International Journal on Artificial Intelligence Tools
Vol. XX, No. X (2007) 1–20
© World Scientific Publishing Company

World Scientific
www.worldscientific.com

DETECTION AND PREDICTION OF RARE EVENTS IN TRANSACTION DATABASES

CHRISTOS BERBERIDIS

Machine Learning and Knowledge Discovery Group,
Department of Informatics, Aristotle University of Thessaloniki,
University Campus, Thessaloniki 54124, Greece
berber@csd.auth.gr

IOANNIS VLAHAVAS

Department of Informatics, Aristotle University of Thessaloniki,
University Campus, Thessaloniki 54124, Greece
vlahavas@csd.auth.gr

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Abstract
Rare events analysis is an area that includes methods for the detection and prediction of events, e.g. a network intrusion or an engine failure, that occur infrequently and have some impact to the system. There are various methods from the areas of statistics and data mining for that purpose. In this article we propose PREVENT, an algorithm which uses inter-transactional patterns for the prediction of rare events in transaction databases. PREVENT is a general purpose inter-transaction association rules mining algorithm that optimally fits the demands of rare event prediction. It requires only 1 scan on the original database and 2 over the transformed, which is considerably smaller and it is complete as it does not miss any patterns. We provide the mathematical formulation of the problem and experimental results that show PREVENT's efficiency in terms of run time and effectiveness in terms of sensitivity and specificity.

Keywords: rare events, prediction, data mining, sequence mining.

1. Introduction

In most studies so far, association rules have not been appreciated for their predictive capacity, because they typically associate events within the same transaction, being essentially intra-transactional. Inter-transaction association rules are a relatively new kind of association rules, that can embed temporal information, being able thus to facilitate prediction. The prediction of rare events from data is a particularly interesting problem, because the result not only has to be accurate but it also has to be delivered in time. By the term "rare events" we mean events of a certain domain that do not happen often or regularly but they have a special meaning or play an important role in the system and they are usually hard to predict. Examples of such events are network intrusions, engine failures, earthquakes and meteorological events such as hail and heat waves.

Event prediction is very similar to time series prediction. Classical time series prediction, which has been studied extensively within the field of statistics, involves predicting the next *n* successive observations from a history of past observations [11]. These statistical techniques involve the building of mathematical probabilistic models, which are based on specific data, since they are strongly dependent on various theoretical assumptions regarding the underlying nature of variation (probability distributions etc.). However, this is not our case. First, we are interested in extracting knowledge from a very broad class of large transaction databases, without any prior information on the variability of the data and therefore without having to state theoretical assumptions. Second, our main goal is not to build certain mathematical models, but to discover patterns, which are related to certain critical events and which are going to provide us an alarm for the early identification of such events.

In this paper we propose PREVENT, a novel algorithm for the production of frequent inter-transactional itemsets, based on a well known prefix tree, namely the FP-Tree [10]. We extend and complement our work in [18], in which a short, preliminary version of this work was presented. Although PREVENT is a general-purpose inter-transactional pattern mining algorithm, we also believe that it optimally fits the demands of the rare events prediction application domain. Our approach differs from other temporal association rule induction approaches largely in that we use inter-transactional patterns instead of rules and we adopt a general framework for the task of prediction, where the prediction is delivered within a specific time window. Moreover, it requires only 2 database scans, outperforming Apriori-based approaches for the production of frequent inter-transactional itemsets and it is complete as it does not miss patterns. In order to measure its efficiency we tested it over a number of datasets, including a real world meteorological dataset for the prediction of heat waves and a daily electric energy consumption dataset.

The paper is organized as follows: The next section presents a review of the literature regarding temporal association rules and sequential patterns. In section 3 we provide the mathematical formulation of the problem, including definitions and theoretical background of our approach, the algorithm we propose, as well as a discussion about its computational complexity and performance. In sub-section 3.4 we explain why we use patterns instead of rules. In section 4 we present the experiments we conducted in order to test and verify the performance of the proposed algorithm. Finally, in section 5 we present our conclusions and propose our ideas for further work.

2. Related Work

In this section we present a brief description of the relative bibliography. First, we refer to the association rules and pattern mining approaches that gave us the inspiration for our approach. Then, in section 2.2 we provide a short survey of other approaches (mostly supervised ones) dealing with the rare class prediction problem. Finally, in section 2.3 we discuss some problems and open issues regarding to this problem.

2.1. *Association Rules and Sequential Patterns for Prediction*

Having a temporal database, we can mine for various types of association rules. One approach is to cluster the data based on time and then discover association rules from each cluster, in order to track how the model changes over time [9]. Traditional association rule analysis was extended to sequence mining, where the members of the series are sets of individual items, called itemsets, from some underlying domain (alphabet). Given a set E of events, an event sequence s is a sequence of pairs (e, t), where $e \in E$ and t is an integer, the occurrence time of the event of type e. Unlike time series, sequences do not require any explicit relationship with time, only that the itemsets are totally ordered. According to Agrawal and Srikant [6] the goal of sequence mining is to find all maximal length sequences with support above a certain threshold. Their first work was a level-wise Apriori-based algorithm [24] and several variations were later proposed. Probably their most influential algorithm is GSP [15], which adopts a sliding window technique in order to extract the frequent sequences.

The basic difference between a time series and sequence, then, is that a time series is a list of ordered values, while a sequence is a list of ordered itemsets or values. Sequence mining aims to discover patterns such as $\{\{A\}, \{B\}, \{C, D\}\}$, where $\{A\}$, $\{B\}$ and $\{C, D\}$ are itemsets in different transactions, within a user-defined time window. Finding the most frequent maximal patterns is a particularly useful task that provides the user with valuable insight about the temporal nature of the data. However, the predictive power of sequential association rules is questionable. Sequence analysis or sequential pattern mining was extensively studied initially by Agrawal et al. [6, 7], where the notions of sequence and subsequence were defined.

An episode rule [8] is a generalization of association rules applied to sequences of events. An event sequence S is an ordered list of events, each one occurring at a particular time. Thus, it can be viewed as a special type of time series. Given the above definitions, an episode a is a partial order of event types. Episodes can be viewed as directed acyclic graphs. There are serial, parallel and non-serial and non-parallel episodes. Episode mining algorithms are searching for episodes or episode rules within a sliding window of user-defined size. What is captured here is the temporal relationship among events that occur within the same window, e.g. "C comes after A and B within a window of size w".

Temporal association rules typically search for correlations among items in transaction data sets, facilitating temporal relationships, such as "A usually occurs some time after B". The thirteen temporal relationships defined by Allen [17] (before, after, during, contains etc.) are usually supported.

Although transactions occur under certain contexts such as time, space, customers, etc., such contextual information has been ignored because this task is intra-transactional, as in standard association rule learning. Association rules aggregate information from a large number of transactions into a rule for a single transaction: "In the millions of transactions of a supermarket, there is a 60% probability to find beers and diapers in the same transaction". However, rules like "If the prices of IBM and SUN go up, Microsoft's will most likely (80% of the time) goes up 2 days later" [3] cannot be captured by the intra-transactional approaches. This kind of rule associates itemsets among different

transactions, along the axis of of a dimensional attribute. The contextual information here is time, which is the dimensional attribute. These rules are called inter-transactional and they can be single or multi dimensional.

Inter-transactional association rules were introduced in [1] and [2]. The authors extend the notion of inter-transactional association rules to the multidimensional space and propose EH-Apriori, an Apriori-based algorithm, for mining such rules. The authors also propose the use of templates and concept hierarchies as a means to reduce the large number of the produced rules. A new set of algorithms is introduced in [3], called FITI (an acronym for "First Intra then Inter"), which outperforms EH-Apriori. In [4] and [5], the authors use inter-transactional association rules for prediction on meteorological and stock market data, correspondingly.

Two approaches that mostly resemble the inter-transaction association rules mining approach are the ones proposed by Agrawal et al. in [6] and Mannila et al. in [8]. More information on how inter-transactional association rules differ from these can be found in [3]. A really concise and informative survey on temporal knowledge discovery in general can be found in [14].

2.2. *Other Approaches on Rare Event Analysis*

Except from the aforementioned approaches, event prediction has also been treated extensively by classification. In the machine learning and statistical literature there is an abundance of rare event prediction approaches, supervised or unsupervised. The former ones require that the data be labeled in order to build a model that is easy to understand, while the latter analyze each event to determine how similar it is to the majority. There are plenty of outlier detection approaches, statistical [35, 36], distance-based [37, 38], density-based [39], clustering [40], neural network and SVM-based [41, 42]. Machine learning has treated the rare event prediction problem as a “class imbalance” problem or as “cost-sensitive classification”. One solution in order to overcome the class imbalance problem is the use of “data record manipulation” techniques. Such techniques involve downsizing (undersampling) the majority class [44] or oversampling the rare class [43]. SMOTE (Synthetic Minority Over-sampling TEchnique) [45] is a popular oversampling method that creates synthetic examples of the minority class based on a majority voting among the nearest neighbors. Another solution is cost-sensitive classification, where a cost is assigned to misclassified instances, according to a cost matrix. AdaCost [48] (a cost-sensitive boosting method) and MetaCost [20] are worth mentioning examples which can be used for mining rare classes with high misclassification cost. AdaBoost [47] is a popular algorithm that allows to combine a number of weak rules into a stronger (more accurate) classifier. Additionally, emerging patterns (an extension to the association rules paradigm) have been used in order to mine for a rare class. These are patterns whose support increases significantly over time [46]. Other approaches to rare event prediction are the temporal analysis of rare events that includes surprising patterns in time series [49], temporal sequence associations [51] and [51] where Vilalta & Ma combine event types in order to build a rule-based system for prediction.

An informative study on how different classifiers behave with respect to class imbalance and how certain solutions affect their accuracy can be found in [29], where Japkowicz et al. show that among three different classifiers, namely C5.0, Multi-layer Perceptrons (MLPs) and Support Vector Machines (SVMs), SVMs are the least sensitive but they are not necessarily more accurate than the others. Their accuracy varies for different problem types and concept complexities. The authors conclude that SVMs seem to be quite robust and accurate in a large variety of problems but they largely depend on the selection of the right kernel function and its variance and they also have really high training times, which makes them impractical for a range of applications. This last drawback holds for Neural Nets and MLPs as well.

Fawcett et al. [25] introduce a problem class called *activity monitoring*, which involves the monitoring of a series of a large population of entities for interesting events that require action. They introduce the use of Activity Operating Monitor Characteristic (AMOC) curve, a modified ROC curve, to accommodate issues that relate to activity monitoring. Torgo et al. [26] propose a new splitting criterion for regression trees for the prediction of extreme and rare values of a continuous target variable. In order to avoid the discretization of the continuous variable and thus the resulting loss of information, the authors utilize the *F* measure [28] in order to choose the best splitting criterion at each node and they achieve noticeable results in some datasets, although in some other ones the model's precision is not satisfactory.

This paper extends and complements the work introduced in [18]. The approach presented here is different from the other approaches proposed so far because it is based on the inter-transaction association rules framework, utilizing the computational advantages of FP-Growth, an efficient, state of the art, intra-transaction association rules mining algorithm. At the same time, we adopt an intuitive prediction framework, described in section 3.2, in order to mine for local patterns instead of rules or global models, which is usually the case with most classification approaches. The result is a fast algorithm that fits the demands of the discrete event prediction task, although it can be used as a general purpose inter-transaction association rules mining algorithm. PREVENT is a low computational cost algorithm that is also complete (no predictive patterns are missed). Additionally, PREVENT's modular nature allows for the utilization of other algorithms and features such as incrementality.

2.3. *Open Issues in Rare Event Prediction*

Despite the ongoing research on rare class classification, there are still open issues that still need to be investigated by the research community. In the last decade, a number of workshops and tutorials [53, 54, 56] have tried to address them. An important issue is the selection of the appropriate sampling methodology. Oversampling the minority class enlarges the training set increasing thus the training time and inserting artificially created data may potentially leading to overfitting, while undersampling the majority class may lead to loss of information. How much to oversample or undersample is usually decided empirically. Smart resampling techniques, such as SMOTE [45] have shown effective results, often being capable of eliminating redundant information.

Another issue is how class distribution affects classifier performance. Weiss et al. [55] attempt to shed some light on which data distribution is the most appropriate, conducting extensive experiments on a decision tree learner (C4.5) using a 26 datasets and proposing an algorithm that takes under consideration the cost of training examples procurement.

In ICML 2003 several papers tried to investigate how various approaches compare to each other. However, the conclusion is that different techniques are shown to be effective on different applications, depending on the context of each problem. Evaluation of imbalanced data classifiers is another issue where ROC curves seem to be the most prevalent approach, however it has been pointed out [57] that a single point on a ROC curve is optimal only if costs are the same for all examples.

3. The PREVENT Algorithm

PREVENT (Prediction of Rare EVENTs) is a general purpose inter-transaction association rule mining algorithm, which combines advantages such as the embedding of explicit temporal information and low computational cost, requiring only 2 database scans. It is often useful to know exactly when to expect something to happen (e.g. "five days later") instead of a fuzzy temporal window (e.g. "some day within 1 week") or a sequence (e.g. B and C will happen after A). PREVENT, unlike other approaches, is not concerned with the discovery of rules but searches for patterns that contain the temporal information required for the task of prediction. We adopt an intuitive framework for the task of prediction, which involves the definition of a prediction (or monitoring) window where the prediction can be useful to the user.

3.1. *Problem Formulation – Definitions*

In our setup we have the following notions:

- The set of *items* $I = \{ i_1, i_2, \ldots, i_v \}$ representing the possible activities we want to keep record of (e.g. items sold in a store or responses to requests by a server).
- The *dimensional variable T* describing the time properties associated with the items. We assume that the variable takes ordinal values representing intervals of equal length (e.g. day, week, month etc.). Note that this variable can be defined to represent various other ordinal measurements such as length, height, etc. It is also possible to have many of these variables (time, distance, etc.), simultaneously describing our data, but in our context we consider only one. Without loss of generality we denote the values of *T* by integers 0, 1, 2, ...
- The *transactions* which are records of the form $J(t)$ where *t* is a value of the time variable *T* and $J(t) \subseteq I$. So, each transaction is represented by a set of certain activities from *I* recorded in time *t*.
- The *transaction database* containing all the transactions recorded over a (usually long) period of time.
- The *transaction sequence*, a time-ordered sequence of transactions, denoted by $S = J(t_1), J(t_2), \ldots, J(t_n)$, which includes *n* transactions recorded in the time interval $[t_1, t_n]$.

- The *target item*, $i^* \in I$, which represents an activity that we are particularly interested in predicting, e.g. failure of a system to respond, network fault, etc. Such an item occurs infrequently with respect to the other items while its occurrence is much more critical than the others'. We also denote by t^* the time interval when the target item occurs.
- The *target transactions* which are transactions containing the target item. Note that for our work here we do not need to consider target items at all but more generally target transactions, since a target transaction may have the meaning of an infrequent combination of items that we are interested to predict.

Therefore, the problem we consider here is to derive inter-transactional patterns that can be used as alarm messages in order to predict the target transactions within a reasonable period of time before the critical target item occurs. For this purpose we associate with every target transaction $J(t^*)$:

- A *prediction period*, which is a time period preceding the target transaction of fixed length defined as $[t^* - m,\ t^* - w]$ where m is the *monitoring time* and w is the *warning time*. We assume that $m > w$.
- A *target - preceding window* W^*, which is a block of $m - w + 1$ continuous time intervals included in the prediction period of the target transaction. Thus, the window consists of all the time intervals from $t^* - m$ to $t^* - w$. Note that it is not necessary for each interval to contain a transaction. These intervals within a window are called *target - preceding subwindows* of W^*. We use non-negative integers to denote the subwindows. So, we denote the subwindow in the beginning of the prediction period by $W^*(1)$, and the following ones by $W^*(1), \ldots, W^*(m - w + 1)$. We also use the same indices to denote the items in each subwindow. Thus, if the item i_k $(1 \le k \le v)$ occurs in *target - preceding subwindow* $W^*(x)$ $(1 \le x \le m - w + 1)$, it will be denoted by $i_k(x)$. Such items are called *extended items*. We denote the set of all possible extended items as follows:
$$I^* = \{ i_k(x) :\ 1 \le k \le v,\ 1 \le x \le m - w + 1 \}.$$
- A *target megatransaction* $M^* \subseteq I^*$ defined as the set of all extended items within W^*, i.e.
$$M^* = \{ i_k(x) :\ i_k \in W^*(x) \}$$
- A measure of inter-transaction patterns F:
$$\textit{support of } F\text{: } s = \frac{N_F^*}{N^*} \qquad (1)$$
where N^* is the number of all the target megatransactions in the database and N_F^* is the number of all target *megatransactions* that contain the set F. We can characterize a set as *frequent* if s exceeds a lower bound, defined by the user.

The purpose of the search is to find all frequent sets of extended items that contain the temporal information required for the prediction task. Those are temporal patterns that contain the target event and therefore can be used for prediction.

3.2. *Algorithm Description*

The general strategy we follow to predict rare events takes into account the fact that it is highly important that a prediction is given in time. Therefore, we assume that there is a time period preceding a target event X_t, when the prediction can be useful (*prediction period* or *monitoring window*). This period starts with a time point that denotes the beginning of the period when the user is interested in having a prediction and ends with a time point after which it is too late and the prediction has no practical meaning (warning time). The concept is illustrated below and has been proposed in [16], where Weiss et al. present an event prediction technique, based on genetic algorithms.

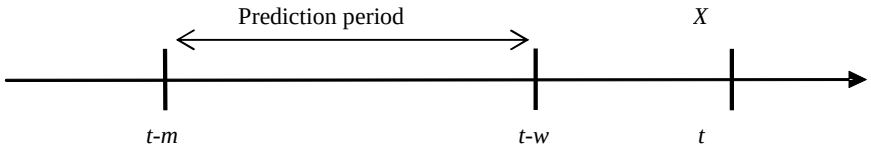

Fig. 1. The prediction period.

In PREVENT, we perform one scan over the database in order to capture and store only the transactions associated with those periods, using a sliding window. The number of such periods (windows) stored is equal to the number of occurrences of the target event, which, in our case is rare. In other words, we capture the corresponding monitoring window of every occurrence of the target event in order to extract the desired knowledge. While capturing those windows, a database transformation takes place in order to map the relative temporal information of every item within the window. The transformation is done according to the definitions given in section 3.1, based on the inter-transactional association rules framework. An example of such transformation is depicted in the following figure.

Example. Assume that the size of the monitoring window is 3 transactions and the set of literals in the database is *{a, b, c, d, e, f, g}*. The corresponding set of possible extended items and their integer mapping are depicted in the table below.

Tid	**Transactions**
1	a, b,d, c, g
2	g, c, d
3	a,e,**f**
4	a, b, c, d
5	a, c, k, **f**
6	a,e,**f**

Transform

target item: f

XTid	**Extended Transactions**
1	$a_1b_1c_1d_1g_1c_2d_2g_2a_3e_3f_3$
2	$a_1e_1f_1a_2b_2c_2d_2a_3c_3k_3f_3$
3	$a_1b_1c_1d_1a_2c_2k_2f_2a_3e_3f_3$

Pattern Mining FP_Growth

min_sup=0.9

Patterns
$a_1c_2a_3f_3$

Fig. 2. A data transformation example.

The original transaction database is transformed into the extended transaction set. The transformed database contains a number of extended transactions equal to the number of occurrences of the target item *f*. Each such transaction consists of the extended items of every moving window instance. For memory efficiency purposes, we map every item instance i_t to an integer and keep the index in order to be able to backtrack later to the original data. An example of such mapping is the following:

Table 1 Integer mapping example.

Set of Extended Items	a_0	a_1	a_2	b_0	b_1	b_2	c_0	c_1	c_2	d_0	d_1	d_2	e_0	e_1	e_2	f_0	f_1	f_2	g_0	g_1	g_2
Integer Mapping:	0	1	2	3	4	4	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

The following step is the mining of the frequent itemsets from the transformed data. We utilize FP-Growth [10], a frequent itemset mining algorithm. FP-Growth builds an FP-tree (Frequent Pattern-tree), which is an extended prefix tree structure that stores crucial information about frequent patterns. FP-tree is actually an efficient way to compress the original database into a much smaller structure that is cost-effective to mine. Each tree node contains a frequent item (itemset of length 1). Each transaction contributes at most one path to the FP-tree, with length equal to the number of frequent items in that transaction. Quoting from [10]: "The tree nodes are arranged in such a way that more frequently occurring nodes will have better chances of sharing nodes than less frequently occurring ones. Our experiments show that such a tree is highly compact, usually orders of magnitude smaller than the original database." Its major advantage is that it reduces the number of database scans to only 2 in order to construct the FP-tree.

FP-Growth, a divide and conquer algorithm, is used to mine the patterns from the FP-tree. It scans the FP-tree once to build a small pattern base for each frequent item a_i, each consisting of the set of transformed prefix paths of a_i. Frequent pattern mining is then recursively performed on the small pattern bases. Pattern bases are usually much smaller than the original FP-tree. While Apriori-based approaches require a large number of repeated scans and the generation of a very large number of candidate sets, which often reaches the levels of a combinatorial explosion, FP-Growth requires only 2 database scans to create the FP-tree. Then it reduces the problem of mining the frequent *k*-itemsets into a sequence of *k* frequent 1-itemset mining problems. FP-Growth avoids the costly generation of candidate itemsets that Apriori-based approaches require. Especially in our setup, where the number of different (extended) items is usually quite large, the application of an Apriori-based algorithm would be extremely costly. The steps of PREVENT are outlined in figure 3, while further information on FP-Growth can be found in [10].

1. Move a sliding window across the transactions of the database until the next occurrence of the target item is found (1st database scan).
    a. For every such occurrence, capture the corresponding monitoring window, transform it as described above and store it in a new database file.
    b. Store the integer-mapping index.
2. Build the FP-Tree (1st scan over the transformed database)
3. Extract the extended frequent itemsets (predictive patterns – 2nd scan over the transformed database).
4. Using the integer-mapping index, convert the extended items, from integer numbers into their original form.

Fig. 3. The PREVENT Algorithm

3.3. *Algorithm analysis and discussion*

When speaking about computational complexity within the data mining context, what is mostly important is the number of database scans. When the main memory is not enough to fit the data, main memory based operations are insignificant compared to operations that require hard disk access. The major advantage of our algorithm is that it requires only 1 scan over the original database, regardless of the size of the database or the number of literals; during this pass (sliding window) the original database is transformed into the set of extended transactions, which is considerably smaller. Then, FP-Growth performs a pass over the transformed database in order to build the FP-Tree and then another pass in order to mine for the patterns.

Moreover, the main memory structures used are small and pose no additional overhead. What is important here is the shrinking factor achieved by the transformation of the original database. In [10] the authors claim that while the shrinking factor of the first FP-tree normally ranges from 20 to 100, the shrinking factor from this FP-tree to the pattern bases is expected to be hundreds of times larger. The size of the sliding window is *m*MaxTransactionSize*, and the size of the integer mapping index is *(Monitoring Window Size)*(Size of the set of extended items)*, both of which easily fit into the main memory. Since there are often a lot of sharing of frequent items among transactions, the size of the tree is usually much smaller than its original database and that of the candidate sets generated in the Apriori-based approaches.

One the most desired features of all data mining algorithms is incrementality. The modular nature of PREVENT allows for the use of any incremental association rules mining algorithm after the first step. The good news is that there are incremental FP-tree based algorithms, such as CATS [21] that can be used directly, so that we don’t have to miss its computational advantages.

3.4. *Using Predictive Patterns Instead of Rules*

According to the prediction framework described in the previous sections, given that the warning time is always *w* time points before the target event X_t, we propose an efficient method for mining predictive patterns within the prediction time period. Alternatively to the original inter-transactional association rules paradigm, we propose the use of patterns instead of the typical If-Then rules for the prediction task. A temporal inter-transactional pattern contains all the necessary information, except from the notion of causality, which, even using rules as a representation form, can be elusive and hard to prove anyway.

Generally, if-then rules are a form of representation, that have specific advantages, such as the fact that are easily understood by people, they are modular (each rule fits a portion of the data) and they relate events in various contexts such as time, space, probability and causality. Rules can be used for description (e.g. "if someone is a doctor then he/she holds a university degree") as well as for prediction (e.g. "if a customer is married then he/she will buy a family car"). However, it is very common within the context of prediction to (consciously or subconsciously) convey causal content. Temporal precedence is normally assumed essential for defining causality and it is one of the most important clues that people use to distinguish causal from other types of associations.

However, the fact that one event occurs after another does not prove a causal relationship between them. Almost three centuries ago (1748), David Hume in his greatest philosophical work *"An Enquiry Concerning Human Understanding"* supported that causality does not really exist: *"We may define a cause to be an object, followed by another, and where all the objects similar to the first, are followed by objects similar to the second"*. In other words, Hume supported that causality is not actually knowable but imagined by our mind to make sense of the observation that A often occurs together with or slightly before B. All we can observe are correlations, not causations. We quote Mazlack on association rules [12]: *"In fact, with association rules all that is discovered is the existence of a statistical relationship. The nature of the relationship is not specified [...] Associations describe the strength of joint co-occurrences. Sometimes the relationship might be causal; for example, if someone eats salty peanuts and then drinks beer, there is probably a causal relationship. On the other hand, it is unlikely that a crowing rooster causes the sun to rise*".

We believe that using association rules for prediction, when we can push inter-transactional temporal information into frequent itemsets, is of no obvious use, especially when causality is not a notion we really try to mine here. The inter-transaction frequent itemsets contain all the information required, without the risk of erroneously implying a causal relationship.

Frequent itemset mining is an inherently unsupervised procedure, which aims to discover informative knowledge from data in exploratory fashion. Embedding inter-transactional dimensional (temporal) information into those co-occurrence patterns can be a useful tool to discover prediction-related information, in cases where the data collected are far from Gaussian, even multimodal. Consider now a typical inter-transactional association rule, such as $A(t_1) \rightarrow X(t_n)$, where $X(t_n)$ is the target event, $A(t_1...t_k)$ is a set of extended items (events occurring at time points $t_1..t_k$) *and* $t_n > t_k$. In the

proposed approach we perform prediction using the frequent extended itemsets derived from the prediction periods of *X*. Since we are interested only in the rules that have *X* in the consequent, extracting only the frequent extended itemsets from the monitoring windows would be enough for predicting the target event. This makes the whole process simpler and faster.

4. Implementation and Performance Results

We implemented our algorithm in C++ and tested it against a number of data sets of different sizes. There were two types of data, real world and synthetic. The real world data were meteorological, containing the hourly measurements of temperature, humidity and THI (temperature-humidity index) from 1954 until 1998. The synthetic datasets were used for uniformly measuring the run time of PREVENT and verifying its completeness.

In some contexts, the search for exact pattern matches could be considered as a drawback, because minor distortions, such as dilation, cannot be captured. Very often, especially in real world sequences, such as meteorological, such effects are common and those patterns missed by exact matching algorithms could result in loss of valuable knowledge. As a result, "loser" patterns, which have a frequency near the user specified minimum threshold, might be possible "winners" if those patterns that are similar to them could somehow be considered to their overall frequency count. We suggest two solutions to this problem that we plan to investigate in the near future.

One approach to this problem is to adopt a different pattern space, using time intervals instead of time points. This way we could utilize the 13 temporal relationships between two intervals defined in [17] (e.g. during, contains, starts, etc.). For example, assume that we have *r*=<A**B*C>, a possible winner pattern ("*" stands for "any single literal") and the search algorithm encounters a pattern *s* =<AIJKBLMC>. In our case, *s* would be ignored in the frequency count, although it contains <A***B**C>, which could be considered as a distorted instance of *r*. However, according to Allen's temporal interval relationships this can be captured by the relationship "*r* starts *s*". Therefore, enabling our algorithm to use such conditions could provide a solution to this problem. The work in [22, 23] are examples of using intervals in pattern mining.

Another solution to the same problem would be considering not only the exact matches but also the patterns that are similar to each other under a certain distance metric. Sequence similarity has been extensively studied in terms of time series mining (discrete or continuous) and within the field of molecular biology (e.g. protein sequence similarity). In the previous example, patterns *r* and *s* are similar, under some distance measure. Various distance measures among those proposed in the literature can be used, depending on the domain of application.

4.1. *Synthetic Data Experiments*

The synthetic data were generated with MATLAB, according to a set of probabilistic pseudo-random parameters, such as the frequency of the rare event, the Monitoring and the Warning Time. The performance of the algorithm depends on the size of the

monitoring window, the number of different items and the frequency of the target event. Below, we present an experimental setup that has the following configuration: There are eleven different items in the database, including the target item. The *Monitoring Time* was set to 6 and the *Warning Time* to 2, which means that, according to the *m*-*w*+1 formula, the size of the *Monitoring Window* is 5. Therefore, the transformed database contains 55 different extended items. The target event frequency was set between 9% and 10%, therefore, the transformed database contained approximately *0.1*DatabaseSize* Mega Transactions. The experiments were taken on a Pentium 4, 2.6 GHz computer with 512MB of RAM and a SCSI hard disk. The apriori implementation we used was taken from [32]. For both approaches, the times measured include only the production of the frequent itemsets, not the association rules. Figure 5 illustrates the performance of our algorithm with respect to the number of *Mega Transactions* of the transformed database.

Table 1 Experimental results on synthetic data

DB Size (transactions)	Mega Transactions	Run Time (seconds)	
		PREVENT	Apriori-based
10000	923	0.39	0.33
50000	4457	1.082	1.21
100000	8943	1.761	2.18
500000	44715	7.46	11.12
1000000	89430	14.24	24.43

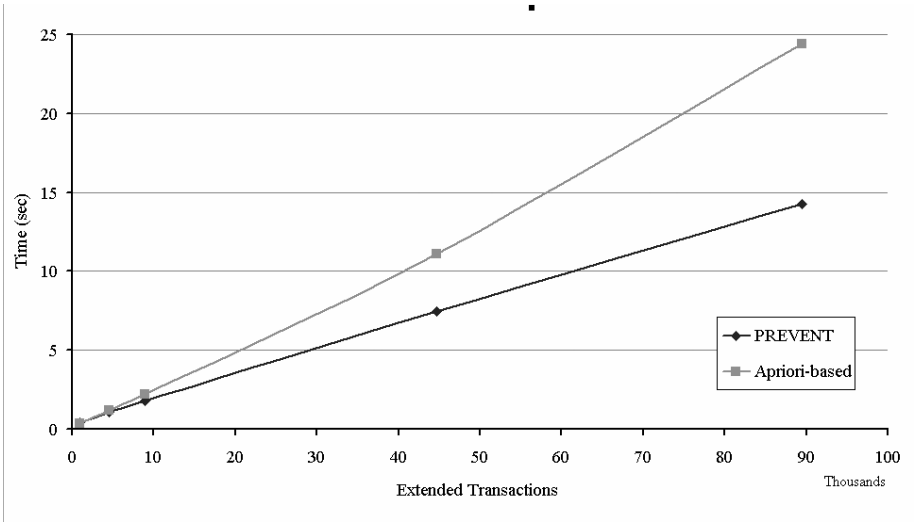

Fig. 5. Run time against the number of Mega Transactions

Our approach is complete due to the completeness of FP-Growth. In our experimental setup, a frequent extended itemset starting from time point 0 and ending at

time point 4 can predict an event that will happen at time point 6. For example, given that the target item is *x*, a frequent extended itemset such as $\{a_0, b_1, b_2, d_4\}$ can be used for the prediction of *x* at time point 6 with some level of support.

4.2. *Real World Data Experiments*

In our experiments we used two real world datasets, one meteorological (heatwaves) and one for electrical energy consumption.

The first dataset contains the hourly measurements of temperature, humidity and *THI* from 1954 until 1998 in the area of Thessaloniki, Greece. *THI* expresses the discomfort that people feel during a heat wave. The data were kindly provided by Prof. T. Karacostas of the Department of Meteorology and Climatology of Aristotle University of Thessaloniki, Greece. There are five *THI* levels [13]:

Table 2 Temperature-Humidity Index levels

THI	**Class**	**Description**
$69 \leq THI < 75$	Mild	Few people feel uncomfortable
$75 \leq THI < 80$	Moderate	About one half of all people feel uncomfortable
$80 \leq THI < 84$	Serious	Nearly everyone feels uncomfortable
$84 \leq THI < 92$	Severe	Rapidly decreasing work efficiency
$THI \geq 92$	Extreme	Extreme danger

We tried a number of different setups, according to some general guidelines provided by the meteorologists. The target event is the occurrence of a Serious, Severe or Extreme heat wave. The hourly data were grouped in 6 hour period averages and discretized into 3-5 classes. The temperature (*T*) is in degrees Celsius and the monitoring window starts 14 6-hour periods before the heat wave and ends 2 6-hour periods before. There were 172 serious (or worse) heat waves in a period of approximately 44 years (approximately 1% probability). Regarding the usefulness of the produced patterns, although preliminary, the domain experts empirically evaluated them as quite interesting and worth investigating. Unfortunately, the domain is too hard to model, especially when only two attributes are available (temperature *T* and humidity *H*), so the patterns were not particularly informative. Below we provide a sample of the patterns produced, with reference to the target event, which occurs at time point 0.

(i) (T > 27, 5 periods before), (T > 27, 3 periods before), support=0.91
(ii) ($50 \leq H \leq 75$, 5 periods before), (T > 27, 5 periods before), (T > 27, 3 periods before), support=0.77
(iii) ($40 \leq H \leq 65$, 5 periods before), (T>27, 5 periods before), (T>27, 3 periods before), support=0.75

Like other pattern mining approaches, PREVENT involves search for local patterns instead of global models. In association rules and pattern mining the notion of accuracy is substituted by other statistics that express strength or interestingness of a pattern, such as the confidence (which is here defined accordingly). However, in event prediction it is always expected to measure the effectiveness of a technique in terms of prediction accuracy. In highly imbalanced datasets, metrics such as the specificity and the sensitivity are considered more appropriate than accuracy. These two metrics are popular in medical research but lately they are also gaining popularity in machine learning [33, 34]. Sensitivity, also known as "recall" in Information Retrieval, and Specificity are defined as follows:

$$Sensitivity = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (3)$$

$$Specificity = \frac{TrueNegatives}{TrueNegatives + FalsePositives} \quad (4)$$

The second dataset contains daily electric energy production measurements (KWhs) from 6 different sources: Hydroelectric, Nuclear, Coal, Petroleum, Alternative sources (eolic, solar, etc). The measurements were taken in Spain, for the whole year 2003 [52]. The predicted variable is the average cost of the KWh in euros. The data were preprocessed (discretized) in order to transform the continuous domain into discrete. All 7 variables were discretized using the "unsupervised.attribute.Discretize" class of the WEKA machine learning library; the 6 input variables were split into 3 bins (high/medium/low) while the class variable was divided into 2 bins (high/low). The following table contains the bins that each input variable was divided and the letters that were assigned to them. Our goal is to predict when to expect a high KWh cost. There were 32 days that the cost of the KWh was high. For the evaluation we performed a standard 10-fold cross validation procedure.

Table 3 Variable discretization for the electric energy dataset

Hydroelectric		
a=(-inf-87266.2]	b=(87266.2-146650.6]	c=(146650.6-inf)
Nuclear		
d=(-inf-138875]	e=(138875-162990]	f=(162990-inf)
Carbon		
g=(-inf-100635.666667]	h=(100635.666667-167734.333333]	i=(167734.333333-inf)
Petroleum		
j=(-inf-22662.166667]	k=(22662.166667-45324.333333]	l=(45324.333333-inf)
Natural gas		
o=(-inf-28150.733333]	p=(28150.733333-56301.466667]	q=(56301.466667-inf)
Alternative		
r=(-inf-8990.333333]	s=(8990.333333-12673.666667]	t=(12673.666667-inf)

Table 4 Predictive patterns for the electric energy dataset

	Pattern	Sensitivity	Specificity
1	carbon_1 = h and petroleum_3 = h and alternative_3 = h, cost_3 = h	0,666667	0,990385
2	petroleum_3 = h and alternative_3 = h, cost_3 = h	0,666667	0,980769
3	carbon_1 = h and petroleum_3 = h, cost_3 = h	0,666667	0,971154
4	nuclear_2 = h and gas_2 = l and hydro_3 = l, cost_3 = h	0,666667	0,971154
5	petroleum_3 = h, cost_3 = h	0,666667	0,961538

In order to measure the effectiveness of PREVENT we measure the sensitivity and specificity of each pattern. Table 4 summarizes the results, displaying the best patterns, according to their specificity and sensitivity scores. The sensitivity and specificity minimum thresholds were set to 65% and 95% correspondingly and PREVENT returned the 5 rules shown in Table 4. The letters h, m and l stand for high, medium and low values correspondingly, while the numbers 1, 2 and 3 indicate the first, second and third day of the prediction window. One can see the very high levels of specificity achieved (96%-99%) that show the very low false positive rate, while the moderately good values of sensitivity (66,7%) means that some true positives might be missed. In other words, in this case study, PREVENT is very unlikely to falsely classify a low cost KWh day while it is possible to miss some of the high cost ones.

5. Conclusions and Further Research

In this paper we proposed PREVENT, a novel data mining approach for predicting rare events in transaction databases in a fast and explicit manner. Our approach is based on the inter-transactional association rules framework and utilizes a state-of-the-art algorithm for classical association rules mining, namely FP-Growth, in order to produce predictive patterns. It involves a database transformation in order to extract only the required information before mining for the predictive patterns. We formulated the problem, proposed a novel algorithm and conducted experiments to test and verify its performance and effectiveness. Our synthetic data experiments showed that it is faster than apriori-based algorithms. However, the synthetic and meteorological datasets were not appropriate for testing its effectiveness, so we experimented with another, publicly available, real world dataset in order to test PREVENT's sensitivity and specificity. It is within our plans to further experiment and evaluate the heat waves data, in a closer collaboration with the meteorologists.

PREVENT features low computational cost, since it requires only one scan over the original database, and two scans over the transformed database, which is considerably smaller. Additionally it is also memory efficient as it uses small memory based structures, except from the FP-tree, which in extreme cases can be relatively large. However, the FP-tree approach is a well established, complete and one of the most

efficient approaches in the literature, which is the reason it was selected. The overall approach is complete due to the completeness of the frequent itemset algorithm.

For future research, a priority issue is improving the sensitivity of our approach. Then, we plan to investigate the idea of dealing with dilation and/or translation effects in sequences, which are very common in sequences such as the meteorological ones. Such effects can be discovered by interval-based approaches [22, 23]. Additionally, we consider conducting a large series of real world experiments on various meteorological data, and extend our approach for distributed databases, such as web databases. Moreover, we are currently experimenting on biological data, namely the prediction of the Translation Initiation Sites in genome sequences [30].

Finally, we would like to kindly thank the three anonymous reviewers for their valuable comments and their insightful contribution on this work.

References

1. Hongjun Lu, Ling Feng, Jiawei Han, Beyond intratransaction association analysis: mining multidimensional intertransaction association rules, ACM Transactions on Information Systems (TOIS) Volume 18 , Issue 4 (October 2000) , ACM Press, NY, USA.
2. Anthony K. H. Tung, Hongjun Lu, Jiawei Han, Ling Feng, Breaking the barrier of transactions: Mining Inter-Transaction Association Rules, In Proc. of the 5th International Conference on Knowledge Discovery and Data Mining (KDD '99), ISSN:1046-8188, Pages: 423 – 454.
3. Anthony K. H. Tung, Hongjun Lu, Jiawei Han, Ling Feng, Efficient Mining of Inter-transaction Association Rules, IEEE Transactions On Knowledge And Data Engineering, Vol. 15, No. 1; January/February 2003, pp. 43-56.
4. Ling Feng, Tharam S. Dillon, James Liu: Inter-transactional association rules for multi-dimensional contexts for prediction and their application to studying meteorological data, Data and Knowledge Engineering, Vol. 37, Issue 1, April 2001, Pages 85-115.
5. H. Lu, J. Han, and L. Feng. Stock movement prediction and n-dimensional inter-transaction association rules. In Proc. of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, pages 12:1--12:7, Seattle, Washington, June 1998.
6. Rakesh Agrawal and Ramakrishnan Srikant. Mining Sequential Patterns. In Proc. of the 11th Int'l Conference on Data Engineering, Taipei, Taiwan, March 1995.
7. R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arning, and T. Bollinger. The Quest data mining system. In Proc. of the 1996 Int. Conf. Data Mining and Knowledge Discovery (KDD'96), pages 244 -249, Portland, Oregon, August 1996.
8. H. Mannila and H. Toivonen and A. I. Verkamo. Discovering Frequent Episodes in Sequences. In Proc. of the First International Conference on Knowledge Discovery and Data Mining (KDD-95). AAAI Press. Montreal, Eds. U. M. Fayyad and R. Uthurusamy , Canada, 1995.
9. Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule Discovery from time series. In Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98). AAAI Press, 1998.
10. J. Han, J. Pei, and Y. Yin, Mining Frequent Patterns without Candidate Generation, In Proc. of the 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00), Dallas, TX, May 2000.
11. P. Brockwell and R. Davis, Introduction to Time Series and Forecasting. Springer-Verlag New York, 1996.

12. L. Mazlack, Causality Recognition For Data Mining in an Inherently Ill Defined Word, International Joint Workshop on Soft Computing for Internet and Bioinformatics, December 2003.
13. J. A. Raffner and F. E. Blair (eds), The Weather Almanac, 2nd edition, Gale Research Company, Detroit, MI, 728 pp.
14. J. F. Roddick and M. Spiliopoulou, A Survey of Temporal Knowledge Discovery Paradigms and Methods, IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 4, July/August 2002.
15. R. Srikant and R. Agrawal, Mining Sequential Patterns: Generalizations and Performance Improvements, In Proc. of the International Conference Extending Database Technology (EDBT '96), P. M. G. Apers, M. Bouzeghoub and G. Gardarin, eds., pp. 3-17, 1996.
16. G. M. Weiss and H. Hirsh, Learning to Predict Rare Events in Event Sequences, In Proc. of the 4th International Conference on Knowledge discovery and Data Mining, AAAI Press, 1998, pp. 359-363.
17. J. F. Allen, Maintaining knowledge about temporal intervals, in Communications of the ACM 26(11), pages 832-843, 1983.
18. C. Berberidis, L. Angelis and I. Vlahavas, PREVENT: An Algorithm for Mining Inter-transactional Patterns for the Prediction of Rare Events, In Proc. of the 2nd European Starting AI Researcher Symposium (STAIRS 04), pp. 128-136, 23-24, Valencia, Spain, 2004.
19. W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan, AdaCost: Misclassification cost-sensitive boosting, In Proc. of the 6th International Conference on Machine Learning (ICML-99), Bled, Slovenia, 1999.
20. P. Domingos, MetaCost: A General Method for Making Classifiers Cost-Sensitive, In Proc. of the Fifth International Conference on Knowledge Discovery and Data Mining (pp. 155-164), 1999. San Diego, CA: ACM Press.
21. W. Cheung and O. R. Zaïane, "Incremental Mining of Frequent Patterns Without Candidate Generation or Support Constraint", In Proc. of the Seventh International Database Engineering and Applications Symposium (IDEAS 2003), pp 111-116, Hong Kong, China, July 16-18, 2003.
22. P. Kam, A. W. Fu, Discovering temporal patterns for interval-based events, In Proc. of the 2nd International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2000). UK, 2000.
23. F. Höppner, Discovery of Temporal Patterns - Learning Rules about the Qualitative Behaviour of Time Series, Proc. of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases, Lecture Notes in Artificial Intelligence 2168, Springer. Freiburg, Germany, pp. 192-203, Sept. 2001.
24. R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September 1994
25. T. Fawcett and F. Provost, Activity Monitoring: Noticing interesting changes in behavior, In Proc of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.53-62, San Diego, California, USA, 1999.
26. L. Torgo and R. Ribeiro, Predicting Outliers, In Proc. of Principles of Data Mining and Knowledge Discovery (PKDD'03), Lecture Notes in Artificial Intelligence 2838, Springer, pp. 447-458, Cavtat-Dubrovnik, Croatia, 2003.
27. L. Torgo, Regression Error Characteristic Surfaces, In Proc. of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05), ACM Press, Chicago, Illinois, USA, 2005.
28. F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In Proc. of the 15th International Conf. on Machine Learning, pp. 445–453. Morgan Kaufmann, San Francisco, CA, 1998.

29. Nathalie Japkowicz, Shaju Stephen. The class imbalance problem: A systematic study. Intelligent Data Analysis. IOS Press, Vol. 6, No. 5: pp. 429-449, 2002.
30. G. Tzanis, C. Berberidis, A. Alexandridou, I. Vlahavas, "Improving the Accuracy of Classifiers for the Prediction of Translation Initiation Sites in Genomic Sequences", In Proc. of the 10th Panhellenic Conference on Informatics (PCI'2005), P. Bozanis and E.N. Houstis (Eds.), Springer-Verlag, LNCS 3746, pp. 426 – 436, Greece, November, 2005.
31. I. H. Witten and E. Frank, "Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers, 2000.
32. Ferenc Bodon, A fast APRIORI implementation, IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'03), Melbourne, Florida, USA, 2003.
33. Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the sensitivity of support vector machines. Proceedings of the International Joint Conference on AI, 55–60.
34. Wu, G. & Chang, E. (2003). Class-Boundary Alignment for Imbalanced Dataset Learning. In ICML 2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC.
35. Eleazar Eskin. Anomaly detection over noisy data using learned probabilitydistributions. In Proceedings of the Seventeenth International Conferenceon Machine Learning (ICML 2000), 2000.
36. W. Lee, et al, Information-Theoretic Measures for Anomaly Detection, In Proc. of te IEEE Symposium on Security and Privacy, IEEE Computer Society Press, pp. 130-143, Oakland (CA), 2001.
37. E. Knorr and R. Ng, Algorithms for Mining Distance-Based Outliers in Large Datasets. In Proc. International Conference on Very Large Databases (VLDB '98), pp. 392-403, 1998.
38. Ramaswamy, S., Rastogi, R., and Shim, K. 2000. Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00). ACM Press, pp. 427-438 Dallas, Texas, United States, May 15 - 18, 2000.
39. M.M. Breunig, H.P.Kriegel, R.T.Ng, J. Sander: LOF: Identifying density-based local outliers. In: Proceedings of SIGMOD'00, pp. 427-438, Dallas, Texas, 2000.
40. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S. A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. Applications of Data Mining in Computer Security, D. Barbara and S. Jajodia (eds), Kluwer Academic Publishers, May 2002.
41. Simon Hawkins , Hongxing He , Graham J. Williams , Rohan A. Baxter, Outlier Detection Using Replicator Neural Networks, Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery, p.170-180, September 04-06, 2002.
42. A. Lazarevic, L. Eroz, V. Kumar, A. Ozgur and J. Srivastava. A Comparative Study of Anomaly Detection Schemes. In Network Intrusion Detection; Proceeding of Third SIAM International Conference on Data Mining, San Francisco, 2003.
43. Charles X. Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In Proceedings 4th International Conference on Knowledge Discovery in Databases (KDD-98), New York, 1998.
44. Kubat, M. and Matwin, S., Addressing the Curse of Imbalanced Data Sets: One-Sided Sampling, in Proceedings of the Fourteenth International Conference on Machine Learning (ICML 97), pp. 179-186, 1997.
45. N. Chawla, K. Bowyer, L. Hall, P. Kegelmeyer, SMOTE: Synthetic Minority Over-Sampling Technique, JAIR, vol. 16, 321-357, 2002.
46. H. Alhammady, K. Rao, The Application of Emerging Patterns for Improving the Quality of Rare-class Classification, In Proc. 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2004) , Springer LNCS, 3056, pp. 207 – 211, Sydney, Australia, 2004.

47. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119--139, August, 1997.
48. W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan. AdaCost: Misclassification cost-sensitive boosting. In Proceedings of the Sixteenth International Conference on Machine Learning, pp. 99-105, 1999.
49. E. Keogh, S. Lonardi, and W. Chiu. Finding surprising patterns in a time series database in linear time and space. Proc. ACM Knowledge Discovery and Data Mining, pp 550-556, 2002.
50. Vilalta, R. and Ma, S., Predicting Rare Events In Temporal Domains. In Proceedings of the 2002 IEEE international Conference on Data Mining (ICDM 02), IEEE Computer Society, Washington, DC, December 09 - 12, 2002.
51. Chen, J., He, H., Williams, G. & Jin, H. (2004), Temporal sequence associations for rare events, in 'Proceedings of 8th Pacific-Asia International Conference on Knowledge Discovery and Data Mining (PAKDD), Lecture Notes in Computer Science (LNAI 3056), Sydney, Australia, pp. 235—239
52. KEEL: Computational Environment for Knowledge Extraction based in Genetic and Evolutionary Learning Algorithms, URL: http://sci2s.ugr.es/keel/datasets.php
53. Proceedings of the AAAI 2000 Workshop on Learning from Imbalanced Data Sets, N. Japkowitz (ed), AAAI Tech. Report WS-00-05, AAAI, 2000.
54. Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data Sets, N. V. Chawla, N. Japkowitz and A. Kolcz (eds), 2003.
55. Gary M. Weiss, Foster J. Provost. Learning When Training Data are Costly: The Effect of Class Distribution on Tree Induction. Journal of Artificial Intelligence Research (JAIR) 19, pp.315-354, 2003.
56. A. Lazarevic, Data Mining for Analysis of Rare Events: A Case Study in Security, Financial and Medical Applications, Tutorial in 8th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD), Sydney, Australia, May 26, 2004.
57. C. Elkan. Invited talk: The Real Challenges in Data Mining: a Contrarian View. http://www-cse.ucsd.edu/users/elkan/realchallenges2.ppt, 2003.