

Performance Evaluation of Two Data Mining Techniques of Network Alarms Analysis

Jacques-H. Bellec, M-Tahar Kechadi, and Joe Carthy

Abstract—In large telecommunication networks, alarms are usually useful for identifying faults and, therefore solving them. However, for large systems the number of alarms produced is so large that the current management systems are overloaded. One way of overcoming this problem is to analyse and interpret these alarms before faults can be located. Two different techniques were developed and setup to reach this goal. In this paper, we study these two approaches and evaluate their performance using data from a live network.

I. INTRODUCTION

Efficient management of an IT infrastructure is usually a critical issue, especially in telecommunication networks in which the size and complexity of both hardware and software have dramatically increased in the last few years. A typical telecommunication system may consist of several thousands of nodes geographically distributed over several countries, and connected together via some networking devices. In addition, these systems are heterogeneous as multiple versions of many manufacturers products (hardware and software) coexist within the same infrastructure. There is a continuous change in devices, firmware versions, operating systems, networking technologies, development technologies, and tools. To make it even more complex, in some telecommunication systems, each business environment is different from the other one. This dynamic and complex infrastructure creates new challenges in the network management area. Mainly, managing the alarms of a network has many constraints making it a very difficult task [1], [2]. These constraints include response time, an enormous amount of alarms, incomplete or incorrect information, severity of alarms, temporal reasoning, etc. The main objective is to maintain the network operational without any significant loss of service and revenue for the company [3], [4].

To maintain a good quality of service, performance indicators and alarms have been set to produce data in order to inform network operators about the behaviour of the network. As a result, a large amount of events are recorded in log files in order to be processed by the management systems. Usually, when a fault occurs, devices can send messages to describe the problem that has been detected. But they only have a local view of the error, and then cannot describe the fault, but just the consequence of the fault. Due to the complex nature of these networks, a single fault may produce a cascade of alarms from the affected network elements. In addition, a fault can trigger other faults, for instance

in the case of overloading. Even though failures in large communication networks are unavoidable, quick detection, identification of causes, and resolution of failures can make systems more robust, more reliable, and ultimately increase the level of confidence in the services that they provide [5], [6].

In this paper we focus on the performance evaluation of two different approaches for identifying correlated alarms. The main objective of the first approach, called Behavioural Proximity (BP), is to reduce considerably the number of alarms by clustering them according to their behaviour, to form events. Then these events are correlated to form clusters via the Event Duration Matching (EDM) algorithm. As a result, only crucial seeds of global events are presented to the network operator [7], [8]. The second technique, called Topographical Proximity (TP) exploits topographical information embedded in alarm data in order to get only the sequences, which are plausible within the context of a live network topology [9]. This technique addresses the lack of plausibility in mined sequences generated by standard algorithms like MINEPI [10].

The paper is organised as follows: in the next section we describe the work which has already been done in this domain. In section 2 we describe our model and the input data. Section 3 describes the BP approach, namely how the identification of relevant alarm sequences works. The TP approach is dealt with in section 4, and section 5 discusses the experimental results. Finally, we conclude in section 6.

II. BACKGROUND

In the past, the network fault localization and management were performed by human experts. The size and complexity of today's networks, however, mean that the levels of human intervention required to perform this function are prohibitively high. Currently, many systems employ event correlation engines to address this issue. The problem of an automatic identification of events for correlation purpose has been tackled from various perspectives. Model traversal approaches aim to represent the interrelations between the components of the network [11] or the causal relations between the possible events in the network [12] or a combination of the two [13]. Correlations are identified as alarms propagate through the model. Rule-based [14] and code-based [15] systems also model the relations between the events in the system, specifying correlations according to a rule-set or codebook. Other AI techniques, such as neural networks [6], [16] or decision trees, have also been applied to the task. These approaches vary in the level of expert

knowledge required to train the system. Neural networks, for example, can require no expert input whereas model-based techniques may be fully reliant on the insights of human experts. The domain of sequential data mining addresses the specific problem of identifying relationships or correlations between events in a raw dataset, which is inherently sequential in nature, such as fault data consisting of a series of time stamped events. The output of this mining process may then be used as an input to a rule-code or model-based approach. The basic objective is to find noteworthy sequences of events, or sequential patterns that suggest relationships between constituent events. In practice, a noteworthy sequence often corresponds to a frequently occurring sequence in the data set. However, in the case of network alarm data, frequency as the sole measure of sequence noteworthiness is not a valid measure, since it may indicate redundancy. Mining for sequential patterns can be viewed as a subset of the problem of mining for associations between dataset elements in general, constrained by the temporal aspects of the data. But to deal with this kind of data, the temporal aspect is not the only one that we have to consider. In fact, the particular nature of telecommunication networks leads to some strong relationships between alarms behaviour that we cannot find in other kind of data sets.

III. DATA ISSUE

Our system model is composed of devices, alarms, faults, and events. A fault is the manifestation of an error in a system. Any malfunctioning may cause a failure, namely the inability of a system, or a system component, to perform a required function within specified conditions [17], [18]. An alarm is an unsolicited message from a device, typically indicating a problem with the system that requires attention. This real-time indication of an abnormal situation usually includes a priority or a severity code. An event is a set of correlated alarms according to a specific fault. But a fault can lead to one global event composed of different alarms, or several local events, which do not seem to be correlated (maybe because we do not have enough information). An event can be composed of several identical alarms. Moreover a fault can lead to other faults. For each alarm there is one and only one associated event. One important problem that makes the classification and identification of the alarms more complex is that usually, the datasets collected are very noisy, as they contain many different types of alarms with incomplete, redundant, or unnecessary information about their creation [7]. For instance, the alarms generated by devices that are only warnings can be eliminated, because they do not bring any relevant information about faults. Sometimes the information collected is not correct, because alarms are recorded in a centralized way, the time stamping process may not respect the order of the alarm appearances. Moreover, due to the unforeseen network congestion or circumstances, some alarms are triggered but never recorded, or recorded after some significant delay. Data cleaning applied on raw log files becomes crucial for subsequent classification and reduction of the number of alarms. This step is part of the

pre-processing phase of the usual data mining process. In the datasets used to run our experiments, the log files were recorded and created on a daily basis. However, these log files should be processed together, because a fault may occur at 11:50 pm, but the alarm triggered may be recorded the day after. The reason for this is that a study of network behaviour for a one-day period is not sufficient. In order to get a good understanding of network behaviour, one has to consider a longer observation period: weeks, or months. Figure 1 shows the hierarchical representation of a 3GPP mobile telecommunication network, with the management nodes that gather the alarms triggered by lower level nodes.

IV. THE BEHAVIORAL PROXIMITY TECHNIQUE

A. Overview

Devices used in a wide network can be different in nature, in model and in policy, so they can have different behaviours when a fault occurs. They can send one alarm each time they detect an abnormal behaviour or receive an error notification, or they can send alarms periodically until the problem is solved, or just do nothing. The probes or devices, which trigger several alarms in a periodical way give us more information about the fault than an aperiodic or a single alarm, because this assumes that a set of periodical and identical alarms is related to a common event, and though to a common fault. However, due to some other factors such as network traffic or node's load, some alarms will be delayed or lost [19], [20]. This makes it even more difficult to identify periodic and similar alarms. We define identical alarms as alarms triggered by the same probe and having the same content. We assume that when a fault occurs, some alarms are triggered periodically until the fault is resolved. This is the most usual policy used for triggering the alarms. The period depends on the devices, components, or probes responsible for these alarms. We identify three types of alarms: periodic alarms, which are triggered at a regular period P , aperiodic alarms, or alarms which do not seem to be triggered with a specific period, and finally, alarms which are triggered only once. These alarms can either be classified as periodic (with an unknown period) or aperiodic. In this study we prefer to classify them separately.

A fault can lead to several events and each event can be composed of several alarms. Furthermore, events can be overlapped, so it is hard to recognize them [7], [8], [21]. Therefore, aperiodic sets can be considered as periodic, assuming that they are composed of overlapped alarms characterized by a different value of their periods. The aim of our approach is to cluster identical alarms according to a common fault, and find all the periodic sets of alarms from some original aperiodic sets. In other words, the goal is to gather all the alarms related to the same event. In [8], we introduced a new segmentation algorithm called Score-Matching (SM) to recognize the periodic behaviour of identical alarms. As we have previously defined, each set of periodic and identical alarms represents one event. After

identifying events, we are able to correlate them with our technique called Event Duration Matching (EDM).

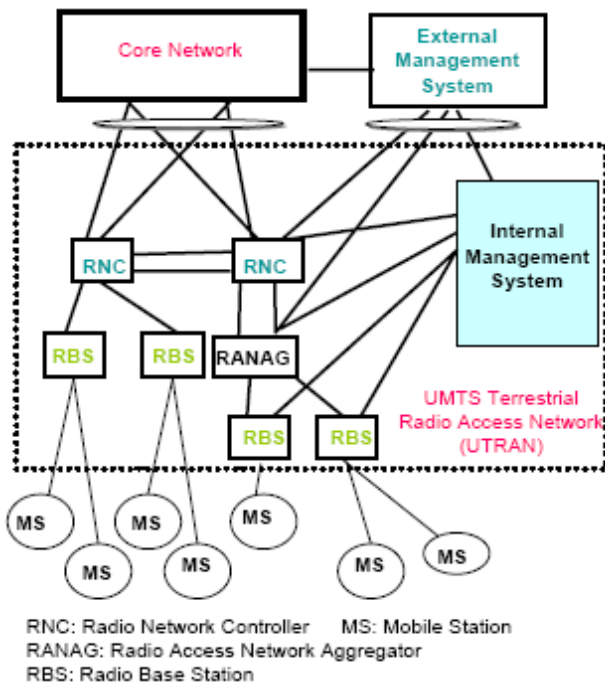


Fig. 1. The 3G Network Structure.

B. The Event Correlation Process

The first part of our process was composed by data preparation and data analysis, detailed in Figure 2. After the identification of the events, we are able to define some rules to correlate them using some new techniques. The trade off between the amount of the available information and the fault distinguishing ability (alarm correlation ability) is made clear if we make the simplifying assumption that two independent faults can not happen at the same location and at the same time. In fact if we miss one overlapped fault in our recognition process, that does not matter because the fault would remain, alarms would be triggered again, and the fault would be identified. We deal with events and not with alarms. It is important to notice that events are composed of correlated alarms. Now, we will correlate events by representing them as time segments, namely the time elapsed between the first and the last alarm of the events. Each segment (event) has a centre of mass, which does not correspond to an alarm but just to a mark to identify the plausible average time of the fault. We consider that the longer the segment is, the more significant it is. We can justify this assumption by the fact that the same alarm is triggered until the problem is solved or until a specified number of occurrences is reached. So we can deduce according to the behaviour that a long segment represents the presence of a fault, contrary to a small segment which does not give sufficient information about the fault.

Figure 3 shows our correlation technique EDM. From the events identified in the first part, namely alarms gathered

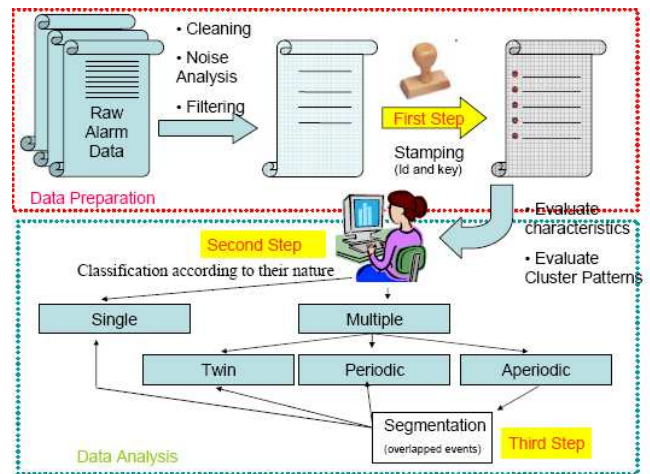


Fig. 2. Data Preparation and Date Analysis.

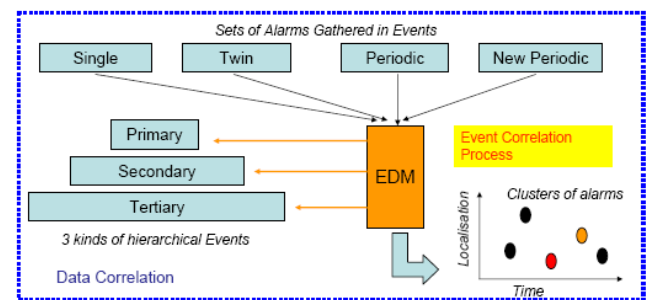


Fig. 3. The Correlation process with EDM.

according to their behaviour, EDM identifies most relevant ones by classifying them into three categories. Primary, secondary and tertiary events are recognized according to their scores and their ranks. The scoring process uses different fields embedded in the alarms which represent the events. These fields are severity, node type, notification type, alarm type code, probable cause, specific problem, event length and the number of alarms, which compose the events. Marks have been accorded to some hypothesis. Firstly we assume that we can have only four kinds of faults, relative to hardware, software, telecom and environment. Secondly we assume that events related to the hardware faults are more important because they may describe the seeds of the faults. On the contrary, events relative to telecommunication faults are less important because they can be just the effect of a hardware or software fault. Thirdly, an important event would be composed by a large number of alarms triggered at a time.

Therefore, all the events receive a score according to these hypotheses and then according to a user-defined rate. Meaningful events are gathered in primary sets, less meaningful in secondary sets and others in tertiary sets, as shown in Figure 3. The algorithm for the Event Classification (EC) is described below for primary events.

The distinction between events is crucial because each primary event will be interpreted as a meaningful event and then used as a root for other less important events. As we

Algorithm 1 EDM Algorithm ; Phase one.

```

1: INPUT: E: Event Set, User-Defined Rate,
2: OUTPUT: P: Primary Events Set
3:  $Av(E) = \sum_{i=0}^n Score(e_i)/n$ ,
4:  $AvL(E) = \sum_{i=0}^n Length(e_i)/n$ ,
5:  $AvO(E) = \sum_{i=0}^n Occurrence(e_i)/n$ ,
6: for all  $e_i \in E$  do
7:   if  $Score(e_i) > (Av(E) * (rate + 1))$  then
8:     if  $Length(e_i) > AvL(E) * (rate + 1)$  then
9:       if  $Occurrence(e_i) > AvO(E) * (rate + 1)$  then
10:         $e_i \in PrimarySet$ 
11:       end if
12:     end if
13:   end if
14: end for
    
```

can notice in Figure 7, in the EDM technique the number of clusters depends directly on the number of primary events identified by the user-defined accuracy rate. The way to evaluate the correlation between two events uses a fuzzy logic approach, with the time distance and the topographical distance. The best link between one primary and one non-primary event is selected according to the best correlation score. Figure 4 illustrates the distribution of the correlation scores with each peak representing a high score.

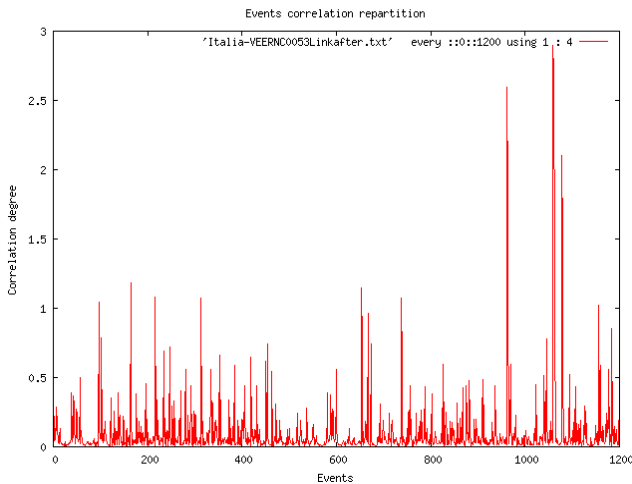


Fig. 4. Events Correlation Scores.

V. TOPOGRAPHICAL PROXIMITY

The standard sequential mining algorithms outlined in section 2 are capable of identifying thousands of sequences in input data. Therefore, post-processing remains an essential component of a useful system whereby sequences which are deemed to be uninteresting, because they are redundant or implausible, are eliminated from the output. This filtering may be automated using templates or performed by domain experts. The Topographical Proximity (TP) algorithm [9]

Algorithm 2 The EDM Algorithm : Phase two

```

1: INPUT P: Primary Events Set, S: Secondary events Set,
2: OUTPUT C: Clusters Set
3: for all  $e_i \in S$  AND  $e_j \in P$  do
4:    $Correlation\_rate(e_i, e_j) = Evaluate\_Link(e_i, e_j)$ ,
5: end for
6: for all  $e_i \in S$  do
7:    $Find\_Best\_Link(e_i)$ 
8: end for
9: for all  $e_j \in P$  do
10:   $Create\_Cluster(e_j)$ 
11: end for
    
```

addresses the problem of determining a reasonable correlation between events in mined sequences at runtime of the mining process. The measure quantifies how closely alarm-generating elements are connected to each other in terms of logical structure of a network. The algorithm uses any information available in alarms which relates to the multiple network topologies, from the topology of physical nodes to any of the multiple views of the management information tree. In practice, the system uses the distinguished names of network entities, both nodes and links, to extract information regarding the relative position of nodes in the hierarchy of the management information tree. The general assumption is: the closer the alarm-generating elements are within this topographical network architecture, the better. BP uses the same idea in its correlation function, namely correlated events depend on their physical distance in the network and the elapsed time between them. The algorithm does not rely on a predefined network configuration as it exploits the topographical information encoded in the alarms. This information is evaluated according to node types and the strength of the possible relationships between node types. The Connections are inferred at runtime between pairs of alarm-generating nodes in the data and a TP measure is assigned. The TP measure is based on the strength of the inferred connection. It is used to reject or promote candidate sequences on the basis of their correlation, i.e. the strength of their connection, thereby reducing the candidate sequence set and optimizing the space and time constraints of the data mining process.

The TP algorithm evaluates the logical distance between two instances of alarm-generating network elements in terms of source node and relationship types. The value has a minimum of zero for nodes that have no logical connection in the network, and a maximum of one for nodes with a direct and close relationship in the network. There is a finite set of TP values for the possible relationships between node types. At runtime, all the available topographical information are extracted from each alarm, and used to infer which relationship may hold between two alarm-generating nodes. An appropriate TP measure is then assigned. The mining algorithm incorporating the TP measure derives from the MINEPI algorithm [10]. It uses a sliding time window to

traverse the data, generating candidate sequences of length n by combining two existing sequences of length $n - 1$ and storing the occurrences of all the sequences above a frequency threshold for subsequent iterations. This threshold is specified by the user. Candidate sequences must conform to user-specified sequence duration, frequency, and topographical proximity parameters.

The output to the mining process is a set of sequences which are both frequent and represent some connection strength in the network.

VI. THEORETICAL COMPARISON

The Behavioural Proximity technique is developed to deal with the alarm correlation problem by outputting only few relevant sets of alarms, namely the major events, to the network operator. On the contrary, the Topographical Proximity is developed as a layer on top of standard data mining algorithms. It aims to check and validate the sequences according to the geographical information embedded in the alarms. The output to the network operator is just the main frequent sequences identified in the data sets, such as the most common scenarios.

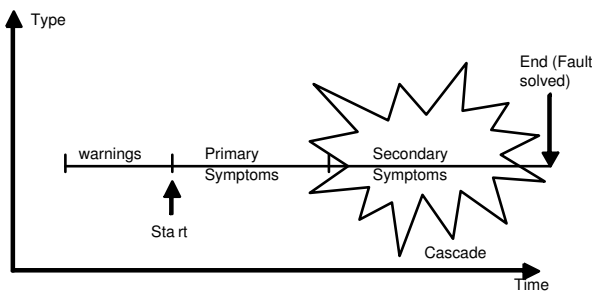


Fig. 5. The Fault life cycle.

Figure 5 shows the life cycle of a fault. We notice that the cascade that follows the crash is more often composed of the same kind of alarms, for example telecommunication failure. However, the primary symptoms may not have been encountered before, they can be unique and then completely transparent to the classical data mining techniques.

In the BP technique, by using behavioural classification, and data analysis, alarms are gathered to form events. The EDM algorithm correlates events to form clusters of events. Each cluster can be seen as a new fault discovered in the system. The standard data mining approach is to find rules of type *If () Then {} Else {}*, (i.e., **IF** communication failure and software failure **Then** high fault rate with degree of trust of 0.90). The main idea in standard data mining is to extract some trends to create some rules. But in the alarm telecommunication problem, common scenarios may not describe the faults neither how to localize them, but just the common alarm cascade which usually follows the faults. Note that BP and TP do not initially answer to the same question. Therefore, a mapping of their output is needed in order to compare the results.

VII. EXPERIMENTAL RESULTS

The two techniques are compared under the same conditions. We use the same samples of data provided by a 3GPP mobile telecommunication network, namely 96,991 individual radio access network alarms. Moreover, we have introduced one sequence composed by four different alarms which has been identified by network experts in the live network. This brings the number of alarms to 10,558. This sequence can be denoted by $\{A, B, C, D\}$. Each letter represents a type of alarm. All the attributes, including the topographical information of these alarms, have been assigned with artificial values. The sequence has always the same period between each sequence element, but the time when it appears is chosen randomly. The results of the EC are shown on Figure 6. We can notice that most of the events are considered to be not important and only a small set of alarms are highly important. By changing the discrimination rate from 0 to 50%, we can get more or less primary and secondary events. For a rate of 0%, we do not have any secondary event, because all the events with a score higher than the average are considered as primary events. On the other hand, with a rate of 50%, all the events with a score higher than the average are secondary events. The number of tertiary events does not change because all the events which do not reach the average score are considered less meaningful. The boundary between the tertiary events and the secondary events is important only if there are no identified primary events, because in this case, all the events would be promoted to the next rank, and the EDM algorithm would identify the clusters according to the new primary events. In Figure 4, the correlation scores are represented for each

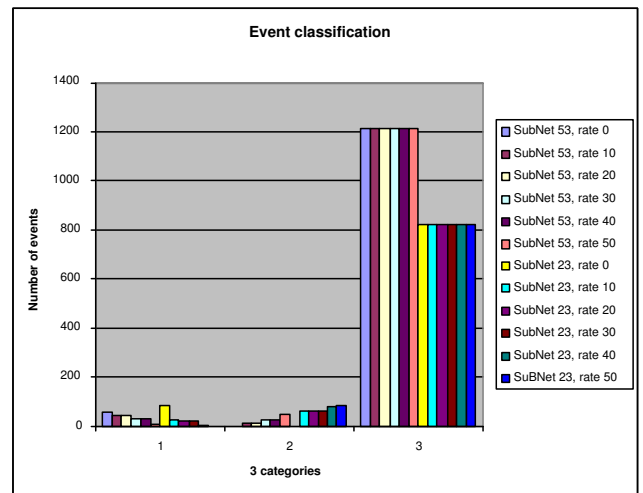


Fig. 6. Events Classification with regards to their importance and type of failures.

possible link between a primary event and all secondary or tertiary events. Each peak represents the highest value found, in other words the elected link between a primary and another event. The number of identified clusters and their sizes, according to a discrimination rate from 0 to 50

are shown in Figure 7 and 8. One can notice that there are neither small clusters nor very large clusters. We can also present to the network operator between 7 to 55 clusters of events depending on the value of the discrimination rate. Each cluster is identified by its total score, its center of mass, its time frame, and its topographical place in the network. So, the network operator can directly identify the faults and makes decision on that particular cluster.

Figure 8 shows experimental results of the BP technique with four subnetworks examined during 72 hours, with different classification rates. The aim of these results is to show approximately what should be a good discrimination rate. It appears that we have a big collapse of the number of identified clusters with rates between 15% and 40%, which can be explained by the fact that there is not a huge number of high scored events in these different data sets. This proves that the scoring function is very efficient. With a rate between 30% and 40%, we have a very small number of identified clusters for each subnetwork. Instead of not having any cluster after a certain rate, namely zero primary event, we chose to upgrade secondary events to primary and tertiary to secondary. By this way, we have the same number of identified clusters with a rate equal to zero and a rate equal to the break point. Each break point depends on the data set and can be defined by the maximum discrimination rate that gets at least one identified cluster. This is shown in Figure 8 and it is between 30% and 40% for these subnetworks. The BP performance plateau is between 20% and 30%.

The data set used in this experimentation contains initially 6000 alarms and with the BP technique 1272 events were identified, and 33 clusters of events were formed with a discrimination rate equal to 10. This means that from 6000 (uncorrelated) potentially events, 33 meaningful events were obtained. This represents 0.55% in this data set and around 1% in average on different subnetworks of this sample.

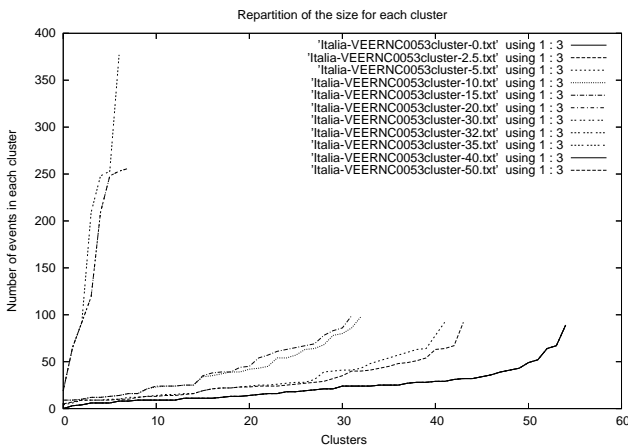


Fig. 7. Size and Number of clusters obtained by BP using different support thresholds from 0 to 50.

In the BP technique, the clusters, which correspond to the virtual sequence $\{A, B, C, D\}$, are $\{A_1 B_1 C_1 D_1, A_2 B_2 C_2 D_2, \dots, A_n B_n C_n D_n\}$, where n

is the total number of $\{A, B, C, D\}$ sequences randomly inserted in the log data. In the TP technique, it is assumed that an alarm A, B, C , or D can appear in any node of the network whereas in the BP algorithm one considers that two identical alarms occurring on different locations are classified separately. In the BP approach, the number of events is equal to the number of days considered multiplied by the number of different alarms inserted and by the number of their corresponding locations. Finally, for the TP approach, the number of patterns is equal to the number of occurrences of each alarm A, B, C , and D multiplied by n and by the number of days. So, the number of identified patterns is equal to the number of occurrences multiplied by the number of clusters obtained by BP.

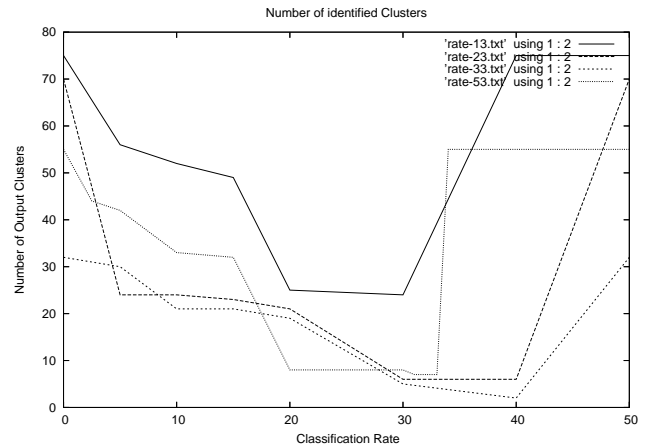


Fig. 8. Number of patterns found according to the user-defined thresholds.

According to [9], the number of patterns identified is between 50 and 5000 depending on the frequency threshold fixed by the user. However, this parameter should not be fixed by the user as some severe alarms usually occur a few times. With BP, the number of clusters appears to be stable enough for different rates used to categorize the events. We can notice that the number of sequence occurrences multiplied by the number of clusters is always higher. Namely 1668 occurrences by 33 clusters gives 55044 patterns but only 6600 patterns are identified by the TP algorithm, which represents 11% of the patterns discovered by the BP technique. On the other hand, the number of clusters obtained by the TP algorithm from 6600 patterns is 4, while the BP algorithm discovered is 33. This shows that the TP technique does not extract all the alarm features. The frequency threshold is too coarse to discriminate between different alarms with different behaviours. These experiments show that the BP algorithm is much more efficient than TP; about 87% better.

VIII. CONCLUSION

The main contribution of this paper is to compare a new technique based on events' correlation, with a more standard technique. We studied two approaches and compared them theoretically and experimentally. We have shown that the choice of correlation measures is crucial as well as the factors

(or dimensions) used to capture the application behaviour. For instance, the TP technique is based of the frequency measure only, which does not reflect the alarm behaviour, whereas in the BP technique we consider the periodicity of the alarms, which reflects better the network behaviour. We show that this new technique is much more efficient than classical statistical approaches. It presents an improvement of 87% compared to the TP approach. For further improvements, we will implement the BP technique with different learning algorithms based. This will need the development of more accurate models for the alarm network behaviour using fuzzy logic, and other learning systems such as neural networks.

ACKNOWLEDGMENT

The authors would like to thank A. Devitt and J. Duffin from Ericsson Ireland for providing us with many datasets from wide telecommunication networks and with a direct access to their previous work and facilities.

REFERENCES

- [1] R. Gardner and D. Harle, "Fault resolution and alarm correlation in high speed networks using database mining techniques," in *Int'l. Conf. on Information, communication and signal processing (ICICS'97)*, Singapore, Sept. 9-12 1997.
- [2] G. Jakobson and M. Weismann, "Real time telecommunication network management: extending event correlation with temporal constraints," in *Proc. of the 4th int'l symposium on integrated network management*, Santa Barbara, California, USA, 1995, pp. 290-301.
- [3] M. Klemitten, H. Mannila, and H. Toivonen, "A data mining methodology and its application to semi-automatic knowledge acquisition," in *8th Int'l. Workshop on database & expert systems applications (DEXA'97)*, Toulouse, France, Sept 1-5 1997.
- [4] E. Aboelela and C. Douligeris, "Fuzzy temporal model for event correlation in network management," in *24th Conf. on Local Computer Networks*, Lowell, Massachusetts, USA, Oct. 1999, pp. 150-159.
- [5] A. Bouloutas, S. Galo, and A. Finkel, "Alarm correlation and fault identification in communication networks," *IEEE Trans. on Communications*, vol. 4, no. 2/3/4, pp. 523-533, Feb/Mar/Apr 1994.
- [6] R. Gardner and D. Harle, "Alarm correlation and network fault resolution using kohonen self-organising map," in *IEEE Global Telecom. Conf.*, vol. 3, New York, NY, USA, 1997, pp. 1398-1402.
- [7] J.-H. Bellec, M.-T. Kechadi, and J. Carthy, "Study of telecommunication system behavior based on network alarms," in *Workshop on Data Mining for Business*, Porto, Portugal, Oct. 3-7 2005.
- [8] J-H.Bellec, M.-T. Kechadi, and J.Carthy, "A new efficient clustering algorithm for network alarm analysis," in *The 17th IASTED Int'l. Conference on Software Engineering and Applications (SEA'05)*, Phoenix, AZ, USA, Nov. 14-16 2005.
- [9] A. Devitt, J. Duffin, and R. Moloney, "Topographical proximity for mining network alarm data," in *ACM SIGCOMM workshop on Mining network data*, Philadelphia, Pennsylvania, USA, Aug 22-26 2005.
- [10] H. Mannila, H. Toivonen, and A. Verkamo, "Discovery of frequent episodes in events sequences," *Data mining and knowledge Discovery*, vol. 1, pp. 259-286, 1997.
- [11] D. Meira and J. Nogueira, "Modelling a telecommunication network for fault management applications," in *Proc. of the Network Operations and Management Symposium, (NOMS'98)*, New Orleans, Louisiana, USA, 1998, pp. 723-732.
- [12] R. Gopal, "Layered model for supporting fault isolation and recovery," in *IEEE/IFIP, Proc. of Network Operation and Management Symposium*, Honolulu, Hawaii, Apr 10-14 2000.
- [13] M. Steinder and A. Sethi, "Non-deterministic diagnosis of end-to-end service failures in a multi-layer communication system," in *Proc. of ICCCN'01*, Scottsdale, Arizona, USA, 2001, pp. 374-379.
- [14] G. Liu, A. Mok, and E. Yang, "Composite events for network event correlation," in *In Proc. IFIP/IEEE International Symposium on Integrated Network Management, (IM'99)*, Boston, USA, 1999, pp. 247-260.
- [15] S. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie, "High speed and robust event correlation," *IEEE Communications Magazine*, vol. 34, no. 5, pp. 82-90, 1996.
- [16] H. Wietgreffe, K.-D. Tuchs, K. Jobmann, G. Carls, P. Frohlich, W. Nejd, and S. Steinfeld, "Using neural networks for alarm correlation in cellular phone networks," in *Proc. of the International Workshop on Applications of Neural Networks to Telecommunications 1997 (IWANN'T*97)*, Melbourne, Australia, 1997.
- [17] K. Julisch, "Clustering intrusion detection alarms to support root cause analysis," *ACM Transactions on Information and System Security*, vol. 6, no. 4, pp. 443-471, 2003.
- [18] K. Yamanishi and Y. Maruyama, "Dynamic syslog mining for network failure monitoring," in *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. New York, NY, USA: ACM Press, 2005, pp. 499-508.
- [19] G. Jakobson and M. Weismann, "Alarm correlation," *IEEE Network*, vol. 7, no. 6, 1993.
- [20] R. Gardner and D. Harle, "Methods and systems for alarm correlation," in *Proc. of Globecom'96*, London, UK, Nov. 18-22 1996, pp. 136-140.
- [21] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmaki, and H. Toivonen, "Time series segmentation for context recognition in mobile devices," in *Proc. of the IEEE International Conference on Data Mining*, San Jose, California, USA, Nov 29 - Dec 2 2001, pp. 203-210.