Topographical Proximity: Exploiting Domain Knowledge for Sequential Data Mining

Ann Devitt
Ericsson
Dublin 4
Ireland
ann.devitt@ericsson.com

Joseph Duffin
Ericsson
Dublin 4
Ireland
joseph.duffin@ericsson.com

Abstract

In today's mobile telecommunications networks, increasingly powerful fault management systems are required to ensure robustness and quality of service of the network. In this context, fault alarm correlation is of prime importance to extract meaningful information from the vast quantities of alarms generated by the network. Existing sequential data mining techniques address the task of identifying possible correlations in frequent sequences of telecoms alarms. These frequent 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 the topographical information encoded in telecommunication alarms in order to address this lack of plausibility in mined alarm 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.

1 Introduction

Given the growing complexity of mobile telecommunications networks, the task of ensuring robustness and maintaining quality of service in the network requires increasingly powerful network management systems. Furthermore, the steady increase in size and complexity of the network produces a corresponding increase in the volume of data generated by network elements (e.g. alarms, performance indicators) placing added strain on management systems. In particular, the area of fault management remains a key problem area 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. In this context, event correlation provides a means of dealing with the large volume of alarm data. Correlations define relations between alarm events that facilitate the processes of alarm filtering, masking and prioritising specified in ITU-T recommendations [7]. 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. Yet alarms encode substantial domain knowledge, in particular topographical information regarding the network elements which generated a given alarm. Furthermore, telecommunications networks, although complex, conform to a well-defined topology of network elements. This paper addresses the challenge of harnessing the latent domain knowledge available in alarm data in order to provide criteria for automatically evaluating the plausibility of mined alarm correlations. Section 2 sets out current approaches in the domain of sequential data-mining addressing the task of event correlation. Section 3 describes the need to exploit topographical attributes of the input data to validate mined sequences and how this has been realised for telecommunications alarm data as the Topographical Proximity (TP) measure. Section 4 describes a set of experiments aimed at providing a qualitative evaluation of the topographical proximity approach for mining telecommunications alarm data. The results are presented and discussed in section 5.

2 Sequential Data Mining

Telecommunications alarm data is inherently temporal and sequential in nature, consisting of a series of timestamped events. The specific problem of identifying relationships between events in a sequential dataset 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. The domain of sequential data mining addresses this problem space with the objective of finding *noteworthy* sequences of events or sequential patterns that suggest relationships between constituent events. In theory, the notion of noteworthiness may be task–specific. In practice, however, a sequence which is noteworthy often equates to a sequence which occurs frequently in the input data. However, frequency as the sole measure of sequence "noteworthiness" is not a valid measure for network alarm data where frequency may indicate redundancy. The research presented here is motivated by the need to establish novel criteria for pattern selection in sequential data mining.

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 sequence 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 frequency or *support* (i.e. normalised frequency) values. 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.

$$ABC + ABD => ABCD \tag{1}$$

The WINEPI [8] and GSP [10] algorithms were among 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. Although the basic premise for the two algorithms is the same, they differ in many design and implementation details. The GSP algorithm was designed for mining transaction data and, therefore, incorporates extra transaction-based constraints on viable candidate sequences. Furthermore, GSP events or items may be organised in a taxonomy allowing events or *their* superordinates in the taxonomy to be used for calculating support values or generating candidate sequences. WINEPI, on the other hand, is optimised for flat sequential data, like telecoms alarm data and addresses the issue of full or partial ordering of event sequences.

Other Apriori-based approaches aim to optimise performance within the same conceptual framework. MINEPI [8] is an extension of the WINEPI algorithm which optimises space and time constraints by compressing event sequences to their minimal occurrence window. FreeSpan [5] 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. Its extensions, PrefixSpan [9] and IncSpan [3], modify the projected database structure and access to optimise the depth-first search of possible candidate sequences. SPADE [13] decomposes the search space and uses lattice-based search strategies to optimise performance.

Apriori-based approaches assume that the aim is to identify highly frequent patterns. Other approaches are designed to extract sequences according to different criteria. Weiss [12] describes a supervised machine learning system using genetic algorithms where the objective is to predict *rare*, rather than frequent, equipment failures events on the basis of alarm sequences, candidates sequences are generated by a combination and/or mutation process. Heierman et al [6] use periodicity and length of sequences as well as frequency in their candidate selection process. Sterritt [11] presents a hybrid approach which combines genetic algorithms and Bayesian belief networks to derive structures based on sequences with a strong cause and effect relationship. The research set out below is based on an Apriori approach but introduces a novel criterion for sequence selection which evaluates sequence plausibility and coherence in terms of network topology.

3 Topographical Proximity

The algorithms outlined in section 2 are capable of efficiently extracting thousands of event sequences in sequential input data. Therefore, post-processing remains an essential component of a usable mining system whereby sequences which are deemed to be uninteresting because they are redundant or simply implausible are eliminated from the output. The Topological Proximity (TP) approach 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 algorithm quantifies how closely alarm-generating elements are connected to each other in terms of the logical structure of a network using topographical information extracted from the alarms themselves. The general assumption is that

the more closely connected the alarm-generating elements, 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. At runtime, a measure of Topographical Proximity is used to reject or promote candidate sequences on the basis of their connectedness. Not only does this ensure that the output sequence set is plausible within the context of the network, but the space and time constraints of the data mining process are optimised as the algorithm uses both frequency and proximity to reduce the dimensions of the candidate sequence set, thereby restricting the search space of possible correlations. The measure may also be used during post-processing to rank sequences in terms of the connectedness of their constituent alarm events. Section 3.1 outlines how the TP measure is calculated based on a generic network topology. Section 3.2 describes how the measure has been integrated into the sequential mining process.

3.1 TP Calculation algorithm

The TP algorithm calculates the logical distance between alarm-generating network elements. The value has a minimum of zero for nodes that have no logical connection in the network and a maximum of one for nodes that have a very clear and close connection. TP calculation is based on the Radio Access Network of a standard UMTS telecommunication network which consists of functional nodes connected by communication interfaces and arranged in a logical, hierarchical structure, represented by the simplified schema in figure 1.[1] Each node in this system has functional subcomponents which may generate fault alarms which are then communicated to a designated Radio Access Network Management Node via a standard interface. Node subcomponents represent a node's internal functionality, the functionality of the interfaces between nodes or logical communications artefacts. The position of an alarm-generating node in the hierarchical structure is encoded in its full distinguished name, included in the source node attribute of each alarm.

In the context of this hierarchical network, the topographical proximity value for network elements on the same branch of the network is automatically assigned the maximum value of 1, to reflect the direct descendancy relation between the network elements, for example between Child1 and ParentNode1_1 or ParentNode2_2 and MasterNode2 in figure 1. For network elements that are not on the same branch of the network (e.g. Child5A and Child1 in figure 1), the topographical proximity value equates to a weighted traversal of the network branches or edges between two network elements. The TP value represents the total number of edges that must be traversed to find a path between the two elements. The weighting reflects the assumption that some hierarchical relations are closer than others. For the purposes of this analysis, logically, Child nodes form tighter clusters around Parent nodes than Parent nodes around Master nodes. This reflects the assumption that alarms on elements lower in the hierarchy may be more likely to share a common cause. Thus, nodes Child1 and Child2 in figure 1 are deemed closer in the context of the network than nodes ParentNode1_1 and ParentNode1_2.

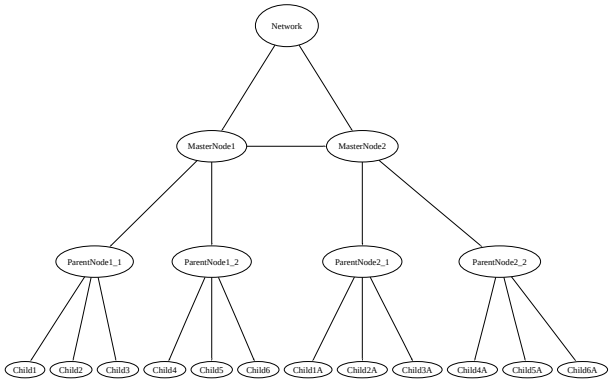

Figure 1. Simplified telecommunications network schema

For any two alarms, the source node attribute of each alarm is parsed to give the inheritance hierarchy of the network element with which that alarm is associated. The Topographical Proximity value for the two associated network elements is then calculated according to algorithm 1. Examples 2 to 4 below provide some sample TP values based on the network elements in figure 1.

$$TPcalculation(Child1, Child3) = 0.8 \quad (2)$$

$$TPcalculation(Child1, Child6) = 0.4 \quad (3)$$

$$TPcalculation(Child1, Child1A) = 0.05 \quad (4)$$

3.2 Integration of TP to the mining algorithm

The current implementation integrates the Topographical Proximity approach with the candidate gen-

[1] The TP calculation algorithm, however, is valid for any network which consists of functional nodes connected by interfaces and arranged in a logical structure. In [4], we describe how proximity values may be predefined as constants and assigned on the basis of shared and disjunct topographical information of alarm-generating nodes.

Algorithm 1 TP calculation algorithm

Input: 2 network elements, E1 and E2
Output: TP value, $0 \geq TP \leq 1$

```
TP = 0
if sameBranch(E1, E2) then
  Return 1
end if
if sharedParentNode(E1, E2) then
  TP+ = 0.4
end if
if sharedMasterNode(E1, E2) then
  TP+ = 0.35
end if
if sharedNetwork(E1, E2) then
  TP+ = 0.05
end if
Return TP
```

eration component of the MINEPI algorithm [8]. MINEPI generates candidate sequences of length n by combining two existing sequences of length $n-1$ and stores the minimal, or most compact, occurrences of all frequent sequences for subsequent iterations. Our algorithm filters all occurrences of candidate sequences on the basis of their connectedness within the network, as represented by the TP value calculated for the alarm-generating network elements. This filtering can be implemented in one of two ways:

1. Store minimal occurrences of all sequences above a given TP threshold;

2. Store the occurrences with the highest TP value of all sequences.

In the first case, the space constraints of the system are optimised for sequence compactness, in the second for sequence connectedness. In order to compare the performance of the original Minepi algorithm with that of the Topographical Proximity approach, the experiment reported in section 4 take the first approach using the TP value to prune the candidate set rather than to explicitly optimise sequence storage. The final step, as with Minepi, prunes the remaining candidate set based on a support (i.e. frequency) threshold.

Each minimal occurrence of a sequence has an associated proximity value. For sequences of length two, the TP value is calculated according to algorithm 1. For longer sequences, the TP value is the mean of the TP values for the two existing occurrences to be merged and the proximity value calculated for the source nodes of the first and last alarms of the new candidate, as in algorithm 2. For example, candidate sequence 7 below

Algorithm 2 calculateSequenceTP

INPUT: $seq, \{alarm_1, alarm_2 \ldots alarm_n\}$
OUTPUT: TPvalue
if $length(seq) == 2$ **then**
  return $calculateTP(alarm_1, alarm_2)$
else
  TP_{seq1} = Retrieve from memory $TP_{alarm_{1\cdots(n-1)}}$
  TP_{seq2} = Retrieve from memory $TP_{alarm_{2\cdots n}}$
  $TP_{new} = calculateTP(alarm_1, alarm_n)$
  return $\frac{TP_{seq1}+TP_{seq2}+TP_{new}}{3}$
end if

is composed of subsequences 5 and 6.[2]

$$Seq\_1 = Child1, Child3, MasterNode1 \quad (5)$$

$$Seq\_2 = \quad Child3, MasterNode1, Child1A \quad (6)$$

$$Seq \quad = Child1, Child3, MasterNode1, Child1A \quad (7)$$

TP_{Seq}, the TP value for the new candidate sequence Seq, is calculated as follows, where the only new TP calculation evaluates the connection between Child1 and Child1A:

$$TP_{Seq} = \frac{TP_{Seq1} + TP_{Seq2} + TPcalc(Child1, Child1A)}{3} \quad (8)$$

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, the cost is offset by the reduction in the search space of candidate sequences at each iteration achieved by imposing a minimum TP value threshold. Unlike a support threshold, the TP threshold is not an arbitrary means of reducing the set of candidate sequences. The TP threshold can be set to reflect domain experts intuitions regarding what connections constitute plausible sequences in their network. A support threshold is imposed after the TP threshold but the frequency constraint can be more flexible given the candidate sequences set is pre-pruned for proximity. Section 5 explores how the use of the topographical proximity threshold interacts with the standard mining parameters of maximum sequence duration and minimum support value to obtain optimum results in a qualitative evaluation of mined sequences.

4 Experiments

A set of experiments was conducted in order to provide a qualitative evaluation of the mining algorithm at

[2]For the purposes of illustrating the TP calculation, the alarms in the sample sequences are represented by their source nodes. The examples refer to the simplified network in figure 1.

different topographical proximity thresholds. To date, research has tended to focus on system performance, justifiably given the intensive computation involved in the mining process. What has been notably lacking, however, is an evaluation of the *quality* of the mined sequences. The experiment described below aims to address this shortfall. To this end, the mining task has been formulated as one of identifying specific target sequences 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 Test Cases

For the purposes of this experiment, the time window and minimum support system parameters were tested within the ranges of 60-600 seconds at 60 second intervals and 25-175 occurrences at intervals of 25, respectively. This gives a total of 70 test cases (10 *time windows* $*$ 7 *support values*) for each Topographical Proximity (TP) threshold value. For each time window and support parameter combination, baseline system performance of Minepi without Topographical Proximity ($TP = 0$) was calculated. Six further test cases for each parameter combination were evaluated at $TP = \{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. The aim was to determine optimum system parameters and TP threshold values from the 490 $(10 * 7 * 7)$ test cases and to establish whether the imposition of a TP threshold improved the quality of the output sequence set.

4.2 Methodology

Most commercially available alarm management systems are fully dependent on the expertise and experience of network analyst to derive rules for filtering and correlating alarms. This experiment aims to provide a global measure of the quality of the performance of the mining algorithm evaluated in the context of the domain knowledge of such experts. This objective has been formulated as the task of identifying in live network data common alarm sequences specified by network analysts.

Dataset. The basic dataset for the experiments consists of 96,991 alarms from the Radio Access Network (RAN) of a live telecommunications network.The alarm format conforms to telecoms 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 details.

Target Sequences Set. The quality of the output frequent event sequences must be evaluated relative to the frequency of known event sequences in the input data. In order to compile a target set of event sequences, a detailed statistical analysis of the alarm data was conducted by network experts. The analysis focused on the most frequently occurring individual alarms in order to identify repeating alarms and suspected correlations among the frequent alarm set. The results was a target set of twenty event sequences consisting of eighteen repeating alarm sequences, nine of length two events and nine of length four, and two inter-event correlations of length two and four. This set of twenty sequences represents a baseline of gold standard sequences which experts extrapolate from the dataset and which the algorithm should identify in the dataset.

Procedure. The mining algorithm was run on the dataset of 96,991 alarms for the 490 test cases set out in section 4.1. For each test case, three performance metrics were calculated based on the number of target sequences from the set of twenty target sequences identified for these parameters and threshold values.

4.3 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 sequences relative to the total number of sequences found by the system.

$$Precision = \frac{\text{Number of target sequences found}}{\text{Total number of sequences found}}$$

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

$$Recall = \frac{\text{Number of target sequences found}}{\text{Number of sequences in the target set}}$$

A high precision value indicates that the algorithm is selective and does not identify many spurious sequences. A high recall value indicates that the algorithm is accurate, successfully identifying most of the target sequences. These two metrics are combined to give a single indicator of system performance, the F Score representing the trade-off between these two indicators of precision and accuracy. A high F Score value indicates that the algorithm is both selective and

accurate with respect to the target sequence set. The F Score is calculated according to the following formula:

- $FScore = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

The performance metrics were calculated for perfect matches of target sequences identified by the system. They 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. Results are presented and discussed in section 5.

5 Results

In order to isolate the impact of the Topographical Proximity value on system performance in this experiment, the effects of the time window and support parameters were analysed. The ten graphs in figure 2 illustrate performance for each of the ten time windows from 60 to 600. Each graph plots the three performance metrics of precision, recall and F Score for all TP value thresholds: the Minepi baseline ($TP = 0$) and $TP = \{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. The figure illustrates that there is little variation in performance across the ten time windows. This would strongly suggest that window size is not a significant factor in the task of identifying target sequences. This can be attributed to the fact that the sequences are short in duration and therefore should be identified at all window sizes above 60 seconds. Figure 2 presents the results derived using a support threshold of 100 but results at all support thresholds exhibit the same characteristics.

The minimum support parameter, however, has a much greater effect on sequence identification, as illustrated in figure 3. The seven subplots demonstrate system performance for support values $\{25, 50, 75, 100, 125, 150, 175\}$ at a time window of 240 seconds. The plots show quite different behaviour for the seven support thresholds. It is therefore in the context of these seven experimental conditions represented by the seven support thresholds that we evaluate the effect of the TP value on system performance.

Figure 3 demonstrates a clear trend across all support value thresholds: as the TP threshold increases, there is a decrease in recall with a corresponding increase in precision, giving an overall increase in F Score value. This trend reflects the trade-off between reproducing the target sequence set in the output and generating a more restricted and, therefore, precise set of output sequences. The trade-off is such that, despite the reduction in recall values, overall performance, represented by the FScore value, improves as higher TP thresholds are enforced. This result validates expectations that restricting the sequence selection process to only accept topographically plausible sequences will significantly reduce the number of spurious sequences identified, thereby reducing the search space at runtime and facilitating post-processing. Furthermore, the reduction in recall values, particularly for $TP \geq 0.7$, may be addressed by employing the second strategy outlined in § 3.2 of optimising sequence storage with reference to proximity rather than sequence duration.

The results reported here would suggest that the use of the topographical proximity value yields a favourable trade-off between accuracy and recall for sequential data mining of telecoms alarm data. Furthermore, we would suggest that the output sequence set for higher TP thresholds more accurately represent the opinion of domain experts that:

- interesting correlations occur on related or connected nodes;
- frequency alone may not be an appropriate criterion for identifying noteworthy sequences in telecommunications data.

6 Future Work

The research reported in this paper suggests two complementary directions for future work. The first is to extend the topographical proximity measure to a broader sequence validation methodology. This can be addressed by identifying those attributes of individual alarms, which are significant, not for classifying individual alarms into types, but at the sequence level for validating alarm sequences. Future implementations aim to exploit attributes such as alarm severity and probable cause to generate a more refined measure of sequence plausibility by which to constrain the sequence generation process. Furthermore, the algorithm described in this paper assumes a simplified and homogeneous network topology. This is an oversimplification which needs to be addressed in future development by exploiting other explicit connections within the telecoms network.

The second key extension to the current research regards the qualitative evaluation of sequential mining algorithms. The experiment described above infers a target sequence set from domain experts analysis of the input data. Devitt et al [4] describe an experiment which uses a silver standard of alarm data with synthetic sequences inserted in known quantities and distributions into the data. A further step would require the development of a gold standard dataset for telecoms alarm data where all significant and interesting

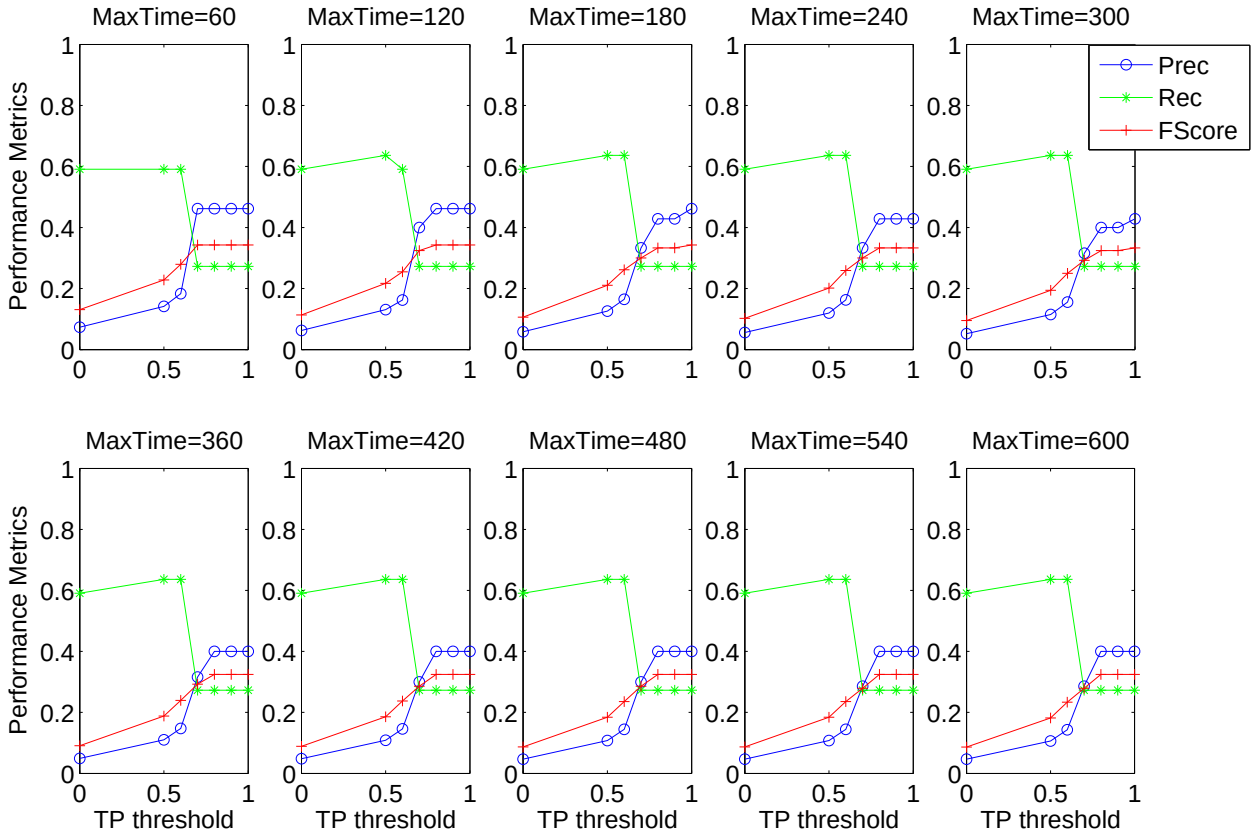

Figure 2. Performance metrics by TP threshold, $60 \leq timeWindow \leq 600$**,** $support = 100$**.**

correlations have been tagged in the data by domain experts.

7 Conclusions

The main contribution of this paper is to introduce the Topographical Proximity (TP) approach for sequential mining of telecommunications alarm data. 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 strongly suggest that the performance of the mining algorithm improves with the inclusion of the TP measure.

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 *Proceedings of the ACM-SIGMOD 1993 International Conference on Management of Data*, pages 207–216, Washington, D.C., may 1993.

[3] H. Cheng, X. Yan, and J. Han. Incspan: Incremental mining of sequential patterns in large databases. In *Proceedings of KDD 2004*, 2004.

[4] A. Devitt, J. Duffin, and R. Moloney. Exploiting network topology in mining sequential patterns from telecommunications alarm data. In *Proc. of SIGCOMM 2005, MineNet Workshop*, pages 179–184, 2005.

[5] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-

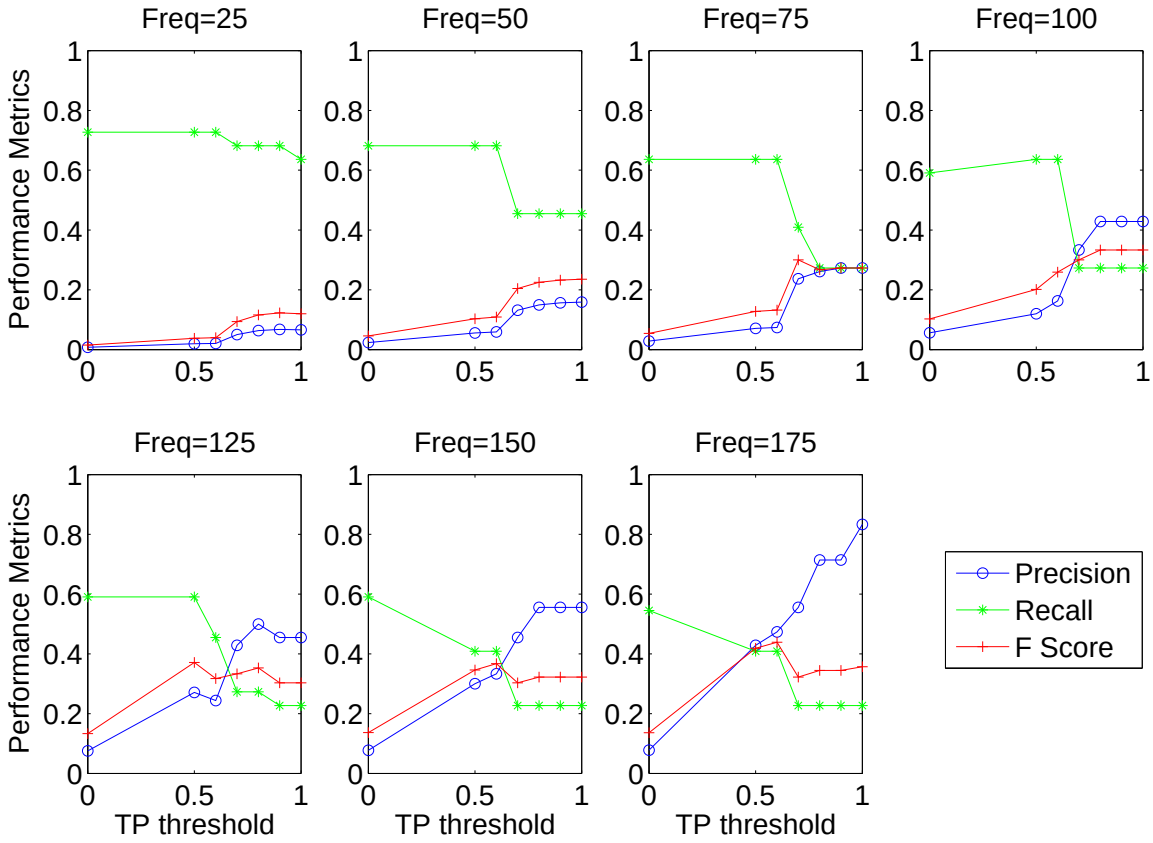

Figure 3. Performance metrics by TP threshold, $25 \leq support \leq 175$**,** $timeWindow = 240$**.**

pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

[6] 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.

[7] ITU. Itu-t recommendations: M.3030 principles for a telecommunication management network, 1988.

[8] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, 1997.

[9] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(10), 2004.

[10] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology, EDBT*, 1996.

[11] R. Sterritt. Discovering rules for fault management. In *Proceedings of the Eighth Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS '01)*, pages 190–196, Apr. 2001.

[12] G. M. Weiss. Predicting telecommunication equipment failures from sequences of network alarms. In W. Kloesgen and J. Zytkow, editors, *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, 2002.

[13] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60, 2001.