

# Topographical Proximity for Mining Network Alarm Data

Ann Devitt

Joseph Duffin

Robert Moloney

Ann.Devitt,Joseph.Duffin,Robert.Moloney@ericsson.com

Network Management Research Centre  
Ericsson R&D Ireland  
Dublin 4, Ireland

## ABSTRACT

Increasingly powerful fault management systems are required to ensure robustness and quality of service in today's networks. In this context, event correlation is of prime importance to extract meaningful information from the wealth of alarm data generated by the network. Existing sequential data mining techniques address the task of identifying possible correlations in sequences of alarms. The output sequence sets, however, may contain sequences which are not plausible from the point of view of network topology constraints. This paper presents the Topographical Proximity (TP) approach which exploits topographical information embedded in alarm data in order to address this lack of plausibility in mined sequences. An evaluation of the quality of mined sequences is presented and discussed. Results show an improvement in overall system performance for imposing proximity constraints.

**Categories and Subject Descriptors:** H.2.8 [Database Management]: Database Applications—*Data Mining*

**General Terms:** Algorithms, Experimentation

**Keywords:** Topographical proximity, event correlation, mined sequential patterns, fault data, network configuration

## 1. INTRODUCTION

Given the growing complexity of today's networks, the task of ensuring robustness and maintaining quality of service requires increasingly powerful network management systems. This steady increase in network size and complexity produces a corresponding increase in the volume of data, such as performance indicators or alarms, to be processed by management systems. In particular, the area of fault management remains a key problem for network operators, as the speed at which faults are handled has very immediate consequences for network performance. The complex, inter-

connected nature of the network means that a single fault may produce a cascade of alarms from affected network elements. Conversely, intermittent, self-clearing alarms may be raised without any attendant fault in the network. Event correlation provides a means of dealing with this large volume of alarm data. Correlations define relations between alarm events that facilitate the process of alarm filtering, masking and prioritisation. While sequential data-mining techniques have evolved to identify possible useful correlations in alarm data, the task of identifying the subset of important and plausible correlations remains heavily dependent on the domain expertise of network equipment manufacturers and operators. On the other hand, alarm data encodes substantial domain knowledge which may be exploited to improve the mining process. In particular, individual alarms contain topographical information identifying the network elements which generated them. This paper addresses the challenge of harnessing this latent domain knowledge in order to provide criteria for automatically evaluating the plausibility of mined alarm correlations.

Section 2 sets out current approaches to event correlation and sequential data-mining. Section 3 outlines the Topographical Proximity algorithm which exploits topographical attributes of alarm data for mining sequential patterns in network alarm data. Section 4 describes experiments undertaken to provide a qualitative evaluation of the topographical proximity approach for mining telecommunications alarm data. The results are presented and discussed in section 5.

## 2. BACKGROUND

In the past, the task of network fault localization and management was 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 automatically identifying events for correlation has been tackled from various perspectives. Model traversal approaches aim to represent the interrelations between the components of the network [8] or causal relations between possible events in the network [3] or a combination of the two [10]. Correlations are identified as alarms propagate through the model. Rule-based [6] and code-based [14] systems also model the relations between events in the system, specifying correlations according to a rule-set or codebook. Other AI techniques, such as neural networks [13] or decision trees, have also been applied to the task.

The approaches noted above vary in the level of expert

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*SIGCOMM'05 Workshops*, August 22–26, 2005, Philadelphia, PA, USA.  
Copyright 2005 ACM 1-59593-026-4/05/0008 ...\$5.00.

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 a human expert. 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 which consists of a series of timestamped events. The output of this mining process may then be used as the 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 equates 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 as frequency may indicate redundancy. The research presented here is motivated by the need to establish novel criteria for pattern selection in sequential data mining.

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. Much of the foundation work in sequential mining techniques shares a common historical origin in the Apriori association rule mining algorithm for transaction data [2]. Apriori is based on the assumption that a frequent set of elements must consist of elements which are themselves frequent. The algorithm generates a set of frequent sequences by iterating through a “generate and count” process, generating candidate sequences of increasing length and pruning the set based on sequence support, i.e. normalised frequency. Candidates are generated by a process of merging two existing sequences of length  $n - 1$  to give a sequence of length  $n$ , as in example 1.

$$ABC + ABD \Rightarrow ABCD \quad (1)$$

The GSP [9] and WINEPI [7] algorithms were the first to adapt the Apriori technique to mine for temporal association rules in sequential data. Both employ a sliding time window with a user-specified duration to traverse the input data, extracting sequences according to user-specified minimum and maximum sequence duration constraints. The two algorithms do, however, differ significantly in design and implementation details. Other Apriori-based approaches aim to optimise performance within the same conceptual framework. MINEPI [7] is an extension of the WINEPI algorithm which optimises the space and time constraints by compressing event sequences to their minimal occurrence time window. FreeSpan [4] focuses on the candidate generation process employing a database of projected sequence extensions to ensure that the system only generates candidates that exist in the data. Extensions of this algorithm modify the projected database structure and access to optimise the depth-first search of possible candidate sequences. SPADE [15] decomposes the search space and uses lattice-based search strategies to optimise performance. Other approaches are designed to identify sequences according to criteria other than frequency such as periodicity [5], causality [11] or predictive power [12]. The research set out below is loosely based on an Apriori approach and introduces a novel criterion for sequence selection which evaluates sequence plausibility and coherence in the context of network topology.

### 3. TOPOGRAPHICAL PROXIMITY

The sequential mining algorithms outlined in section 2 are capable of efficiently identifying thousands of event sequences in sequential 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 introduced in this paper constitutes a means of determining the plausibility of a 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 the logical structure of a network. The algorithm is intended to be eclectic using any information available in alarms which relates to the multiple topologies of a network, from the topology of physical nodes to any of the multiple views of the management information tree. In practice, the system uses the fully distinguished names (FDNs) of network entities, both nodes and links, to extrapolate information regarding the relative positioning of nodes in the containment hierarchy of the management information tree. The general assumption is that the closer the alarm-generating elements within this topographical realisation of the network, the more plausible, and hence interesting, the relationship between the alarms and the greater likelihood that there is some cause and effect relationship between them.

The algorithm is not reliant on a pre-defined network configuration as it exploits the topographical information encoded in the alarms themselves. This information is evaluated relative to a specification of node types and of the strength of possible relationships between node types. Connections are inferred at run-time between pairs of alarm-generating nodes in the data and a Topographical Proximity (TP) measure is assigned based on the strength of the inferred connection. The TP measure is used to reject or promote candidate sequences on the basis of their plausibility, i.e. the strength of their connection, thereby reducing the candidate sequence set and optimising the space and time constraints of the data mining process. The TP measure may also be used at post-processing to rank sequences in terms of the connectedness of their constituent alarm events. Section 3.1 outlines the assumptions underlying the topographical proximity approach and details how the topographical proximity measure is calculated. Section 3.2 sets out how the TP measure has been integrated into the sequential mining process.

#### 3.1 TP Calculation Algorithm

The TP algorithm assumes a network which consists of functional nodes connected by communication interfaces and arranged in a logical co-operative and hierarchical structure, as represented by the simplified schema in figure 1. According to this schema, alarm-generating or source node types are generalised to Master, Servant and Child Nodes. These nodes have functional subcomponents which may generate fault alarms. Node subcomponents represent a node’s internal functionality, the functionality of the interfaces between nodes or logical communications artefacts. Aspects of source node location and functionality relevant to a particular alarm are presented as alarm attributes. In particu-

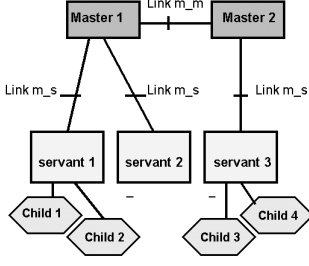


Figure 1: Simplified Network Schema

lar, the source node FDN attribute identifies the node’s ancestors within the containment hierarchy. Other attributes specify further topographical information, such as relevant links or resources. The relationships defined between source node types are categorised as node identity, adjacency, sisterhood and subsumption. The type of relationship and the degree of certainty that there exists such a relationship between alarm-generating nodes in the network is defined according to the level of shared, reciprocal or disjunct topographical information extracted from the two alarms, as set out in the table below.

Relationship category	Information overlap
Identity	Full Matching FDNs
Adjacency	Reciprocal information
Subsumption, sisters	Shared information
Unrelated nodes	No shared information
Unrelated alarms	Disjunct information

The TP calculation algorithm evaluates the logical distance between two instances of alarm-generating network elements on the basis of the two source node types and the relationship type extrapolated from the topographical information in the alarms. A finite set of TP values for the possible relationships between different node types was defined by domain experts and is stored in a relationship table for use at run-time. 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 clear and close relationship in the network. At run-time, each alarm is parsed for all available topographical information. This is used to determine which relationship may hold between two alarm-generating nodes and an appropriate Topographical Proximity (TP) value from the relationship table is assigned, as set out in algorithm 1.

The closest relationship is node identity ( $TP = 1$ ), e.g. two alarms originating from the *Master1* node in figure 1. The loosest connection is where two nodes have only a shared subnetwork in common ( $TP = 0.2$ ), e.g. two alarms originating from nodes *Child1* and *Child4* in figure 1. An adjacency relation holds between alarms originating on nodes *Master1* and *Master2* if there is a reference to *LinkM\_M*,  $TP = 0.9$ . If both alarms refer to the link,  $TP = 1$ . All other connections are assigned values ( $0 \leq TP \leq 1$ ) to reflect the closeness of the relationship.

### 3.2 TP Mining Algorithm

The mining algorithm incorporating the Topographical Proximity measure derives from the MINEPI algorithm [7].

---

#### Algorithm 1 calculateTP

---

**INPUT:**  $alarm_1, alarm_2$ : two alarm events

**OUTPUT:** TPvalue

Identify  $node_1$ , source node type of  $alarm_1$

Identify  $node_2$ , source node type of  $alarm_2$

Identify relationship between  $node_1$  and  $node_2$ , given topographical information available

Look up TPvalue in the predefined relationship table.

Return TPvalue for this relationship

---

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 occurrences of all sequences above a user-specified frequency threshold for subsequent iterations. However, where MINEPI is optimised for time, storing the most compact, or minimal, occurrences of all frequent sequences, the TP algorithm is optimised for sequence connectedness, storing sequence occurrences whose TP value exceeds a given TP threshold parameter. The algorithm to calculate topographical proximity for candidate sequences is set out in algorithm 2. The TP value for an occurrence of a candidate sequence is the mean of the TP values for the two existing sequence occurrences to be merged and the proximity value calculated for the first and last alarms of the new candidate. This latter TP calculation is necessary to evaluate the new connection in the candidate sequence, i.e. the connection which is not present in the subsequences.

The full candidate generation algorithm is set out in algorithm 3. Candidate sequences must conform to user-specified sequence duration, frequency and topographical proximity parameters. The added cost of the TP computation is minimal as, for each occurrence of a new candidate sequence, only one new TP calculation is carried out. Furthermore, even this cost is offset by the reduction in search space of candidate sequences at each iteration achieved by imposing a minimum TP value threshold. The output to the mining process is a set of sequences which are both frequent and represent plausible connections in the network. Section 5 explores the impact of the topographical proximity measure on the accuracy of the mining algorithm.

---

#### Algorithm 2 calculateSequenceTP

---

**INPUT:**  $seq, \{alarm_1, alarm_2 \dots alarm_n\}$

**OUTPUT:** TPvalue

**if**  $length(seq) == 2$  **then**

    return  $calculateTP(alarm_1, alarm_2)$

**else**

$TP_{seq1} =$  Retrieve from memory  $TP_{alarm_1 \dots (n-1)}$

$TP_{seq2} =$  Retrieve from memory  $TP_{alarm_2 \dots n}$

$TP_{new} = calculateTP(alarm_1, alarm_n)$

    return  $\frac{TP_{seq1} + TP_{seq2} + TP_{new}}{3}$

**end if**

---

## 4. EXPERIMENTS

This TP algorithm was implemented for alarm data from a 3GPP mobile telecommunications network. A set of experiments has been conducted in order to provide a qualitative evaluation of the mining algorithm at different topographical proximity thresholds. The absence of a gold standard dataset with target sequences marked up in the data has

---

**Algorithm 3** TP mining algorithm

---

**INPUT:**  $seq_1, seq_2$ : 2 sequences of length  $n - 1$ , e.g. ABC and BCD  
**OUTPUT:** newSeq: sequence of length  $n$ , e.g. ABCD  
**for all**  $o_1$  such that  $o_1$  isa occurrence of  $seq_1$  **do**  
  **for all**  $o_2$  such that  $o_2$  isa occurrence of  $seq_2$  **do**  
    Posit occurrence  $o_{newSeq}$  from start  $o_1$  to end  $o_2$   
    **if**  $Duration_{o_{newSeq}} \leq maximumDuration$  **then**  
       $TP_{newSeq} = calculateSequenceTP(o_{newSeq})$   
      **if**  $TP_{newSeq} \geq TPthreshold$  **then**  
        store  $o_{newSeq}$   
      **end if**  
    **end if**  
  **end if**  
**end for**  
**if**  $\#occurrences_{newSeq} \leq frequencyThreshold$  **then**  
  Prune newSeq  
**end if**

---

meant that research to date has tended to focus on system performance in terms of computational complexity rather than accuracy. The experiment described below aims to address this shortfall with an evaluation of the quality of the output of the mining algorithm. A synthetic gold standard dataset was created by introducing synthetic alarms into real network alarm data. It was possible then to evaluate system performance in identifying sequences which are present in known quantities and distributions in the data. The experiment was run on a Pentium 4 3.2 GHz processor with 2 GB of RAM running Microsoft Windows XP Professional version 2002.

## 4.1 Performance Metrics

The metrics used to determine performance in the experiment reported below are the measures of precision and recall borrowed from the Information Retrieval domain. In the context of this mining experiment, the measures are defined as follows:

- **Precision:** the number of correctly identified target sequence instances relative to the total number of sequence instances found by the system.

$$Precision = \frac{\text{No. of correct instances found}}{\text{Total no. of instances found}} \quad (2)$$

- **Recall:** the number of correctly identified target sequences relative to the total number of inserted target sequences.

$$Recall = \frac{\text{No. of correct instances found}}{\text{No. of instances in the target set}} \quad (3)$$

These two metrics may be combined to give a single indicator of system performance, the F Score, which may be interpreted as a measure of the accuracy and precision of the result set. F Score is calculated according to the following formula:

$$FScore = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

These metrics focus on the performance of the mining algorithm in terms of its ability to identify patterns known to exist in the data while restricting these patterns to ones which represent plausible connections in a telecommunications network.

## 4.2 Dataset

The basic dataset for each experiment consists of 96,991 individual radio access network (RAN) alarms from a live telecommunications network plus 10,538 inserted alarms.<sup>1</sup> The alarm format conforms to 3GPP FM standards [1] and includes a timestamp with a granularity of milliseconds and thirteen attributes relating to four broad categories of alarm timing, event lifecycle, alarm type and alarm source information. The RAN contained 2 RNC nodes, 304 RBS nodes and 1800 cells.

### 4.2.1 Synthetic alarm sequences

The synthetic alarm sequences were designed to be unique in the data but yet correspond to known correlation patterns in telecommunications alarm data. The two sequences conform to the timing and network element constraints of two common inter-event correlations identified by network experts in telecommunications alarm data. All attributes, including topographical information, of the component alarms, however, have been assigned synthetic values which do not occur in the original dataset. Synthetic sequence A consists of four alarms on the same network node, three within 10 milliseconds followed by a fourth alarm after 200 seconds. Sequence B consists of two alarms on the same Master Node occurring 200 seconds apart.

### 4.2.2 Synthetic Dataset

The two synthetic sequences were introduced into the original dataset of 96,991 alarms. Synthetic sequences were inserted at random within sub-blocks of 10,000 alarms, between 100 and 200 times per sub-block. From a random start time, all sequences are interleaved with existing alarms according to the timing constraints specified for the synthetic alarm sequence. The resulting dataset comprises 107,529 alarms: 96,991 original alarms plus 10,538 synthetic alarms, 1,668 instances of Sequence A and 1,933 of Sequence B.

## 4.3 Test Cases

The sequence length, time window and minimum support system parameters for this experiment were dictated by the constraints on target sequence insertion into the original alarm data. Maximum sequence length was fixed at four alarms per sequence. The time window parameter was fixed at 240 seconds as both synthetic sequences have a duration of approximately 200 seconds. The minimum support parameter, which specifies the required pattern frequency per block of 10,000 alarms, was tested with the range of 50–175 occurrences at intervals of 25 occurrences. For each of the six support threshold values, baseline system performance was calculated using the basic MINEPI algorithm,  $TP = 0$ , and seven further test cases were evaluated at  $0.4 \leq TP \leq 1.0$ , giving a total of 48 test cases. The aim was to investigate the interactions between the support and TP threshold system parameters and establish optimum parameter values for the identification of known target sequences in the data.

## 4.4 Procedure

The mining algorithm was run on the set of 107,529 alarms for the 48 test cases. Precision, recall and F Score values

---

<sup>1</sup>Core network alarms were excluded from this analysis, as, although they may include interesting correlations, the core network is separate from the structure defined in section 3 which is the basis of the TP calculation.

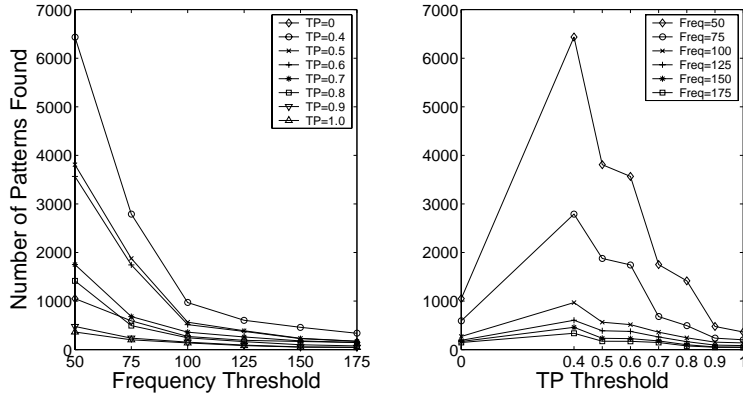


Figure 2: Number of Patterns Found at  $timeWindow = 240$  according to TP and Support Frequency thresholds

were calculated based on the number of occurrences of the synthetic target sequences found relative to the number of known occurrences in the synthetic dataset. The results are presented and discussed in section 5.

## 5. RESULTS

The performance of the system is evaluated firstly in general terms relating to the size of the output sequence set identified for each test case and secondly with a precise qualitative evaluation on the task of identifying known sequence instances in the data.

### 5.1 Performance Overview

The two plots in figure 2 give a general overview of system performance represented by the number of patterns found during the mining process for this dataset. In the absence of a gold standard, it is not possible to know the *correct* number of patterns which a mining algorithm should identify. In order to merit the computation of the mining process, however, the output set should consist of a set of good sequences which is significantly smaller than the input data set. The TP approach is designed to ensure that the output sequences are “good” in the sense of plausible within the network. Figure 2 examines how the system parameters of support and TP threshold impact on the size of the output sequence set. The first plot illustrates the expected outcome that at high support thresholds there is a marked reduction in the number of patterns identified in the data. Indeed, the number of patterns found at high support thresholds converges for all TP values. It is perhaps more interesting to look at the behaviour of the algorithm at low support thresholds:

- Low TP values ( $0.4 \leq TP \leq 0.6$ ): there is a significant increase in the size of the output set relative to the MINEPI baseline;
- Mid TP values ( $0.7 \leq TP \leq 0.8$ ) exhibit similar characteristics to the baseline performance;
- High TP values ( $TP \geq 0.9$ ) the output set is consistently smaller than the baseline.

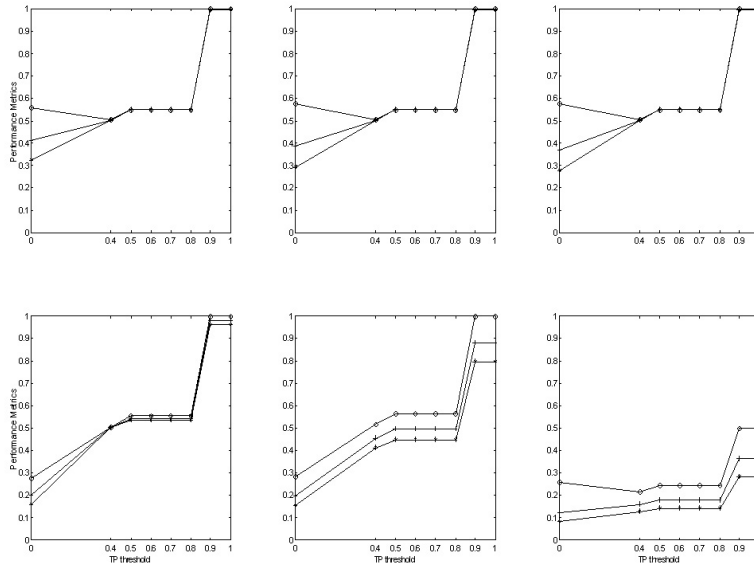
These observations would suggest firstly that a low TP threshold is not sufficiently powerful to restrict the output sequence set. This factor is offset by the quality of output sequences which should reflect some plausible connection

between constituent alarm-generating nodes. Secondly, for mid-range TP thresholds, baseline performance is achieved with the added benefit of plausible sequences. Finally, for high TP thresholds, the support parameter has little impact on the size of the output set. Therefore, infrequent sequences may be identified in the data without an attendant explosion in the number of patterns found. This would suggest that improvements in system performance may be achieved by refining the node and connection type definitions which determine TP assignment values at run-time. These definitions may also be customised to specific mining tasks.

### 5.2 Target Sequence Identification Task

The six plots in figure 3 present system performance on the synthetic sequence identification task for each of the six support parameter values. There is a very clear trend across all support value thresholds: as the TP threshold increases, there is marked increase in precision, recall and F Score value with perfect performance, where all sequences are correctly identified, for  $TP = 0.9, 1$  at  $50 \leq support \leq 125$ . There is a reduction in the performance measures for  $support \geq 150$ . The decrease in recall for  $support = 150$  represents a reduction in the number of target sequences found. This result reflects the random nature of the sequence insertion process, where not all blocks of 10,000 alarms contain the same number of sequences and individual sequences may not pass the support threshold for some processing blocks. There is a marked reduction in performance for  $support = 175$  as no instances of synthetic sequence A were found in the data. In fact, there is an average of 166 instances of this sequence per 10,000 alarms in the data, therefore, it should not be found at this support threshold. This point highlights the fact that the topographical optimisation applied in this approach maintains a direct mapping between the support threshold parameter and the actual frequency of the constituent alarms in the data, while this mapping may be lost in the compacting process of the MINEPI algorithm.

The key point to note here, however, is that, for all frequency levels, the use of a TP threshold results in improved performance on the task of identifying specific instances of prototypical target sequences in the dataset. The performance plateau for  $0.5 \leq TP \leq 0.8$  would suggest that there is a baseline TP value between 0.4 and 0.5 assigned for many



**Figure 3: Performance metrics at  $timeWindow = 240$ ,  $TP = \{0, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ ,  $100 \leq SupportFrequency \leq 175$ . Precision:  $-*$ -, Recall:  $-O$ -, F Score:  $-+-$ .**

of the permutations of the target sequences in this data. This gives further support to the hypothesis that even loose topographical proximity constraints will ensure that mined sequences converge with sequence types defined by network analysts. Furthermore, there is no apparent trade-off between precision and recall values for this approach suggesting that the mining process is both selective and accurate, maintaining tight control on candidate sequence promotion while exploring the full range of sequence instances.

## 6. CONCLUSIONS

The main contribution of this paper is to introduce the Topographical Proximity (TP) measure for mining alarm data for event correlations. This measure exploits the topographical information encoded in alarms to validate all candidate sequences at run-time with respect to the plausibility of the possible correlation they represent. The second significant contribution is to provide a qualitative evaluation of the performance of the mining algorithm. The evaluation results would strongly suggest that the performance of the mining algorithm improves with the inclusion of the TP measure.

## 7. REFERENCES

- [1] 3GPP. 3rd generation partnership project technical specification group services and system aspects. Telecommunication management. Fault Management. Part 2: Alarm Integration Reference Point (IRP), Information Service (IS), (Release 6) 3GPP TS 32.111-2 V6.3.0, 3GPP, 2004.
- [2] R. Agrawal, T. Imielinski, and A. N. Swami. Mining associations between sets of items in massive databases. In *Proc. of ACM-SIGMOD 1993 International Conference on Management of Data*, pages 207–216, May 1993.
- [3] R. Gopal. Layered model for supporting fault isolation and recovery. In *Proc. of NOMS'00*, Hawaii, 2000.
- [4] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.
- [5] E. O. Heierman, G. M. Youngblood, and D. J. Cook. Mining temporal sequences to discover interesting patterns. In *Proceedings of KDD 2004, Workshop on mining temporal and sequential data*, 2004.
- [6] G. Liu, A. K. Mok, and E. J. Yang. Composite events for network event correlation. In *IM'99*, pages 247–260, 1999.
- [7] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, 1997.
- [8] D. M. Meira and J. M. S. Nogueira. Modelling a telecommunication network for fault management applications. In *Proc. of NOMS'98*, pages 723–732, 1998.
- [9] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proc. of EDBT'96*, 1996.
- [10] M. Steinder and A. S. Sethi. Non-deterministic diagnosis of end-to-end service failures in a multi-layer communication system. In *Proc. of ICCCN'01*, pages 374–379, Arizona, 2001.
- [11] R. Sterritt. Discovering rules for fault management. In *Proc. of ECBS'01*, pages 190–196, 2001.
- [12] G. M. Weiss. Predicting telecommunication equipment failures from sequences of network alarms. In W. Kloesgen and J. Zytow, editors, *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, 2002.
- [13] H. Wietgreffe, K.-D. Tuchs, K. Jobmann, G. Carls, P. Frohlich, W. Nejdil, and S. Steinfeld. Using neural networks for alarm correlation in cellular phone networks. In *Proc. of IWANN'T 1997*, 1997.
- [14] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie. High speed and robust event correlation. *IEEE Communications Magazine*, 34(5):82–90, 1996.
- [15] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 0:1–31, 2000.