i.d.), then $\mathit{support}(x \cup y) = \mathit{support}(x) \cdot \mathit{support}(y)$ and $\mathit{lift}(x \Rightarrow y) = 1$. Instead, the more x and y are correlated, the more $\mathit{lift}(x \Rightarrow y)$ is higher. In a nutshell, if the lift is larger than 1, this lets us know the degree to which those two

occurrences are dependent to each another, and makes this rule potentially useful for predicting the consequent in future data sets.

In order to find the most frequent itemsets, we apply the FP-Growth algorithm [18]. It efficiently calculates all frequently-occurring itemsets, using a data structure known as FP-tree. FP-Growth utilizes a depth-first search and uses a pattern-growth approach, which results in low memory consumption and fast execution time [19]. The output of FP-Growth operator is frequent items which are the suitable input for creating association rules that we compute using the software RapidMiner[2].

To select which rules to present, we select a minimum threshold on support and confidence to focus on rules that are often satisfied. Among the found rules, we analyze the ones with the highest lift.

B. Definitions of transactions

The first step is to define what kind of relations we are interested to study and then define the transactions accordingly. In a transaction database, each row corresponds to a transaction whereas each column indicates a possible item. We will use binary transaction matrices, where the value can be either 0, i.e, representing the absence of an alarm, or 1, i.e., the presence of an alarm. Different definitions of transactions and items have different results and their own advantages and drawbacks.

We want to understand what is likely to happen 1) within close time and 2) in close space. For time aggregation, we define an appropriate time-window where we aggregate alarms in a single transaction. As we shrink the window, we get patterns with a faster dynamic. From domain knowledge, we consider non-overlapping windows of 2 hours. Each transaction then contains the set of alarms observed during an interval of 2 hours.

For space, we decided to cluster the network devices according to the geographical coordinates of their site. For each Italian province in the dataset, we apply the clustering algorithm K-means [20] with K of the same order of magnitude of the number of radio network controllers in that province. Then, we put alarms from devices that belong to different clusters in different transaction matrices. For example, in Fig. 3, we report the positions of all the network elements in the province of Turin and we color them according to the cluster they belong to (result of K-means with $K = 5$).[3]

As a result, each transaction matrix contains alarms that happen in the same region and within 2 hours. For a month, in the province of Turin, we have 5 clustered regions (matrices), each with 372 transactions (rows), that correspond to 12 2-hours long time intervals per day for May 2017.

For defining the items of the matrix, i.e., its columns, we identify two complementary approaches.

[2]https://rapidminer.com/

[3]Unfortunately we do not have the physical topology of the network to cluster devices.

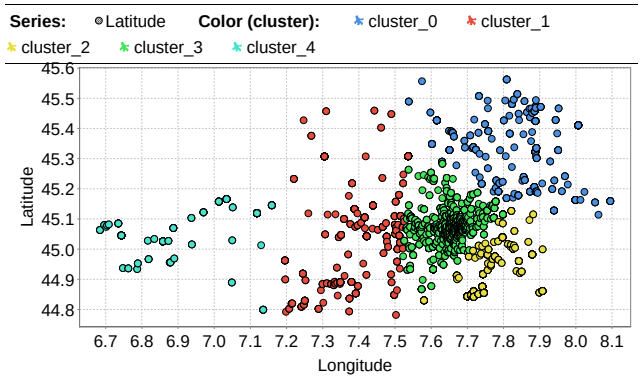

Fig. 3: Example of division in space of Turin province (covering 6 821 km^2). Devices are spatially divided into 5 clusters using K-means algorithm.

First, we want to understand the correlations among specific devices. In the province of Turin, we observe 1 407 devices. We consider each of them as an item, and then the transaction matrix results with 1 407 columns. If a device has fired at least an alarm in the 2 hours of time in a cluster of space, its transaction row will have value 1 for that device. With this, the rules are useful to analyze which are those devices (items) that frequently fire alarms at the same time, in the same region.

In the second approach, we want to understand correlation among alarms and the device type raising them. For this, we use only the device type, i.e., equipment and technology, and we add the alarm type (as defined in Sec. III). In this second case we have 271 combinations of equipment, technology, alarm type, which are the columns of the transaction matrix (we observe 23 combinations of equipment and type, and 64 types of Probable Cause + Specific Problem).

By considering this second type of transactions, we get (possibly less) rules that may be subsequently generalized in other regions.

V. Analysis of patterns of single separated devices

In the first transaction definition we consider each network device as an item and look for specific recurrent patterns involving the same set of devices.

To reduce the heterogeneity of data, we focus on datasets of Milan and Turin provinces, separately, and in May and September, separately. Let us focus first on the province of Turin in May. Table I shows the most frequent itemsets. The GBSCTO033 and GBSCTO034 have support equal to 0.861, meaning that they appear in 86.1% of the transactions in their region (cluster). They appear jointly in 75.5% of cases, and also with GBSCTO0032 in 57.9% of transactions.

Itemsets alone offer little information, since they report only the co-occurrences of alarms. Rule mining instead would provide not only the co-occurrence probability, i.e., the confidence, but also extract the correlation, measured by the lift, allowing us to extract the important itemsets. To avoid extracting too many rules, we select a minimum confidence

TABLE I: 6 most frequent itemsets in Turin, May 2017.

Support	Itemset
0.861	GBSCTO033
0.861	GBSCTO034
0.755	GBSCTO033, GBSCTO034
0.666	UBTSTO109
0.579	GBSCTO032, GBSCTO033, GBSCTO034
0.503	GBSCTO033 UBTSTO109

equal to 0.7 and a minimum support of 0.085. With these parameters, 9 214 rules are extracted for Turin in the May dataset.

We rank these with decreasing lift, and look for closed rules, i.e., we select those rules with the highest number of items, for which the probability of observing the consequent is much higher than an i.i.d. assumption would offer. These rules are marked as situations. We checked the topmost ones with the TIM analysts, who were able to confirm that these were indeed problems that were causally related, some of which were already known, while others were new. In the following, we report a significant example of a rule.

Table II shows the situation with the highest lift: 3 antecedent items are linked to 7 consequent items. The antecedent holds true for 33 time bins. The consequent is present 32 times, leading to a confidence of 97% and a lift of 11.2, a value much higher than 1. To let the analyst investigate the incident, Fig. 4 highlights the strong temporal correlation among alarms network devices generated. Each row represents a device identifier (NeId). A blue dot is reported if that device was firing alarms in that 2-hours time bin. The plot focuses on May 6th, 2017 during which a clear synchronized pattern emerges. Fig. 5 investigates the spatial dimensions. It reports the 10 devices (belonging to 8 sites) on a map. The maximum distance among these devices is 13 km, while the maximum distance within the cluster is much larger, i.e., 40 km. All of the devices involved are BTSes working either with UMTS/800LTE/1800LTE technology. Almost every Probable Cause raised is a Quality of Service alarm "sync reference PDV problem". After analyzing this situation, TIM domain experts confirmed the correlation of these alarms within these devices, confirming as well the situation to be a typical event of a link failure that involves a specific region. Other cases (not reported for brevity) highlighted major events due to the failure of a BSC causing a major outage involving several BTSs, or the loss of synchronization due to NTP server failure.

TABLE II: Example of a situation in Turin with *confidence* = 0.97 and *lift* = 11.2 where 10 devices are involved.

Antecedent	UBTSTO27F, UBTSTO08E, UBTSTO384
Consequent	UBTSTO0B7, UBTSTO14A, 8BTSTO384 1BTSTO0B7, 8BTSTO0B6, 1BTSTO156 1BTSTO00D

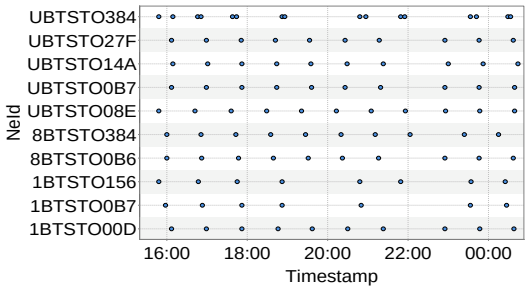

Fig. 4: Scatter plot of timestamps of the example situation involving the 10 devices for the day of May 6th, 2017.

Fig. 5: Geographical location of the 10 devices, in 8 sites (red dots), involved in the example rule. They are all within 13 km.

VI. Analysis of Patterns of Device Type

With the definition of transaction used in Sec. V, we have pinpointed specific correlations among specific device IDs. Here we generalize the approach and consider the second definition of an item which describes both the device type/technology and the alarm type. In order to filter significant rules, we set the minimum confidence again to 0.7, but we lower the minimum support to 0.05. 4 681 rules are extracted for Turin in May 2017.

We let the reader appreciate the expressiveness of these. Fig. 6 shows the scatter plot of the lift vs. the inverse of support, with the red line showing the average lift. Observe how lift grows to values much higher than one – albeit for rules that have limited support (rightmost part of the Figure). In a nutshell, some specific rules exhibit a very high lift, consequently appearing much more frequently than by chance - pinpointing very high correlation. These rules hold true in few time bins, but enough to emerge as frequent patterns. These are the most interesting situations.

As before, we showcase a significant rule - detailed in Table III. This rule has a confidence of 0.9 and a lift of 7. Spatial representation of all single devices of that type that were involved showcases a dense area covering the whole city (not reported for brevity). The domain experts confirm that this correlation is due to a failure of a specific type of BSC device of a specific vendor, that are same for the month of

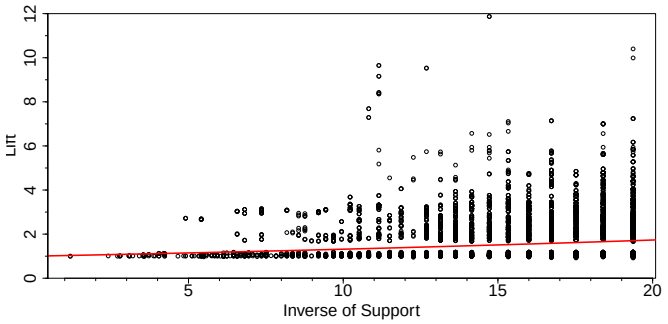

Fig. 6: Scatter plot of lift and inverse of support considering device type and alarm type for Turin in May 2017. Moving average is showed in red.

TABLE III: Example of a situation extracted considering device type, technology and alarm type as items. This rule has *confidence* $= 0.9$ and *lift* $= 7$.

Antecedent	UBTS-equipmentMalfunction GBTS-Cell Logical Channel Availability Supervision GBSC-Data Output AP Transmission Fault UBTS-UtranCell_ServiceUnavailable
Consequent	UBTS-Heartbeat Failure

May and September and are present in both the provinces of Turin and Milan.

As we have decoupled the rule definition from the specific devices, we can now study if rules hold true in different provinces, and at different time periods. For this, we extracted rules considering four separated datasets – in the Turin and Milan province, and in May and September 2017. We then check for the subset of rules that hold true in all four datasets.

We end up with 70 common rules, which are thus generic situations that may hold true at a different time, in different places. The previously described rule is one of these.

In summary, the rule mining approach we propose in this paper results in a nice and flexible tool to analyze logs and automatically identify important situations.

VII. Conclusions

We faced the problem to aid the domain experts by presenting correlation and patterns in verbose alarm logs. For this, we investigated the adoption of rule mining solutions, where ingenuity is required to find a proper definition of items, and transactions.

We prepared the data via data exploration and item definition, including spatial clustering of alarms, before applying frequent pattern mining and association rule mining. We then ordered rules based on lift and closeness to select the most interesting situations. These were verified by the TIM Network Operations Center team, who confirmed that those were indeed significant and recurrent situations, some of which were unknown to them. We believe this is a first step in designing an automatic methodology to extract knowledge from alarm logs and simplify network maintenance.

References

[1] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, 1993.

[2] J. L. Hellerstein, S. Ma, and C.-S. Perng, "Discovering actionable patterns in event data," *IBM Systems Journal*, vol. 41, pp. 475–493, 2002.

[3] T. Qiu, Z. Ge, D. Pei, J. Wang, and J. Xu, "What happened in my network: mining network events from router syslogs," in *Internet Measurement Conference*, 2010.

[4] G. Jakobson and M. Weissman, "Alarm correlation," *IEEE Network*, vol. 7, no. 6, pp. 52–59, 1993.

[5] ——, "Real-time telecommunication network management: extending event correlation with temporal constraints," in *Proceedings of the Fourth Symposium on Integrated Network Management*, 1995, pp. 290–301.

[6] K. Hätönen, M. Klemettinen, H. Mannila, P. Ronkainen, and H. Toivonen, "Knowledge discovery from telecommunication network alarm databases," in *Proceedings of the Twelfth International Conference on Data Engineering*, 1996, pp. 115–122.

[7] ——, "TASA: Telecommunication Alarm Sequence Analyzer or: How to enjoy faults in your network," in *IEEE/IFIP 1996 Network Operations and Management Symposium (NOMS'96)*, 1996, pp. 520–529.

[8] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, 1995, pp. 3–14.

[9] L. Burns, J. Hellerstein, S. Ma, C. S. Perng, D. A. Rabenhorst, and D. Taylor, "A systematic approach to discovering correlation rules for event management," in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, 2001, pp. 345–359.

[10] R. Vaarandi and M. Pihelgas, "LogCluster - a data clustering and pattern mining algorithm for event logs," in *2015 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 1–7.

[11] P. Tee, G. Parisis, and I. Wakeman, "Vertex entropy as a critical node measure in network monitoring," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 646–660, 2017.

[12] S. Kobayashi, K. Otomo, K. Fukuda, and H. Esaki, "Mining causality of network events in log data," *IEEE Transactions on Network and Service Management*, vol. 15, pp. 53–67, 2018.

[13] T. Kimura, K. Ishibashi, T. Mori, H. Sawada, T. Toyono, K. Nishimatsu, A. Watanabe, A. Shimoda, and K. Shiomoto, "Spatio-temporal factorization of log data for understanding network events," in *Proceedings - IEEE INFOCOM*, 2014, pp. 610–618.

[14] L. Charlet and A. K. D, "Market basket analysis for a supermarket based on frequent itemset mining," *IJCSI International Journal of Computer Science Issues, www.IJCSI.org*, vol. 9, pp. 1694–0814, 2012.

[15] T. Raeder and N. V. Chawla, "Market basket analysis with networks," *Social Network Analysis and Mining*, vol. 1, pp. 97–113, 2010.

[16] D. Gunopulos, R. Khardon, H. Mannila, S. Saluja, H. Toivonen, and R. S. Sharma, "Discovering all most specific sentences," *ACM Trans. Database Syst.*, vol. 28, no. 2, pp. 140–174, 2003.

[17] X. Wei, Z. Li, T. Zhou, H. Zhang, and G. Yang, "IWFPM: Interested weighted frequent pattern mining with multiple supports," *Journal of Software JSW*, vol. 10, pp. 9–19, 2015.

[18] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *SIGMOD Rec.*, vol. 29, no. 2, pp. 1–12, 2000.

[19] S. Latha and N. Ramaraj, "Algorithm for Efficient Data Mining," in *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, vol. 2, 2007, pp. 66–70.

[20] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.