A Case Study in Sequential Pattern Mining for IT-Operational Risk

Valerio Grossi, Andrea Romei, and Salvatore Ruggieri

Dipartimento di Informatica, Università di Pisa
Largo Bruno Pontecorvo 3, 56127 Pisa ITALY
{vgrossi,romei,ruggieri}@di.unipi.it

Abstract. IT-operational risk management consists of identifying, assessing, monitoring and mitigating the adverse risks of loss resulting from hardware and software system failures. We present a case study in IT-operational risk measurement in the context of a network of Private Branch eXchanges (PBXs). The approach relies on preprocessing and data mining tasks for the extraction of sequential patterns and their exploitation in the definition of a measure called *expected risk*.

Keywords: sequential pattern, pre and post-processing, operational risk.

1 Introduction

According to the International Convergence of Capital Measurement and Capital Standards, known as Basel II [5], *operational risk* is "the risk of loss resulting from inadequate or failed internal processes, people and systems, or from external events". In the specific event of business disruption and system failures (e.g., hardware and software failures), the term Information-Technology (IT) operational risk is adopted.

Operational risk management consists of identifying, assessing, monitoring and mitigating the most potentially adverse risks [2, 4, 7]. On the basis of the risk management evaluation of an organization, the board of directors, regulations, shareholders, or the highly competitive market may require the organization to revise its internal processes, to set aside capital, to subscribe insurance policies, or to make investments in order to cover the residual risk.

Risk identification and assessment is conducted at the level of business units or processes by self-assessment against a checklist of potential vulnerabilities, or by collecting a set of statistics or metrics called risk indicators, or – by increasing the sophistication level – by formal risk quantification against measures of the distributions of frequency and impact of losses. In this sense, the risk of an event is formally defined as the "probability of the event" $\times$ "loss due to the event".

Risk monitoring and mitigation consists of regularly monitoring operational loss events, providing early warning indicators of an increased risk of future losses, and promptly mitigating the risk by reducing the exposure to, or the frequency and/or the impact of loss events.

The literature on operational, financial and market risk assessment accounts for contributions from statistics and simulation. Statistical models are based on the characterization of loss distributions or, at least, of certain parameters such as the expected loss and the tail loss. High frequency low impact risks (such as transaction processing errors) are modelled by the expected loss and the standard deviation of loss. Risk mitigation consists of acting on the organization processes, infrastructure and personnel. Low frequency high impact risks (such as frauds or earthquakes) are modelled by the tail of the loss distribution. Risk mitigation consists of setting aside capital or to subscribe an insurance. Existing statistical approaches consider mainly the low frequency high impact risks, such as in the approaches of Value at Risk [8], coherent measures [3], and extreme value theory [9]. Bayesian Networks have been adopted [2, Chapter 14] [6] as a powerful tool to cope with shortage of data (as in rare events), to integrate qualitative prior knowledge (such as expert opinions), and to make what-if scenario analyses.

In this paper, we concentrate on the high frequency low impact class of risk by reporting a case study in IT-operational risk in the context of a network of Private Branch eXchanges (PBX). We adopt sequential pattern mining on weighted sequences for the purpose of defining and validating the notion of *expected risk* as a predictive measure of the risk in managing the network of PBX's. We report the problems found and the solutions adopted both in the data preprocessing task and in the sequential pattern extraction and deployment task. To the best of our knowledge, this is the first paper reporting the implementation of a KDD process in the IT-operational risk context.

2 Case Study Specification

2.1 Monitoring a Network of PBX's

We introduce a case study concerning the management of a network of PBX's by a Communication Service Provider (CSP). The customers of the CSP are small-medium enterprisers requiring both voice and data lines at their premises at different contractually agreed quality of services. The customers externalize to the CSP the maintenance of the PBX's and the actual management of the communication services. When a malfunctioning occurs, customers refer to the CSP call center, which can act remotely on the PBX, e.g., to reboot the system. If the problem is not recoverable remotely, as in the case of hardware failure, a technician is sent on-site. Both call center contacts and technician reports are logged in the CSP customer relationship management database.

A PBX is doubly redundant, i.e., it actually consists of two independent communication apparatuses and a self-monitoring software. Automatic alarms produced by the equipment are recorded in the PBX system log. Call center operators can access the system log to control the status of the PBX. Also, a centralized monitoring software collects on a regular basis system logs from all the installed PBX's.

Among the operational risk events, PBX malfunctioning may have different impact on the CSP. At one extreme, the call center operator can immediately solve the problem. At the other extreme, a technician intervention may be required, with the customer's offices inoperative in the meantime, and the risk that the agreed quality of service could not be guaranteed. To record the impact of a malfunctioning, the severity level of the problem occurred is evaluated and documented by the technician. In this context, our case study aims at:

- the Extraction, Transformation and Loading (ETL) into a merged database of the available sources of data, including customer type information, call center logs, technicians reports, and PBX system logs;
- the characterization of early warnings of problems in terms of typical sequences of alarms that lead to them;
- the exploitation of the sequential patterns extracted for automatic on-line malfunctioning prediction and risk quantification.

2.2 Data Sources

Data has been provided by a leading regional CSP. The CSP's customer relationship management system records in the [TECH-DB] table the history of technician interventions at the customer premises. For each problem, at least the following attributes of information are available.

Attribute	**Description**
`Date`	Problem opening date and time
`PBX-ID`	Unique ID of the Private Branch eXchange
`CType`	Customer line of business
`Tech-ID`	Unique ID of technician's intervention
`Severity`	Problem severity recorded after problem solution
`Prob-ID`	Problem type recorded after problem solution

Table 1. [TECH-DB] Log of technician on-site interventions.

The `Date` attribute consists of the problem opening date and time, defined as the time the call center receives a customer call reporting the notice of a malfunctioning. The `PBX-ID` is a unique identifier of the involved PBX. If a customer has installed more than one PBX, this is determined by the call center operator based on the customer description of the problem and the available configuration of PBX's installed at the customer premise. `CType` is the customer line of business, accordingly to a CRM categorization including: banks, health care, industry, insurance and telecommunication businesses. The `Tech-ID` attribute is a unique identifier of the technician intervention: during a same intervention one or more problems may be tackled. `Severity` is a measure of the impact of the problem. It is defined on a scale from 1 to 3 as follows:

3 critical, service unavailable;
2 medium, intermittent service interruptions;
1 low level problem.

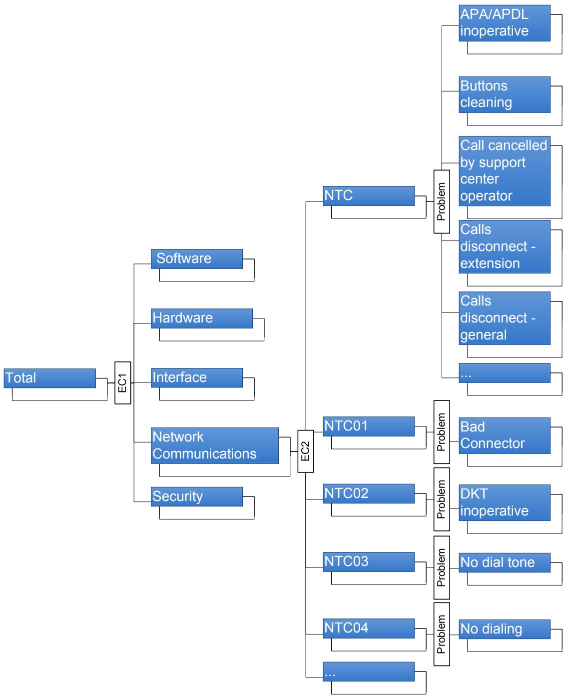

Fig. 1. A hierarchy of problem types for PBX malfunctioning.

Finally, the `Prob-ID` attribute is a coding of the malfunctioning solved by the technician. Two hundred problem descriptions are codified.

The second data source is the collection of logs generated by PBX's. Logs are downloaded into a centralized repository, called [ALARM-DB], on a regular round-robin basis.

Attribute	**Description**
`PBX-ID`	Unique ID of the Private Branch eXchange
`TestDate`	Date and time log was downloaded
`Alarms`	Sequence of alarms raised since last log download

Table 2. [ALARM-DB] Log of PBX alarms.

For a given PBX identifier `PBX-ID` and date-time of log download `TestDate`, [ALARM-DB] stores the set of alarms raised by the PBX since the previous log download. Sixteen distinct alarms are available in the data. Unfortunately, the precise time an alarm is raised is not stored in the PBX log.

3 Preprocessing IT-operational Logs

PBX logs stored in the [ALARM-DB] table and technician's reports stored in the [TECH-DB] table are not directly suitable as an input for sequential pattern mining. In this section, we report two main issues with pre-processing those data in order to yield a database of sequences. The first issue is concerned with the granularity level in the analysis of PBX problem types. The second one with the problem of building the sequences of alarms related to a PBX problem. Preprocessing has been automated as a collection of ETL data flows developed using the data integration module of the Pentaho suite [11]. A GUI written in Java puts together the various data flows in a stand-alone application.

3.1 A Hierarchy of Problem Types

Since two hundreds problem types it is too fine-grained detail, problem types are organized in a three level hierarchy, which is partly shown in Figure 1. The lowest level is the problem type. The highest level (EC1) consists of the Basel II event categories for IT-operational risk: software, hardware, interface, security and network communications. The middle level (EC2) is an intermediate categorization left at the choice of the user during the preprocessing phase.

Every problem type readily falls in one of the five EC1 level members. However, the mapping cannot be automated for a new problem type, and then the preprocessing GUI asks the user to provide the EC1 and EC2 levels for a previously unseen problem type. In the rest of this paper, we will concentrate our attention at the level of EC1, but we point out that, if large volume of data is available, the overall approach can be followed at finer levels of detail.

3.2 An Heuristics for Joining Alarm and Problem Logs

For a technician intervention, the sequence of alarms generated by the involved PBX has to be reconstructed in order to relate a malfunctioning of the PBX with the alarms raised by it. Unfortunately, the problem cannot be directly solved. While a technician intervention records the timestamp it occurred, the alarm log contains the timestamp an alarm is downloaded to the central repository, not the timestamp the alarm is raised. Since alarm log collection may occur once every a few days, this is an issue.

We adopt the following heuristics. Let *Opening* be the timestamp a problem for a PBX is opened. This is available in the [TECH-DB] table. We leave the user the choice to join the problem with the sequence of alarms collected in the time interval [*Opening-Diff1*, *Opening+Diff2*], where *Diff1* and *Diff2* are parameters of the preprocessing procedure.

3.3 A Database of Sequences

The preprocessing of available input produced a database of 1899 sequences, spanning over a period of four months, of the form:

```
Date: 22/02/2007 8.36
CType: Bank
PBX-ID: 90333
Prob-ID: Hardware
Severity: 2
AlarmSequence:
   {CARD}:19/02/2007 18:34:01;
   {}:20/02/2007 18:12:53;
   {PCM TIME SLOT}:21/02/2007 17:15:21
--
Date: 21/02/2007 16.54
CType: Health
PBX-ID: 91993
Prob-ID: Security
Severity: 2
AlarmSequence:
   {DIGITAL TRUNK CARD,DKT SUBUNIT}:19/02/2007 17:56:47;
   {POWER_SUPPLY}:21/02/2007 09:10:07
```

The attributes `Date`, `CType`, `PBX-ID` and `Severity` are directly taken from [TECH-DB]. `Prob-ID` is obtained by lifting the problem description to its EC1 category in the hierarchy of Figure 1. Finally, the sequence of sets of alarms, labelled by the date each set is collected, is obtained by joining alarm and problem logs. In the first sequence above, a `CARD` alarm is raised on 19/02/2007, then no alarm on the next day, a `PCM TIME SLOT` alarm on the 21/02/2007, and finally a malfunctioning is reported to the call center on 22/02/2007. As parameters for the joining heuristics, we have set *Diff1* to three months and *Diff2* to one day. Setting *Diff1* as large as possible is desirable, but it is important not to overlap with the time interval of a past problem for the same PBX. Concerning *Diff2*, it should be set to the average period that alarms are collected into the centralized repository. Assuming the period is one day, an alarm raised the same day of a customer call could or could not be already processed by ETL flows at the time of the call. Without any further information, and considering that such alarms are the most valuable for prediction, we assume they have been processed.

4 Sequential Pattern Mining for Risk Assessment

4.1 Sequences and Sequential Patterns

Let us recall a few standard definitions [1, 10]. Given a finite set $\mathcal{I}$ of items, an itemset T is a subset of $\mathcal{I}$. A sequence is an ordered list of itemsets $\langle T_1, \ldots, T_n \rangle$. For brevity, we write $\langle t_1, \ldots, t_n \rangle$ if for $i = 1 \ldots n$, $T_i = \{t_i\}$, i.e., all T_i's are singletons. A sequence $s_1 = \langle T_1, \ldots, T_n \rangle$ is a sub-sequence of another sequence $s_2 = \langle S_1, \ldots, S_m \rangle$ (or s_2 is a super-sequence of s_1), denoted as $s_1 \sqsubseteq s_2$, if there exists $1 \leq p_1 < \ldots < p_n \leq m$ such that for $i = 1 \ldots n$, $T_i \subseteq S_{p_i}$. The concatenation of two sequences s_1 and s_2 is denoted by $s_1 \cdot s_2$. A sequence database $\mathcal{D}$ is a collection of sequences. The (relative) support (w.r.t. $\mathcal{D}$) of a sequence s is defined as the fraction of sequences in $\mathcal{D}$ which are sub-sequences of s, i.e., $supp(s) = |\{s' \in \mathcal{D} \mid s' \sqsubseteq s\}|/|\mathcal{D}|$. A sequential pattern is a sequence $SP = \langle T_1, \ldots, T_n \rangle$ such that $supp(SP) \geq \xi$, where ξ is a fixed minimum support threshold. For notational purposes, it is convenient to label sequential patterns with their support, and to differentiate them from sequences. Therefore, we write the sequential pattern SP in the form $T_1 \rightarrow T_2 \rightarrow \ldots T_n[s]$, where $s = supp(SP)$. A sequence $s \in \mathcal{D}$ supports SP if $SP \sqsubseteq s$. A sequential pattern is maximal if there is no other sequential pattern that is a super-sequence of it. We denote by $Max\ \mathcal{S}$ the set of maximal sequential patterns in $\mathcal{S}$. Restricting to maximal sequential patterns alleviates the problem of dealing with an exponential number of extracted patterns. Several algorithms have been proposed in the literature to extract (maximal) sequential patterns [10].

4.2 Adapting Sequences and Sequential Patterns

For the problem under consideration, we will make use of a variant of sequential pattern mining, where sequences and sequential patterns are weighted.

In our context, we fix the set of items $\mathcal{I}$ to include alarms identifiers, problem type items `Prob-ID`$= \beta$ and business line items `CType`$= \alpha$, where β is an EC1-level problem type and α is the customer line of business. A sequence is now of the form:

$$\texttt{CType} = \alpha\ ,\ \texttt{AlarmSet}_1\ ,\ \ldots \texttt{AlarmSet}_k\ ,\ \texttt{Prob-ID} = \beta\ \texttt{[}sev\texttt{]}$$

For an occurrence of a PBX problem recorded in the technician's database, a sequence models the temporal succession of alarms $\texttt{AlarmSet}_i$ raised by the PBX. The heuristics described in Sect. 3.2 is adopted in order to join a problem occurrence with the succession of alarms related to it.

A sequence starts with `CType` $= \alpha$, ends with `Prob-ID` $= \beta$, and it is labelled with the problem severity *sev*:

- Starting with `CType` $= \alpha$ is a work-around to differentiate the sequences on the basis of the type of business of customers. This is motivated by the requirement that risk management has to differentiate risk for line of business or processes. Therefore, we are interested in modeling specific patterns of problems due to the different usages of the PBX network by different businesses.
- Similarly, ending the sequence with `Prob-ID` $= \beta$ allows for isolating patterns that lead to a specific problem from patterns that hold in general.
- Labelling the sequence with the problem severity is a weighting strategy, based on the impact of the problem occurred. For a sequence s, we write $SEV(s)$ to denote its severity label.

As a result of the above definition of sequences, we are interested in extracting from a database of sequences $\mathcal{TS}$, which we call the training set, sequential patterns SP of the form:

$$\texttt{CType} = \alpha \rightarrow \texttt{AlarmSet}_1 \rightarrow \ldots \texttt{AlarmSet}_n \rightarrow \texttt{Prob-ID} = \beta\ \texttt{[}supp\texttt{,}sev\texttt{]} \qquad (\star)$$

where:

- the severity label sev is $(\Sigma_{s \in \mathcal{TS}, SP \sqsubseteq s} SEV(s))/|\{s \in \mathcal{TS} \mid SP \sqsubseteq s\}|$, i.e., the average severity of sequences in the training set supporting the sequential pattern;
- the support label $supp$ is $supp(SP)/supp(\langle \texttt{CType} = \alpha,\ \texttt{Prob-ID} = \beta \rangle)$, i.e., the relative support of the sequential pattern w.r.t. the number of sequences starting with $\texttt{CType} = \alpha$ and ending with $\texttt{Prob-ID} = \beta$.

Sequential patterns of the form $(\star)$ can be extracted by the following procedure. First, split the sequence database into one database for each distinct pair $(\texttt{CType} = \alpha, \texttt{Prob-ID} = \beta)$; then run any sequential pattern extraction algorithm from the literature on each sequence database and for a specified minimum support threshold; then remove extracted sequential patterns not including an item Prob-ID as the last item; finally, calculate severity of a sequential pattern by averaging severities of the sequences supporting it. Alternatively, multidimensional [12] or a context-based [13] approaches to sequential pattern mining could be adapted.

4.3 Mean Risk

Consider an initial fragment $s_1 = \langle \texttt{CType} = \alpha_1, \texttt{AlarmSet}_1, \ldots, \texttt{AlarmSet}_k \rangle$ of a sequence. Assume for the moment that we do not or cannot exploit any sequential pattern. How can we then define the risk that s is followed by $\texttt{Prob-ID} = \beta$? Since the risk of an event is "probability of the event" $\times$ "loss due to the event", we approximate:

- the "probability of the event" as the ratio:

$$p_{\alpha,\beta} = \frac{supp(\langle \texttt{CType} = \alpha, \texttt{Prob-ID} = \beta \rangle)}{supp(\langle \texttt{CType} = \alpha \rangle)},$$

  i.e., the confidence that a sequence in the training set starting with $\texttt{CType} = \alpha$ will be followed by $\texttt{Prob-ID} = \beta$;
- the "loss due to the event" as the average severity of sequences in the training set starting with $\texttt{CType} = \alpha$ and ending with $\texttt{Prob-ID} = \beta$:

$$l_{\alpha,\beta} = \frac{\Sigma_{s \in \mathcal{TS}, \langle \texttt{CType} = \alpha, \texttt{Prob-ID} = \beta \rangle \sqsubseteq s} SEV(s)}{|\{s \in \mathcal{TS} \mid \langle \texttt{CType} = \alpha, \texttt{Prob-ID} = \beta \rangle \sqsubseteq s\}|}.$$

We define the *mean risk* that the initial fragment s_1 starting with $\texttt{CType} = \alpha$ will end with $\texttt{Prob-ID} = \beta$ as $p_{\alpha,\beta} \times l_{\alpha,\beta}$. If s_1 does not start with $\texttt{CType} = \alpha$, i.e. $\alpha_1 \neq \alpha$, then the mean risk is zero, as one could expect. Formally:

$$\textrm{MR}\textsc{isk}(\alpha, \beta, s_1) = \begin{cases} p_{\alpha,\beta} \times l_{\alpha,\beta} & \text{if } \langle \texttt{CType} = \alpha \rangle \sqsubseteq s_1 \\ 0 & \text{otherwise.} \end{cases}$$

Summary mean risk for a line of business w.r.t. all problem types is defined as the sum of the risk for individual problem types:

$$\mathrm{SMR}\textsc{isk}(\alpha, s_1) = \Sigma_{\beta \in dom(\texttt{Prob-ID})} \mathrm{MR}\textsc{isk}(\alpha, \beta, s_1).$$

When $\alpha = \alpha_1$, it can be rewritten as: $1 \times \frac{\Sigma_{s \in \mathcal{TS}, \langle \texttt{CType}=\alpha \rangle \sqsubseteq s} SEV(s)}{|\{s \in \mathcal{TS} \mid \langle \texttt{CType}=\alpha \rangle \sqsubseteq s\}|}$, namely "probability of any problem" $\times$ "average severity" for the line of business. Similarly, we define the summary mean risk for a problem type w.r.t. all lines of business:

$$\mathrm{SMR}\textsc{isk}(\beta, s_1) = \mathrm{MR}\textsc{isk}(\alpha_1, \beta, s_1) = \Sigma_{\alpha \in dom(\texttt{CType})} \mathrm{MR}\textsc{isk}(\alpha, \beta, s_1).$$

The three definitions above extend to a set S of initial fragments by summing up the individual risk of each element in S.

4.4 From Sequential Patterns to Expected Risk

Consider now a specific initial fragment $s = \langle$`CType=Bank`, `CARD` , `CARD SUBUNIT`, `DIGITAL TRUNK CARD`$\rangle$ of a sequence. We would like to adopt the sequential patterns extracted from the training set to the purpose of enhancing the notion of mean risk to a notion called *expected risk*. Intuitively, we start by looking at the sequential patterns supported by s (with the exclusion of the last item in the sequential pattern – i.e., of the `Prob-ID` item). Assume that the set $\mathcal{SP}$ of such sequential patterns consists of:

SP$_0$ `CType=Bank` → `CARD SUBUNIT` → `Prob-ID=Software` [0.4, 3]
SP$_1$ `CType=Bank` → `CARD` → `Prob-ID=Hardware` [0.2, 2]
SP$_2$ `CType=Bank` → `CARD SUBUNIT` → `Prob-ID=Hardware` [0.3, 3]
SP$_3$ `CType=Bank` → `CARD` → `CARD SUBUNIT` → `Prob-ID=Hardware` [0.1, 3]
SP$_4$ `CType=Bank` → `DIGITAL TRUNK CARD` → `Prob-ID=Hardware` [0.2, 2]

For `Prob-ID=Software`, there is only one supported sequential pattern, namely SP_0. By recalling that the risk of an event is "probability of the event" $\times$ "loss due to the event", we set the expected risk (to have a problem with software) to $0.4 \times 3 = 1.2$.

Consider now `Prob-ID=Hardware`. First, we observe that SP_1 is a sub-sequence of SP_3, hence it is superseded by SP_3, and similarly for SP_2. Therefore, we restrict to maximal sequential patterns in $\mathcal{SP}$, namely to SP_3 and SP_4. We now define the expected risk to have a hardware problem by averaging the severities of SP_3 and SP_4 based on their support, i.e., as $(0.1 \times 3 + 0.2 \times 2)/(0.1 + 0.2) = 0.7/0.3 = 2.33$. Notice that we scale the average severity by dividing by the sum of the supports of the maximal sequential patterns. The motivation is twofold. On the one side, two (maximal) sequential patterns may have some common supporting sequence, hence the sum of their supports can be strictly greater than one. On the other side, the set of maximal sequential patterns may not cover all possible sequences (rare ones cannot be modeled by sequential patterns by definition), i.e., the sum of their supports can be strictly lower than one.

Finally, let us consider `Prob-ID=Interface`. There is no sequential pattern supported by s, at least for the fixed minimum support threshold. Therefore, the reasoning followed so far cannot be applied. Intuitively, we fall in the case that no sequential information is available, i.e., on the notion of mean risk, and then we set the expected risk to the mean risk.

Let us introduce some notation. Let $\mathcal{SP}$ be the set of sequential patterns extracted from the training set $\mathcal{TS}$. For a given initial fragment s_1 and problem type β, we define the set of sequential patterns supported by s_1 and that are maximal as:

$$\mathcal{M}(s_1, \beta) = Max\{SP \in \mathcal{SP} \mid SP \sqsubseteq s_1 \cdot \langle \texttt{Prob-ID} = \beta \rangle\}.$$

The *expected risk* is then defined as:

$$\mathrm{E}\textsc{Risk}(\alpha, \beta, s_1) = \begin{cases} \dfrac{\Sigma_{SP \in \mathcal{M}(s_1,\beta)} supp(SP) \times sev(SP)}{\Sigma_{SP \in \mathcal{M}(s_1,\beta)} supp(SP)} & \text{if } \mathcal{M}(s_1, \beta) \neq \emptyset \\ \mathrm{M}\textsc{Risk}(\alpha, \beta, s_1) & \text{otherwise.} \end{cases}$$

The definition readily extends to a set S of initial fragments, to summaries for line of business and problem type as follows:

$$\begin{aligned} \mathrm{E}\textsc{Risk}(\alpha, \beta, S) &= \Sigma_{s_1 \in S} \mathrm{E}\textsc{Risk}(\alpha, \beta, s_1) \\ \mathrm{SE}\textsc{Risk}(\alpha, S) &= \Sigma_{\beta \in dom(\texttt{Prob-ID})} \mathrm{E}\textsc{Risk}(\alpha, \beta, S) \\ \mathrm{SE}\textsc{Risk}(\beta, S) &= \Sigma_{\alpha \in dom(\texttt{CType})} \mathrm{E}\textsc{Risk}(\alpha, \beta, S). \end{aligned}$$

Notice that in the special case that there is no sequential pattern, i.e. $\mathcal{SP} = \emptyset$, these definitions collapse to the ones for the mean risk.

One could consider removing non-maximal sequential patterns off-line when sequential patterns are extracted, whilst now they are removed on-line when the expected risk is calculated. Unfortunately, this approach does not lead to the same results. In fact, consider an initial fragment $s = \langle \texttt{CType=Bank}, \texttt{ALARM}_1 \rangle$. If non-maximal sequential patterns are removed off-line, then SP_1 above could not be taken into consideration. Moreover, no maximal sequential pattern SP_3-SP_4 is supported by s. Summarizing, we have no sequential pattern to exploit in defining the expected risk of s.

4.5 Actual Risk

Once an initial fragment completes to a full sequence s, i.e., it is terminated by a problem type item $\texttt{Prob-ID} = \beta$ and labelled by severity $SEV(s)$, it is easy to calculate the involved risk. We define the *actual risk* of the sequence as $1 \times SEV(s)$, and this value contributes only to problem type β:

$$\mathrm{A}\textsc{Risk}(\alpha, \beta, s) = \begin{cases} SEV(s) & \text{if } \langle \texttt{CType} = \alpha, \texttt{Prob-ID} = \beta \rangle \sqsubseteq s \\ 0 & \text{otherwise.} \end{cases}$$

The measure readily extends to sets of sequences, and to summaries for lines of business and problem types as done for mean risk and expected risk.

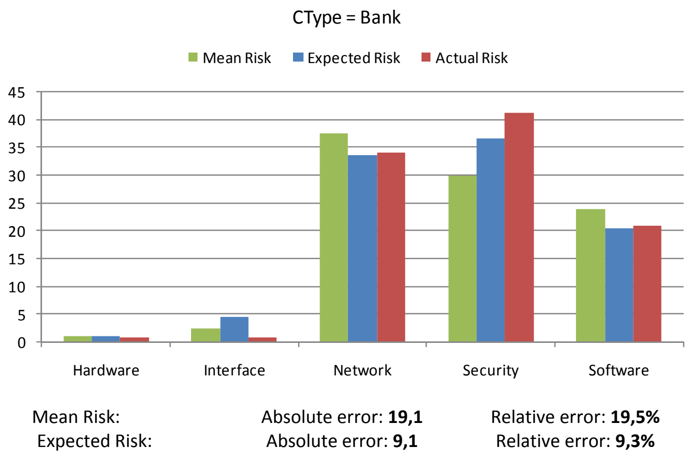

Fig. 2. Expected risk vs actual risk: detail for the `Bank` customer line of business.

5 Deploying Sequential Patterns

5.1 Application Scenario

Consider a set of sequential patterns extracted from a training set of past sequences. How can the notion of expected risk be turned into practice for risk assessment and mitigation? Of course, the analysis of the sequential patterns by a domain expert might highlight previously unknown patterns of alarms leading to malfunctioning with high average severity.

Besides this descriptive usage, we concentrate here on the deployment of extracted patterns in on-line risk assessment. Let us assume an application scenario where a call center operator receives a call from a customer reporting a malfunctioning. A ticket is open for dealing with the malfunctioning. At the time of the call, the following information is known: the customer (line of business), the involved PBX, the sequence of alarms of that PBX collected up to that time. In other words, an initial fragment s_1 is available, as assumed in Section 4.4.

A decision support system can then exploit the notion of expected risk to estimate the overall risk of the currently open tickets. This information is useful to assess the current level of risk for the various customer lines of business and problem types. Operatively, it can help in determining the needed effort in terms of required expertise for technician's interventions, e.g., how many software and hardware technicians should be alerted.

When the tickets are closed and the technician reports are available, the actual risk can be calculated. The comparison between the expected risk and the actual risk provides a measure of the accuracy of the notion of expected risk. Also, since expected risk is a refinement of mean risk, it is worth evaluating the improvement of accuracy of expected risk over mean risk. The next two subsections report experimental results on those two accuracy issues.

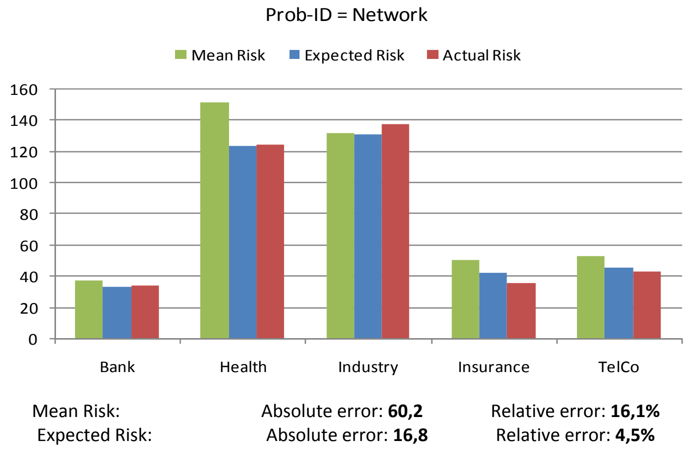

Fig. 3. Expected risk vs. actual risk: detail for `Network` problem type.

In particular, the experimental results are obtained by partitioning the available database of sequences (see Section 3.3) into temporally separated training set $\mathcal{TS}$ and test set $\mathcal{TE}$. The split date-time was set to obtain about a 75%-25% partitioning[1]. Moreover, we set the minimum support threshold in the sequential pattern extraction (see Section 4.2) to 5%.

5.2 Expected Risk vs Mean Risk vs Actual Risk

The overall expected risk for a business line α is represented as an histogram chart with the distinct values β for problem type `Prob-ID` on the X-axis and the value of $\text{ER}\textsc{isk}(\alpha, \beta, \mathcal{TE})$ on the Y-axis. The distributions of the mean risk and of the actual risk measures are represented in the same manner. The difference area between the expected and the actual risk histograms measures the error of adopting expected risk for estimating actual risk. More formally, the absolute error is defined as:

$$\Sigma_{\beta \in dom(\texttt{Prob-ID})} abs(\text{ER}\textsc{isk}(\alpha, \beta, \mathcal{TE}) - \text{AR}\textsc{isk}(\alpha, \beta, \mathcal{TE}))$$

and the relative error is its ratio over $\Sigma_{\beta \in dom(\texttt{Prob-ID})} \text{AR}\textsc{isk}(\alpha, \beta, \mathcal{TE})$. Similar definitions can be stated for the mean risk.

Figure 2 shows the histogram charts of mean, expected and actual risks for `CType=Bank`, namely for the bank subnetwork of PBX's. Apart from the `Interface` problem type, the expected risk measure provides a much better estimation of actual risk than the mean risk. The relative errors for the various business lines are summarized in the following table:

[1] We observe that, in a more realistic scenario, the number of open tickets at a certain time is typically low, especially if compared to closed ones, since problems are usually solved within 48 hours time.

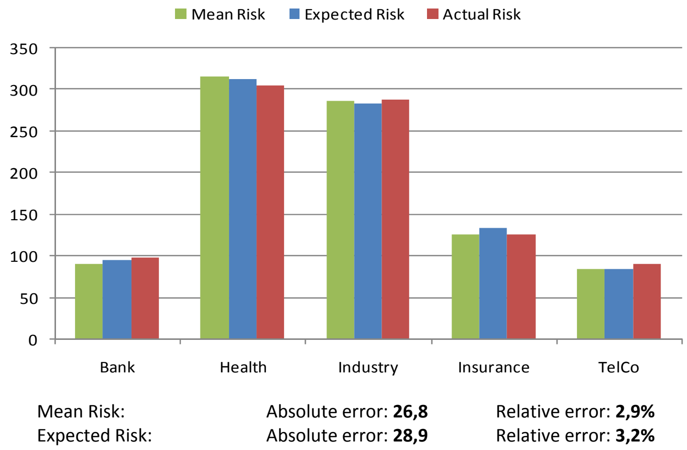

Fig. 4. Expected risk vs. actual risk: summary for lines of business.

CType	**No. Seq.**	**Actual Risk**	**Relative Error**	
			Mean Risk	**Expected Risk**
Bank	51	98	19.5%	9.3%
Health	185	306	17.0%	9.4%
Industry	159	289	6.8%	4.9%
Insurance	68	127	47.5%	25.0%
TelCo	47	91	31.6%	16.8%

As a general observation, the relative error of expected risk is near the half of the error of the mean risk. Figure 3 shows the histogram charts of mean, expected and actual risks for `Prob-ID=Network`, namely for the problem types related to the communication network of PBX's, and for the various customer business lines. Expected risk turns out to improve over mean risk for all business lines. Now the difference area between expected and actual risk charts is obtained as:

$$\Sigma_{\alpha \in dom(\texttt{CType})} abs(\text{ERisk}(\alpha, \beta, \mathcal{TE}) - \text{ARisk}(\alpha, \beta, \mathcal{TE})).$$

The relative errors for the various problem types are summarized next:

Prob-ID	**Actual Risk**	**Relative Error**	
		Mean Risk	**Expected Risk**
Hardware	35	20.6%	27.3%
Interface	24	42.6%	69.0%
Network	374	16.1%	4.5%
Security	241	28.2%	11.0%
Software	237	14.5%	12.5%

For the problem types `Hardware` and `Interface` there is a degradation of performances of expected risk over mean risk, whilst for the other problem types

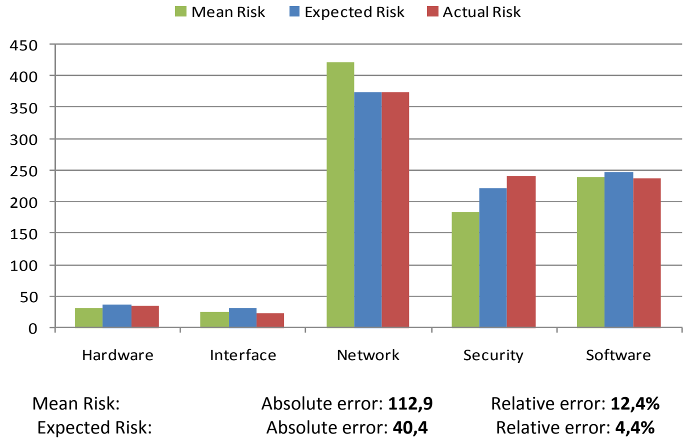

Fig. 5. Expected risk vs. actual risk: summary for problem types.

there is a gain. Notice, however, that there is a very low actual risk for `Hardware` and `Interface`, and hence a low number of sequences.

Consider now the summary measure $\text{SER}\textsc{isk}(\alpha, \mathcal{TE})$, which provides for a business line α the expected total risk w.r.t. all problem types. Figure 4 shows the histogram charts of mean, expected and actual risks. By averaging over a whole line of business, the predictive power of mean risk improves, and its refinement to expected risk yields no additional benefit. As stated in the introduction, this confirms that simple statistics, such as mean and standard deviation, are enough to deal with high-frequency (and necessarily low impact) loss events. As soon as we go into the details of a specific line of business, frequency decreases, and then sequential information leads to better predictive power.

Finally, Figure 5 shows the charts for the summary measure $\text{SER}\textsc{isk}\ (\beta, \mathcal{TE})$, which provides for a problem type β the expected total risk w.r.t. all business lines. By averaging on the problem type, expected risk improves over mean risk considerably.

5.3 Tuning the Parameters

Let us consider here a few issues concerning the choice of parameters in pattern extraction and deployment. Figure 6 shows how the expected risk error is affected by the minimum support threshold. The figure reports the relative error:

$$\frac{abs(\text{ER}\textsc{isk}(\texttt{Health}, \beta, \mathcal{TE}) - \text{AR}\textsc{isk}(\texttt{Health}, \beta, \mathcal{TE}))}{\text{AR}\textsc{isk}(\texttt{Health}, \beta, \mathcal{TE})}$$

for two sample β, and the total relative error:

$$\frac{\Sigma_{\beta \in dom(\texttt{Prob-ID})} abs(\text{ER}\textsc{isk}(\texttt{Health}, \beta, \mathcal{TE}) - \text{AR}\textsc{isk}(\texttt{Health}, \beta, \mathcal{TE}))}{\Sigma_{\beta \in dom(\texttt{Prob-ID})} \text{AR}\textsc{isk}(\texttt{Health}, \beta, \mathcal{TE})}$$

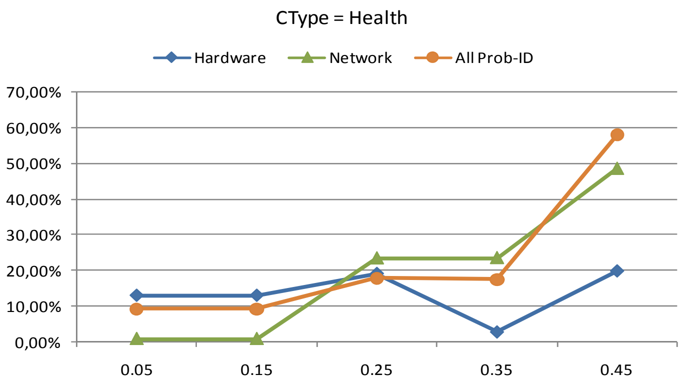

Fig. 6. Expected risk relative error by varying the minimum support.

for various minimum support thresholds. It is immediate to observe that lower minimum supports lead to lower errors. However, after reaching some minimum, the error does not improve and it eventually starts increasing.

Another choice we made was to partition the training set based on pairs (`CType` $= \alpha$, `Prob-ID` $= \beta$), and to extract sequential patterns from each partition. An alternative choice is to partition with respect to `CType` $= \alpha$ only, i.e., not forcing the extraction of sequential patterns for every problem type. This alternative leads to extract sequential patterns where only frequent problem types appear. We are more accurate for them, but less accurate for minority problem types. As an example, the relative error of expected risk for the `Bank` line of business in Figure 2 degrades to 23.2%.

Moreover, we have also conducted experiments on the maximum number of sequential patterns to be considered for a pair (`CType` $= \alpha$, `Prob-ID` $= \beta$) among those having a minimum support. In the experiments reported so far, the number was set to the top 5 (w.r.t. the support) for all pairs. Extensive experimentation shows that, for a same minimum support threshold, a large number of sequential patterns (> 20) leads to poorer performances, and that the optimal number might vary for each pair (`CType` $= \alpha$, `Prob-ID` $= \beta$). Therefore, a form of self-tuning of parameters might improve the reported results.

6 Conclusions

Is data mining suitable for IT-operational risk management? We believe that we provided an affirmative answer to the question – yet, preliminary and for a specific case study. For high frequency - low impact loss events, the extraction of frequent (sequential) patterns from past log of data can improve basic measures of risk, which rely only on simple statistics, such as in the case of the mean risk. The improvement consists of more accurate predictions for lower frequency events, as in the case of a specific line of business or problem type. Nevertheless, a frequent pattern approach, like the one proposed here, cannot deal with very low frequency events, which have very few occurrences in databases or none at all. Hence, our approach is complementary to the ones proposed in the statistical and simulation literature for low frequency - high impact loss events.

A further work we intend to pursue is to enhance the basic statistics with a classification based approach: all in all, the mean risk measure is a "decision stump" classifier on the `CType` attribute. A better classifier could be trained by using additional predictive attributes, such as the PBX hardware/software version, or, in order to evaluate the gain due to sequential information, the presence/absence of alarms.

Acknowledgments. This work has been partly supported by *MUlti-industry, Semantic-based next generation business INtelliGence (MUSING)*, a European RTD project (2006-2010) founded by the 6^{th} Framework Programme - Information Society Technologies, Contract No. FP6-027097, `http://www.musing.eu`.

References

1. R. Agrawal and R. Srikant. Mining sequential patterns: Generalizations and performance improvements. In *Proc. of EDBT 1996*, volume 1057 of *LNCS*, pages 3–17. Springer, 1996.
2. C. Alexander, editor. *Operational Risk: Regulation, Analysis and Management.* Prentice Hall, 2003.
3. P. Artzner, F. Delbaen, J.M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9:203–228, 1999.
4. Basel Committee on Banking Supervision. *Sound Practices for the Management and Supervision of Operational Risk.* BIS, 2003. `http://www.bis.org/publ/bcbs96.htm`.
5. Basel Committee on Banking Supervision. *International Convergence of Capital Measurement and Capital Standards: A Revised Framework.* BIS, 2006. `http://www.bis.org/publ/bcbs128.htm`.
6. R. G. Cowell, R. J. Verrall, and Y. K. Yoon. Modeling operational risk with bayesian networks. *Journal of Risk & Insurance*, 74:795–827, 2007.
7. E. Davis, editor. *Operational Risk: practical approaches to implementation.* Incisive Media Investments, 2005.
8. P. Jorion. *Value at Risk: the new benchmark for managing financial risk.* McGraw-Hill, 2007.
9. C. Klüppelberg. Risk management with extreme value theory. In B. Finkenstädt and H. Rootzén, editors, *Extreme Values in Finance, Telecommunications and Environment*, pages 101–168. Chapman & Hall / CRC.
10. T.-R. Li, Y. Xu, D. Ruan, and W.-M. Pan. Sequential pattern mining. In *Intelligent Data Mining*, volume 5 of *Studies in Computational Intelligence*, pages 103–122. Springer, 2005.
11. Pentaho Corporation. Pentaho: Open Source Business Intelligence, Version 1.6, 2008. `http://www.pentaho.com`.
12. H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In *Proc. of CIKM 2001*, pages 81–88. ACM, 2001.
13. R. Ziembinski. Algorithms for context based sequential pattern mining. *Fundamenta Informaticae*, 76(4):495–510, 2007.