Data mining of temporal sequences for the prediction of infrequent failure events : application on floating train data for predictive maintenance

Wissam Sammouri

Université PARIS-EST

Ècole doctorale Mathmatiques et Sciences et Technologies de l'Information et de la Communication (MSTIC)

THÈSE

présentée en vue de l'obtention du Grade de

Docteur de l'Université Paris-Est

Spécialité: Signal, Image et Automatique

par

Wissam Sammouri

Data mining of temporal sequences for the prediction of infrequent failure events: Application on Floating Train Data for predictive maintenance

Fouille de séquences temporelles pour la maintenance prédictive. Application aux données de véhicules traceurs ferroviaires

Jury :

AbdelHakim ARTIBA	Professeur Université de Valenciennes	Rapporteur
Walter SCHÖN	Professeur Université de Technologie de Compiègne	Rapporteur
Said MAMMAR	Professeur Université d'Evry Val d'Essonne	Examinateur
Latifa OUKHELLOU	Directrice de Recherche IFSTTAR	Directrice de thèse
Etienne CÔME	Chargé de Recherche IFSTTAR	Examinateur
Patrice AKNIN	Directeur Scientifique SNCF	Examinateur
Charles-Eric FONLLADOSA	Chef de Projet R&D ALSTOM	Invité

À ma famille,

Acknowledgements

Je voudrais tout d'abord exprimer mes plus profonds remerciements à ma directrice de thèse Mme Latifa Oukhellou pour ses précieux conseils, support et encouragements. J'aimerais aussi remercier mon encadrant Etienne Côme pour son écoute.

Je remercie tous les membres du jury de m'avoir fait l'honneur de participer à l'évaluation de mes travaux de thèse. Je remercie notammenet Mr Walter Schon et Mr AbdelHakim Artiba, qui ont accepté de rapporter sur ces travaux, pour leur lecture attentive du manuscrit et leurs observations toujours constructives. Merci également à tous ceux qui ont contribué à me faire avancer et qui m'ont transmis leurs connaissances, les nombreux professeurs qui ont marqué ma scolarité et m'ont fait choisir cette direction.

Je remercie Alstom de m'avoir confié à une mission enterprise, ce qui était une contribution majeure à cette thèse. Il n'y a pas de fouille de données sans données.

Plus généralement, je tiens à adresser mes remerciements les plus chaleureux à l'ensemble des membres de l'équipe de recherche du GRETTIA avec qui j'ai eu le plaisir d'échanger et aussi de me détendre. Avec une pensée particulière pour Hani El Assaad. Merci également à Allou S., Olivier F., Moustapha T., Annie T., Andry R., Carlos D-M., Laura P. et Ferhat A. pour leur aide.

Je souhaite remercier mes amis pour leur support et leurs encouragements en particulier pendant la période de rédaction. Cette phase aurait été beaucoup plus difficile sans vous. En particulier Mahmoud Sidani, Ghaydaa Assi et Elena Salameh d'avoir été là dans les moments les plus difficiles. Je pense aussi à Hiba F., Hasan S., Kamar S., Diala D., Youmna C., Christel M-M. et Aura P..

Je terminerai ce préambule en remerciant mes parents et ma famille pour leur amour et support inconditionnel, pour leurs encouragements infaillibles et pour une infinité de choses. Sans vous rien n'aurait été possible.

Wissam Sammouri
Paris, 20 Juin 2014

Abstract

Data mining of temporal sequences for predictive maintenance: Application on floating train data

In order to meet the mounting social and economic demands, railway operators and manufacturers are striving for a longer availability and a better reliability of railway transportation systems. Commercial trains are being equipped with state-of-the-art on-board intelligent sensors monitoring various subsystems all over the train. These sensors provide real-time flow of data, called floating train data, consisting of georeferenced events, along with their spatial and temporal coordinates. Once ordered with respect to time, these events can be considered as long temporal sequences which can be mined for possible relationships. This has created a necessity for sequential data mining techniques in order to derive meaningful associations rules or classification models from these data. Once discovered, these rules and models can then be used to perform an on-line analysis of the incoming event stream in order to predict the occurrence of target events, i.e, severe failures that require immediate corrective maintenance actions. The work in this thesis tackles the above mentioned data mining task. We aim to investigate and develop various methodologies to discover association rules and classification models which can help predict rare tilt and traction failures in sequences using past events that are less critical. The investigated techniques constitute two major axes: Association analysis, which is temporal and Classification techniques, which is not temporal. The main challenges confronting the data mining task and increasing its complexity are mainly the rarity of the target events to be predicted in addition to the heavy redundancy of some events and the frequent occurrence of data bursts. The results obtained on real datasets collected from a fleet of trains allows to highlight the effectiveness of the approaches and methodologies used.

Keywords: Data mining, Temporal sequences, Association rules, Pattern recognition, Classification, Predictive maintenance, Floating Train Data.

Abstract

Fouille de séquences temporelles pour la maintenance prédictive. Application aux données de véhicules traceurs ferroviaires.

De nos jours, afin de répondre aux exigences économiques et sociales, les systèmes de transport ferroviaire ont la nécessité d'être exploités avec un haut niveau de sécurité et de fiabilité. On constate notamment un besoin croissant en termes d'outils de surveillance et d'aide à la maintenance de manière à anticiper les défaillances des composants du matériel roulant ferroviaire. Pour mettre au point de tels outils, les trains commerciaux sont équipés de capteurs intelligents envoyant des informations en temps réel sur l'état de divers sous-systèmes. Ces informations se présentent sous la forme de longues séquences temporelles constituées d'une succession d'événements. Le développement d'outils d'analyse automatique de ces séquences permettra d'identifier des associations significatives entre événements dans un but de prédiction d'événement signant l'apparition de défaillance grave. Cette thèse aborde la problématique de la fouille de séquences temporelles pour la prédiction d'événements rares et s'inscrit dans un contexte global de développement d'outils d'aide à la décision. Nous visons à étudier et développer diverses méthodes pour découvrir les régles d'association entre événements d'une part et à construire des modèles de classification d'autre part. Ces règles et/ou ces classifieurs peuvent ensuite être exploités pour analyser en ligne un flux d'événements entrants dans le but de prédire l'apparition d'événements cibles correspondant à des défaillances. Deux méthodologies sont considérées dans ce travail de thèse: La première est basée sur la recherche des règles d'association, qui est une approche temporelle et une approche à base de reconnaissance de formes. Les principaux défis auxquels est confronté ce travail sont principalement liés à la rareté des événements cibles à prédire, la redondance importante de certains événements et à la présence trés fréquente de "bursts". Les résultats obtenus sur des données réelles recueillies par des capteurs embarqués sur une flotte de trains commerciaux permettent de mettre en évidence l'efficacité des approches proposées.

Mots clés: Fouille de données, Séquences temporelles, Règles d'associations, Classification, Maintenance Prédictive, Véhicules traceurs ferroviaires.

Contents

Chapter 1

Introduction

1.1 Context and Problematic

In order to meet the mounting social and economic demands as well as the pressure to stand out within fierce global competitivity, railway operators and manufacturers are striving for a longer availability and a better reliability of railway transportation systems. A permissive and lax maintenance strategy such as “run-to-failure” can lead to sizable maintenance costs not to mention the loss of public credibility and commercial image. Also, a systematic schedule-based maintenance policy can be uselessly time and resource consuming. From an intuitive point of view, the most intelligent maintenance policy exploits the functional lifetime of a component till the end. We thus speak of opportunistic maintenance which refers to the scheme in which preventive maintenance is carried out at opportunities based on the physical condition of the system. The automatic diagnosis of the physical condition of systems allows to detect degradation or failures either prior or directly upon their occurrence. Diagnosis is a term which englobes at the same time the observation of a situation (monitoring of an industrial system) and the relevant decisions to be taken following this observation (system degraded or not, etc.). It is a vast research field uniting researchers of multiple scientific communities such as control, signal processing, statistics, artificial intelligence, machine learning, etc.

In order to establish this maintenance policy in the railway domain, probe train vehicles equipped with intelligent sensors dedicated for the monitoring of railway infrastructure (rail, high-voltage lines, track geometry, etc.) have been widely used in the recent years. However, these vehicles require certain logistic measures since they cannot circulate all the time. This shifted railway operators and manufacturers towards instrumenting commercial trains with sensors for the same purpose. While a commercial train is operating, these sensors monitor different systems and send information in real time via wireless technology to centralized data servers. This new approach thus allows the constant and daily diagnosis of both vehicle components and railway infrastructure. However, the high number of commercial trains to be equipped demands a trade-off between the equipment cost and their performance in order to install sensors on all train components. The quality of these sensors reflects directly on the frequency of data bursts and signal noise, both rendering data analysis more challenging. The main advantage of this approach lies in the huge quantity of obtained data, which if exploited and mined, can contribute to the benefit of the diagnosis process.

1.2 Positioning, objectives and case study of the thesis

The recent leaps in information technology have reflected a boost in the capacity to stock data as well as in both processing and computational powers. This has leveraged the use of intelligent monitoring systems which paved the way for automatic diagnosis procedures. Similar to floating car data systems which are now broadly implemented in road transportation networks, floating train data systems have also been recently developed in the railway domain. Commercial trains equipped with state-of-the-art on-board intelligent sensors provide real-time flow of data consisting of georeferenced events, along with their spatial and temporal coordinates. Once ordered with respect to time, these events can be considered as long temporal sequences which can be mined for possible relationships. This has created a necessity for sequential data mining techniques in order to derive meaningful association rules or classification models from these data. Once discovered, these rules and models can then be used to perform an online analysis of the incoming event stream in order to predict the occurrence of target events, i.e, severe failures that require immediate corrective maintenance actions.

The work in this thesis tackles the above mentioned data mining task. We aim to investigate and develop various methodologies to discover associations (association rules and episode rules) and classification models which can help predict rare failures in sequences. The investigated techniques constitute two major axes: **Association analysis**, which is temporal, and aims to discover rules of the form $A \longrightarrow B$ where B is a failure event using significance testing techniques (T-Patterns, Null models, Double Null models) as well as Weighted association rule mining (WARM)-based algorithms, and **Classification techniques**, which is not temporal, where the data sequence is transformed using a methodology that we propose into a data matrix of labeled observations and selected attributes, followed by the application of various pattern recognition techniques, namely K-Nearest Neighbours, Naive Bayes, Support Vector Machines and Neural Networks to build a classification model that will help predict failures.

The main challenges confronting the data mining task and increasing its complexity are mainly the rarity of the target events to be predicted in addition to the heavy redundancy of some events and the frequent occurrence of data bursts.

Industrial subsystems susceptible to be the most monitored are those presenting strong security requirements and low intrinsic reliability. Within a railway context, in a train, the tilt and traction systems correspond to this type of description. A failure in any of these subsystems can result in an immediate stop of the vehicle which can heavily impact the whole network both financially and operationally. The present work will focus on the prediction of these two types of target events using past events that are less critical. The real data upon which this thesis work is performed was provided by Alstom transport, a subsidiary of Alstom. It consists of a 6-month extract from the TrainTracer database. TrainTracer™ is a state-of-the-art **C**entralized **F**leet **M**anagement (CFM) software conceived by Alstom to collect and process real-time data sent by fleets of trains equipped with on-board sensors monitoring 31 various subsystems such as the auxiliary converter, doors, brakes, power circuit and tilt.

1.3 Organization of the dissertation

This document consists of six chapters.

In Chapter 2, we introduce the context and the problematic of the study. We precise where the work of this thesis stands in the corresponding research field and identify the objectives and the applicative case study. First, we discuss the field of Data Mining and explain its general process, we then highlight the different types of maintenance policies while emphasizing on predictive maintenance in which the context of this thesis lies. We present an extended state of the art survey on data mining approaches applied to the railway domain. Following that, we then tackle the applicative context of the thesis. We introduce TrainTracer, from which the data extracts used in this thesis were furnished and describe how data is organized. We then invoke the major constraints and expected difficulties. Finally, we converge the above tackled subjects into formally defining the applicative and theoretical contexts in which the thesis lie.

Chapter 3 introduces our first contribution in this thesis. We first formally define the association rule mining problem and discuss its two most influential breadth-first and depth-first approaches used. In this chapter, two hypothesis-test-based significance testing methods are especially adapted and compared to discover significant co-occurrences between events in a sequence: **Null models** and **T-Patterns algorithm**. In addition to that, a bipolar significance testing approach, called **Double Null Models (DNM)** is proposed, applied and confronted with the above mentioned approaches on both synthetic and real data.

In Chapter 4, We focus on the problem of Episode rule mining in sequences. We formalize the problem by introducing basic notations and definitions and then discussing related work in this context. Following an extensive literature survey, we formally define the weighted association rule mining problem and adapt it to the problem of mining episode rules in temporal sequences. We propose a methodology called **Weighted Winepi** aimed to find significant episode rules between events and an approach derived from it to better include infrequent events in the mining process. We also propose **"Oriented Weighted Winepi"** which is more suitable to the applicative problematic of this thesis which is to find episodes leading to target events. Methods are confronted and tested on synthetic and real data.

In Chapter 5, we first introduce the general principle of pattern recognition. We explain briefly the principal approaches used in our work: ***K*-Nearest Neighbours**,

Naive Bayes, **Support Vector Machines** and **Neural Networks**. We propose a methodology to transform data sequence into a labelled data matrix of labelled observations and selected attributes. We then propose a hypothesis-testing-based approach to reduce the dimensionality of the data. Results obtained by all classifiers on real data are confronted and analyzed.

In the last part of this thesis in Chapter 6, we review and conclude the contributions of our work and discuss research perspectives as well as arising issues.

Chapter 2

Applicative context: Predictive maintenance to maximize rolling stock availability

Contents

2.1 Introduction

Computers have promised us a fountain of wisdom but delivered a flood of data.
– A frustrated MIS executive (*Fayyad et al., 1996*)

The recent couple of decades have witnessed an unprecedented advancement in information technologies which has leveraged a gigantic upgrade in data storage capacities. For instance, the amount of data that can be stocked in hard drives has increased from the order of Kilobytes (10^3 bytes) in the 1980s to Petabytes (10^{15} bytes) very recently. This on-the-go advancement did not only concern hardware but software as well. Database utilities have been revolutionized in its various functional aspects such as: data collection, database creation, data management (storage and retrieval), advanced data analysis (ERP software, data warehousing and mining) and visualization. This technology reflected in the boost of the database and information industry, and led to the abundancy of huge volumes of databases and information repositories which can be analyzed and mined for valuable information within various domains and applications. However, the enormous amounts of data have far exceeded our human analysis ability, thus transforming data repositories into data archives that are rarely consulted. This situation can best be described as a *data rich but information poor situation.* This has created a need for data mining techniques to do an automated analysis that is at the same time powerful and fast.

In this thesis, we are particularly interested in the railway transportation field. System failures and general upkeep of rolling stock aren't just costly to operators because of repairs. The time loss in maintenance depots as well as in corrective maintenance procedures affects negatively the quality of the operator's service due to reduced fleet availability. Diagnosing the problem, ordering replacement parts, troubleshooting: all of these time-consuming activities are the greatest impediments keeping operators from peak efficiency, and thus from optimal profit. The recent leaps in information and communication technologies have revolutionized support techniques for rail equipment maintenance and triggered an evolution in preventive maintenance strategies towards more optimized and cost effective solutions that aim to provide longer availability and better reliability of transportation systems. Similar to smart vehicles, commercial trains are being equipped with positioning and communication systems as well as on-board intelligent sensors monitoring various subsystems such as tilt, traction, signalling, pantograph, doors, etc. These sensors provide a real-time flow of spatio-temporal data consisting of georeferenced alarms, called events, which are transferred wirelessly towards centralized data servers where they are stocked and exploited within a specially-conceived data-warehousing and analysis system called ***Floating Train Data system (FTD)***. The information extracted from these data are used to establish a unified preventive (condition-based) maintenance management as well as a more-advanced predictive maintenance approach which consists of performing an online analysis of the incoming event stream in order to predict and alert the imminent arrival or the increased probability of occurrence of severe failure events, i.e., failures requiring immediate corrective maintenance actions, also called target events.

In this chapter, we introduce the applicative context of this thesis which is the maximization of rolling stock availability by mining floating train data sequences within a predictive maintenance framework. We first define Data Mining and explain its general process in section 2.2. We then highlight the different types of maintenance policies in 2.3.1 while emphasizing on predictive maintenance in which the context of this thesis lies. In 2.3.2 we present an extended literature survey on data mining approaches applied to the railway domain. We then tackle the applicative context of the thesis in 2.4. We introduce TrainTracer, from which the data extracts used in this thesis were furnished and describe how data is organized in 2.4.1. We then invoke the major constraints and difficulties in 2.4.2 and the approaches used to clean data in 2.4.3. Finally, we converge the above tackled subjects into positioning the work of this thesis by formally defining the applicative and theoretical contexts in 2.5.

2.2 Data Mining: Definition and Process Overview

Data mining refers to the extraction or "mining" of knowledge from large amounts of observed data. It is a vast domain of diverse algorithms and techniques which comply with different types of data types and problems. It involves an integration of techniques and methods from multiple disciplines such as database and warehouse technology, statistics, probability, pattern recognition (Neural Networks, Support Vector Machine, K-Nearest Neighbours, Decision Trees, etc.), data visualization, etc. It is an essential step in the process of **K**nowledge **D**iscovery in **D**atabases (KDD).

KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad et al., 1996) and consists of an iterative sequence of steps that can be classified into three main categories: pre-processing, data mining and post-processing, which will be briefly defined next.

1. **Data Pre-processing**

   This stage is considered as preparing the data for mining. It consists of a set of measures and operations executed over data in order to improve its quality, which will directly reflect on mining results. This step consists of three main types of techniques precised by (Han et al., 2006) to be: data integration, data cleaning and data reduction.

   - Data Integration

   Data integration consists of concatenating and merging data from different sources. The main obstacles are mainly homogenizing data by neutralizing possible structural differences behind the different data sources without losing valuable information, in addition to the identification and removal of redundancies, i.e duplicate data records.

   - Data Cleaning

   Once the data integration is achieved, noise and errors should be handled. Data is cleaned in order to enhance mining quality. The cleaning process aims to eliminate inconsistent or missing values, outliers and noise either manually or automatically. The approaches used to deal with this problem consist mainly of either ignoring/deleting records that contain missing or inconsistent values or correcting these records by injecting acceptable values.

- Data Reduction

One of the most important challenges facing data mining algorithms is scalability. Practically all learning algorithms consider in advance that data are fit to main memory and disregard how extremely sizable databases could be dealt with when only a limited portion of data can be viewed. In addition to that, computational time increases significantly with the size or complexity of the data. For this reason, data reduction is a real need to simplify the mining process. Several possible approaches can be carried out to reduce dimensionality including removing redundancies and irrelevant data, data sampling, feature selection, windowing, batch-incremental mode and parametric modelling for a lighter presentation of the data.

2. **Data Mining**

   This step is the core and the major challenge in the KDD process. It involves the selection and application of appropriate algorithms and methodologies to extract patterns or knowledge from the data. One big challenge, which is also the main motivation behind this thesis, is to adapt existing algorithms, exploiting all known information and imposing constraints, on "dirty and ambiguous" data in order to focus the discovery process to comply with users expectations as well as with the applicative context. Another related challenge is the analysis of complex temporal data, which will be discussed more in details in the next chapter.

   Data mining is a vast domain of algorithms and techniques that can be very diversified to comply with different types of problems such as mining frequent or rare patterns in different types of data such as transaction databases or sequences of events that can be temporal or non temporal. These techniques can be assembled into three main categories: classification/prediction, clustering and association rule mining, succinctly described below.

- Classification and Prediction

Classification is the task of assigning observations to one of several predefined data classes. In order to describe and differentiate between these classes, classification methods aim at learning a classifying model or function, called classifier c, which can be defined as a function from a set of instances each with an attribute set x and a set of class labels y ($c : x \longrightarrow y$). For the model to be derived within a supervised context, a set of already labelled observations, called *training set*, is used. This model will then be implemented to predict the class of unlabelled observations. A classification model can be viewed as a black box that automatically assigns a class label when presented with the attribute set of an unknown observation. Book references: (Bishop, 2006; Han et al., 2006; Hastie et al., 2003)

- Cluster analysis

Cluster analysis divides data into clusters that are meaningful, useful or both (Tan et al., 2005). It is used in situations where the privilege of having a training set with known class labels does not exist, which means that the training is performed in an *unsupervised* framework. Clustering techniques analyze similarities and dissimilarities between instances and partition data into natural classes, *clusters*, which are then used to predict labels. The similarity measure used to identify clusters, also called *distance function*, is a major challenge to define, especially in the presence of complex data types. Book references: (Han et al., 2006; Hastie et al., 2003)

- Association analysis: Association and Episode rule mining

Association rule mining is the discovery of existing reliable (and not necessary frequent) dependencies between items in transaction data or events in a sequence. These dependencies are called association rules. More formally, an association rule is an implication of the form $A \implies B$, where A_i (for $i \in 1, ..., m$) and B_j (for $j \in 1, ..., n$) are attribute-value pairs, indicating that when A occurs, B occurs with a certain probability $P(B|A)$, called *confidence*. Association analysis, although initially developed for market basket analysis problems, is now widely used for transaction data analysis and more recently in sequence analysis to discover what is called *episode rules* between events (Mannila and Toivonen, 1996). Most of the developed algorithms are frequency-oriented, i.e, discover frequent itemsets and episodes first before generating the rules, which leads to a sizable number of discovered associations that are not necessarily significant and interesting. Strong but rare associations remain usually undetected, mainly due to frequency constraints set to reduce time complexity. This problem is a major motivation in this thesis and will be discussed further. Book references: (Han et al., 2006; Hastie et al., 2003; Ye, 2003).

3. **Post-processing**

With insufficient ground truth or the lack of it, data mining operations may lead classification and clustering algorithms to discover various models that can explain or describe the data very differently. Furthermore, after a learning system discovers concept models from the training set, their evaluation should take place on a testing set using several criteria and measures such as classification accuracy, mean square error, correct classification rate, scalibility, etc.

Similarly, association rule mining algorithms are destined to estimate a huge number of associations of which the majority is of low significance and utility. It is vital to analyze results in order to select the best ones to the final users. This has motivated a large number of researchers to develop *interestingness measures* that are used to evaluate results. Interestingness measures can be assembled into two main categories: Objective and Subjective. Objective interestingness measures are generally probability-based and are usually functions of a 2x2 contingency table. A survey of those measures can be found in (Nada Lavrac et al., 1999; Tan and Kumar, 2002) as well as in (Lenca et al., 2004; Ohsaki et al., 2004).

In some cases, the information provided by objective measures might not be sufficient to judge if a rule is significant enough to be considered and thus a subjective point of view is needed. A subjective interestingness measure takes into account both the data and the user's knowledge. Such a measure is appropriate when: (1) the background knowledge of users varies, (2) the interests of the users vary, and (3) the background knowledge of users evolve. Subjective measures cannot be represented by simple mathematical formulas because the user's knowledge may be expressed in various forms such as visualization, experience, etc. Instead, they are usually incorporated into the mining process.

Although we agree that data mining is a step in the knowledge discovery process, however in reality, the term data mining is used by industry, media and research to describe the whole knowledge discovery process instead of just a step in it. Therefore in this thesis, we choose to adopt the latter view since it broadens the data mining functionality and is more appropriate for the industry-oriented applicative nature of this work.

2.3 Railway Context

The recent leaps in information and communication technologies have revolutionized support techniques for rail equipment maintenance and triggered an evolution in preventive maintenance strategies toward more optimized and cost effective solutions. These processes aim to provide longer availability and better reliability of transportation systems. Similar to smart vehicles, commercial trains are being equipped with positioning and communication systems as well as on-board intelligent sensors monitoring various subsystems such as tilt, traction, signalling, pantograph, doors, etc. These sensors provide a real-time flow of spatio-temporal data consisting of georeferenced alarms, called events, which are transferred wirelessly towards centralized data servers where they are stocked and exploited within a specially-conceived data-warehousing and analysis system called Floating Train Data system (FTD). The information extracted from these data are used to establish a unified preventive (condition-based) maintenance management as well as a more-advanced predictive maintenance approach which consists of performing an on-line analysis of the incoming event stream in order to predict and alert the imminent or increased probability of occurrence of severe failure events, i.e., failures requiring immediate corrective maintenance actions, also called target events.

2.3.1 Existing Maintenance Policies

Maintenance costs are a major portion of the total operating costs of all manufacturing or production plants. Depending on the specific industry, these costs can represent between 15 and 60 percent of the costs of goods produced (Mobley, 2002). The recent development of microprocessor and computer-based instrumentation that can be used to monitor the operating condition of equipment and systems have provided the means to eliminate unnecessary repairs, prevent catastrophic machine failures and reduce the negative impact of maintenance operations on the profitability of manufacturing and production plants.

To understand what predictive maintenance is, traditional policies should first be considered. Figure 2.1 shows the evolution of maintenance strategies in time. The earliest technique (and the most frequent up-till-now), corrective maintenance (also called Run-to-failure or reactive maintenance), is a simple and straightforward procedure which consists of waiting till the failure occurs to replace defected pieces. The main disadvantages of this approach include fluctuant and unpredictable production as well as the high costs of un-planned maintenance operations. The advancement in industrial diagnosis instrumentation led to the emergence of time-driven preventive maintenance policies such as schedule-based preventive maintenance where pieces are replaced before their formally-calculated **M**ean **T**ime **T**o **F**ailure (MTTF) is attained. In order to efficiently implement this periodic maintenance policy, an operational research is required to find the optimal maintenance schedule that can reduce operation costs and increase availability. This scheduling takes into consideration the life cycle of equipment as well as man power and work hours required. In many cases, maintenance policies are still based on the maintenance schedules recommended by the user, which are usually conservative or are only based on qualitative information driven by experience and engineering rationale (Zio, 2009). Several approaches were developed to assess the performance of a maintenance policy, especially in case of complicated systems. For instance, in (Marseguerra and Zio, 2002; Zio, 2013), Monte Carlo simulation is used in order to avoid the introduction of excessively simplifying hypotheses in the representation of the system behavior. This framework was extended in (Baraldi et al., 2011) by combining it with fuzzy logic in the aim of modelling component degradation in electrical production plants.

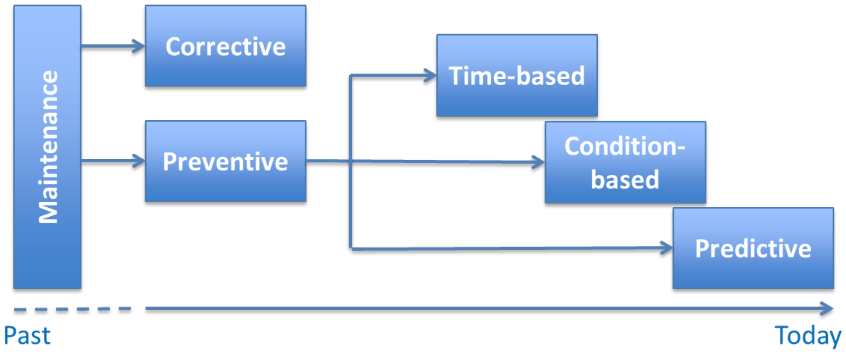


Figure 2.1: Evolution of maintenance policies in time

The development of intelligent sensors and condition-assessment tools have paved the way for a condition-based maintenance policy. The operating state of a system is constantly monitored by means of a dedicated monitoring tool. Once degradation is detected, the system is replaced. This method increases the component operational lifetime and availability and allows preemptive corrective actions. On the other hand, it necessitates an increased investment in efficient monitoring equipment as well as in maintenance staff training.

The steady progress of computer hardware technology as well as the affordability and availability of computers, data collection equipment and storage media has reflected a boost in the amount of data stocked by people and firms. However, the abundance of these data without powerful analysis tools has led to *data rich but information poor situations* where data repositories became data archives that are seldomly visited. The presence of this data has inspired researchers to develop algorithms which automatically analyze data in order to find associations or models that can help predict future failures. These algorithms have established what is now called predictive maintenance. Predicting system degradation before it occurs may lead to the prevention or at least the avoidance of bad consequences.

Predictive maintenance can be defined as the measurements which detect the commencement of system degradation and thus an imminent breakdown, thereby allowing to control or eliminate causal stressors early enough to avoid any serious deterioration in the component's physical state. The main difference between predictive maintenance and schedule-based preventive maintenance is that the former bases maintenance needs on the actual condition of the machine rather than on some predefined schedule and hence it is condition-based and not time-based. For example, the *VCB (Vacuum Circuit Breaker)*, whose role is to isolate the power supply of high voltage lines when there is a fault or need for maintenance is replaced preventively every 2 years without any concern for its actual condition and performance capability. It is replaced simply because it is time to. This methodology would be analogous to a time-based preventive maintenance task. If, on the other hand, the operator of the train, based on formerly acquired experience, have noticed some particular events or incidents which frequently precede the failure of the VCB, then, after insuring that safety procedures are being respected, he/she may be able to extend its replacement until these events or incidents appear, and thus optimizing the usage of material and decreasing maintenance costs. Figure 2.2 shows a comparison between maintenance policies in terms of total maintenance cost and total reliability. Predictive maintenance, in cases where it can be applied efficiently, is the least expensive and assures an optimal reliability with respect to other policies.

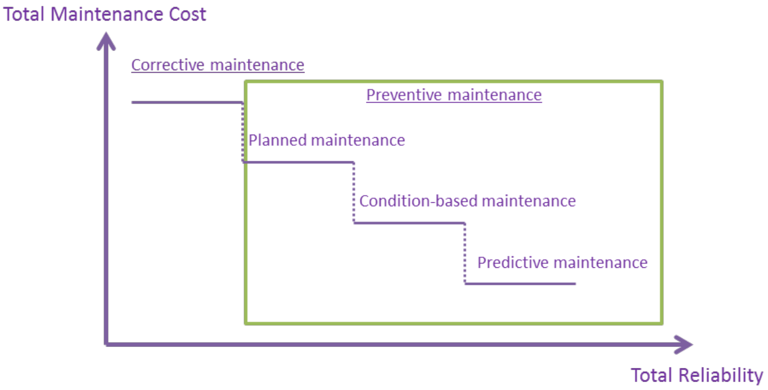


Figure 2.2: Comparison of maintenance policies in terms of total maintenance costs and total reliability

2.3.2 Data mining applied to the railway domain: A survey

The mounting socio-economic demand on railway transportation is a challenge for railway network operators. The availability and reliability of the service has imposed itself as the main competitivity arena between railway manufacturer moguls. Decision-making for maintenance and renewal operations is mainly based on technical and economic information as well as knowledge and experience. The instrumentation of railway vehicles as well as infrastructure by smart wireless sensors has provided huge amounts of data that, if exploited, might reveal some important hidden information that can contribute to enhancing the service and improving capacity usage. For this reason, recent years have witnessed an increasing uprise in applicative research and projects in this context. Research fields receiving the biggest focus were railway infrastructure and railway vehicles. Other fields include scheduling and planning as well as predicting train delays to increase punctuality. All of which we discuss below.

1. Monitoring railway infrastructure

   Railway infrastructure maintenance is a major concern for transportation companies. Railway infrastructure is constantly subjected to traffic and environmental effects that can progressively degrade track geometry and materials. It is vital to discover these degradations at early stages to insure safety as well as comfort of the service. There are three main systems that can be used for inspection (Bocciolone et al., 2002, 2007; Grassie, 2005): *Portable manual devices* (which can give information on a relatively low length of track, operated by maintenance technicians), *Movable devices* (which are fully mechanically driven and can cover long distances), and very recently the *implementation of sensors on active commercial or probe vehicles* to perform various types of measures in order to identify track irregularities. The advantage of the latter system is that data can be collected more frequently and at anytime.

Several tools and methods exploiting collected data for infrastructure condition inspection (tracks, track circuit, rail switches and power supply system) have been proposed in the recent years. Table 2.1 summarizes some of the recent works.

Reference	Subsystem concerned	Methodologies Used
(Insa et al., 2012), (Vale and Lurdes, 2013), (Andrade and Teixeira, 2012), (Weston et al., 2006, 2007), (Rhayma et al., 2011, 2013),(Bouillaut et al., 2013), (Yella et al., 2009), (Fink et al., 2014)	Track defects	Statistical methods, Stochastic probabilistic model, Bayesian Networks, Probabilistic approaches, Stochastic finite elements methods, Monte-Carlo simulation procedure, Bayesian networks, Multilayer feedforward neural networks based on multi-valued neurons, Pattern recognition, Classification
(Kobayashi et al., 2013; Kojima et al., 2005, 2006; Matsumoto et al., 2002; Tsunashima et al., 2008)	Track inspection using probe vehicles	Signal processing
(Oukhellou et al., 2010), (Chen et al., 2008), (Lin-Hai et al., 2012)	Track circuit	Neural networks and decision tree classifiers, Neuro-fuzzy system, Genetic algorithm
(Chamroukhi et al., 2010), (Samé et al., 2011)	Rail switches	Mixture model-based approach for the clustering of univariate time series with changes in regime, Regression model
(Cosulich et al., 1996), (Wang et al., 2005), (Chen et al., 2007)	Power supply system	Probabilistic approach based on stochastic reward nets, Radial basis neural networks, finite element analysis with Monte Carlo simulation, Fault Tree Analysis

Table 2.1: Examples of recent research work along with the methodologies used for the condition inspection of various train infrastructure subsystems

2. Monitoring railway rolling stock

   The recent years have witnessed the development of numerous approaches for the monitoring of railway rolling stock material. One of the subsystems receiving a lot of focus is doors. In general, doors of public transportation vehicles are subject to exhaustive daily use enduring a lot of direct interactions with passengers (pushing and leaning on the doors). It is important to note that malfunctions encountered with doors are usually due to mechanical problems caused by the exhaustive use of components. Each train vehicle is equipped with two doors from each side which can be either pneumatic or electric.

Reference	Subsystem	Methodologies
(Miguelanez et al., 2008), (Lehrasab, 1999; Lehrasab et al., 2002), (Roberts et al., 2002), (Dassanayake, 2002), (Dassanayake et al., 2009),(Han et al., 2013)	Doors	Ontology-based methods, Neural networks, Classification, fuzzy logic, statistical learning
(Bruni et al., 2013)	Axle	Statistical methods
(Randall and Antoni, 2011),(Capdessus et al., 2000),(Zheng et al., 2013),(Antoni and Randall, 2006),(Pennacchi et al., 2011)	Rolling element bearings	Envelope analysis, Squared envelope spectrum, 2nd order cyclostationary analysis, Spectral kurtosis , Empirical mode decomposition, Minimum entropy deconvolution
(Wu and Thompson, 2002),(Pieringer and Kropp, 2008),(Belotti et al., 2006),(Jia and Dhanasekar, 2007),(Wei et al., 2012),(Liang et al., 2013)	Wheels	Dynamic modelling, Signal processing, Wavelet transform methods, Fourier Transform, Weigner-Villa Transform

Table 2.2: Examples of recent research work along with the methodologies used for the condition inspection of various train vehicle subsystems

   Other subsystems receiving focus are the axle and the rolling element bearings since they are the most critical components in the traction system of high speed trains. Monitoring their integrity is a fundamental operation in order to avoid catastrophic failures and to implement effective condition based maintenance strategies. Generally, diagnosis of rolling element bearings is usually performed by analyzing vibration signals measured by accelerometers placed in the proximity of the bearing under investigation. Several papers have been published on this subject in the last two decades, mainly devoted to the development and assessment of signal processing techniques for diagnosis.

With the recent significant increases of train speed and axle load, forces on both vehicle and track due to wheel flats or rail surface defects have increased and critical defect sizes at which action must be taken are reduced. This increases the importance of early detection and rectification of these faults. Partly as a result of this, dynamic interaction between the vehicle, the wheel, and the rail has been the subject of extensive research in recent years.

Table 2.2 resumes some of the important works on train vehicle subsystems in the recent years.

3. Other projects related to railway predictive maintenance

   Numerous projects have been developed in the railway domain that are not only related to railway infrastructure and vehicles but to other applications as well. For example, in (Ignesti et al., 2013), the authors presented an innovative Weight-in-Motion (WIM) algorithm aiming to estimate the vertical axle loads of railway vehicles in order to evaluate the risk of vehicle loading. Evaluating constantly the axle load conditions is important especially for freight wagons, which are more susceptible to be subjected to risk of unbalanced loads which can be extremely dangerous both for the vehicle running safety as well as for infrastructure integrity. This evaluation could then easily identify potentially dangerous overloads or defects of rolling surfaces. When an overload is detected, the axle would be identified and monitored with non-destructive controls to avoid and prevent the propagation of potentially dangerous fatigue cracks. Other examples include the work in (Liu et al., 2011), where the Apriori algorithm is applied on railway tunnel lining condition monitoring data in order to extract frequent association rules that might help enhance the tunnel's maintenance efforts. Also, in (Vettori et al., 2013), a localization algorithm is developed for railway vehicles which could enhance the performances, in terms of speed and position estimation accuracy, of the classical odometry algorithms.

   Due to the high cost of train delays and the complexity of schedule modifications, many approaches were proposed in the recent years in an attempt to predict train delays and optimize scheduling. For example, in (Cule et al., 2011), a closed-episode mining algorithm, CLOSEPI, was applied on a dataset containing the times of trains passing through characteristic points in the Belgian railway networks. The aim was to detect interesting patterns that will help improve the total punctuality of the trains and reduce delays. (Flier et al., 2009) tried to discover dependencies between train delays in the aim of supporting planners in improving timetables. Similar projects were carried out in the Netherlands (Goverde, 2011; Nie and Hansen, 2005; Weeda and Hofstra, 2008), Switzerland (Flier et al., 2009), Germany (Conte and Shobel, 2007), Italy (De Fabris et al., 2008) and Denmark

(Richter, 2010), most of them based on association rule mining or classification techniques.

In the next section, we present the applicative context of this thesis.

2.4 Applicative context of the thesis: TrainTracer

TrainTracer is a state-of-the-art centralized fleet management (CFM) software conceived by Alstom to collect and process real-time data sent by fleets of trains equipped with on-board sensors monitoring various subsystems such as the auxiliary converter, doors, brakes, power circuit and tilt. Figure 2.3 is a graphical illustration of Alstom's TrainTracer^{TM}. Commercial trains are equipped with positioning (GPS) and communications systems as well as on-board sensors monitoring the condition of various subsystems on the train and providing a real-time flow of data. This data is transferred wirelessly towards centralized servers where it is stocked, exploited and analyzed by the support team, maintainers and operators using a secured intranet/internet access to provide both a centralized fleet management and unified train maintenance (UFM).

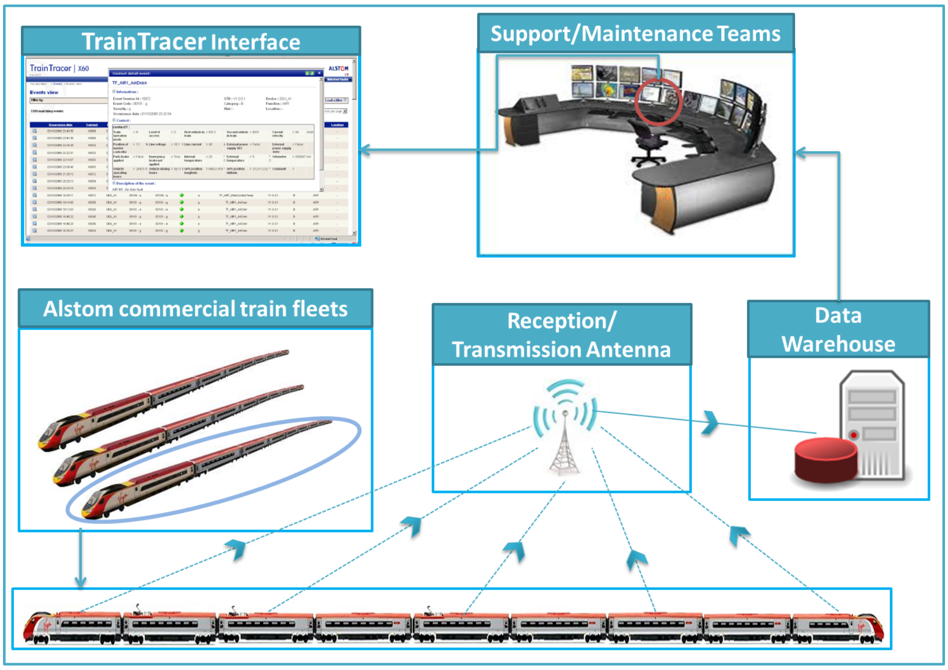


Figure 2.3: Graphical Illustration of Alstom's TrainTracer^{TM}. Commercial trains are equipped with positioning (GPS) and communications systems as well as onboard sensors monitoring the condition of various subsystems on the train and providing a real-time flow of data. This data is transferred wirelessly towards centralized servers where it is stocked and exploited.

2.4.1 TrainTracer Data

The real data on which this thesis work is performed was provided by Alstom transport, a subsidiary of Alstom. It consists of a 6-month extract from the TrainTracer database. This data consists of series of timestamped events covering the period from July 2010 to January 2011. These events were sent by the Trainmaster Command Control (TMCC) of a fleet of pendolino trains that are currently active. Each one of these events is coupled with context variables providing physical, geographical and technical information about the environment at the time of occurrence. These variables can be either boolean, numeric or alphabetical. In total, 9,046,212 events were sent in the 6-month period.

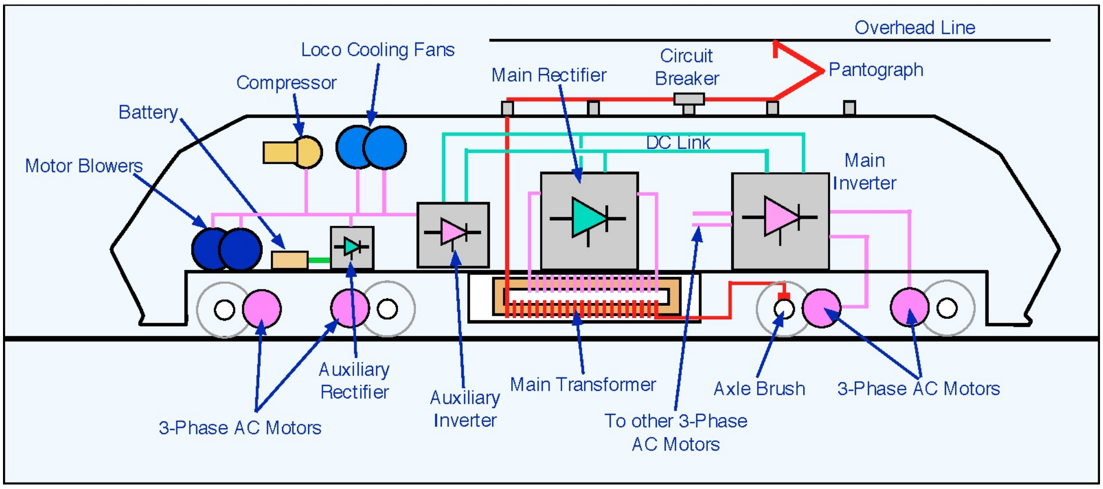


Figure 2.4: Design of a traction-enabled train vehicle (http://railway-technical.com)

- **Subsystems**

  Although all events are sent by the same unit (TMCC) installed on the vehicles, they provide information on many subsystems that vary between safety, electrical, mechanical and services (consider figure 2.4). There are 1112 distinct event types existing in the data extract with varying frequencies and distributions. Each one of these event types is identified by a unique numerical code.

- **Event Criticality Categories**

  Events belonging to the same subsystem may not have the same critical importance. Certain events can indicate normative events (periodic signals to indicate a functional state), or are simply informative (error messages, driver information messages) while others can indicate serious failures, surpass of certain thresholds whose attributes were fixed by operators or even unauthorized driver actions. For this reason, events were divided by technical experts into various intervention categories describing their importance in terms of the critical need for intervention.

The most critical category is that of events indicating critical failures that require an immediate stop/slow down or redirection of the train by the driver towards the nearest depot for corrective maintenance actions. Example: the ”Pantograph Tilt Failure” event. These events require high driver action and thus we refer to their category by “Driver Action High”.

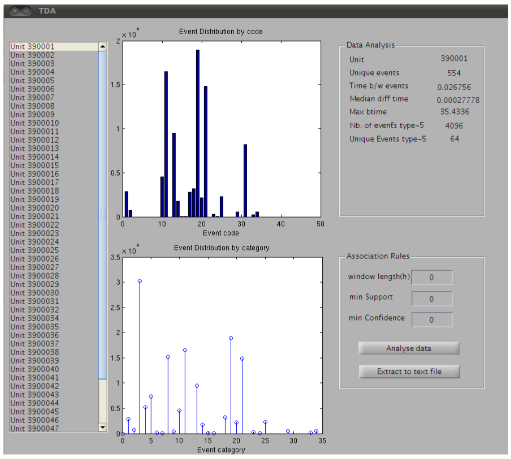
- **Target Events**

  As mentioned before, events are being sent by sensors monitoring subsystems of diverse nature: passenger safety, power, communications, lights, doors, tilt and traction etc. Among all events, those requiring an immediate corrective maintenance action are considered as target events, that is mainly, all “Driver Action High” events. In this work, we are particularly interested in all subsystems related to tilt and traction. The tilt system is a mechanism that counteracts the uncomfortable feeling of the centrifugal force on passengers as the train rounds a curve at high speed, and thus enables a train to increase its speed on regular rail tracks. The traction system is the mechanism responsible for the train’s movement. Railways at first were powered by steam engines. The first electric railway motor did not appear until the mid 19th century, however its use was limited due to the high infrastructure costs. The use of Diesel engines for railway was not conceived until the 20th century, but the evolution of electric motors for railways and the development of electrification in the mid 20th century paved the way back for electric motors, which nowadays, powers practically all commercial locomotives (Faure, 2004; Iwnicki, 2006). Tilt and traction failure events are considered to be among the most critical, as they are highly probable to cause a mandatory stop or slowdown of the train and hence impact the commercial service and induce a chain of costly delays in the train schedule.

  In the data extract under disposal, Tilt and Traction driver action high failure events occur in variable frequencies and consist a tiny portion of 0.5% of all events. Among them, some occur less than 50 times in the whole fleet of trains within the 6-month observation period.

2.4.2 Raw data with challenging constraints

In order to acquire a primary vision of the data and to identify the unique characteristics of target events, a graphical user interface (GUI) was developed using Matlab environment. This interface enabled the visualization of histograms of event frequencies per train unit as well as in the whole data and provided statistics about event counts and inter-event times (Figure 2.5).



Unit 390001
Unit 390002
Unit 390003
Unit 390004
Unit 390005
Unit 390006
Unit 390007
Unit 390008
Unit 390009
Unit 3900010
Unit 3900011
Unit 3900012
Unit 3900013
Unit 3900014
Unit 3900015
Unit 3900016
Unit 3900017
Unit 3900018
Unit 3900019
Unit 3900020
Unit 3900021
Unit 3900022
Unit 3900023
Unit 3900024
Unit 3900025
Unit 3900026
Unit 3900027
Unit 3900028
Unit 3900029
Unit 3900030
Unit 3900031
Unit 3900032
Unit 3900034
Unit 3900035
Unit 3900036
Unit 3900037
Unit 3900038
Unit 3900039
Unit 3900040
Unit 3900041
Unit 3900042
Unit 3900043
Unit 3900044
Unit 3900045
Unit 3900046
Unit 3900047

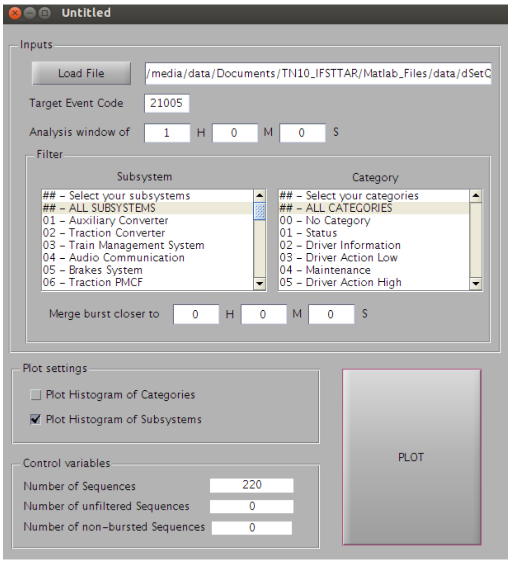
Data Analysis

Unit	390001
Unique events	554
Time b/w events	0.026756
Median diff time	0.00027778
Max btime	35.4336
Nb. of events type-5	4096
Unique Events type-5	64

Association Rules

window length(h) 0
min Support 0
min Confidence 0

Figure 2.5: GUI designed for data visualization, implemented in Matlab. It enabled the visualization, for a selected event and train unit, of various histograms and plots, along with various statistics concerning counts and inter-event times



Inputs

/media/data/Documents/TN10_IFSTTAR/Matlab_Files/data/dSetC

Target Event Code 21005

Analysis window of 1 H 0 M 0 S

Filter

Subsystem
– Select your subsystems
– ALL SUBSYSTEMS
01 – Auxiliary Converter
02 – Traction Converter
03 – Train Management System
04 – Audio Communication
05 – Brakes System
06 – Traction PMCF

Category
– Select your categories
– ALL CATEGORIES
00 – No Category
01 – Status
02 – Driver Information
03 – Driver Action Low
04 – Maintenance
05 – Driver Action High

Merge burst closer to 0 H 0 M 0 S

Plot settings

□ Plot Histogram of Categories
☑ Plot Histogram of Subsystems

Control variables

Number of Sequences 220
Number of unfiltered Sequences 0
Number of non–bursted Sequences 0

Figure 2.6: GUI designed for data visualization, implemented in Matlab. It enabled the request and visualization, for a selected target event T, of various histograms and plots, along with various statistics concerning events and their inter-event times

Another graphical interface was developed by a masters degree intern (Randriamanamihaga, 2012) working on the same data and was also used to visualize the ensemble of sequences preceding the occurrences of a given target event. This interface is shown in Figure 2.6. Figure 2.7 is one of many examples of data visualization we can obtain. In this figure, we can visualize a sequence of type $(S_T, t_T - t, t_T)$ where S_T is the sequence of events preceding target event (T, t_T).

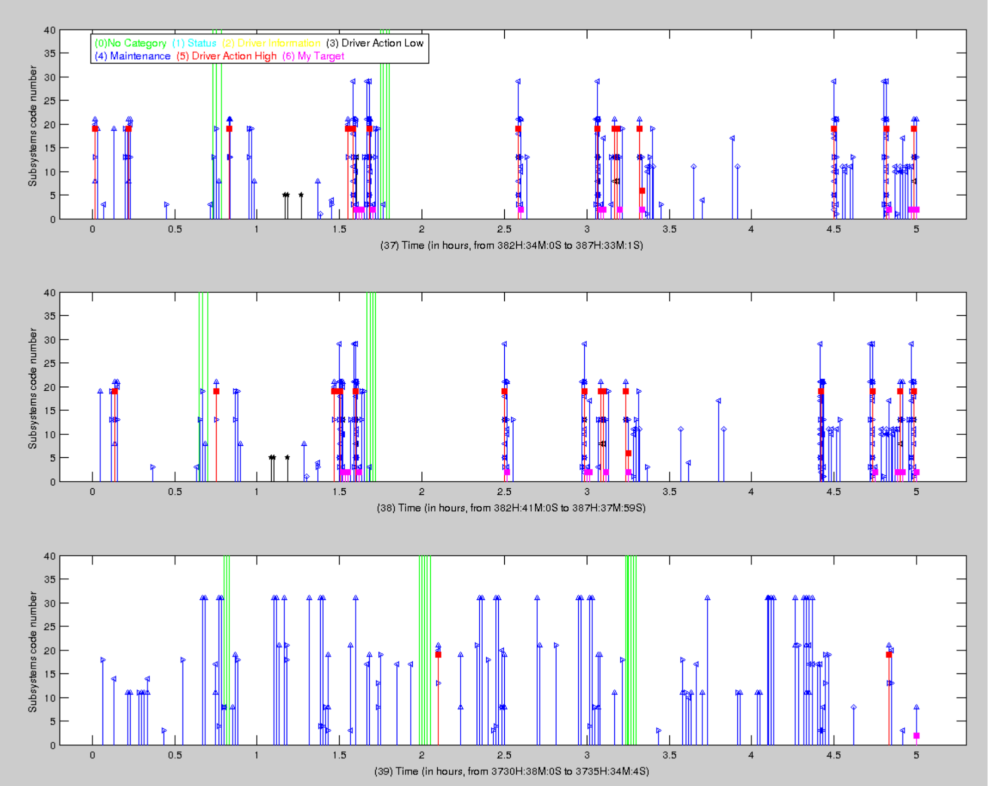


Figure 2.7: Example of a visualized sequence, after request of target event code 2001. The y-axis refers to subsystems, the colors represent different intervention categories and the length of each event designate its count

Both tools developed to visualize data lead to the following interpretation: many obstacles are to be considered and confronted, namely the rarity and redundancy of events.

- **Rarity:**

  The variation in event frequencies is remarkable. Some events are very frequent while others are very rare. Out of the 1112 event types existing in the data, 444 (≈ 40%) have occurred less than 100 times on the fleet of trains in the whole 6-month observation period (see Figure 2.8). These events, although rare, render the data mining process more complex.

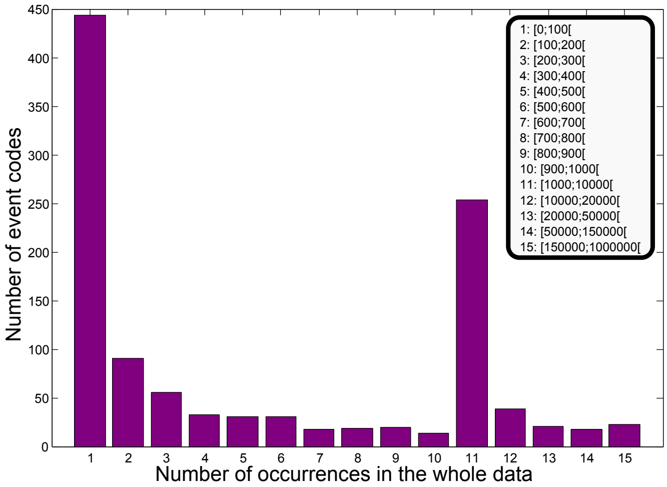


Figure 2.8: Histogram of event frequencies in the TrainTracer data extract under disposal

- **Redundancy:**

  Another major constraint is the heavy redundancy of data. A sequence $w \downarrow [\{A\}]$ of the same event A is called redundant (also called **bursty**), see Figure 2.9, if in a small lapse of time (order of seconds for example), the same event occurs multiple times. More formally, if $w \downarrow [\{A\}] = \langle\ (A, t_1), (A, t_2), \ldots, (A, t_n)\ \rangle$ is a sequence of n A events subject to a **burst**, then

$$\exists\, t = t_{fusion} \text{ such as } \forall\, (i, j) \in \{1, \ldots, n\}^2\ , |\, t_i - t_j\, | \leq t_{fusion} \tag{2.1}$$

  The reasons to why these bursts occur are many. For example, the same event can be detected and sent by sensors on multiple vehicles in nearly the exact time. It is obvious that only one of these events needs to be retained since the others do not contribute with any supplementary useful information. These bursts might occur due to emission error caused by a hardware or software failure, as well as reception error caused by similar factors.

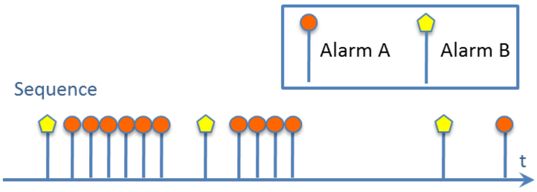


Figure 2.9: Illustration of the concept of bursts on event A

Figure 2.10 illustrates data bursts in a sequence. We can identify two types of bursts. The first type consists of a very dense occurrence of multiple types of events within a short time lapse. Such bursts can occur normally or due to a signalling/reception error. The second type on the other hand consists of a very dense occurrence of a single event type within a short period of time, usually due to a signalling or reception error as well (event sent multiple times, received multiple times). Bursty events can be generally identified by a typical form of the histogram of inter-event times depicted in Figure 2.11. This latter has a peak of occurrences (usually from 0 to 15 seconds) that we can relate to bursts. For example, 70% of all the occurrences of the code 1308 [1] (an event belonging to category 4 and appears in the data 150000 times) are separated by less than one second!

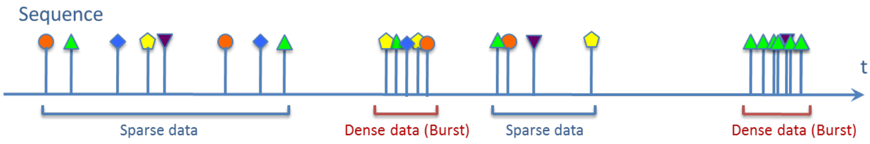


Figure 2.10: Illustration of the two types of bursts

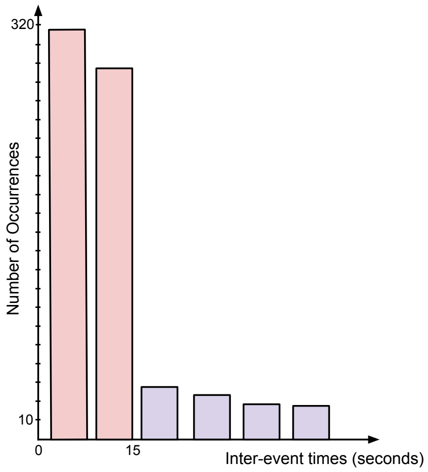


Figure 2.11: Histogram of inter-event times of a bursty event (Randriamanamihaga, 2012)

[1] DC Load Shed

2.4.3 Cleaning bursts

Several pre-treatment measures have been implemented to increase the efficiency of data mining algorithms to be applied. For instance, 13 normative events that are also very frequent were deleted from the data. Erroneous event records with missing data or outlier timestamps were also neglected in the mining process. The work by (Randriamanamihaga, 2012) during a masters internship on the TrainTracer data has tackled the bursts cleaning problem and applied tools based on finite probabilistic mixture models as well as combining events of the same type occurring very closely in time (≤ 6 seconds, keeping the first occurrence only) to decrease the number of bursts. This cleaning process has decreased the size of data to 6 million events (instead of 9.1), limited the number of distinct event codes to 493 (instead of 1112), and the number of available target events to 13 (instead of 46). Although a significant proportion of data was lost, the quality of the data to be mined was enhanced, which leads to a better assessment of applied algorithms and obtained results. For this reason, the resulting "cleaned" data was used in this thesis work.

2.5 Positioning our work

In the railway domain, instrumented probe vehicles that are equipped with dedicated sensors are used for the inspection and monitoring of train vehicle subsystems. Maintenance procedures have been optimized since then so that to rely on the operational state of the system (Condition-based maintenance) instead of being schedule-based. Very recently, commercial trains are being equipped with sensors as well in order to perform various measures. The advantage of this system is that data can be collected more frequently and anytime. However, the high number of commercial trains to be equipped demands a trade-off between the equipment cost and their performance in order to install sensors on all train components. The quality of these sensors reflects directly on the frequency of data bursts and signal noise, both rendering data more challenging to analyze. These sensors provide real-time flow of data consisting of georeferenced events, along with their spatial and temporal coordinates. Once ordered with respect to time, these events can be considered as long temporal sequences that can be mined for possible relationships.

This has created a necessity for sequential data mining techniques in order to derive meaningful associations (association and episode rules) or classification models from these data. Once discovered, these rules and models can then be used to perform an on-line analysis of the incoming event stream in order to predict the occurrence of target events, i.e, severe failures that require immediate corrective maintenance actions.

The work in this thesis tackles the above mentioned data mining task. We aim to investigate and develop various methodologies to discover association rules and classification models which can help predict rare failures in sequences. The investigated techniques constitute two major axes: **Association analysis**, which is temporal, and aims to discover association rules of the form $A \longrightarrow B$ where B is a failure event using significance testing techniques (T-Patterns, Null models, Double Null models) as well as Weighted association rule mining (WARM)-based algorithm to discover episode rules, and **Classification techniques**, which is not temporal, where the data sequence is transformed using a methodology that we propose into a data matrix of labelled observations and selected attributes, followed by the application of various pattern recognition techniques, namely K-Nearest Neighbours, Naive Bayes, Support Vector Machines and Neural Networks to train a static model that will help predict failures.

We propose to exploit data extracted from Alstom's TrainTracer database in order to establish a predictive maintenance methodology to maximize rolling stock availability by trying to predict failures prior to their occurrence, which can be considered as a technological innovation in the railway domain. Once association rules or classification models are found, both can then be implemented in rule engines analyzing arriving events in real time in order to signal and predict the imminent arrival of failures. In the analysis of these sequences we are interested in rules which help predict tilt and traction "driver action high" failure events, which we consider as our target events.

To formalize the problem, we consider the input data as a sequence of events, where each event is expressed by a unique numerical code and an associated time of occurrence.

Definition 2.5.1. ***(event)*** Given a set E of event types, an event is defined by the pair (R, t) where $R \in E$ is the event type (code) and $t \in \Re^{+}$ its associated time of occurrence, called timestamp.

Definition 2.5.2. ***(event sequence)*** An event sequence S is a triple (S, T_s, T_e), where $S = \{(R_1, t_1), (R_2, t_2), ..., (R_n, t_n)\}$ is an ordered sequence of events such that $R_i \in E \ \ \forall i \in \{1, ..., n\}$ and $T_s \leq t_1 \leq t_n \leq T_e$.

2.5.1 Approach 1: Association Analysis

Definition 2.5.3. ***(Association rule)*** We define an association rule as an implication of the form $A \longrightarrow B$, where the antecedent and consequent are sets of events with $A \cap B = \phi$.

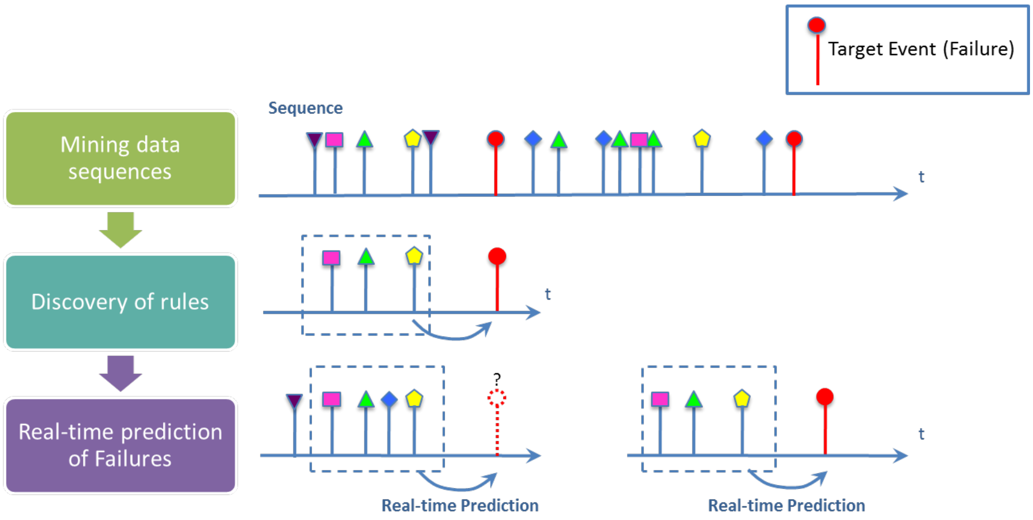


Figure 2.12: A graphical example of mining data sequences for association rules. Discovered rules are then used for real-time prediction of failures

Consider the Figure 2.12, in the analysis of sequences we are interested in discovering associations between normal events and target events. Once such relationships are discovered, they can be integrated in an online rule engine monitoring the incoming event stream, hence allowing the prediction of target events (failures). Once a target event is predicted, the maintenance teams are alerted to initiate predictive maintenance procedures. In order for this prediction to be effective, it is subject to two important constraints:

- First, target events should be predicted within a time delay that should be sufficient enough to allow logistic and maintenance measures to be taken, such as directing a train towards a stand-by maintenance team in a nearby depot, thus avoiding the costly consequences of a train breaking down in-between stations. This time delay is called warning time (See Figure 2.13).

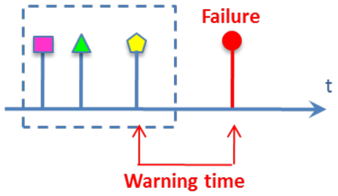


Figure 2.13: An illustration of warning time prior to the occurrence of a target event

- Secondly, prediction accuracy should be high enough due to high intervention costs in the case of false predictions. The whole process is depicted in Figure 2.14 below.

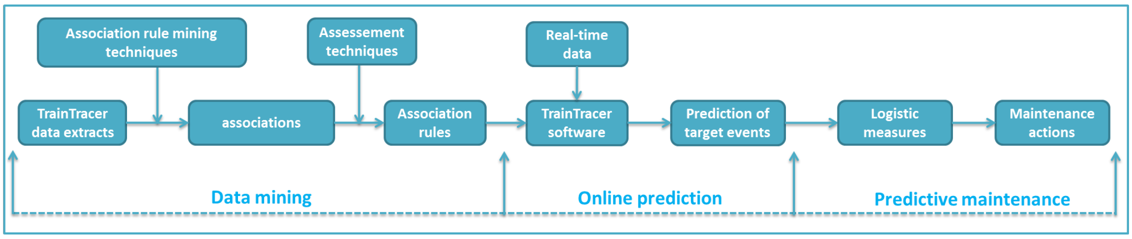


Figure 2.14: Mining association rules for predictive maintenance. Discovering significant associations between events and validating their abidance to the warning time and accuracy constraints leads to association rules. These rules are validated by railway experts and integrated in online monitoring tools to predict target events.

However, due to the rareness of target events, the existing sequence mining techniques cannot be used since they are designed to discover frequent patterns and associations between frequent events and not rare ones. Rare events would be directly pruned out in the process and any correlation between them and other frequent/infrequent events would remain undiscovered. This problem was the main motivation in orienting and structuring the theoretical work of this thesis.

This thesis hence tackles the problem of mining temporal sequences for association rules between infrequent events, an important problem that has not received much academic attention until recently. In data mining literature, the use of the term "pattern" is attributed to the repetitive series of correlated events discovered in a sequence and mostly signifies frequent patterns. However, since in this thesis we are interested in infrequent patterns, we preferred using the term association and episode rules.

2.5.2 Approach 2: Classification

The second approach which we will adopt is pattern recognition and classification methods. Figure 2.15 illustrates the process. In this approach we neglect the temporal aspect of the sequence and transform it into a data matrix of labelled observations and selected attributes. For each occurrence of a target event, events observed within a time window of width w preceding this event are considered as an observation and given a label 1. This window is equivalent to a *bad* window followed by a failure. Random windows are chosen and constitute the *good* windows each with a label 0. A random window is omitted if it contained a failure event. Since the training data are labelled, the problem in hand is a supervised learning classification problem. All observations are used to train and learn a classification model using multiple pattern recognition methods, namely K-Nearest Neighbours, Naive Bayes, Support Vector Machines and Neural Networks.

Once the classification model is learned, it is tested using a cross validation technique. Events arriving in real time are considered as an input attribute set (x) to the classification model which outputs a class label 1 if a nearby occurrence of a target event is predicted and a label 0 if not.

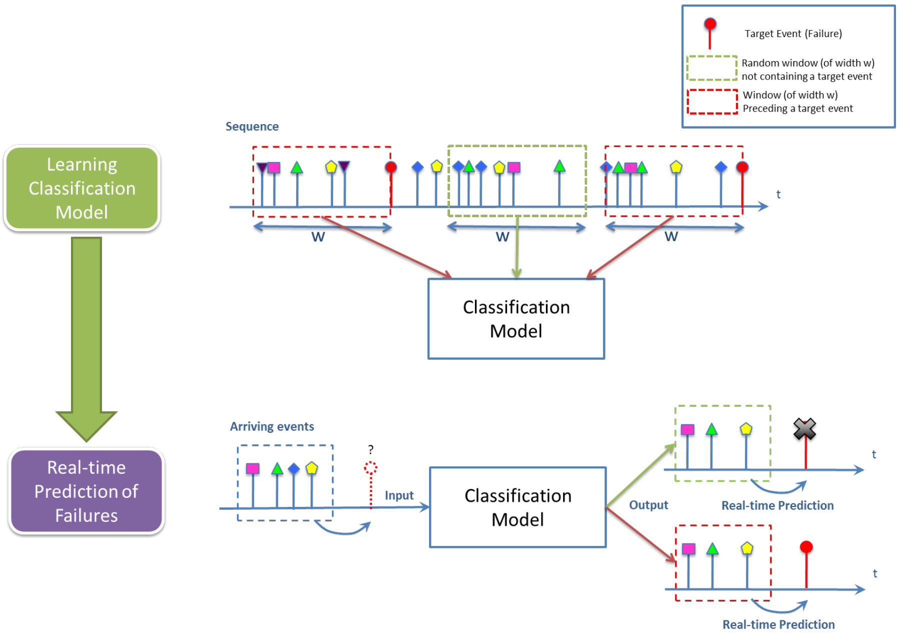

Target Event (Failure)
Random window (of width w) not containing a target event
Window (of width w) Preceding a target event
Sequence
Learning Classification Model
t
W
W
W
Classification Model
Real-time Prediction of Failures
Arriving events
?
Input
Classification Model
Output
Real-time Prediction
t
t
Real-time Prediction

Figure 2.15: A graphical illustration of how a classification model can be trained from a data sequence and then used to predict the near-by occurrence of failures

At the end, the performance of both approaches: **Association analysis** and **Classification methods** are confronted and compared.

Chapter 3

Detecting pairwise co-occurrences using hypothesis testing-based approaches: Null models and T-Patterns algorithm

Contents

3.1 Introduction

Mining statistically significant rules directly from sequences is an important but neglected problem. In most traditional frequency-confidence framework-based association rule approaches, the actual statistical interestingness of the rule is evaluated using interestingness measures after it has been discovered. Quite likely, the reason is practical: statistical significance is not a monotonic property and therefore it cannot be used for pruning the search space in the same manner as the frequency (Hamalainen and Nykanen, 2008). This leads to a lot of spurious results, without forgetting that frequency-based approaches are not capable to mine rules between infrequent items without the use of a low support threshold which in its turn causes a huge computational complexity.

In chapter 2, we have discussed the complexity of the TrainTracer data extracts on which the case study of this thesis is focused. One of the main raised issues was the rareness of the target events, which immediately rules out most of the existing association rule and pattern mining algorithms such as Apriori (Agrawal et al., 1996) and its derivatives and FP-tree (Han et al., 2004) and its derivatives due to their dependence on frequency. Another raised issue was the strong presence of redundancy and noise in the data, an additional constraint which renders the mining process more vulnerable to false discoveries. In this chapter, we focus our work on the discovery of pairwise co-occurrences between events A and B. Once discovered, these co-occurrences would then be assessed to derive length-2 association rules $A \longrightarrow B$, where B is a target event.

In order to discover these co-occurrences, two different methodologies have been adapted to the problem: Null models and T-Patterns algorithm. The first, null models, consists of randomization techniques followed by the calculation of various co-occurrence scores. The second, T-Patterns, exploits the temporal dimension by investigating the statistical dependence between inter-arrival times of couples of events in order to highlight possible relationships and build trees of hierarchical temporal dependencies. In addition to that, a bipolar significance testing approach, called double null models (DNM) is proposed. The motivation behind this approach is to render null models more resistant to spurious results and to integrate the directionality aspect into the mining process instead of assessing it in post-mining steps. All approaches were applied and confronted on both synthetic and real data.

Once significant co-occurrent event couples are found, they will be assessed by means of objective interestingness measures: Recall and Precision which will assess the directionality aspect and the accuracy, as well as subjective interestingness measures which mainly evaluate the inter-event times between events A and B. Significant couples abiding the constraints discussed in chapter 2 (accuracy and inter-event time) are considered as statistically significant association rules. This methodology is illustrated in Figure 3.1 below.

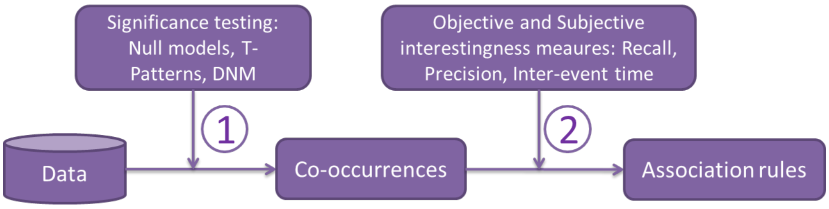

Significance testing: Null models, T-Patterns, DNM
Objective and Subjective interestingness meaures: Recall, Precision, Inter-event time
1
2
Data
Co-occurrences
Association rules

Figure 3.1: Illustration of the methodology adapted in this chapter to discover association rules. Significance testing approaches: Null Models, T-Patterns and Double Null Models (DNM) are used to discover significant co-occurrences. Once found, these co-occurrences are assessed using objective and subjective interestingness measures in order to derive association rules.

This chapter is organized as follows: In Section 3.2, we formally define association analysis and the association rule mining problem. We also give the basic notations and define the notorious ARM problem and its two pillar approaches. In Section 3.3 we describe single null models and propose the double null model approach in 3.3.5. We then explain the T-Patterns algorithm in 3.4. The interesting measures used to evaluate results and derive rules from discovered co-occurrences are discussed in Section 3.5. In Section 3.6, experiments are then performed on synthetically generated data in order to assess the performance of null models and to derive the best combination of double null models. The most performant single null models, double null models as well as the T-Patterns algorithm are then applied on real TrainTracer data in Section 3.7. Finally, we resume and conclude in Section 3.8.

3.2 Association analysis

3.2.1 Introduction

Association analysis is an important omnipresent data mining problem that aims to discover particular relationships between items in a transaction database or events in a data sequence. The aim is to discover association rules, which are also referred to as *Episode rules* in cases when temporal relationships between events in sequences are described. In this thesis, we use the term association rule for both temporal and non-temporal rules in both transaction and sequential data.

The original motivation behind the initial association rule mining algorithms was the need to analyze supermarket basket transaction data (Agrawal and Srikant, 1994) in order to examine customer behavior in terms of purchased products. Association rules were used to describe how often items are purchased together. For instance, the rule Fries $\longrightarrow$ Ketchup (75%) states that three out of four customers who have bought fries have also purchased Ketchup. Such rules can be useful for decisions concerning product pricing, promotions, or store layouts. However, recent years have witnessed a vast and extensive development of association analysis approaches. These approaches were applied in a wide range of domains such as environmental monitoring (Tan et al., 2001), bioinformatics (Haiminen et al., 2008), recognition of human behavior (Ermes et al., 2008; Honda et al., 2007) and telecommunication alarm log management and interaction (Mannila et al., 1997) etc. The developed techniques are very diversified to comply with different types of problems in transaction databases or sequences of events.

In general, the association rule mining process consists of 2 main steps:

1. Mining associations (itemsets, patterns, episodes, co-occurrences)

2. Evaluating these associations using interestingness measures to generate and validate rules

3.2.2 Association Rule Discovery: Basic notations, Initial problem

In this section, we define some basic notions and concepts of association rule mining. We briefly describe how the field evolved as well as its pillar algorithms. As mentioned earlier, the initial motivation behind the development of the first association rule mining algorithms was the analysis of market basket data, which consisted of transactions. Hence the algorithm was initially conceived for transaction data.

Let I be a set of items, also called *itemset.* A set $X = \{i_1, i_2, .., i_k\} \subseteq I$ is called a *k-itemset* if it contains k distinct items. Let $\mathcal{D}$, the transaction database, be a set of transactions over I, where each transaction is a couple $T = (tid, X)$ where tid is the transaction identifier and X an itemset. Given the following definitions.

Definition 3.2.1. (Association Rule): For a given transaction database $\mathcal{D}$, an association rule is an expression of the form $X \longrightarrow Y$, where X and $Y \subset I$ are itemsets and $X \cap Y = \{\}$. Such a rule indicates that if a transaction contains all items in X then it is highly probable that it also contains all items in Y. X is called the body or antecedent and Y is called the head or consequent of the rule. An association rule is accompanied by mathematical statistics that describe the relationship.

Definition 3.2.2. (Support): A transaction T is said to *support* an item i_k if it contains that item. T is said to support a subset of items $X \subseteq I$ if it supports each item i in X. The support of an itemset X in $\mathcal{D}$ can be defined in various manners. For instance, the percentage support refers to the percentage of transactions supporting X. The fractional support (also called frequency) is the proportion of transactions containing X. The absolute support, also called support count, refers to the absolute number of transactions supporting X in $\mathcal{D}$. In this thesis, unless specified otherwise, we assume the support to be percentage support, which is formalized by the following equation:

$$support(X, \mathcal{D}) := \frac{|\{I | I \in \mathcal{D} \wedge X \subseteq I\}|}{|\mathcal{D}|} \times 100 \tag{3.1}$$

An itemset is said to be *frequent* if its support is no less than a given minimal support threshold *minsup*. The support of an association rule $X \longrightarrow Y$ in $\mathcal{D}$, is the support of $X \cup Y$ in $\mathcal{D}$.

Definition 3.2.3. (Confidence): Formally, the confidence of an association rule $X \longrightarrow Y$ in D is the conditional probability of having Y contained in a transaction, given that X is contained in that transaction as well:

$$confidence(X \longrightarrow Y, \mathcal{D}) := P(Y|X) = \frac{support(X \cup Y, \mathcal{D})}{support(X, \mathcal{D})} \tag{3.2}$$

The rule is called confident if $P(Y|X)$ exceeds a given minimal confidence threshold $minconf$, with $0 \leq minconf \leq 1$.

The original association rule mining approach introduced in (Agrawal and Srikant, 1994) as well as most of the existing association rule mining techniques are based on a frequency-based support-confidence framework, where the problem consists of mining all rules that have support and confidence greater than or equal to the user-specified minimum support and confidence thresholds. Step 1 of the association rule mining process thus consists of frequent itemset mining where the goal is to discover itemsets occurring repetitively. Let $\mathcal{D}$ be a transaction database over a set of items I, and *minsup* a minimal support threshold. The collection of frequent itemsets in $\mathcal{D}$ with respect to *minsup* is denoted by

$$F(\mathcal{D}, minsup) := \{X \subseteq I | support(X, \mathcal{D}) \geq minsup\} \tag{3.3}$$

Once discovered, these frequent itemsets are then evaluated mainly by the confidence measure or other interestingness measures defining to which extent the candidate rule holds. Supposing confidence is the interestingness measure to be used to evaluate discovered frequent itemsets, The ARM problem consists of finding $\mathcal{R}(\mathcal{D}$,*minsup*,$minconf)$, where

$$\begin{aligned} \mathcal{R}(\mathcal{D}, minsup, minconf) := \{X \longrightarrow Y | X, Y \subseteq I, X \cap Y = \{\}, X \cup Y \in F(\mathcal{D}, minsup), \\ confidence(X \longrightarrow Y, D) \geq minconf\} \end{aligned} \tag{3.4}$$

Most research in the area of association rule discovery has focused on the subproblem of efficient frequent item set discovery (Han et al., 2004; Park et al., 1995; Pei et al., 2001a; Savasere et al., 1995). The task of discovering all frequent itemsets is quite challenging since its computational requirements are generally more important than those of rule generation. The search space is exponential in the number of items occurring in the database. Consider Figure 3.2 below, a lattice structure can be used to illustrate the list of all possible itemsets for $I = \{a, b, c, d, e\}$. In general, a dataset containing k items can potentially generate up to $2^k - 1$ frequent itemsets, excluding the null set. In many applications, k can be very large which might implicate an explosion in the search space of itemsets to be explored. For this reason, not all itemsets should be explored. The support threshold limits the output to a hopefully reasonable subspace. Also, the number of transactions contributes to the complexity of the problem.

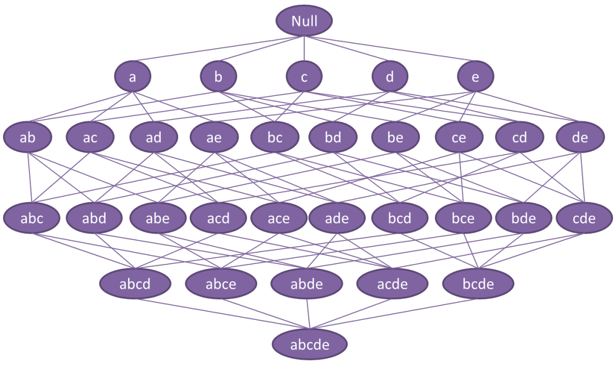


Figure 3.2: An itemset lattice

Definition 3.2.4. (Candidate itemset) Given a transaction database $\mathcal{D}$, a minimal support threshold *minsup*, and an algorithm that computes $F(\mathcal{D}, minsup)$, an itemset I is called a candidate if its support is being evaluated by an ARM process.

Since the release of the first association rule mining algorithms, many approaches were developed to reduce the computational complexity of frequent itemset generation. These approaches can be grouped under two major axes:

1. Breadth-first algorithms: which are mostly based on the downward closure property of the Apriori algorithm, described further in this section. These approaches target the reduction of the number of candidate itemsets to be evaluated.
2. Depth-first algorithms: mostly based on the FP-tree algorithm and its derivatives, these approaches attempt to reduce the number of comparisons between itemsets and transactions (database scans) by elaborating more advanced data structures to compress the dataset while conserving the information.

The two main breadth-first and depth-first algorithms are Apriori and FP-tree, both which we briefly describe below.

Apriori algorithm: A breadth-first approach

The first support-confidence based algorithm was the AIS algorithm, introduced by (Agrawal et al., 1993). It generated all frequent itemsets and confident association rules and was proposed along with the introduction of the association rule mining problem. Shortly after that, the algorithm was improved by the same team and renamed Apriori. The Apriori algorithm was the first to tackle the computational time problem of association rule mining by reducing the number of item sets that are considered. It was the first approach to use support-based pruning as well as the downward closure property (Agrawal and Srikant, 1994; Srikant and Agrawal, 1995). The same technique was independently proposed by (Mannila et al., 1994). Both works were joined afterwards in (Agrawal et al., 1996). The algorithm is guided by the following principle, called the *downward closure property*, also called the *anti-monotonicity property*.

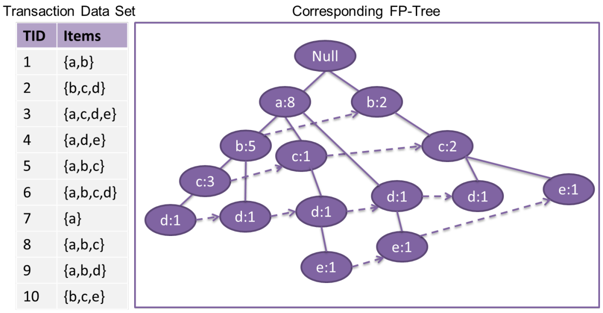
Definition 3.2.5. (Downward Closure Property) Let I be a set of items, and $J = 2^I$ be the power set of I. A measure f is downward closed if $\forall X, Y \in J : (X \subseteq Y) \longrightarrow f(Y) \leq f(X)$, which means that if an itemset Y is frequent, then all of its subsets X must also be frequent since a superset cannot have a frequency exceeding that of its subsets.

Let C_k denote the set of candidate k-itemsets, F_k denote the set of frequent k-itemsets and $X[i]$ represent the ith item in X. The algorithm initially makes a single path over the data set to determine the support of each item and obtain frequent length-1 itemsets F_1, hence C_1 consists of all items in I. The algorithm performs a breadth-first search through the search space of all itemsets by iteratively generating candidate itemsets C_{k+1} of size $k + 1$, starting with $k = 1$. As mentioned before, an itemset is *candidate* if all of its subsets are known to be frequent. To count the supports of all candidate k-itemsets, the database is scanned one transaction at a time, and the supports of all candidate itemsets that are included in that transaction are incremented. All itemsets that turn out to be frequent are inserted into F_k and used to generate C_{k+1}. The algorithm halts when there are no new frequent itemsets to be generated, that is, when $F_k = \{\phi\}$.

Once the frequent itemsets are obtained, association rules are generated via a confidence-based process.

The FP-Growth algorithm: A depth-first approach

The first algorithm to generate frequent itemsets in a depth-first manner was the Eclat algorithm proposed in (Zaki et al., 1997)(Zaki, 2000). Later on, several other depth-first algorithms have been proposed (Agarwal et al., 2000a,b; Han et al., 2004) of which the FP-growth algorithm by (Han et al., 1999, 2004) is the most well known and considered to be the most influential. The algorithm does not implement the generate-and-test paradigm of the Apriori algorithm. It consists of two steps. First, it represents the data differently using a compact structure called an FP-tree and a minimum support threshold *minsup*. The construction of the tree requires only two passes over the whole database. The algorithm then uses this structure instead of the transaction database to extract frequent itemsets directly. Figure 3.3 gives an example of a transaction dataset consisting of ten transactions and five items and its corresponding FP-tree.



Transaction Data Set

TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}

Figure 3.3: Construction of an FP-tree from transaction data consisting of ten transactions and five items. Each node of the tree represents an item along with a counter showing the number of transactions mapped onto the given path.

As mentioned earlier, both approaches Apriori and FP-tree are frequency-based and are thus not suitable for the discovery of association rules leading to infrequent events. For this reason, we have decided to adopt a different approach based on hypothesis testing methods: Null-models and T-Patterns. In the next section, we tackle the null models approach.

3.3 Null models

3.3.1 Formalism

A null model is a sequence-generating model which generates randomizations of data sequences while conserving their general statistical characteristics. Certain elements of the data are held constant while others are allowed to vary stochastically. These models evaluate relationships between couples of events by means of a statistical hypothesis test, where the null hypothesis refers to the significance of a particular relationship in the original data sequence. To solve this test, the initial data sequence is randomized using a null model and the co-occurrence scores of each randomization are calculated. In order to understand how significant are the scores of the scrutinized couple in the initial sequence, the empirical p-value which is equal to the fraction of randomizations with a higher co-occurrence scores than the initial data is calculated and compared to a pre-defined threshold. If it is inferior to the threshold, the event couple under scrutiny is considered to be statistically significant, and highly probable to have not occurred by chance.

Null models were initially developed in the 1970s for applications in ecology. Their dominant application domain remains to be ecology, although it has been also applied in various fields such as genetics (Hannenhalli and Levy, 2002; Klein and Vingron, 2007), physiology (Bellwood et al., 2002; Lavorel and Garnier, 2002), sports (Ermes et al., 2008), etc.

(Gotelli and Graves, 1996) was the first to introduce and formalize the following definition:

A null model is a pattern-generating model that is based on randomization of ecological data. Certain elements of the data are held constant and others are allowed to vary stochastically. The randomization is designed to produce a pattern that would be expected in the absence of a particular ecological mechanism..

Essentially, there were two views of null models: (Connor and Simberloff, 1983) considered them as statistical descriptions of randomized data while (Colwell and Winkler, 1984) and (Gotelli and Graves, 1996) considered them as simulations of random assembly processes. The latter definition is the one that will later dominate the concerned scientific community. Numerous techniques for randomizing data were proposed in the last years, varying between simple models (Hannenhalli and Levy, 2002), (Levy et al., 2001) and more complex ones based on SWAP randomization (Gotelli and Entsminger, 2001; Lehsten and P., 2006) or Markov Chains Monte Carlo-based methods (Hanhijrvi et al., 2009; Ojala et al., 2009).

In this section, we define and evaluate three single null models with 2 different co-occurrence scores. We also introduce double null models (DNM), a bi-polar significance testing approach based on single null models.

3.3.2 Co-occurrence scores

In order to assess the relationship between a couple of events, a co-occurrence score is needed. There are several possible scores that can quantify the degree of co-occurrence of an event couple. Given a set E of event types and $S = \{(R_1, t_1), (R_2, t_2), ..., (R_n, t_n)\}$ is a temporal sequence of length l time units, $R_i \in E$ and $t_i \in \Re^+$. Consider the event couple under scrutiny (A, B) where $A \in E$ and $B \in E$, let $N(A)$ be the number of times an event type A occurs in the sequence S and denote $f(A) = N(A)/n$. Consider w as the maximum co-occurrence distance (or scanning window width).

- The directed co-occurrence count score $D(A, B, S)$ is the number of events of type A that are followed by at least one event of type B within a maximum co-occurrence distance w. The values of $D(A, B, S) \in \{0, ..., N(A)\}$.

- The precedence count score $P(A, B, S)$ for event types A and B is the number of events of type B that are preceded by at least one event of type A within a maximum co-occurrence distance w. $P(A, B, S) \in \{0, ..., N(B)\}$.

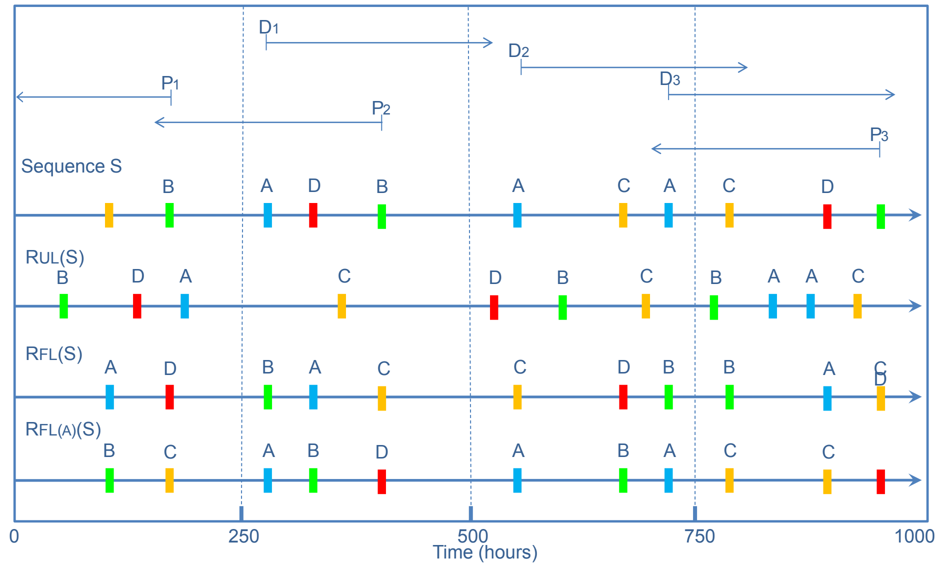


Figure 3.4: Graphical illustration of the UL, FL and $FL(A)$ single null models with the P and D scores for a given event couple (A, B) in an event sequence S of length $l = 1000h$ with 4 event types A, B, C and D

3.3.3 Randomizing data: Null models

In order to evaluate the significance of a co-occurrence score, a null model is needed. We describe three different single null models: the Uniform Locations (UL) model, the Fixed Locations (FL) model and the Fixed Locations Fixed Event Type ($FL(R)$) model. We also propose a double model approach that will be explained in 3.3.5. The randomization technique defined by each model to generate randomized versions of a given event sequence S is explained below:

1. The Uniform Locations UL null model, as its name suggests, consists of generating sequences resulting from the randomization of both the timestamps and the event codes in the sequence. A similar model is applied in (Levy et al., 2001).

2. The randomized sequence $R_{FL}(S)$ is obtained by the Fixed Locations FL null model by keeping the event timestamps fixed, and assigning event types at random on these locations according to their frequencies in the original sequence. This model was applied in (Hannenhalli and Levy, 2002; Klein and Vingron, 2007).

3. The randomized sequence $R_{FL(R)}(S)$ for a sequence S and an event type R is defined similarly to $R_{FL}(S)$, with the exception that the occurrences of events of type R are kept unchanged. This model was introduced in (Haiminen et al., 2008).

3.3.4 Calculating p-values

For a given sequence S and a null model $M \in \{UL, FL, FL(R)\}$, the empirical p-value for an event couple (A, B) is the fraction of randomizations in which the D (or P) score in the randomized sequences $R_M(S)$ exceeds the D (or P) score of the original sequence S:

$$p_D(A, B, M, S) = \frac{\#(D(A, B, S) \leq D(A, B, R_M(S)))}{\#(D(A, B, R_M(S)))} \tag{3.5}$$

3.3.5 Proposed Methodology: Double Null Models

As formally defined and explained in (Haiminen et al., 2008), the UL, FL and FLR null models with the D and P scores evaluate the variation of the predecessor event A with respect to a successor event B and not vice versa. However, since the problematic of this work is to discover rules of the form $A \longrightarrow B$ where B is a target event, the evaluation of the A event with respect to B is also important due to the directionality aspect imposing the presence of the target event B after the A event since events succeeding a failure do not help in predicting it.

The proposed methodology consists of a bipolar approach for discovering significant couples in a way that best assesses recall and precision and renders the mining process more resistant to spuriousness, hence decreasing the number of discovered couples. For a given couple (A, B) under scrutiny, this approach first evaluates the variation of successor event B with respect to the predecessor event A by means of a UL, FL or $FL(A)$ null model approach with the D score **(step 1)**. Then it evaluates the variation of the predecessor event A with respect to successor event B using a UL, FL or $FL(B)$ null model approach with the P score **(step 2)**. Couple discovered by both models are considered to be statistically significant (See Algorithm 1). This renders results more robust against spuriousness resulting from the randomness factor as well as data bursts, as it will be shown in Section 3.6. The main challenge is to find the most optimal combination of models that can most probably lead to the best results. This will also be tackled in 3.6. In the next section, a different approach that is also based on hypothesis testing, the T-Patterns algorithm, is described.

Algorithm 1 Pseudo code of the Double Null Model (DNM) algorithm

Inputs: Data, w: maximum co-occurrence distance, A-list: list of all non-target events, B-list: list of target events, number of randomizations, p-value threshold, NM1: Null model 1, NM2: Null model 2

```
for every possible couple of events $(A_i, B_j)$ do
  Compute $D_0$ and $P_0$ scores of the initial data sequence
  for n=1:number of randomizations do
    Generate $NM1$ randomized data sequence
    Compute $D_n$ for the couple in $NM1$
    Generate $NM2$ randomized data sequence
    Compute $P_n$ for the couple in $NM2$
  end for
  Compute p-value $p_D$ :  $p_D(A,B,NM1,S) = \frac{\#(D(A,B,S) \leq D(A,B,R_{NM1}(S)))}{\#(D(A,B,R_{NM1}(S)))}$
  Compute p-value $p_P$:  $p_P(A,B,NM2,S) = \frac{\#(P(A,B,S) \leq P(A,B,R_{NM2}(S)))}{\#(P(A,B,R_{NM2}(S)))}$
  if $p_D$, $p_P$ $\leq$ p-value threshold then
    Couple $(A_i, B_j)$ is statistically significant
  end if
end for
```

Output: List of all the discovered couples

3.4 T-Patterns algorithm

The T-patterns algorithm (TP) is based on the concept that events can be considered as temporal processes upon modeling their timestamps in the sequence. The aim is then to scrutinize the statistical independency between the timestamps of couples of events by means of a statistical hypothesis test. Two temporal point processes A and B are considered to be independent if the arrival of an A-event (event type A) does not lead to an increase in the probability of occurrence of a B-event (event type B). The hypothesis test is of the form:

- H_0: A and B are independent processes
- H_1: A and B are dependent

To solve this test, (Magnusson, 2000) assumed that the two processes A and B are independent random poisson processes distributed over the observation period with a constant intensity that is equal to the average number of occurrences of each of these events per unit time interval. Now that the expected number of B-events in an interval of time is known, the algorithm asserts that after an occurence of an A-event at a time instant t, there is an interval $[t + d_1, t + d_2]$, where $(d_2 \geq d_1 \geq 0)$, that tends to contain more occurrence of B than would be normally expected by chance. This interval is called critical interval (CI) and presented for simplicity as $[d_1, d_2]$. To evaluate an (A, B) couple within a CI, the standard p-value is computed, which is the probability, under the null hypothesis, of having more B-events in the CI than what was observed.

If the calculated p-value is inferior to a predefined significance threshold (ex. 1%), the null hypothesis is rejected.

This algorithm was enhanced by (Tavenard et al., 2008) and (Salah et al., 2010) who asserted that if A and B temporal processes were to be independent, then whenever an A-event occurs between two successive B-events, it will be uniformly distributed in that interval (Salah et al., 2010). The non-uniformity of A within the B-intervals increases the odds of the dependency of the two temporal processes and thereby makes it worthy to start looking for critical intervals.

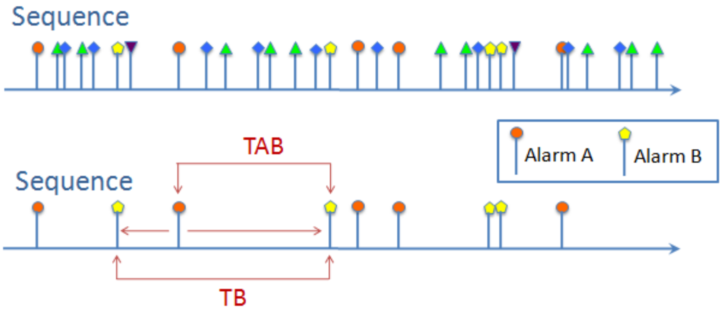


Figure 3.5: Graphical illustration of T_{AB} and $\widetilde{T}_B$

Consider Figure 3.5 above. Denote by $t_A = (t_{A_1}, t_{A_2}, ..., t_{A_n}, ...)$ and $t_B = (t_{B_1}, t_{B_2}, ..., t_{B_n}, ...)$ the (ordered) sequence of times at which an A-event and a B-event occur respectively. $T_A(n) = t_{A_n} - t_{A_{n-1}}$ represents inter A-event time-intervals and $T_B(n) = t_{B_n} - t_{B_{n-1}}$ inter B-event time-intervals. The combination of an A-event and the first subsequent B-event is referred to as AB-event. The time-interval separating these two events is denoted by T_{AB}

$$T_{AB}(k) = t_{B_{k^*}} - t_{A_k}, \tag{3.6}$$

where $k^* = \arg\min\{j \mid t_{B_j} > t_{A_k}\}$. Considering $\widetilde{T}_B$ as the set of T_B intervals in which at least one A-event occurs, T_{AB} should then be uniformly distributed between 0 and $\widetilde{T}_B$:

$$T_{AB} \sim U(0, \widetilde{T}_B) \tag{3.7}$$

To obtain a standard uniform distribution, the ratio vector U of the time between each event A_k and the first succeeding B event to the B-interval length containing A_k is calculated:

$$U(k) = \frac{T_{AB}(k)}{\widetilde{T}_B(k)} = \frac{t_{B_{k^*}} - t_{A_k}}{t_{B_{k^*}} - t_{B_{k^*-1}}} \tag{3.8}$$

To validate the null hypothesis (independence), U should be uniformly distributed between 0 and 1. The test can be solved using a standard Kolmogorov-Smirnov (KS) test (Tavenard et al., 2008).

Algorithm 2 Pseudo code of the T-patterns algorithm

Inputs: $A - List$: List of all non-target events occurring in data, $B - List$: List of target events occurring in data, Significance level α of the KS test= 1%

```
for every possible couple of events $(A_i, B_j)$ do
  Compute vector $t_A$ the ordered sequence of times at which the $A$-event occurred
  Search for the first $B$-event succeeding every $A$-event and Calculate $T_{AB}$ using
  equation (3.6)
  Compute $\widetilde{T_B}$ which is the time distance between the two B-events within which the
  $A$-event occurred
  Calculate the ratio vector $U$ using equation (3.8)
  if $U$ is not uniformly distributed using a Kolmogorov Smirnov statistical test
  with significance level $\alpha$ then
    Couple $(A_i, B_j)$ is statistically significant
  end if
end for
```

Outputs: List of all discovered (A, B) couples

3.5 Deriving rules from discovered co-occurrences

Finding co-occurrences is just the first step of the association rule mining process. The second step is the analysis of these relationships in order to deduce interesting association rules. Any large sequence with a big number of event types can lead to the discovery of a very large number of relationships and co-occurrences of which many are uninteresting, trivial or redundant. Interestingness measures are used to prune those who are uninteresting so as to narrow the search space and focus the analysis.

3.5.1 Interestingness measures in data mining

As mentioned before, discovered associations (co-occurrences, itemsets, episodes and patterns) need to be analyzed in order to deduce those which are solid enough to be considered as rules. Many of these rules can be uninteresting, trivial or redundant.

Example 3.5.1. An example of a Trivial rule is: $Pregnant \longrightarrow Female$ with confidence (accuracy) =100%!

The challenge hence is to select rules that are interesting and significant. In the original formulation of association rules as well as in most Apriori-based methods, support and confidence are the main measures used. However, confidence by itself is not sufficient, since for example if all transactions include item A, then any rule $A \longrightarrow B$ will have a confidence 100%. In this case, other interestingness measures are necessary to filter rules.

The past few years have witnessed the introduction of many interestingness measures in literature. Some measures are dedicated to specific applications and not for others. Using interestingness measures facilitates a general and practical approach to automatically identifying interesting patterns. Two recent studies have compared the ranking of rules by human experts to the ranking of rules by various interestingness measures, and suggested choosing the measure that produces the ranking which most resembles the ranking of experts (Ohsaki et al., 2004; Tan and Kumar, 2002). These studies were based on specific datasets and experts, and their results cannot be taken as general conclusions (Geng and Hamilton, 2006).

During the data mining process, interestingness measures can be used in three ways, which we call the roles of interestingness measures and are illustrated in Figure 3.6. First, measures can be used to prune uninteresting patterns during the mining process so as to narrow the search space and thus improve mining efficiency. For example, a threshold for support can be used to filter out patterns with low support during the mining process (Agrawal and Srikant, 1994). Similarly, for some utility-based measures, a utility threshold can be defined and used for pruning patterns with low utility values (Yao et al., 2004). Measures can also be used to rank patterns according to the order of their interestingness scores or during the postprocessing step to select interesting patterns. For example, we can use the chi-square test to select all rules that have significant correlations after the data mining process (Bay and Pazzani, 1999).

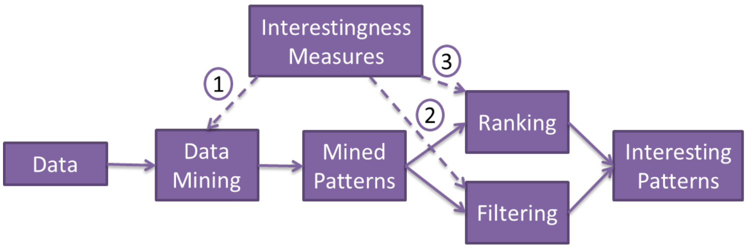


Figure 3.6: Roles of interestingness measures in the data mining process

In general, interestingness measures can be divided into two main categories: Objective and Subjective.

3.5.2 Objective interestingness measures

Objective interestingness measures are generally probability-based such as support, confidence, lift, Gini, Piatetsky-Shapiro, J-measure and many others (Lenca et al., 2004; Nada Lavrac et al., 1999; Ohsaki et al., 2004; Tan and Kumar, 2002).

Since directionality is a major aspect to be considered in the association rules to be derived in this thesis, we have decided to use recall and precision interestingness measures to assess the robustness of discovered co-occurrences and rules (Ohsaki et al., 2004). These measures are defined as follows:

$$\text{Recall} = \frac{\#\ \text{Predicted}\ \ \text{target}\ \ \text{events}}{\#\text{Total}\ \ \text{target}\ \ \text{events}} \tag{3.9}$$

$$\text{Precision} = \frac{\#\ \text{True\ \ predictions}}{\#\ \text{Total\ \ predictions}} \tag{3.10}$$

For an event couple (A, B), recall represents $P(A/B)$, that is the fraction of target events that were predicted. The precision represents $P(B/A)$ which is the number of true predictions over the total number of predictions, i.e, the fraction of correct predictions. A high recall (i.e. low rate of false negatives) means that a few target events were missed while a high precision reflects a high predictive capability and indicates a low rate of false positives. A false positive case corresponds to a wrong prediction and a false negative situation is when no prediction occurs prior to a target event. Since a high value of both interestingness measures is required considering the high cost of useless maintenance intervention in case of false positive predictions and the high cost of corrective maintenance in case of false negatives, a trade-off should be established to decide whether a significant event couple or pattern under scrutiny can be considered as an association rule or not.

When calculated in real data, due to the fact that some events have occurred very frequently in a limited number of trains only, the calculation of the recall/precision is affected negatively and leads to erroneous high values. To overcome this inconvenience, a filter was introduced prior to the calculation of interestingness measures of couples discovered by both the T-patterns and the null models. This filter identifies trains where the frequency of an event is greater than $\overline{x} + 3\sigma$, where $\overline{x}$ refers to the mean frequency of an event among all trains and σ its standard deviation. For an event couple (A, B), trains in which the frequency of event A or B is greater than the threshold are neglected. This procedure renders recall and precision values more robust.

3.5.3 Subjective Interestingness measures

In some cases, the information provided by objective measures might not be sufficient to judge if an itemset or pattern is significant enough to be considered as an association rule and a subjective point of view is needed based on experience and constraints. A subjective interestingness measure takes into account both the data and the user's knowledge. Such a measure is appropriate when: (1) The background knowledge of users vary, (2) the interests of the users vary, and (3) the background knowledge of users evolve. Unlike the objective measures considered in the previous subsection, subjective measures may not be representable by simple mathematical formulas because the users knowledge may be represented in various forms. Instead, they are usually incorporated into the mining process.

Within the applicative context of this thesis, discovered rules should respect two major constraints: a high global accuracy on one hand and a sufficiently large warning time on another. Target events should be predicted within a warning time sufficient enough to allow logistic and maintenance actions to be taken.

A prediction is considered to be correct if a target event occurs within its prediction period $[W, M]$, also called critical interval, which is defined by a warning time, W, and a monitoring time, M. The warning time is the time delay before a target event becomes highly probable to occur. The monitoring time determines how far into the future the prediction extends (see Figure 3.7).

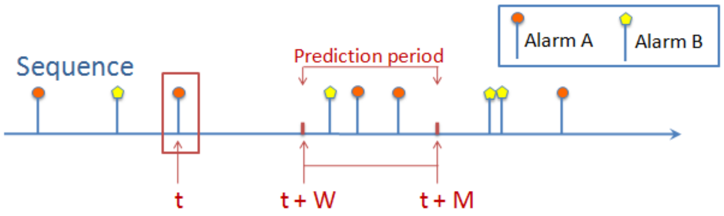


Figure 3.7: Illustration of warning time and monitoring time

Many approaches were proposed in (Magnusson, 2000; Tavenard et al., 2008) to model critical intervals that are based either on shrinking or splitting techniques till finding those with a significant p-value. Gaussian mixture models were also used to fit the histogram of the T_{AB} vector, consisting of the temporal distance between each A pattern and the first succeeding B event and thus deduce the most optimal critical interval (Salah et al., 2010). In the applicative case study of this thesis, railway experts have asked for an inter-event time of 30 minutes (d_1) and a prediction period extending to 24 hours (d_2). Thus, the critical interval was fixed to [30 minutes , 24 hours] which limited the approaches that can be used. In order to evaluate inter-event time, we compute the T_{AB} vector for each discovered rule $A \longrightarrow B$ (where B is a target event and A can either be a single event or pattern), which is equivalent to the vector of the time distance between every occurrence of the antecedent A pattern and the first succeeding B target event. The rule is usually considered to have a sufficiently acceptable inter-event time if the median of this vector is $\geq$ 30 minutes.

In the end, discovered couples abiding the inter-event time constraints and having acceptable recall and precision values are considered as significant association rules. It is important to mention that burstiness and redundancy in data can lead to the discovery of false but statistically-significant co-occurrences in the first step of the mining process. A physical subjective analysis by technical experts can identify such false associations and prune them out of the search process.

In the next section, experiments on synthetic data are performed in order to derive the most performant single null model as the well as the best double null model combinations which would be applied on real data along with the T-Patterns algorithm in 3.7.

3.6 Experiments on Synthetic Data

In order to determine the most powerful double null model combination, we need to search for the most performant single null models with the D and P scores, an experimental study was conducted on synthetic data sequences. This study is based on two diagnostics: (1) the efficiency to discover the planted co-occurrence or pattern (true positive) and (2) the ability to discard non-existing co-occurrences (false positives). The aim is also to analyze the effect of the window size parameter, i.e, the maximum co-occurrence distance w, as well as the number of bursts and the frequency of the planted co-occurrences on the performance of the null models in temporal data sequences. This knowledge is used to tune w to the value that will mostly contribute to optimal results on the TrainTracer data.

3.6.1 Generation Protocol

The generative model of the data is as follows. The generated sequences were randomly divided into sparse and un-overlapped dense segments of various lengths both constituted of m event types. The timestamps of each event type are generated randomly and separately by means of a Poisson process of parameter λ_j, where $j \in \{1, ..., m\}$ over a period of l hours. λ values are unique for each event type and are generated by means of a uniform distribution on the interval $[L1_{min}; L1_{max}]$ for sparse segments and $[L2_{min}; L2_{max}]$ for dense (burst) segments, where $L2_{max} << L1_{min}$. In the generated sequences, we have established a directed co-occurrence pattern between two event types A and B denoted (A, B) or $A \longrightarrow B$ that is defined by randomly generated Recall and Precision values. These values precise the percentage of A events followed by B events and of B events preceded by A events. B events succeeding the A events are planted within a temporal delay T_{AB} generated from a uniform distribution on a defined interval. We have focused on directed co-occurrences since our main goal on the real train data is to discover directed association rules of the form $A \longrightarrow B$ where B is a target event.

3.6.2 Experiments

In all of the performed experiments, 100 sequences of length 1000 hours were generated and randomized 100 times. The p-value threshold was fixed to 1%, $T_{AB} \in [0, 1]$ hours. The sequences consisted of 10 event types numbered from 1 to 10. A directed co-occurrence relationship was established between events 8 and 9 (injected pattern $8 \longrightarrow 9$). The length of dense segments (bursts) varied between 10 minutes and 1 hour.

- It is important to evaluate the performance of single and double null models under different values of the maximum co-occurrence window w in order to tune its value for the remaining experiments on synthetic and real data. Table 3.1 shows the results for the experimental diagnostics 1 and 2 of the single null models UL, FL and $FL(R)$ with the D and P scores (see Figure 3.4). Table 3.2 shows the results of the double null models. Recall and Precision values varied between 0.5 and 1 (50% and 100%). The value of w was varied between 30 minutes and 20 hours. Experimental diagnostic I represents the mean number of pairs that were discovered by the models in the 100 generated sequences and reflects the false positive rate. Diagnostic II represents the number of sequence generations where the injected pattern $8 \longrightarrow 9$ was discovered and reflects the prediction rate.

Table 3.1: Results of single null models on 100 generated data sequences of l = 1000 hours and varying values of w.

I - Mean number of discovered event couples in 100 generations						
w(h)	UL-D(A)	UL-P(B)	FL-D(A)	FL-P(B)	FL(A)-D	FL(B)-P
0.5	90	90	7	8	2	2
1	90	90	7	6	2	2
5	6	6	4	4	3	4
10	1	0	3	4	3	3
20	0	0	1	1	0	1
II - Number of generations where (a,b) was found significant						
w(h)	UL-D(A)	UL-P(B)	FL-D(A)	FL-P(B)	FL(A)-D	FL(B)-P
0.5	100	100	100	87	100	100
1	100	100	100	99	100	100
5	98	95	100	88	100	99
10	50	43	87	71	83	75
20	0	0	11	13	10	12

Table 3.2: Results of double null models on 100 generated data sequences of l = 1000 hours and varying values of w.

I - Mean number of discovered event couples in 100 generations									
w(h)	UL-D, UL-P	UL-D, FL-P	UL-D, FL(B)-P	FL-D, UL-P	FL-D, FL-P	FL-D, FL(B)-P	FL(A)-D, UL-P	FL(A)-D, FL-P	FL(A)-D, FL(B)-P
0.5	90	8	2	7	1	1	2	1	1
1	90	6	2	7	1	1	2	1	1
5	1	1	1	1	1	1	1	1	1
10	0	0.5	0	0	1	1	0	1	1
20	0	0	0	0	0	0	0	0	0
II - Number of generations where (a,b) was found significant									
w(h)	UL-D, UL-P	UL-D, FL-P	UL-D, FL(B)-P	FL-D, UL-P	FL-D, FL-P	FL-D, FL(B)-P	FL(A)-D, UL-P	FL(A)-D, FL-P	FL(A)-D, FL(B)-P
0.5	100	87	100	100	87	100	100	87	100
1	100	99	100	100	99	100	100	99	100
5	94	87	97	95	88	99	95	88	99
10	38	50	49	43	69	70	42	67	68
20	0	0	0	0	5	5	0	5	5

Knowing that the expected mean number of couples to be discovered is 1, results in tables 3.1 and 3.2 show that a small value of w leads to a high number of false positives and a very high value leads to a low prediction rate since it is more probable to obtain randomizations with scores that are higher than the initial scores. A trade-off value of 5 hours leads to the best results in both false positive and prediction rate. Results show that the false positive rate of double null models is inferior to that of single null models and hence more precise. The double null models leading to best results with the variation of the w were the $(FL-D, FL(B)-P)$ and $(FL(A)-D, FL(B)-P)$.

- Although the aim of our work is to discover relevant co-occurrences with high significance and correlation, it is interesting however to test how far would the double null models go in the discovery of co-occurrences with variable strength defined by Recall and Precision values. Consider Figure 3.8 below, we have tested the single and double null models with the D and P score on zones 1, 2 and 3. In each case, 100 sequences were generated with a relationship between events 8 and 9 that is defined by recall and precision values that are randomly generated on an interval defined for each zone. For example, in zone 1, both recall and precision values vary on the interval $[0.5; 1]$ and thus the relationship between events 8 and 9 is strong. In zones 2 and 3 one of these measures is high ($\in [0.5; 1]$) while the other is weak ($\in]0; 0.5]$), which might be possible in real cases due to burstiness or transmission/reception error. In Zone 4, the relationship is very weak and the performance of single and double null models in this zone is not of much interest and will not be discussed. Results in tables 3.3 and 3.4 show that the double null models which gave the best results in both the prediction rate and false positive rate are the $(FL-D, FL(B)-P)$ (the FL null model with the D score combined with the $FL(B)$ null model with the P score) and the $(FL(A)-D, FL(B)-P)$ (the $FL(A)$ null model with the D score combined with the $FL(B)$ null model with the P score). We can also see from the results that the double null models succeed in discovering the injected co-occurrences (with 0 false positives) in zones 1 and 3. In zone 2 with low recall value, the 9 events generated are much more frequent than the 8 events (in order to respect the recall value) and most of them are not preceded by the 8 events in the synthetically generated initial data, thus there is a high probability to obtain a higher score in the randomized data, which explains why the null models had low prediction rate.

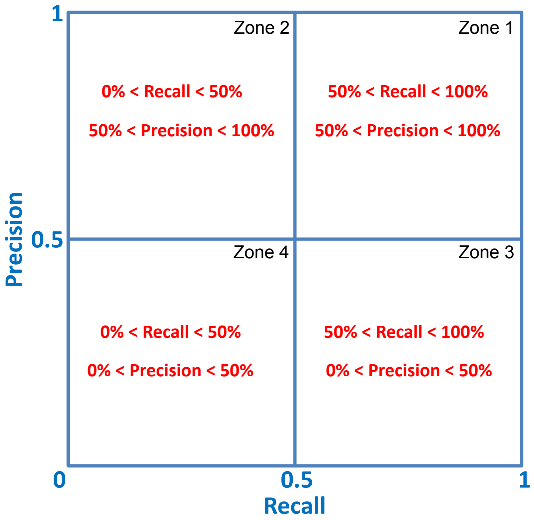


Figure 3.8: Graphical Illustration of the four test zones

Table 3.3: Results $l = 1000$ hours, $w = 5$ hours and varying values of recall and precision.

I - Mean number of discovered event couples in 100 generations						
Zone	UL-D(A)	UL-P(B)	FL-D(A)	FL-P(B)	FL(A)-D	FL(B)-P
1	7	7	5	5	4	4
2	4	4.5	5	4	1	1
3	11	12	4	4	3	3
II - Number of generations where (a,b) was found significant						
Zone	UL-D(A)	UL-P(B)	FL-D(A)	FL-P(B)	FL(A)-D	FL(B)-P
1	99	92	100	88	99	95
2	15	1	39	1	33	16
3	100	80	86	73	92	79

Table 3.4: Results $l = 1000$ hours, $w = 5$ hours and varying values of recall and precision

I - Mean number of discovered event couples in 100 generations									
Zone	UL-D, UL-P	UL-D, FL-P	UL-D, FL(B)-P	FL-D, UL-P	FL-D, FL-P	FL-D, FL(B)-P	FL(A)-D, UL-P	FL(A)-D, FL-P	FL(A)-D, FL(B)-P
1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0
3	2	1	1	1	1	1	1	1	1
II - Number of generations where (a,b) was found significant									
Zone	UL-D, UL-P	UL-D, FL-P	UL-D, FL(B)-P	FL-D, UL-P	FL-D, FL-P	FL-D, FL(B)-P	FL(A)-D, UL-P	FL(A)-D, FL-P	FL(A)-D, FL(B)-P
1	92	88	94	92	88	95	91	88	94
2	1	1	6	1	1	14	1	1	14
3	80	73	79	74	68	72	76	70	76

- In order to test the efficiency of the single and double null models on bursty data sequences, the number of bursts was varied. Results in 3.5 and 3.6 show that as the number of bursts increases, the UL null model loses its predictability as so do the UL-based double null models. The $(FL-D$, $FL(B)-P)$ and the $(FL(A)-D$,

$FL(B)-P$) double models have showed to give the best results considering both the false positive rate and the prediction rate. These two null models both outperform the single null models as well as the other tested double null models in giving more precise results with lower false positive rate while conserving approximately the same prediction rate as single null models when predicting strong and significant co-occurrences. These two null models will hence by applied on the real TrainTracer data extracts to discover significant co-occurrences between couples of events. In the following, the $(FL-D, FL(B)-P)$ double null model will be referred to as DNM1 and the $(FL(A)-D, FL(B)-P)$ double models as DNM2.

Table 3.5: Results of single null models for sequences of l = 1000 hours, w = 5 hours and varying burstiness.

I - Mean number of discovered event couples in 100 generations						
Nb. bursts	UL-D(A)	UL-P(B)	FL-D(A)	FL-P(B)	FL(A)-D	FL(B)-P
10	7	7	4	4	3.5	3
20	2	1	6	6	4.5	5
30	1	0	9	9	6.5	6
40	1	0	9.5	9	7	8
II - Number of generations where (a,b) was found significant						
Nb. bursts	UL-D(A)	UL-P(B)	FL-D(A)	FL-P(B)	FL(A)-D	FL(B)-P
10	98	93	100	94	100	97
20	90	68	100	74	100	90
30	63	36	100	48	99	70
40	53	19	99	47	99	67

Table 3.6: Results of double null models for sequences of l = 1000 hours, w = 5 hours and varying burstiness

I - Mean number of discovered event couples in 100 generations									
Nb.	UL-D,	UL-D,	UL-D,	FL-D,	FL-D,	FL-D,	FL(A)-D,	FL(A)-D,	FL(A)-D,
bursts	UL-P	FL-P	FL(B)-P	UL-P	FL-P	FL(B)-P	UL-P	FL-P	FL(B)-P
10	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1
30	0	0	1	0	0.5	1	0	1	1
40	0	0	1	0	0	1	0	1	1
II - Number of generations where (a,b) was found significant									
Nb.	UL-D,	UL-D,	UL-D,	FL-D,	FL-D,	FL-D,	FL(A)-D,	FL(A)-D,	FL(A)-D,
bursts	UL-P	FL-P	FL(B)-P	UL-P	FL-P	FL(B)-P	UL-P	FL-P	FL(B)-P
10	93	94	95	93	94	97	93	94	97
20	68	74	85	68	74	90	68	74	90
30	36	48	59	36	48	70	36	48	70
40	19	44	51	19	47	67	19	47	67

3.7 Experiments on Real Data

In this section, the results obtained by the most performant single null models ($FL-D$, $FL(A)-D$ and $FL(B)-P$), the two proposed DNMs as well as by the T-patterns algorithm (TP) are presented and discussed. The main issue rendering experiments on real data difficult lies in the fact that no ground truth is available, and thus we cannot evaluate the results easily (for example compare the discovered association rules to ones that are already known). The adopted procedure is hence the following: The discovered couples are evaluated by means of objective interestingness measures (Recall and Precision) as well as subjective (Inter-event time). Couples abiding the accuracy and inter-event time constraints are considered as statistically significant association rules and are presented to technical experts to obtain a physical analysis and feedback.

In Section 3.6, Experiments have shown that the efficiency of null models decreases with high values of w. However, knowing that a small value of w signifies a short co-occurrence scanning distance, this would mean that all pairwise co-occurrences with an inter-event time longer than w will be neglected. Thus, a trade-off was considered for experiments with real TrainTracer™ data sequences, and w was fixed to 5 hours.

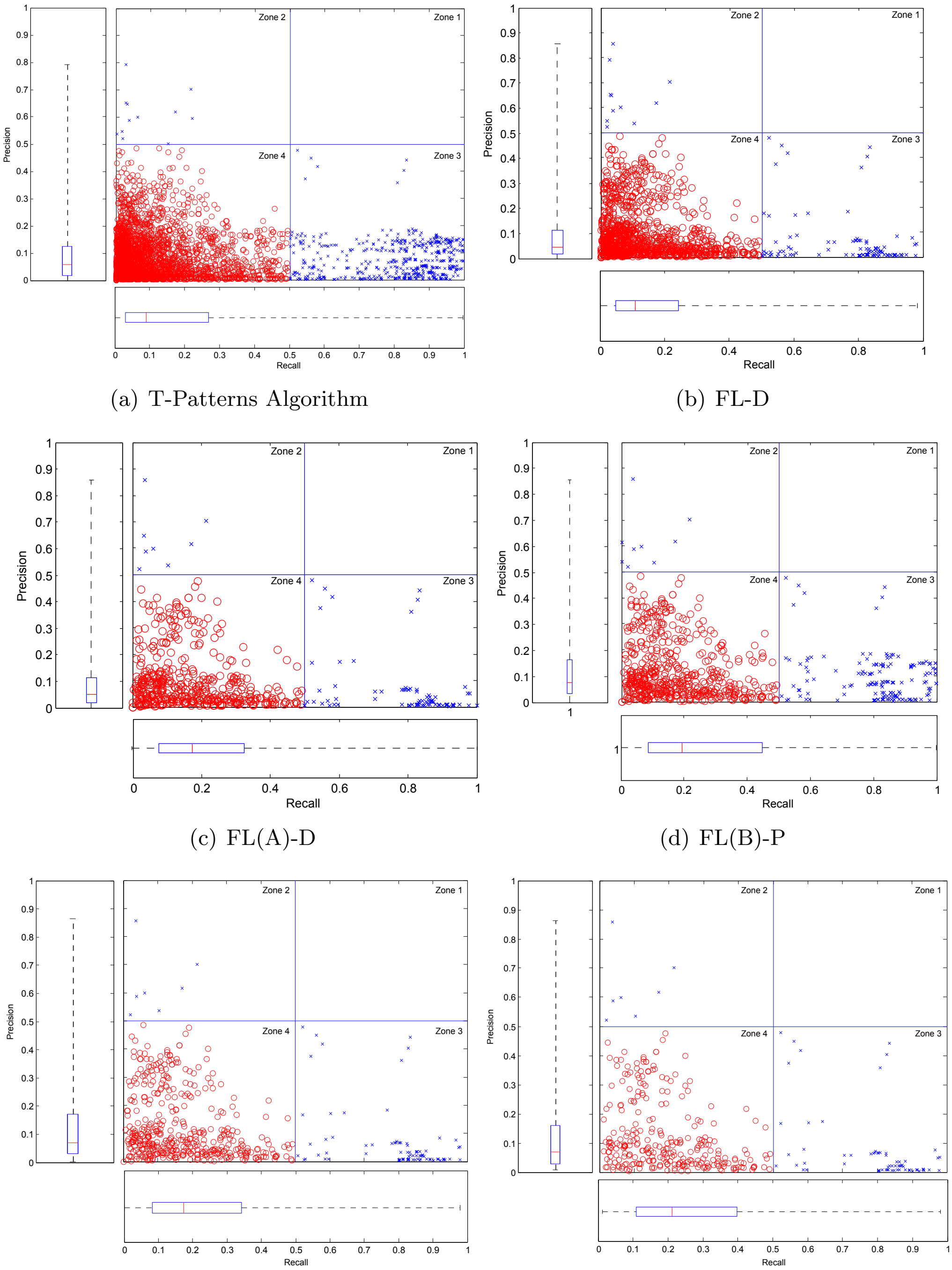
In Table 3.7, Results (1) represents the number of significant couples discovered by the T-patterns algorithm (TP), the FL and $FL(A)$ null models with the D score, the $FL(B)$ model with the P score and the double null models DNM1 $(FL-D, FL(B)-P)$ and DNM2 $(FL(A)-D, FL(B)-P)$ with a p-value threshold = 1%.

Table 3.7: (1) Number of significant event couples discovered by the T-patterns algorithm (TP)($\alpha = 1\%$) as well as single and double null models (p-value threshold = 1%) respectively in the TrainTracer™ data sequences. (2) Number of significant event couples abiding the inter-event time constraint discovered by the T-patterns algorithm (TP) ($\alpha = 1\%$) as well as single and double null models (p-value threshold = 1%) respectively in the TrainTracer™ data sequences.

	TP	FL-D	FL(A)-D	FL(B)-P	DNM1	DNM2
(1)	3667	1454	639	776	598	404
(2)	3608	1300	567	714	547	367

All of the above mentioned discovered couples were subjected to several evaluation processes in order to determine those satisfying the inter-event and accuracy constraints and thus would then be considered as reliable rules. These processes consisted of modeling inter-event times in addition to the calculation of recall and precision measures. Since railway experts from Alstom have asked for a critical interval of $[30min, 24hours]$, mining was focused on couples with inter-event times at least equal to 30 minutes. Results (2) in 3.7 represent the number of significant event couples abiding the inter-event time constraint. In total, 547 couples were discovered by DNM1 consisting of a merge between $FL-D$ and $FL(A)-D$, while 367 couples were discovered by DNM2 consisting of a merge between $FL(A)-D$ and $FL(B)-P$. These couples are hence considered to be statistically significant and are thus retained to be further scrutinized and post-treated. Figure 3.9 show the Recall/Precision scatter plots for couples discovered by the

T-Patterns algorithm, the single null models as well as by the proposed DNM models.



(a) T-Patterns Algorithm

(b) FL-D

(c) FL(A)-D

(d) FL(B)-P

(e) DNM1: FL-D,FL(B)-P

(f) DNM2: FL(A)-D,FL(B)-P

Figure 3.9: Recall/Precision scatter plot of all couples with T_{AB} median value ≥ 30 minutes discovered by the various approaches. Recall threshold = 50%, Precision threshold = 50%

As shown in the scatterplots of Figure 3.9, four zones can be defined according to 50% thresholds on both recall and precision. Couples of events belonging to zone 1 are statistically very relevant and hence can be considered as plausible association rules with high interestingness. Zone 2 and 3 contain all couples with either a high recall or precision value. Event couples belonging to these two zones are considered to be possibly relevant enough to be association rules considering that the weakness of one of the two measures might be a result of the complexities occurring in the data such as redundancy or bursts. Zone 4 contains all couples with low recall and precision values. These couples are considered to be statistically insignificant. The scatterplots show that the double null models, single null models and the T-patterns algorithm have discovered the same interesting couples. The difference however, is in the low number of spurious insignificant couples that were discovered by both algorithms. The proposed DNM approach have neglected most of the insignificant couples of zone 4 which were discovered by single null models and the T-Patterns algorithm and hence has shown to be more robust against spuriousness. Table 3.8 shows the number of discovered couple per zone.

Table 3.8: Number of couples of events per zone discovered by the T-Patterns algorithm (TP), DNM1 (FL-D, FL(B)-P), DNM2 (FL(A)-D, FL(B)-P) for Recall threshold = 50%, Precision threshold = 50%

Zone	TP	FL-D	FL(A)-D	FL(B)-P	DNM1	DNM2
1	0	0	0	0	0	0
2	12	11	8	9	7	7
3	562	115	85	166	82	71
4	3034	1174	474	539	458	289

The analysis of the discovered association rules had to be both statistical and physical with the help of railway maintenance experts in order to identify among them those having a real physical meaning. Indeed, useful dependencies between elements indirectly connected in the subsystem can be found as well as spurious and normative association rules which are omitted by knowledgeable experts if they don't have any technical significance. Due to confidentiality agreements, it is not possible to provide examples of the discovered association rules.

It is important to indicate that both recall and precision values may be negatively affected by data bursts in a specific train or at a certain period of time where failures were frequent due to infrastructure problems or other factors. That is why, prior to presenting the rule to technical experts, we have considered the recall and precision values of the association rule per train as well as the distribution of the two events of the couple amongst trains over the 6-month observation period (example Figure 3.10). The observation of unusual distributions decreased the chances of a rule to be credible.

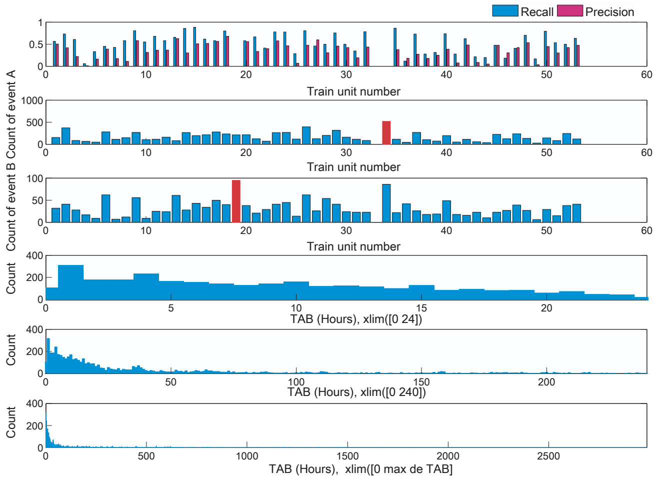


Figure 3.10: Example of the distribution of Recall/Precision values of an association rule A ⟶ B per train as well as the distribution of both events and the histograms of all T_{AB} values of the rule visible within variable time scales

3.8 Conclusion

In this chapter, two different significance testing approaches were applied to discover significant co-occurrences leading to length-2 rules of the form $A \longrightarrow B$, where B is a target event. Although co-occurrences are decided by means of hypothesis test in both methods, the approaches used to calculate the p-value used by the test are completely different. The first, Null Models, consists of randomization techniques followed by the calculation of various co-occurrence scores. The second, T-Patterns, exploits the temporal dimension by investigating the statistical dependence between inter-arrival times of couples of events in order to highlight possible relationships. In addition to that, a new approach called Double Null Models (DNM) was proposed which consists of a bipolar null model approach emphasizing on detecting interesting recall-precision relationship between events. The main challenge faced by the algorithms is to discover significant co-occurrences between infrequent events in bursty sequences, while maintaining a low false positive rate.

In order to test the efficiency of single null models and to identify the best double null model combination, synthetically generated data sequences with implemented co-occurrences and bursts were generated. The aim of these tests was also to tune some parameter values such as the maximum co-occurrence window w. The proposed null model approach in its two best combinations was found to outperform single null models in false positive rate while conserving a high prediction level.

The most performant single null models, double null models and the T-Patterns algorithm were then applied on real TrainTracer data. The resulting discovered co-occurrences were then assessed using objective interestingness measures (Recall and

Precision) and subjective interestingness measures (inter-event time) in order to evaluate their abidance to the warning time and accuracy constraints imposed by the applicative problematic. Those who did were considered as significant rules.

Significance testing algorithms based on statistical hypothesis tests are efficient methods to discover relationships and co-occurrences between infrequent events in both bursty and non-bursty sequences, unlike the dominating frequency-based approaches. The main inconvenience lies in the heavy computational time (few days for complex TrainTracer sequences) as well as the pattern length which is limited to 2. In the case of null models, the co-occurrence window w should be tuned appropriately. The T-Patterns algorithm, although efficient in discovering temporal dependencies, is sensitive to spuriousness and burstiness. Knowing that the main target of this chapter was to mine statistically significant co-occurrences leading to length-2 rules, it is always interesting to search for longer rules. This will be the main task of chapter 4.

Chapter 4

Weighted Episode Rule Mining Between Infrequent Events

Contents

4.1 Introduction

Association rule mining has been developed for transaction data problems. However, when working with temporal long sequences, the term episode rule mining is mostly employed. The difference between Episode Rules and Association Rules is that the former takes timestamps of events into account and the order does not have to be important (in case of parallel episodes). Association rule mining algorithms, in general, although effective in mining frequent rules, are vulnerable to the rule explosion problem. For example, using a low support threshold with Apriori-based algorithms in order to explore rules between infrequent items or events leads to an infinite number of combinations of rules growing exponentially with the number of items or events in the sequence. This places a sizable burden on the decision maker who has to analyze these rules in order to find those who are of actual interest. One of the recently emerging approaches which have been developed to overcome this problem and to assist rule mining algorithms in producing interesting rules is Weighted association rule mining WARM (Cai et al., 1998; Sun and B., 2008; Wang et al., 2000; Yan and Li, 2006). This approach was developed mainly for transaction data problems. Unlike the classical model of association rule mining where all items are treated with equal importance, WARM suggests to substitute an item's support with a weighted form of support as a mean of numerical prioritization to favor items over others. Hence, the higher the weight of an item, the more important it is. These weights can be assigned manually by field experts or automatically by means of recently proposed techniques. For example, in market basket analysis problems, high weights are attached to items of high importance such as high profit items.

In this Chapter, we first tackle the problem of Episode Rule mining in sequences in 4.2 and give an extensive literature study. Afterwards, we discuss the weighted association rule mining (WARM) problem in 4.3. We formally define it in 4.4 and adapt it to the problem of mining episode rules in temporal sequences. We then explain WINEPI in 4.5.2, a frequent episode rule mining algorithm proposed in (Mannila et al., 1997) consisting of a sliding window transforming a temporal data sequence into a series of overlapped windows. We also define the valency model proposed in (Koh et al., 2010) to calculate weights for items in a transaction data based on their interactions with their neighbours in 4.5.4. We propose an approach based upon WINEPI and the valency model which we call Weighted WINEPI aimed to find significant episode rules between infrequent items in 4.5.3 and an approach derived from it in 4.5.5 to better include infrequent events in the mining process. We also propose "Oriented Weighted WINEPI" in 4.5.6, which is more suitable to the applicative problematic of this thesis and tune the mining process towards discovering rules leading to target events. Methods are confronted and tested on synthetic and real data in 4.5.7 and 4.5.8. Finally, we resume and conclude in 4.6.

4.2 Episode rule Mining in Sequences

Many data mining and machine learning techniques are adapted towards the analysis of unordered collections of data such as transaction databases. However, numerous important applications require the analyzed data to be ordered with respect to time, such as data from telecommunication networks, user interface actions, occurrences of recurrent illnesses, etc. These datasets are composed of large temporal sequences of events, where each event is described by a date of occurrence and an event type. The main aim behind analyzing such sequences is to find *episode rules*. Informally, an episode rule is a causal relationship reflecting how often a particular group of event types tends to appear close to another group. Once these relationships are found, they can be used to perform an on-line analysis to better explain the problems that cause a particular event or to predict future events. Episode mining has been of important interest in various applications, including internet and network anomaly intrusion detection (Luo and Bridges, 2000; Qin and Hwang, 2004; Su, 2010; Wang et al., 2008), Biology (Bouqata et al., 2006; Casas-Garriga, 2003b; Méger et al., 2004; Patnaik et al., 2008), stock market prediction and finance (Ng and Fu, 2003), climatology (Harms et al., 2001b), chiller management (Patnaik et al., 2011) and many others. In this section, we formally define the problem of sequence mining for episode rules. We then present a literature review of the most notorious algorithms and approaches that have been conceived and used for this purpose.

4.2.1 Notations and Terminology

In the following, we define the standard notions used in the problems of episode mining in sequences. Consider the input as a sequence of events, where each event is expressed by a unique numerical code and an associated time of occurrence.

Definition 4.2.1. ***(event)*** Given a set E of event types, an event is defined by the pair (R, t) where $R \in E$ is the event type and $t \in \Re^+$ its associated time of occurrence, called timestamp.

Definition 4.2.2. ***(event sequence)*** An event sequence S is a triple (S, T_s, T_e), where $S = \{(R_1, t_1), (R_2, t_2), ..., (R_n, t_n)\}$ is an ordered sequence of events such that $R_i \in E \;\; \forall i \in \{1, ..., n\}$ and $T_s \le t_1 \le t_n \le T_e$. Figure 4.1 illustrates an example.

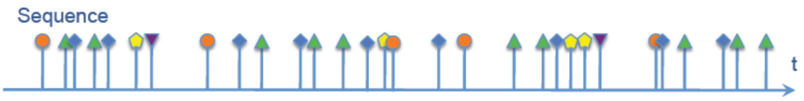


Figure 4.1: A graphical example of a sequence

Definition 4.2.3. ***(Episode)*** An episode is a partially ordered collection of events occurring together. Episodes can be described as directed acyclic graphs. Consider, for instance, episodes α, β and γ in figure 4.2.

Episode α is a serial episode: it occurs in a sequence only if there are events of types A and B that occur in this order in the sequence. Other events can occur between these two. The event sequence, for instance, is merged from several sources, and therefore it is useful that episodes are insensitive to intervening events. Episode β is a parallel episode: no constraints on the relative order of A and B are given. Episode γ is an example of non-serial and non-parallel episode: it occurs in a sequence if there are occurrences of A and B and these precede an occurrence of C; no constraints on the relative order of A and B are given.

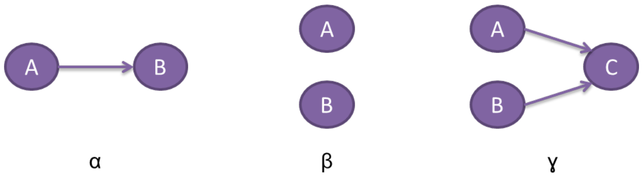


Figure 4.2: illustration of serial (α), parallel (β) and composite (γ) episodes

Definition 4.2.4. ***(Episode rule)*** We define an episode rule as an implication of the form $A \longrightarrow B$, where the antecedent and consequent are sets of events (episodes) with $A \cap B = \phi$.

In this thesis, we employ the term "association rules" when referring to episode rules as wells.

4.2.2 Literature review

1. **Initial algorithms: mining sequences in transaction databases**

   The problem of mining sequential patterns was initially introduced in (Agrawal and Srikant, 1995) and applied to database sequences. The problem was formulated as follows: "Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given a user-tuned *minsup* threshold, the aim of sequential pattern mining is to find all frequent subsequences, whose occurrence frequency in the database of sequences is no less than *minsup*" (Agrawal and Srikant, 1995). The main difference between frequent itemsets and sequential patterns is that a sequential pattern considers the order between items, whereas frequent itemset does not specify the order. The initial algorithm as well as most of the existing ones are frequency-oriented.

   In (Srikant and Agrawal, 1996), an Apriori-based algorithm, GSP (Generalized Sequential Patterns) was proposed to mine sequential patterns. The approach depends on the Apriori monotonicity property adapted to sequences. This property states that, if a sequence S is not frequent, then none of the super-sequences of S can be frequent. For example if sequence $\{a, b\}$ is infrequent, this implies that its super-sequences such as $\{a, b, c\}$ or $\{a, c, b\}$ would be infrequent too. The disadvantage of this approach, as all Apriori-based appoaches, is the time complexity in situations with high number of frequent patterns, long patterns or low *minsup* threshold, due to the huge number of candidate sets to evaluate as well as the repetitive database scans to be performed. To overcome this problem, the Prefix Span algorithm was proposed in (Pei et al., 2001b) by extending the concept of FP-tree (Han et al., 1999) into a prefix-projected pattern growth. A notorious algorithm called SPADE is proposed in (Zaki, 2001). SPADE is a vertical format sequential pattern mining method which maps the sequence database into a vertical id-list database format <Sequence ID, Event ID>, defining each item. SPADE then uses combinatorial properties to decompose the original problem into smaller sub-problems which can be independently solved using efficient lattice search techniques and joint operations. Subsequences are expanded using Apriori candidate generation strategy. Other algorithms were proposed that use constraints in order to limit and focus the mining operation, hence affecting what is called constraint based sequential pattern mining, where the monotonocity and anti-monotonocity properties can be applied to constraints. Monotonicity means that if an item set satisfies the rule constraint, then all of its supersets satify as well, whereas Anti-monotonicity means that if an item set does not satistify the rule constraint, then none of its supersets do as well.

2. **Mining episodes in long sequences**

   The concept of finding patterns in long sequences of timestamped events as well as the first algorithm to tackle this problem was described by (Mannila et al., 1997). Patterns are described as episodes and can be **parallel**, where the order in which events occur is irrelevant, or **serial**, where events occur in a particular order, or a combination of the two. This problem can be view as a constrained mining problem since episodes (with their directionality) constrain events in the form of acyclic graph. The **standard** episode association rule mining problem is based on the anti-monotonic Apriori of frequent patterns and tends to find all episode rules satisfying given frequency and confidence constraints.

   Two main approaches were proposed in (Mannila et al., 1997). The first, WINEPI, slides a window of fixed length over the sequence, and each window containing the episode counts towards its total frequency, which is defined as the proportion of all windows containing it. The confidence of an association rule $X \longrightarrow Y$, denoted $conf(X \longrightarrow Y)$, is defined as the ratio of the frequency of $X \cup Y$ and the frequency of X. Once the frequent episodes have been found, rules between them are generated in the traditional manner. The second approach, MINEPI, searches for frequent episodes based on their minimal occurrences. Here, however, association rules are of the form $X[win1] \Longrightarrow Y[win2]$, meaning that if itemset X has a minimal occurrence in a window W_1 of size win_1, then X$\cup Y$ has a minimal occurrence in a window W_2 of size win_2 that fully contains W_1. Both approaches have been designed to function using a maximum window size constraint which specifies the maximum elapsed time between the first and the last event of the occurrences of the patterns. To be more precise, in the case of WINEPI, the algorithm uses a single windows size constraint. If the user wants to perform an extraction with a different window size, the algorithm must be executed again. The other algorithm, MINEPI, neccessitates a maximal window size constraint to restrict fairly the search space in practice, but is capable to derive rules for several window sizes that are inferior to this maximal window size.

Other approaches based on different types of constraints were introduced, for example, episode rule mining based on maximum gap constraint, as defined by (Casas-Garriga, 2003a), was done by (Méger and Rigotti, 2004), but only for serial episodes.

Most other related work were based on the WINEPI definitions, and mainly attempted to find the same rules (or representative subsets) more efficiently (Das et al., 1998; Harms et al., 2001a), by either decreasing computational time or reducing the number of unimportant rules obtained. The perfomance of these algorithms dramatically degrades in the case of mining long sequential patterns in dense databases or when using a low minimum support threshold. (Yun, 2008) presented a weighted sequential pattern mining algorithm (WSpan). This algorithm uses weight constraints to reduce the number of unimportant patterns. During the mining process, weights are used to evaluate patterns in addition to support. (Chen and Huang, 2008) proposed an algorithm called PTAC (sequential frequent Patterns mining with Tough Aggregate Constraints) which embodies two strategies. The first evaluates patterns by means of a “promising-ness” feature as well as the validity of the corresponding prefix. The second process prunes unpromising patterns. (Garofalakis et al., 2002) proposed SPIRIT, a family of algorithms which use relaxed constraints with different properties such as anti-monotonicity in order to filter out some unpromising candidate patterns in the early stage of the sequential pattern mining process. In (Pei et al., 2007), authors propose a pattern-growth method for constraint-based sequential pattern mining problems. They show that all the monotonic and anti-monotonic constraints are prefix-monotone and thus can be pushed deep into pattern-growth-based mining. Other sequential pattern mining methods include the work of (Laur et al., 2007) in which a new statistical approach was introduced based on statistical supports to enhance the mining precision and improve the computational efficiency of the incremental mining process. This approach maximizes either the precision or the recall and limits the degradation of the other criterion. In (Lin et al., 2008), the notion of positive and negative sequential patterns was introduced. When a particularly-monitored itemset is found to be a subset of a pattern, this pattern is called positive pattern. Negative patterns represent those lacking this itemset. In (Kuo et al., 2009), a K-means algorithm was used to acquire better computational efficiency for fuzzy sequential pattern mining.

3. **Mining rules between infrequent events**

   However, most of the existing algorithms for mining sequential data as well as the ones existing above are frequency-based and oriented towards the discovery of frequent patterns. A major limitation of the frequent pattern approach is that it relies on the existence of a meaningful minimum support level that is sufficiently strong to reduce the number of accepted patterns to a manageable level. However, in some data mining applications relatively infrequent associations are likely to be of great interest and might exhibit strong coherence. These associations are likely to be pruned out by the minimum support threshold used by most approaches. An example of such association is the Vodka and Caviar problem, presented by (Cohen et al., 2000). Associations between expensive items such as Ketel Vodka and Beluga Caviar are likely to be of interest due to the high value of both products but will be infrequent and probably undetected by frequent itemset mining algorithms. Another important disadvantage of relying strictly on frequency constraints is that it treats every item with equal significance, i.e, every item fulfilling the frequency threshold would be allowed to survive, regardless of its informative value, which results in the discovery of numerous spurious patterns. Significant patterns that are not frequent enough can rarely be detected unless a very low frequency threshold is used which would imply in its turn a very heavy computational cost.

   To address this problem, recent years have witnessed the uprisal of other algorithms focusing on mining rules between infrequent items and events without the use of a minimum support constraint. Several algorithms for rule-space search directly explore the space of potential rules (Bayardo, 1999; Bayardo et al., 2000; Webb, 2000). This framework allows a wide range of criteria to be applied to identify interesting rules other than the minimum support criterion used by the Apriori approach. One of the major approaches to handle this problem is constraint-based association rule mining which increase the level of user engagement in the mining process (Grahne et al., 2000; Ng et al., 1998; Srikant et al., 1997) as well as weighted association rule mining techniques which value the importance of items by assigning them weights either manually (using expert knowledge) or automatically using models based on the quality of interactions and connections between items (Koh et al., 2010; Pears et al., 2013). Other interesting approaches aimed to discover particular associations as pairwise co-occurrences such as null models, randomization algorithms that are followed by the calculation of different scores and a statistical hypothesis test to assess the significance of occurrences in (Haiminen et al., 2008; Hannenhalli and Levy, 2002; Klein and Vingron, 2007; Levy et al., 2001) as well as the T-patterns algorithm which exploits the temporal dimension by investigating the statistical dependence between inter-arrival times of couples of events in order to highlight possible relationships (Magnusson, 2000; Salah et al., 2010).

4.3 Weighted Association Rule Mining: Relevant Literature

The classical association rule mining scheme (Apriori, FP-tree and their variants) is based primarily on the frequency of items to decide which will be pruned or not. This strict adherence on support and confidence framework was not designed to deal with the rare items problem (Koh and Rountree, 2005). Items which are rare but co-occur together with high confidence levels are unlikely to reach the minimum support threshold and are therefore pruned out. Once of recently emerging approaches to overcome this problem is weighted association rule mining (Cai et al., 1998; Ramkumar et al., 1998; Songfeng et al., 2001; Sun and B., 2008; Tao et al., 2003; Wang et al., 2000; Yan and Li, 2006), developed for transaction data problems. The general principle behind this approach is to substitute an item's support with a weighted support, reflecting individual importance depending on the problematic. Thus rules and itemsets containing high weight items will have proportionately higher weighted support. This provides an alternative ranking process to the classical support and confidence framework used for rule ranking. The notion of weighted support was first introduced to association rule mining in (Ramkumar et al., 1998) by assigning weights manually to both items and transactions. In their approach rules whose weighted support is larger than a given threshold are kept for candidate generation, much like in traditional Apriori (Agrawal et al., 1993). In (Cai et al., 1998), weighted support was defined in a similar way except that weights were applied only to items and not to transactions. Two different ways were proposed to calculate the weight of an itemset, either as the sum of all the constituent items weights or as the average of the weights. Both of these approaches invalidated the downward closure property (Agrawal and Srikant, 1994), which resulted in additional complexity and time consumption.

This led (Tao et al., 2003) to propose a "weighted downward closure property" that can be retained by using weighted support. In their approach called WARM, two types of weights were assigned: item weight and itemset weight. The aim of using weighted support is to make use of the weight to steer the mining process towards the selection of targeted itemsets according to their perceived significance in the dataset, influenced by that weight, rather than by their frequency alone.

Many other techniques have been proposed in the recent years, mostly for transaction data problems. In (Wang et al., 2000), transactions are transformed into a directed graph where nodes denote items and links represent association rules. A generalized version of Kleinberg's HITS link-based model was then applied on the graph to deduce item rankings. (Yun and Leggett, 2006) introduced the Weighted Interesting Pattern (WIP) approach based on a novel measure called w-confidence. The w-confidence for a given pattern P is defined as the ratio of the minimum weight of items in P to the maximum weight within P. Patterns with w-confidence and weighted support are greater than user defined minimum thresholds were mined from an FP-tree that was built by ordering items in weight ascending order. (Farhan Ahmed et al., 2011) also used an FP-tree approach within a market basket analysis context using retail transaction data defined a utility measure that took into account the number of units purchased per item in a market basket. They used the unit price and the unit profit for each item to define a utility value for an item over a transaction. The utility value for an item/itemset over the entire dataset was then obtained by summing up the utilities over all transactions in the dataset. They showed that the utility measure satisfied the downward closure property and this was exploited when mining the prefix tree that was built.

Until now, in most weighted association rule mining approaches, weight assignment process relies on users subjective judgments. The weight of each item is fixed from the beginning (Cai et al., 1998; Ramkumar et al., 1998; Tao et al., 2003; Vo et al., 2013; Wang et al., 2000) and depends heavily on expert knowledge. Research in this area has concentrated on formulating efficient algorithms for exploiting these pre-assigned weights. For example, recently (Li et al., 2007) introduced a system for incorporating weights for mining association rules in communication networks. They made use of a method based on a subjective judgements matrix to set weights for individual items. Inputs to the matrix were supplied by domain specialists in the area of communications networks.

The major issue when relying on subjective input is that rules generated only encapsulate known patterns, thus excluding the discovery of unexpected but nonetheless important rules. Another issue is that the reliance on domain specific information limits the applicative scope to only those domains where such information is readily available. Indeed, having such domain specific knowledge would provide an accurate representation of the current reality. However, many application domains exist where such knowledge is either unavailable or impractical to obtain.

It is not until recently that dynamic weights have received interest. Yan and Li (Yan and Li, 2006), working in the Web mining domain, asserted that weights can be assigned on the basis of the time taken by a user to view a web page. Weights are allowed to vary according to the dynamics of the system, as pages became more popular (or less popular) the weights would increase (or decrease). (Li and Li, 2010) used an FP-tree approach to perform weighted association rule mining in the domain area of telecommunication networks. The k-support of an itemset was defined as the weighted support of the itemset relative to a user defined weighted minimum support threshold, *wminsup*. The main novelty of their work was to use a neural network to adapt item weights when concept drift in the data caused the weights of items to change.

Recent research has shown that it is possible to deduce the relative importance of items automatically based on their interactions with each other with no need for explicit domain-specific knowledge above what is needed by the classical Apriori approach. (Koh et al., 2010) were the first to propose a totally automated weight-assignment model in transaction dataset based on an item's interaction with other items. This model, called valency model, is based on a linear combination between two main factors: purity and connectivity. The purity of an item is determined by the number of items that it is associated with over the entire transaction database; the greater the number the lower the purity and vice versa, whereas connectivity describes the strength of the interactions between items. (Koh et al., 2010) applied Principal Component Analysis to quantify the rule base generated and showed that the rule base captures a higher degree of variation across the underlying dataset than those generated by Apriori. The valency model was extended in (Pears et al., 2013) by expanding the field of interaction beyond immediate neighborhoods by using a graph based connectivity model.

In the following section, we define formally the weighted association rule mining problem for transaction data, and then adapt it for temporal sequences in 4.5.

4.4 The Weighted Association Rule Mining Problem

The weighted association rule mining problem was initially defined for frequent itemset mining in transaction datasets and not for sequences. The aim of weighted association rule mining is to steer the focus of the mining process towards significant relationships involving items with significant weights rather than to try avoid the combinatorial explosion of spurious insignificant relationships.

Given a set of items, $I = \{i_1, i_2, ..., i_n\}$, a transaction may be defined as a subset of I and a dataset as a set D of transactions. A set X of items is called an itemset. The support of X, $sup(X)$, is the proportion of transactions containing X in the dataset. For an association rule of the form $X \longrightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \phi$, the support of the rule is $s = sup(XY)$. This rule holds in the transaction set D with confidence c where $c = conf(X \longrightarrow Y) = sup(XY)/sup(X)$.

Given a transaction database D, a support threshold $minsup$ and a confidence threshold $minconf$, the task of Apriori-based association rule mining algorithms is to generate all association rules that have support and confidence above the user-specified thresholds.

However, in weighted association rule mining approaches, a weight w_i is assigned to each item i, where $-1 \leq wi \leq 1$, reflecting the relative importance of an item over other items that it is associated with. This weight can be assigned manually or generated automatically. The weighted support of an item i is $w_i sup(i)$, which helps control the frequency factor of items. Similar to traditional association rule mining algorithms that are a based on support and confidence framework, a weighted support threshold and a confidence threshold is assigned to measure the strength of the association rules produced.

The weight of a k-itemset, X, is given by:

$$(\Sigma w_i) sup(X) \tag{4.1}$$

Here a k-itemset, X, is considered a frequent itemset if its weighted support is greater than the user-defined minimum weighted support (wminsup) threshold.

$$(\Sigma w_i) sup(X) \geq wminsup \tag{4.2}$$

The weighted support of a rule $X \longrightarrow Y$ is:

$$(\Sigma w_i) sup(XY) \tag{4.3}$$

Algorithm 3 Weighted Association Rule Mining (WARM)

```
Input: Transaction database D, weighted minimum support $wminsup$, universe of items $I$
$k \leftarrow 1$
$L_k \leftarrow \{\{i\}|i \in I, weight(c) * support(c) > wminsup\}$
while ($|L_k| > 0$) do
  $k \leftarrow k + 1$
  $C_k \leftarrow \{x \cup y|x, y \in L_{k-1}, |x \cap y| = k - 2\}$
  $L_k \leftarrow \{c|c \in C_k, weight(c) * support(c) > wminsup\}$
end while
$L_k \leftarrow \bigcup_k L_k$
Output: Weighted frequent itemsets
```

An association rule $X \longrightarrow Y$ is considered interesting if $X \cup Y$ is a frequent itemset and the confidence of the rule is greater than or equal to a minimum confidence threshold. A general weighted association rule mining algorithm 3 is shown above. The algorithm requires a weighted minimum support to be provided. In this algorithm L_k represents the frequent itemsets also known as the large itemsets and C_k represents the candidate itemsets. The mining process begins by assigning/calculating the weight of all unique items in the data. The length-2 candidate itemsets are then generated and evaluated using frequent items whose weighted support is superior to the *wminsup* threshold. Itemsets whose weighted support exceeds the weighted minimum support are considered frequent itemsets and will be used to build candidate itemsets of length-3 included in the rule generation phase and so on until no frequent rules can be generated. Thus it can be seen that item weighting enables items with relatively low support to be considered interesting and inversely, items which have relatively high support may turn out to be uninteresting (not frequent). This adds a new dimension to the classical association rule mining process and enables rules with high weights in their rule terms to be ranked ahead of others, thus reducing the burden on the end user in sifting through and identifying rules that are of the greatest value.

4.5 Adapting the WARM problem for temporal sequences

Although the original motivation behind the association rule algorithms was to analyze market basket transaction data, they have been extensively used across a wide range of different application areas which include bioinformatics, text mining, web usage mining, telecommunications, medical disease diagnosis, etc. However, almost all ARM algorithms were conceived for transaction data and not for temporal sequences. (Mannila et al., 1997) was the first to propose an apriori-based approach for ARM mining in temporal sequences called WINEPI consisting of the same support-confidence framework having segmented the sequence to a set of transactions using a sliding window of predefined length. The WINEPI algorithm, as Apriori, was conceived to mine frequent patterns and not rare ones. In this section, we first present the WINEPI algorithm as defined in (Mannila et al., 1997). We then propose adapting WINEPI to the weighted association rule mining problem by integrating weights into the mining process. We call this approach *weighted WINEPI*. The aim is thus to direct the mining focus to significant episode rules which take into account infrequent events. This will give better integrity to results and help decision makers in their post-analysis task.

4.5.1 Preliminary definitions

In this section, we follow the standard notions of event sequence, episode and support previously defined in sections 4.2 and 4.4. Consider the input as a sequence of events, where each event is expressed by a unique numerical code and an associated time of occurrence.

Definition 1. ***(operator $\sqsubset$)*** Let α and β be two ordered sequences of events. α is a subsequence of β, denoted $\alpha \sqsubseteq \beta$ iff α can be obtained by removing the last element (event) of β.

Definition 2. ***(Episode)*** We define an episode as an ordered collection of events occurring repetitively. Formally, an episode α is a k-tuple of the form $\alpha = \langle R_1, R_2, ..., R_k \rangle$ with $e_i \in E$ for all $i \in \{1, ..., k\}$. In the following, we will use the notation $R_1 \longrightarrow R_2 \longrightarrow ... \longrightarrow R_k$ to denote the episode $\langle R_1, R_2, ..., R_k \rangle$, where $\longrightarrow$ is read as 'followed by'. We denote the empty episode by ϕ.

Definition 3. ***(Size of an episode)*** Let $\alpha = \langle R_1, R_2, ..., R_k \rangle$ be an episode. The size of α is denoted by $|\alpha|$ is equal to the number of elements of the tuple α, i.e. $|\alpha|$=k.

Definition 4. ***(Suffix and Prefix of an episode)*** Let $\alpha = \langle R_1, R_2, ..., R_k\rangle$ be an episode. The suffix of α denotes an episode composed only of the last element of the tuple α, i.e, suffix(α)=$\langle e_k\rangle$. The prefix of α is the order of events except the last event, that is episode $\langle R_1, R_2, ..., R_{k-1}\rangle$. We denote it as prefix(α).

Definition 5. ***(episode rule)*** Let α and β be episodes such that prefix(β)=α. An episode rule built on α and β is the expression $\alpha \Longrightarrow$ suffix(β).

For example, if $\alpha = R_1 \longrightarrow R_2$ and $\beta = R_1 \longrightarrow R_2 \longrightarrow R_3$; the corresponding episode rule is denoted $R_1 \longrightarrow R_2 \Longrightarrow R_3$. In this work, the episode rules are restricted to those having a single event type in their right hand side (suffix), a target event.

4.5.2 WINEPI algorithm

The WINEPI algorithm was developed by (Mannila et al., 1997) to tackle the problem of mining frequent episode rules in temporal sequences. To be considered interesting, the events of an episode must occur close enough in time. The user defines how close is close enough by giving the width of the time window within which the episode must occur. A window is defined as a slice of an event sequence. Hence, an event sequence can be defined as a sequence of partially overlapping windows (see Figure 4.3).

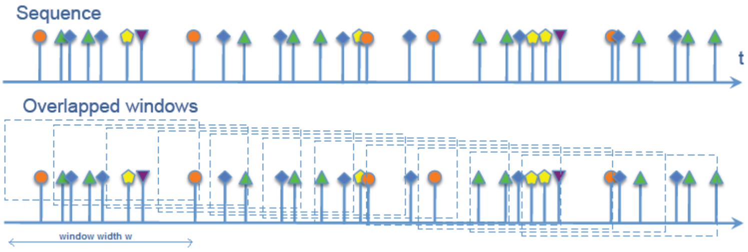


Figure 4.3: Overlapping windows

Formally, a window on an event sequence $S = (S, T_s, T_e)$ is a subsequence $win = (w, t_s, t_e)$, where $t_s < T_e$ and $t_e > T_s$, consisting of all events (R, t) from S where $t_s \leq t \leq t_e$. The time span $t_e - t_s$ is called the *width* of the window w. By definition, the first and the last windows on a sequence extend outside the sequence, so that the first window contains only the first time point(event) of the sequence and the last window contains only the last event. With this definition an event close to either end of a sequence is observed in equally many windows as an event in the middle of the sequence. Given an event sequence S and window width w, the number of possible windows of width w in S is $T_e - T_s + w - 1$, and the set of all windows is denoted by $W(S, w)$.

The WINEPI algorithm uses an apriori-based breadth-first strategy to discover all frequent episodes. Once these episodes are known, they can be used to derive episodes rules. The search starts from episodes with only one event. On each level the algorithm first computes a collection of candidate episodes, and then checks their frequencies from the event sequence.

The frequency of an episode (similar to support in transaction data mining) is defined as the fraction of windows in which the episode occurs. More formally, given an event sequence S and a window width w, the frequency of an episode α in S is

$$fr(\alpha, S, w) = \frac{|w \in W(S, w)|\alpha \ occurs \ in \ w|}{|W(S, win)|} \tag{4.4}$$

Given a frequency threshold min_{fr}, α is frequent if fr$(\alpha, S, w) \geq min_{fr}$.

In the candidate generation episode, Apriori's downward closure property is used to enhance the pruning process and decrease computational time. This property is defined by the following.

Definition 6. ***(Downward Closure Property)*** If an episode α is frequent in an event sequence S, then all subepisodes $\beta \sqsubset \alpha$ are frequent.

Once the frequent episodes are identified, they are assessed to obtain rules that describe connections between events in the given event sequence. Similar to the Apriori method, the measure used to assess these episodes is the *confidence*. Formally, for an episode rule $\alpha \Longrightarrow \beta$, where α and β are episodes such that $\alpha \sqsubset \beta$, the confidence can be defined as:

$$conf(\alpha \Longrightarrow \beta) = \frac{freq(\beta)}{freq(\alpha)} \tag{4.5}$$

which can be interpreted as the conditional probability of β occurring in a window, given that α occurs. The reason why confidence is calculated is because it shows the connections between events more clearly than frequency alone. It is for the same reason that the mining process does not stop upon the discovery of frequent episodes and continues towards deriving episode rules.

4.5.3 Weighted WINEPI algorithm

In the proposed Weighted WINEPI algorithm, the event sequence is considered as a sequence of partially overlapping windows, similar to the WINEPI algorithm. However, the sliding window process was modified to be adapted for sequences where timestamps are not necessarily integers. Consider an event sequence S and the first event occurring in that sequence to be (R_1, t_1). Unlike the WINEPI algorithm where the first window contains the R_1 only, the first window in the weighted WINEPI algorithm contains events occurring between $t_1 + 1 - w$ and $t_1 + 1$ hours, the second window consists of events between $t_1 + 2 - w$ and $t_1 + 2$ and so on. This way, each hour in the sequence is scanned by the same number of windows. Figure 4.4 gives an example of this concept. In the following example, window width $w = 2$ hours. The corresponding windows are given in table 4.1.

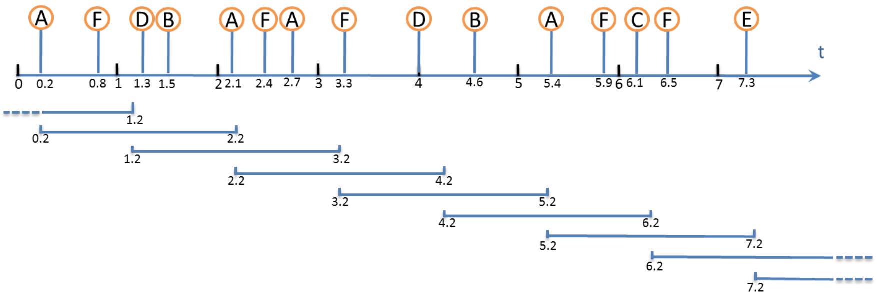


Figure 4.4: An illustration of the overlapping window strategy adopted in the weighted WINEPI algorithm

Table 4.1: Windows database (left), Unique events and their respective neighbours (right)

Window ID	Events
100	A, F
200	A, F, D, B, A
300	D, B, A, F, A
400	F, A, F, D, D
500	F, D, B
600	B, A, F, C
700	A, F, C, F
800	F, E
900	E

Event	Related unique events
A	B, C, D, F
B	A, C, D, F
C	A, B, F
D	A, B, F
E	F
F	A, B, C, D, E

The support of episodes is replaced by weighted support with weights based on the valency model introduced in (Koh et al., 2010), with a few differences in order to give the algorithm the capacity to scrutinize and discover relations between rare events.

Considering the set of windows as a set of transactions, the weighted WINEPI algorithm calculates the weighted support of all events (episodes of length-1) to deduce those who are the most significant (not necessarily frequent). These events will be then used to generate length-2 candidate episodes. The weighted support of these episodes is then computed and so on.

Given an event sequence S and a window width w, the weighted support of an episode α in S is

$$weightedsup(\alpha, S, w) = w_\alpha * \frac{|w \in W(S,w)|\alpha \; occurs \; in \; w|}{|W(S,w)|} \tag{4.6}$$

An episode is considered interesting if its weighted support is superior to a predefined *wminsup* threshold. Once the interesting episodes are discovered from the sequence, they can be evaluated using interestingness measaures such as the confidence which is used by both Apriori and WINEPI in order to derive episode rules or others such as Recall and Precision.

4.5.4 Calculating weights using Valency Model

The Valency model, as proposed by (Koh et al., 2010) is based on the notion that an item should be weighted based on how strong his connections to other items as well are, as well as on the number of items that it is connected to. Two items are considered to be connected (and are called neighbours) if they have occurred together in at least one transaction. Items which appear often together when compared to their individual support have a high degree of connectivity and are thus given a more important weight. In addition to that, an item that is contained in a small clique of items is considered to have a high degree of purity and is given a proportionally higher weight as well. Given two items i and k that co-occur together $count(ik)$ times, the connectivity between i and k is defined as:

$$c(k,i) = \frac{count(ki)}{count(k)} \tag{4.7}$$

where $count(k)$ is the support of item k. Given an item k which is connected to n items in its neighberhood, the connectivity, c_k is defined as:

$$c_k = \sum_{i=1}^{n} \frac{count(ik)}{count(k)} \tag{4.8}$$

The higher the connectivity between item k and its neighbours, the higher its weight should be. However, it cannot be considered as the main factor in the weighting scheme because a very frequent item existing in most transactions would naturally have a high connectivity without necessarily having a significance to its neighbours. With this in mind, the purity measure is introduced to define the extent to which an item could be said to be distinctive. The smaller the number of items that a given item interacts with, the more interesting it would be and the higher the purity value it will have and vice versa. An item would not be allowed to acquire a high weight unless it had a high purity value, regardless of its connectivity. The role of purity is thus to ensure that only items with high discriminating power could be assigned a high weight. Formally, the purity of a given item k is defined as:

$$P_k = 1 - \frac{log_2(n_k)}{log_2(U)}, \quad 0 \leq P_k \leq 1 \tag{4.9}$$

where U represents the number of unique items in the dataset and n_k the number of unique items which co-occur with item k (number of neighbours).

The valency of an item k, denoted by v_k is defined as the linear combination of both the purity and connectivity components:

$$v_k = \beta.p_k + (1-\beta).\sum_{i=1}^{n_k} c(k,i).p_i \tag{4.10}$$

where β is a parameter which measures the relative contribution of the item k over the items that it is connected to in the database and is given by:

$$\beta = \frac{1}{n_k}\sum_{i=1}^{n_k} c(k,i) = \frac{c_k}{n_k} \tag{4.11}$$

The valency of an item is taken as its weight w_α. The two main advantages of this model are its automated mechanism for weight fitting and its simplicity. It produces rules with greater diversity than an un-weighted association rule miner.

4.5.5 Adapting Weighted WINEPI to include infrequent events

Weighted WINEPI, as explained above, is efficient for discovering frequent episodes having a high support. Indeed, episodes between infrequent events have higher chances to be discovered using Weighted WINEPI than with the original WINEPI but still, the effect of their high weight value may be suppressed by their low support, and thus many will be pruned out. To overcome this problem, it is important to identify distinctive events from the beginning by excluding from the mining process not those with low weighted support but with low distinctivity, which can be defined by the valency, and then using the remaining events to create length-2 candidate episodes that will be scrutinized by their weighted support to keep the most interesting ones and so on.

In order to do that, we replace the minimum weight threshold *wminsup* by a threshold called *tp* (signifying top percentage). The weighted WINEPI will be applied as follows. First, Valency is calculated for all events in the sequence. The tp % events with the highest valency value are kept for length-2 candidate generation, the others are pruned out. Length-2 candidate episodes are generated and evaluated by means of weighted support to derive length-2 episodes and then generate length-3 candidates and so on.

Since Tilt and Traction "Driver Action High" events which are our target events in this thesis are very rare and thus have a very low support, there is a high risk that they will not be included in the mining process, even with Weighted WINEPI. For this reason we propose *Oriented Weighted WINEPI*, an approach that will be explained in the following section 4.5.6.

4.5.6 Adapting Weighted WINEPI to focus on target events: Oriented Weighted WINEPI

In this section, we propose *Oriented weighted WINEPI*, which can be considered as a constraint-based sequential episode mining algorithm. The aim is to mine serial episodes α leading to a specific target event T. For this reason, the distinctivity of events with respect to this specific target event should be evaluated. In order to do so, we propose a measure that we call: *cruciality* $cr(A,T)$, that will be injected in the weighted WINEPI process.

Definition 7. ***(Cruciality measure)*** Consider an event sequence S where $S = \{(R_1,t_1),(R_2,t_2),...,(R_n,t_n)\}$. $E = \{R_1,R_2,...,R_n\}$ is the set of event types occurring in S. Let A and T be two events in S where T is a target event. The cruciality of A with respect to T is equal to:

$$cr(A,T) = P_T(A \longrightarrow T) * (1 - \frac{log_2(N_A)}{log_2(|E|)}) \tag{4.12}$$

where $0 \leq cr(A,T) \leq 1$ $|E|$ is the number of distinct event types in the data sequence and N_A represents the number of distinct events which co-occur with A (number of neighbours). $P_T(A \longrightarrow T)$ is the fraction of windows containing events A and T with T succeeding A relative to the number of windows in which event T occurs. The value of the cruciality can be also defined by $cr(A,T) = Recall(A \longrightarrow T) * Purity(A)$. This measure expresses the particular importance of an event A for a target event T. The recall is the fraction of target event T that was preceded by an event A within a co-occurrence distance w equivalent to the window width. The purity states how much importance should the recall value be accorded. Since, for example, an event with high number of neighbours occurs frequently all over the sequence, and hence would unlikely have a particular specificity towards the target event. Thus its high recall value is suppressed by its low purity, and vice versa, in cases where an event has few number of neighbours and an acceptable recall value, it might be interesting to conserve it.

Oriented Weighted Winepi consists of integrating Cruciality in the mining process by using it as a filter measure for length-1 candidate events in order to conserve only events which are significant to the target event. Those events would build up length-2 candidate that are then evaluated using valency-based weighted support and so on.

4.5.7 Experiments on synthetic data

We test the previously described approach on simulated data sequences of variable lengths. The length of the injected pattern varies from 2 to 4 and so does the number of times it was injected. In each experiment, 100 sequences are generated using the same protocol. Four approaches are then applied on these sequences. Experimentation I in tables 4.2, 4.3 and 4.4 shows the mean number of discovered event couples in 100 generations. This reflects the false positive rate of each approach. Since we are interested in serial episodes (directed patterns), the expected number of discovered couples is 1. Experimentation II gives the number of generations where the injected pattern was discovered. This reflects the predictive capability of each approach.

The four tested approaches are:

1. WINEPI algorithm ($minsup = 1\%$)
2. Weighted WINEPI 1 (WW1): using valency as weight and a *wminsup* threshold ($wminsup = 0.1\%$)
3. Weighted WINEPI 2 (WW2): using a *tp* threshold (tp=25%) on weighted support
4. Weighted WINEPI 3 (WW3): using *tp* threshold on valency for length-1 and weighted support (w-support) for length-2 and above.

Similar to the algorithms discussed in the previous chapters, the value of the w, the maximum co-occurrence distance (scanning window width) was set to 5 hours.

Table 4.2: Results of WINEPI, WW1, WW2 and WW3 on synthetic data with length-2 injected patterns, $w = 5h$ and $tp = 25\%$

I - Mean number of discovered patterns in 100 generations					
Sequence Length (hours)	No. patterns	WINEPI	WW1	WW2	WW3
1000	10	61	65	1	**1**
	50	66	66	1	**1**
	100	66	66	1	**1**
	200	66	66	1	**1**
5000	10	45	46	1	**1**
	50	64	66	1	**1**
	100	66	66	1	**1**
	200	66	66	1	**1**
10000	10	45	45	1	**1**
	50	46	63	1	**1**
	100	66	65	1	**1**
	200	66	66	1	**1**
20000	10	45	45	1	**1**
	50	45	45	1	**1**
	100	45	45	1	**1**
	200	46	56	1	**1**

II - Number of generations where the injected pattern was discovered					
Sequence Length (hours)	No. patterns	WINEPI	WW1	WW2	WW3
1000	10	100	100	0	**100**
	50	100	100	0	**100**
	100	100	100	100	**100**
	200	100	100	100	**100**
5000	10	0	51	0	**100**
	50	100	100	0	**100**
	100	100	100	0	**100**
	200	100	100	0	**100**
10000	10	0	0	0	**100**
	50	100	100	0	**100**
	100	100	100	0	**100**
	200	100	100	0	**100**
20000	10	0	0	0	**100**
	50	0	0	0	**100**
	100	0	33	0	**100**
	200	100	100	0	**100**

Table 4.3: Results of WINEPI, WW1, WW2 and WW3 on synthetic data with length-3 injected patterns, $w = 5h$ and $tp = 25\%$

I - Mean number of discovered patterns in 100 generations					
Sequence Length (hours)	No. patterns	WINEPI	WW1	WW2	WW3
1000	10	84	412	1	**1**
	50	200	925	1	**1**
	100	440	1269,5	1	**1**
	200	799	1410	1	**0**
5000	10	18	396	1	**1**
	50	21	476	1	**1**
	100	45,5	583	1	**1**
	200	92,5	816,5	1	**1**
10000	10	14	445	0	**1**
	50	9	429	0	**1**
	100	12	465,5	0,5	**1**
	200	30	539	1	**1**
50000	10	5	459	0	**1**
	50	3	391	0	**1**
	100	2	444,5	0	**1**
	200	6	441	0	**1**

II - Number of generations where the injected pattern was discovered					
Sequence Length (hours)	No. patterns	WINEPI	WW1	WW2	WW3
1000	10	100	100	0	**100**
	50	100	100	0	**100**
	100	100	100	98	**99**
	200	100	100	100	**7**
5000	10	0	21	0	**100**
	50	100	100	0	**100**
	100	100	100	0	**100**
	200	100	100	0	**100**
10000	10	0	0	0	**100**
	50	100	100	0	**100**
	100	100	100	0	**100**
	200	100	100	0	**100**
50000	10	0	0	0	**100**
	50	0	0	0	**100**
	100	0	3	0	**100**
	200	100	100	0	**100**

Table 4.4: Results of WINEPI, WW1, WW2 and WW3 on synthetic data with length-4 injected patterns, $w = 5h$ and $tp = 25\%$

I - Mean number of discovered patterns in 100 generations

Sequence Length (hours)	No. patterns	WINEPI	WW1	WW2	**WW3**
1000	10	5	298	0	**1**
	50	131	1920	0	**1**
	100	510	4452	1	**1**
	200	1790	7994	1	**1**
5000	10	0	1	0	**1**
	50	1	76	0	**1**
	100	26	162	0	**1**
	200	100	615	0	**1**
10000	10	0	0	0	**1**
	50	1	13	0	**1**
	100	1	76	0	**1**
	200	23	139	0	**1**
20000	10	0	0	0	**1**
	50	0	0	0	**1**
	100	0	0	0	**1**
	200	1	3	0	**1**

II - Number of generations where the injected pattern was discovered

Sequence Length (hours)	No. patterns	WINEPI	WW1	WW2	WW3
1000	10	100	100	0	**100**
	50	100	100	0	**100**
	100	100	100	98	**100**
	200	100	100	100	**75**
5000	10	0	41	0	**100**
	50	100	100	0	**100**
	100	100	100	0	**100**
	200	100	100	0	**100**
10000	10	0	1	0	**100**
	50	100	100	0	**100**
	100	100	100	0	**100**
	200	100	100	0	**100**
20000	10	0	0	0	**100**
	50	0	0	0	**100**
	100	0	27	0	**100**
	200	100	100	0	**100**

Knowing that the number of patterns that are expected to be discovered is 1 since we have injected one type of patterns in the sequence, results in tables 4.2, 4.3 and 4.4 show that using *tp* threshold on valency for length-1 and weighted support (w-support) for length-2 and above (WW3) yields better results than the classical WINEPI, the wminsup-based weighted WINEPI (WW1) and weighted WINEPI with **tp** on w-support (WW2) in both false positive rate and predictive ability. As the injected pattern becomes more rare (example: injected 10 times in sequences of length 20000 hours), the other approaches lose their capacity to predict the injected pattern. In addition to that, the WINEPI algorithm as well as WW1 and WW2 lead to a rather high number of false positives. The results comply for injected patterns of length 3 and 4 as well.

In conclusion, Weighted WINEPI with a *tp* threshold on valency for length-1 and weighted support (w-support) for length-2 and above is an efficient approach for mining patterns between infrequent events in long sequences. In the next section, we apply this approach as well as WINEPI on the real TrainTracer data sequences.

4.5.8 Experiments on real data

Results on traintracer data are presented in Table 4.5 below. The 2 most performant approaches among the 4 tested on generated data were launched on real data: WINEPI and WW3.

The WINEPI algorithm with a *minsup* threshold = 5% and 1% did not discover patterns longer than length-2. None of these patterns ended with a target event. As for WW3, with tp = 5%, mining was stopped at length-4. Analyzing the discovered length-4, length-3 and length-2 patterns have showed that none of them ended with a target event as well. Target events are very rare that even if they pass the first level because of their high valency value, the weighted support calculated to evaluate length-2 candidates consisting of these events will be very low and thus will be pruned out from the mining process. This was the main motivation behind proposing the "Oriented Weighted Winepi" approach proposed in 4.5.6.

Table 4.5: Results of WINEPI with *minsup* = 5% and WW3 with tp = 5% on Train-Tracer data

	Episodes discovered		
Approach	Length-2	Length-3	Length-4
WINEPI	4443	0	0
WW3	2453	846	235

We now apply the Oriented Weighted Winepi. With this approach we are sure to obtain rules leading to a specific target event. In order to evaluate these rules, we use the precision and recall values. Within a Weighted WINEPI window-sliced-sequence context, we define these two measures by the following: Let α and β be episodes such that prefix(β)=α and $\alpha \longrightarrow \beta$ is an episode rule.

$$Precision = \frac{|w \in W(s, win)|\beta \in w|}{|w \in W(s, win)|\alpha \in w|} \tag{4.13}$$

$$Recall = \frac{|w \in W(s, win)|\beta \in w|}{|w \in W(s, win)|suffix(\beta) \in w|} \tag{4.14}$$

In other words, the precision is the fraction of windows containing the prefix and the suffix of the discovered episode (order of events taken into account) over the total number of windows containing the prefix. It expresses the fraction of the prefix episodes that have been succeeded by a target event. If an episode occurs multiple times in a window, it is only considered once. The recall however is the fraction of windows containing the prefix and the suffix (target event) of the discovered episode (order of events taken into account) over the total number of windows containing the suffix. It expresses the fraction of occurrences of a target event that have been preceded (predicted) by the prefix. If a window contains multiple occurrences of a target event, only one occurrence is considered.

Due to confidentiality agreements, it is not possible to provide examples of the discovered rules.

4.6 Conclusion

In this chapter, we have tackled the problem of Weighted Association rule mining. We proposed Weighted WINEPI, an episode rule mining algorithm based on the fusion of both the WINEPI frequent episode mining algorithm (Mannila et al., 1997) consisting of a sliding window transforming a temporal data sequence into a series of overlapped windows as well as the valency model proposed in (Koh et al., 2010) that we specially adapted for the problem of data sequences instead of transaction data. The Weighted WINEPI algorithm was modified to be able to integrate more efficiently infrequent events into the mining process. Several tests on synthetic data have showed that Weighted WINEPI outperforms the classical WINEPI algorithm in detecting patterns between infrequent events. However, due to the particular applicative problematic tackled in this thesis, where the aim is to discover patterns leading to rare target events in the TrainTracer data, we also proposed a constraint-based approach derived from Weighted WINEPI that we called Oriented Weighted WINEPI. The weighted WINEPI algorithm uses a measure called cruciality that we propose as a primary filter to conserve events that are useful for the target event in the mining process and neglect the others. The Oriented Weighted WINEPI, when applied on TrainTracer data discovered episode rules with acceptable recall and precision. Due to the lack of ground truth knowledge on whether rules actually exist in the data extract under disposal or not, the physical significance of the obtained rules needs to be analyzed by technical experts.

Chapter 5

Pattern recognition approaches for predicting target events

Contents

In this chapter, we adopt a non-temporal approach to tackle the problem of the prediction of infrequent target events in sequences based on pattern recognition techniques. First, we propose a methodology to transform the data sequence into a set of labelled observations constituting the dataset upon which various classification methods are applied: K-Nearest Neighbours, Naive Bayes, Support vector machine and Artifical Neural Networks. An attribute selection approach based on hypothesis testing is then proposed to decrease the number of attributes by selecting those which are most significant and contributive to the classification performance.

This chapter is organized as follows. We first introduce the general principle of pattern recognition in Section 5.1 and then present briefly the four main classifiers we are going to use in Section 5.2: K-Nearest Neighbours in 5.2.1, Naive Bayes in 5.2.2, Support Vector Machines in 5.2.3 and Artificial Neural Networks in 5.2.4. In Section 5.3 we propose a methodology to transform the temporal data sequence into a labelled dataset consisting of *good* and *bad* observations. Following that in Section 5.4, we introduce an attribute-selection methodology based on hypothesis testing in order to select the most informative attributes and the most significant to the classification process. Finally, the performed experiments and the obtained results are detailed and discussed in Section 5.5 before concluding in Section 5.6.

5.1 Pattern Recognition

5.1.1 Introduction

The monitoring of the operating state of industrial systems, such as railway rolling stock and infrastructure, robots, sorting machines, etc. enables the enhancement of their productivity and availability which reflects positively by a decrease in production costs. When a failure is detected, the monitoring system executes a corrective procedure to restore normal functionality. Several methods can be used to realize the system monitoring task. The choice of the method depends on many factors such as the dynamics of the system (discrete, continuous, hybrid) and its complexity, the environment constraints, the representation of information (quantitative or/and qualitative) and the information available on the system (structural, analytic, heuristic knowledge, etc.).

When the existing knowledge about the physical behavior of a process is not sufficient to construct an analytic model and when only the measures of its operating state are available, pattern recognition methods become interesting to monitor the operating states of systems, i.e. fault diagnosis. In such case, the system is considered as a black box, where no mathematical equations are necessary to model its functioning. The methods use exclusively a set of measures and/or heuristic knowledge of the functioning of a process in order to build a classification/regression model able to map the observation space into a decision space. The results of the classification depend on the method as it is, the existing knowledge, the attributes characterizing the system and the quality of data (Bishop, 2006; Han et al., 2006; Hastie et al., 2003; Tan et al., 2005; Ye, 2003).

5.1.2 Principle

Generally, a pattern recognition approach is used to build a diagnosis tool aiming to classify new data using a classifier that generates a membership function for each class. These methods can mainly be divided into two categories: parametric methods and non parametric methods. Parametric methods consider the learning dataset as independant data, abiding the same probability law, similar to bayesian classifiers. Non parametric methods generate the membership function of classifiers, either by estimating the conditional probability density function for each class, like in the Parzen window method and K-Nearest Neighbours, or by constructing, via learning, the decision regions, such as in Neural Networks and SVM.

Pattern recognition can be realized in two phases: learning from known data (training phase) and the classification of new ones (test phase). In addition to these two phases, a preprocessing step is used to find the minimal set of the most informative attributes, i.e, an appropriate representation space. The set of major steps of a pattern recognition approach is presented in Figure 5.1 below.

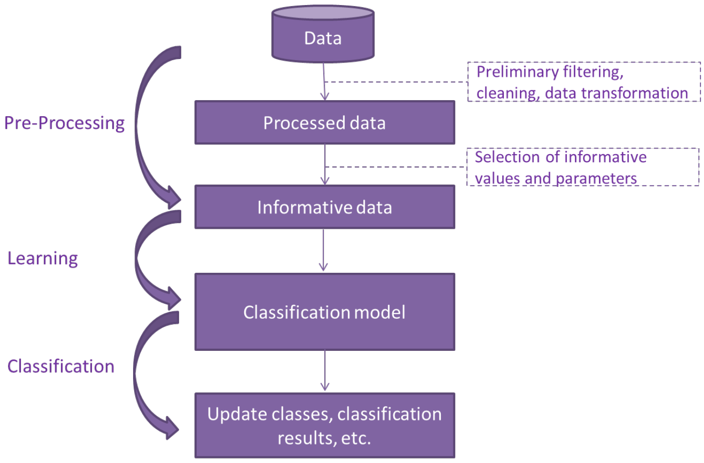

Data
Preliminary filtering, cleaning, data transformation
Pre-Processing
Processed data
Selection of informative values and parameters
Informative data
Learning
Classification model
Classification
Update classes, classification results, etc.

Figure 5.1: Pattern recognition process: principal steps

5.1.3 Preprocessing of data

The first phase of preprocessing mainly consists of cleaning and filtering data redundancies and errors as well as data concatenation in certain cases when data is collected from several sources (like in our applicative example with sequences from various trains). In the second phase of preprocessing, the most discriminative attributes are selected. Two types of method exist to establish the representation space, namely parameter selection methods which choose a subset of attributes among the original ones that are most informative. Second, parameter extraction methods which create a subset of new attributes by combining the original ones. To realize these phases of attribute selection/extraction, data analysis techniques can be used such as Principal Component Analysis (PCA) (Jolliffe, 1986), Fisher's linear discriminant (Fisher, 1936), Multidimensional scaling (Sammon, 1969), Wrapper techniques (Uncu and Turksen, 2007), etc. The more discriminative are the attributes, the better would be the results of the classification algorithms. The set of attributes provided by these methods represent the attributes allowing to characterize each form. Each class is associated to an operational mode (in our situation: normal functioning or failure). These forms, with their class labels, constitute the learning set. They are represented by d characteristics, or attributes, which enables to see them as vectors of d dimensions, i.e., points in the representation space.

Depending on the application field, data collected on a system can be static or dynamic when their characteristics evolve with time. Static data is represented by a point in the representation space while dynamic data consists of a suppelementary dimension which is time (Angstenberger, 2000). Similarly, classes (operating modes) can be static or dynamic. Static classes are represented by restraint zones of similar data in the representation space. Numerous pattern recognition methods exist to treat static data. For example, K-Nearest Neighbours (Dasarathy, 1991), Support Vector Machine (Vapnik, 1995), Bayesian methods (Freedman and Spiegelhalter, 1989), Principal component analysis (PCA) (Jolliffe, 1986), Fuzzy Pattern Matching (Cayrol et al., 1982; Dubois and Prade, 2012), Fuzzy C-means (FCM) (Bezdek, 1981), as well as numerous derivatives of these methods. In certain cases, systems are in constant evolution between their different operational modes. We hence speak of evolutive systems (Angelov et al., 2010; Lughofer and Angelov, 2009) upon which it is necessary to use dynamic classification methods (Kifer et al., 2004; Tsymbal, 2004), which will not be handled in this thesis. In this thesis, we tackle the problem of static learning of static data.

5.1.4 Learning and classification

Once the representation space is defined, the next step consists of building a classifier. Depending on the apriori information available on the system, three types of pattern recognition methods can be used: Supervised methods, Non-supervised and Semi-supervised. When the data are labelled (classes are known for each sample (observation) in the dataset), learning is performed within a supervised framework (Therrien, 1989). The training set is used to construct a classifier which separates in the best possible way the different known classes, in the aim to minimize the classification error (Cristianini and Shawe-Taylor, 2000). Once the training is performed, an inferring function allows the inference of a new data point to a class. In cases where no information is available on classes of the system (data points are unlabelled), the learning is considered to be unsupervised (Bezdek, 1981; Bishop, 2006; Frigui and Krishnapuram, 1996, 1997; Gancho, 2009; Hastie et al., 2003). The methods are mainly based on similarity functions. When data points with similar characteristics appear, they are assigned to the same class and vice versa when their characteristics are different a new class is created by the classifier. The third type of learning is semi supervised (Cozman et al., 2003; Gabrys and Bargiela, 2000; Stephen et al., 2000) which uses known information, i.e, known classes and labelled data, to estimate the characteristics of classes and their inferreing functions while using unsupervised learning to detect new classes and learn their inferring functions.

In this chapter, we use four different supervised classification methods: k-Nearest Neighbours, Naive Bayes, Support Vector Machines and Artifical Neural Networks, all which are tackled in the following section.

5.2 Supervised Learning Approaches

5.2.1 K-Nearest Neighbours Classifier

The K-Nearest Neighbour method first appeared in the early 1950s but did not gain popularity until the 1960s with the advancement of computing power. This made it possible for it to be applied on large training sets. The method has been widely used in the area of pattern recognition since then. The main principle behind Nearest-Neighbour classifiers is learning by analogy, that is, by comparing a given test data point with training data points that resemble it. The training data points are described by d attributes. Hence, all of the training data points are stored in a d-dimensional pattern space. When given an unknown data point for testing, a K-Nearest Neighbours classifier searches the pattern space for the k training data points that are closest to this unknown data point. These k training data points are its k "nearest neighbours". How close a data point X_2 to a test point X_1 is determined by calculating a distance defined by the following equation:

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{d}(x_{1i} - x_{2i})^2} \tag{5.1}$$

After the k nearest neighbours are discovered, the unknown data point is assigned to the most common class among its neighbours.

The optimal value of k, the number of neighbours, can be determined experimentally. Starting with $k = 1$, a test set is used to estimate the error rate of the classifier. This process can be repeated each time by incrementing k to allow more neighbours. The value of k which will lead to the minimum error rate is then selected. One major issue about nearest-neighbour classifiers is that the choice of a distance metric can be critical. Using a distance metric which gives equal importance to each attribute can lead to poor accuracy, in situations where some attributes are noisy or irrelevant. The method, however, has been modified to incorporate attribute weighting and the pruning of noisy data points. The Manhattan (city block) distance, or other distance measurements, may also be used (Bishop, 2006; Han et al., 2006).

5.2.2 Naive Bayes

A Naive Bayes classifier (Neapolitan et al., 2004; Nielsen and Jensen, 2009) is a statistical classification model that can predict class membership probabilities, i.e, the probability that a given observation belongs to a particular class. It assumes that the effect of an attribute value on a given class is independent of the values of other attributes. This assumption is called class conditional independence (Han et al., 2006). Naive Bayesian classifiers are based on the Bayes theorem. This latter calculates the posterior probability of an attribute vector x to belong to a class C_i, using the following equation:

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)} = \frac{P(x|C_i)P(C_i)}{\Sigma_j P(x|C_j)P(C_j)} \tag{5.2}$$

where $P(C_i)$ is the *à priori* probability of a class C_i, P(x) the probability to observe characteristic vector x and $P(x|C_i)$ the probability to observe a characteristic vector x knowing that its corresponding class is C_i. In practice, we are interested only in the numerator part. The denominator of Bayes's theorem does not depend on C_i. The probabilities $P(C_i)$ of each class as well the distributions $P(x|C_i)$ should be estimated using a training dataset. An attribute vector x of an observation is assigned to the class which maximizes most the posterior probability, that is:

$$\forall j \neq i, P(C_i|x) > P(C_j|x) \tag{5.3}$$

With the Naive Bayes classifier, the input characteristics are supposed to be independent from each other. Considering this hypothesis, and considering n to be the number of observations (data points), it is possible to express the likelihood function of each class as the product of n simple probability density functions. These functions are generally expressed using the normal unidimensional distributions.

5.2.3 Support Vector Machines

In this section, a general presentation of Support Vector Machines (SVM) is given. SVM is a well-known method for both linear and nonlinear classification problems. Although the groundwork for this method has been mentioned in (Vapnik, 1979), but the first paper was presented in the early ninties (Boser et al., 1992). SVMs are reputable for their high accuracy with a high capability to model complex nonlinear decision boundaries and lower sensibility to overfitting than other methods. SVMs can be used for both prediction and classification and have been applied extensively in various areas such as object and voice recognition, handwriting character recognition, etc. In general, a support vector machine works as follows. It uses nonlinear mapping to transform the original data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane, i.e. a decision boundary separating the data points (observations) of one class from another. This hyperplane is found using support vectors (training points) and margins defined by support vectors (Han et al., 2006). We will briefly investigate these concepts further below within two cases: when data are linearly separable, and when data are not linearly separable.

Case 1: Data are Linearly Separable

We first consider a two-class problem where the classes are linearly separable. Let the training data set D consist of l training examples given as (X_i, y_i), $i = 1, ..., l$, where each example has d inputs, i.e, $x_{i,j} \in R^d$, and a class label y_i with one of two values ($y_i \in \{-1, +1\}$). Since there is no doubt an infinite number of separating hyperplanes in R^d which can be drawn to separate data points corresponding to different classes, the aim is to find the best one, that is, the one with the minimum classification error when tested on previously unseen data points. SVM tackles this problem by searching for the maximum marginal hyperplane (See Figure 5.2), i.e., the hyperplane with the largest margin, which is logically expected to have the best accuracy.

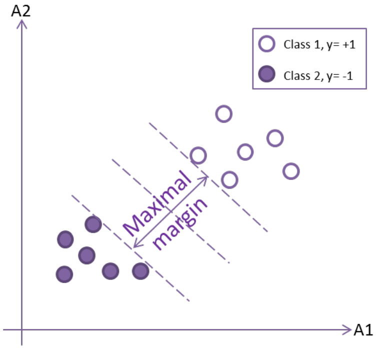


Figure 5.2: An example of a 2-class, 2-attribute (2-D) training data. Within an infinite number of possible separating hyperplanes, the one with the largest marginal width is considered to be the best

All hyperplanes in R^d are parametrized by a vector W and a constant b, and can be written as

$$W.X + b = 0 \tag{5.4}$$

where $W = \{w_1, w_2, ..., w_n\}$ is a vector orthogonal to the plane, called weight vector, n is the number of attributes; and b is a scalar, called bias. Given a hyperplane (W, b) which separates the data, the aim is to find a function $f(x)$ which classifies the training data and later on the testing data.

$$f(x) = sign(W.X + b) \tag{5.5}$$

Knowing that a given hyperplane represented by (W, b) is equally expressed by all pairs $\{\lambda W, \lambda b\}$ for $\lambda \in R^+$, we define a canonical hyperplane as the hyperplane which separates the data from the hyperplane by a "distance" of at least 1. Thus, the weights may be adjusted so that the hyperplanes defining the sides of the margin can be written as

$$H_1 : W.X + b \geq +1 \text{ when } y_i = +1 \tag{5.6}$$

$$H_2 : W.X + b \leq -1 \text{ when } y_i = -1 \tag{5.7}$$

Hence, any data point which falls on or above H_1 belongs to class +1 while any data point that falls on or below H_2 belongs to class -1. Combining the two inequalities of the above equation, we obtain

$$y_i(W.X + b) \geq +1 \ \forall i \tag{5.8}$$

A *support vector* defined to be any training data point located on hyperplanes H_1 and H_2. These data points are the most difficult to classify and are the most informative from a classification perspective. From the above, we can obtain the formula for the size of the maximal margin. The geometric distance from the separating hyperplane to any point on H_1 or H_2 is $\frac{1}{\|W\|}$, where $\|W\|$ is the euclidean norm. The maximal margin is thus $\frac{2}{\|W\|}$. The distance between the hyperplane and any given data point is thus,

$$d((W, b), X) = \frac{y_i(W.X + b)}{\|W\|} \geq \frac{1}{\|W\|} \tag{5.9}$$

Intuitively, we are searching for the hyperplane which maximizes the geometric distance to the nearest data point. The above equation can be rewritten using a Langrangian formulation and then solved using Karush-Kuhn-Tucker (KKT) conditions. The MMH can then be rewritten as the decision boundary (Bishop, 2006; Han et al., 2006).

$$d(X^T) = \sum_{i=1}^{l} y_i \alpha_i X_i X^T + b_0 \tag{5.10}$$

where y_i is the class label of support vector X_i; X^T is a test data point; α_i (lagrangian multipliers) and b_0 are numeric parameters that were determined automatically by the optimization or SVM algorithm above, and l is the number of support vectors. Given a test data point, X^T, we calculate the distance and check to see the sign of the result. This will indicate on which side of the hyperplane the test data point falls. It is important to indicate that the complexity of the learned classifier is characterized by the number of support vectors and not the dimensionality of the data, which explains why SVM are less vulnerable to overfitting than some other methods. Furthermore, the support vectors found can be used to compute an upper bound on the expected error rate of the SVM classifier, which is also independent of the data dimensionality.

Case 2: Data are Linearly Inseparable

We have tackled the problem of classifying linearly separable data. However, in many cases, data can be non linearly separable, as in Figure 5.3. The approach described for linear SVMs has been extended to create nonlinear SVMs which are capable of finding nonlinear decision boundaries (i.e., nonlinear hypersurfaces) in input space. These approaches mainly consist of two major steps. In the first step, the original input data is transformed into a higher dimensional space using nonlinear mapping. Once the data have been transformed into the new higher space, the second step searches for a linear separating hyperplane in the new space. We again end up with a quadratic optimization problem that can be solved using the linear SVM formulation. The maximal marginal hyperplane found in the new space corresponds to a nonlinear separating hypersurface in the original space (Han et al., 2006).

When solving the quadratic optimization problem of the linear SVM, the training data points appear only in the form of dot products, $\phi(X_i).\phi(X_j)$, where $\phi(X)$ is the nonlinear mapping function applied to transform the training data points. This dot product can be replaced by a kernel function, $K(X_i, X_j)$ applied to the original input data. That is,

$$K(X_i, X_j) = \phi(X_i).\phi(X_j) \tag{5.11}$$

The procedure to find a maximal separating hyperplane is similar to that described for linearly separable data, although it involves placing a user-specified upped bound C (best determined experimentally), on the Lagrange multipliers α_i. Other kernels include:

$$\text{Polynomial kernel of degree h:}\quad K(X_i, X_j) = (X_i.X_j + 1)^h \tag{5.12}$$

$$\text{Radial basis function kernel:}\quad K(X_i, X_j) = e^{-\|X_i - X_j\|^2}/2\sigma^2 \tag{5.13}$$

$$\text{Sigmoid kernel:}\quad K(X_i, X_j) = tanh(\kappa X_i.X_j - \delta) \tag{5.14}$$

Each of these kernels leads to a different nonlinear classifier in the original input space. So far, we have focused on linear and nonlinear SVMs for binary (i.e, two-class classification) since it is the case of the data used in this thesis. However, SVM classifiers can be combined for multiclass cases as well. Given m classes, m classifiers are trained, one for each class, returning positive value for that class and negative for the rest. A test data point is assigned to the class corresponding to the largest positive distance.

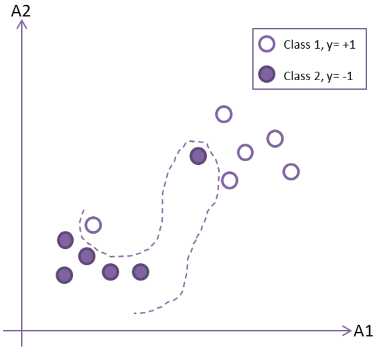


Figure 5.3: A 2-D case showing linearly inseparable data. Unlike the linear separable data of Figure 5.2, the decision boundary here is nonlinear.

5.2.4 Artificial Neural Networks

The idea of Artificial Neural networks was inspired from the neuro-physiological structure of the brain. They have been applied in numerous fields for pattern classification purposes such as classification of handwritten characters, image classification, sound processing and recognition, etc. (Bishop, 1995; Cristianini and Shawe-Taylor, 2000; Han et al., 2006).

An Artificial Neuron

A neuron is the elementary unit of an artifical neural network which, if activated, produces an output signal equal to the weighted sum of the its input signals coming from other neurons. In the following, we consider a particular type of neurons, the perceptron.

A perceptron is a processing unit with the following characteristics:

- It possesses $d + 1$ inputs that we note $x_i \in \{0, ..., d\} \in \mathcal{R}$. The input x_0 is particular and is called the bias and is equal to 1
- It possesses an output S
- Each input is weighted by a weight $w_i \in \{0, ..., d\} \in \mathcal{R}$
- An activation function, $\phi(.)$, which determines the value of S with respect to the

weighted sum of its inputs $\sum_{i=0}^{i=d} w_i x_i$, that is:

$$S = \phi(\sum_{i=0}^{i=d} w_i x_i) \tag{5.15}$$

In a classification task, the output indicates the class predicted for an input data. Hence, once the input data has been fixed, the output of the perceptron depends only on its weights. The learning process of a perceptron consists of tuning the values of the weights to those which lead to the best results in terms of minimal classification rate.

There are different types of activation functions. It can simply be the linear function:

$$\phi(\nu) = \nu \tag{5.16}$$

In this case, the output of the neuron is the weighted sum of its inputs. We talk of linear neuron. The output takes a value in $\mathcal{R}$.

It can also be the Heaviside function:

$$\phi(\nu) = \begin{cases} 1 \ si \ \nu \geq 0 \\ 0 \ si \ \nu < 0 \end{cases} \tag{5.17}$$

if $s \in \{0, 1\}$, or

$$\phi(\nu) = \begin{cases} 1 \ si \ \nu \geq 0 \\ -1 \ si \ \nu < 0 \end{cases} \tag{5.18}$$

if $s \in \{-1, 1\}$

It can also be a sigmoid function

$$\phi(\nu) = \frac{1}{1 + e^{-\alpha\nu}} \tag{5.19}$$

with $a \in \mathcal{R}$ if $s \in [0, 1]$, or hyperbolic tangent

$$\phi(\nu) = tanh(\nu) \tag{5.20}$$

if $s \in [-1, 1]$.

In classification, neural networks of the type "multi-layer perceptron" (Broadbent and Lucas, 1989; Dietz et al., 1989) permit nonlinear separation between classes depending on their architecture as well as the choice of the activation function. Neurons are distributed among multiple layers: input neurons associated to data, output neurons associated to each class and hidden neurons which are intermediates between the input and output neurons (see Figure 5.4).

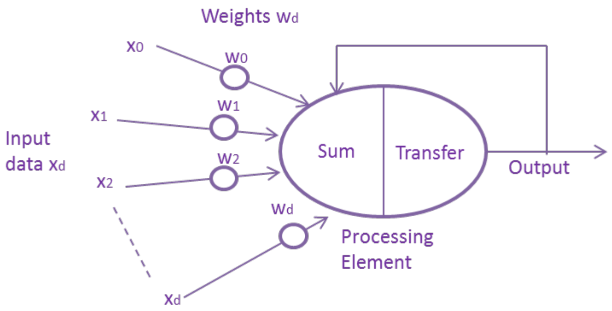


Figure 5.4: Schema illustrating the principle of a neuron

Multi-Layer Perceptrons

Multi-Layer perceptrons are the combination of neurons into layers, mainly to solve more-complex non-linear classification problems. A multi-layer perceptron consists of the following elements (see Figure 5.5):

- An input layer L_0: d inputs if the data are described by d attributes (which should be numerical)
- An output layer L_q: which can contain multiple neurons.
- One or multiple intermediary layers. Each layer is named L_i where i varies between 1 and $q-1$, each constituting a certain number of perceptrons $|C_i|$.
- Each connection between two units is characterized by a real weight value

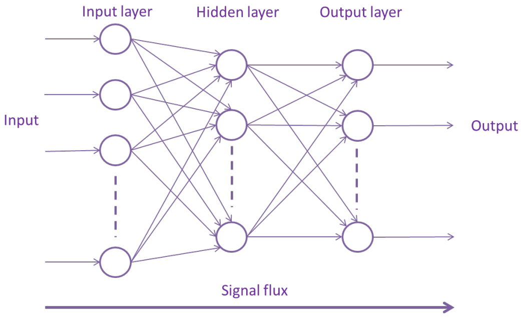


Figure 5.5: The architecture of a Multi-Layer Perceptron

The learning process of a multi-layer perceptron consists of updating the weights of the network using non-linear optimization methods. The aim is to minimize a cost function which defines the gap between the solution given by the network and the expected solution (Han et al., 2006).

5.3 Transforming data sequence into a labelled observation matrix

The four classifiers K-Nearest Neighbours, Naive Bayes, Artificial Neural Networks and Support Vector Machine cannot be applied on the sequence of events as it is. For this reason, it is vital to transform the sequence S into an observation matrix. This matrix will be unique for each target event T to be predicted. In this section, we propose a specific methodology to transform long temporal data sequences into labelled data observations.

From the sequence S constituting all events, for each target event T, two types of sequences are extracted. The first type of sequences, the *bad* sequences W_T, correspond to the sequences of length w, subsequences of S preceding all occurrences of the target event T in the sequence. The second type, selected randomly, are the *good* sequences W_R. These sequences do not exhibit any presence of T nor are they succeeded by T. Once the sequences W_T and W_R are extracted, we can define the observation space $X = \{X_1, ..., X_{l+m}\}$ consisting of both *bad* and *good* sequences $(X \cup \{W_T, W_R\})$, where l is the number of *bad* windows and m the number of *good* windows, and $Y = \{Y_1, ..., Y_{l+m}\}$ the label vector. We attribute a label $Y_i = 1$ for *bad* sequences and $Y_i = 0$ for *good* ones. Each observation X_i consists of d attributes, where d is the number of uncritical events types occurring in the data. The value of each attribute $x_{i,j}$, $i = \{1, ..., l+m\}$, $j = \{1, ..., d\}$ is equal to the number of occurrences of uncritical event type U_j in window X_i.

In the data extract under disposal, there are 436 distinct uncritical event types $U_1, ..., U_{436}$ which can occur in the window which are not target events, thus there are 436 different attributes. We consider the number of occurrences of each one of these events U_i in the observation window (whether *good* or *bad* sequence) as the attribute value (see Figure 5.6). In the end, we obtain a data matrix (dataset) of dimensions $(l + m) \times 436$

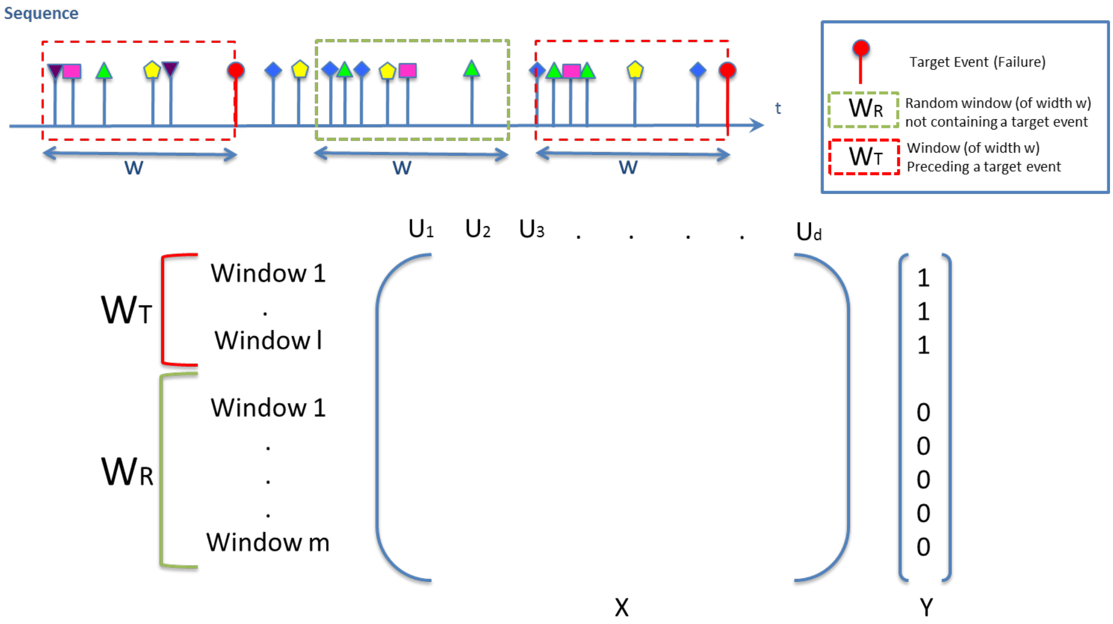

Sequence
t
w
w
w
Target Event (Failure)
WR Random window (of width w) not containing a target event
WT Window (of width w) Preceding a target event
U1 U2 U3 . . . . . Ud
WT
Window 1
Window l
WR
Window 1
Window m
1
1
1
0
0
0
0
0
X
Y

Figure 5.6: The methodology adopted to transform data sequence into labelled observation matrix constituting the dataset upon which pattern recognition methods can be applied

In the next section, we attempt to limit the dimensionality of the data matrix by reducing the number of attributes. We propose a methodology based on statistical hypothesis testing that will help choose the most significant attributes to be maintained for the classification process.

5.4 Hypothesis testing: choosing the most significant attributes

In this section, we propose a methodology to identify events which are most likely to occur prior to a target event. This methodology is based on a hypothesis test which we shall explain in the following.

Consider n independent and identically distributed (i.i.d.) random variables $X_1, X_2, ..., X_n$ abiding the law $X \sim \mathcal{B}er(p_1)$ and m independent and identically distributed random variables $Y_1, Y_2, ..., Y_m$, $Y \sim \mathcal{B}er(p_2)$. Bernoulli law, binary in nature, allows us to test the presence or not of an event in a given window. We note $x = (x_1, x_2, ..., x_n)$ as the data points (observations) and $y = (y_1, y_2, ..., y_n)$ as their respective labels.

We are interested in the following hypothesis test for equality of proportions:

$$\begin{cases} H_0: & p_1 = p_2 \\ H_1: & p_1 > p_2 \end{cases} \tag{5.21}$$

The unknown proportions p_1 and p_2 are estimated directly from observations by the frequencies $\overline{X}$ and $\overline{Y}$ defined by:

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \text{ and } \overline{Y} = \frac{1}{m}\sum_{i=1}^{m} Y_i \tag{5.22}$$

According to Annex 1, we propose the critical region to be:

$$W = \left\{ (x;y) | z = \frac{\overline{x} - \overline{y}}{\sqrt{\frac{\overline{x}+\overline{y}}{n+m}(1 - \frac{\overline{x}+\overline{y}}{n+m})(\frac{1}{n} + \frac{1}{m})}} > \phi_{-1}(1-\alpha^*) \right\} \tag{5.23}$$

where $\phi(.)$ is the distribution function of the standard normal distribution. The value of $\phi_{-1}(1-\alpha^*)$ can be deduced directly from statistical tables. For $z > \phi_{-1}(1-\alpha^*)$, the hypothesis H_0 is rejected and the alternative hypothesis $p_1 > p_2$ is adopted.

We apply this test on our data in order to obtain the events which are most likely to occur prior to a target event. We define the null hypothesis H_0 as "the probability of an event to occur prior to a target event (p_1) is equal to its probability to occur in a random sequence (p_2)". The alternative hypothesis is: "the probability of an event to occur prior to a target event (p_1) is greater than its probability to occur in a random sequence (p_2)". The test is applied between each event type and the target event as follows.

For a target event T, we derive the set of sequences W_T and W_R similarly as described in the section 5.3, with the sole difference that W_R may contain target events as well. Having derived both, we can now precise X_i and Y_j. If we consider A_{test} to be the event type which we are testing to validate whether it can potenially predict a target event T or not. We hence consider the following:

$$X_i = 1 \Longleftrightarrow \exists (A,t) \in w_{T,i} \text{ such that } A = A_{test} \tag{5.24}$$

$$Y_j = 1 \Longleftrightarrow \exists (A,t) \in w_{R,i} \text{ such that } A = A_{test} \tag{5.25}$$

X_i thus marks the existence of event A_{test} (with a probability $p = p_1$) in the i^{th} sequence preceding target event T. Similarly for Y_i, with $p = p_2$ but applied on randomly selected sequences. The corresponding $\overline{X}$ and $\overline{Y}$ defined in 5.22 hence correspond to the respective frequencies of A_{test} in W_T and W_R.

The algorithm 4 resumes the different steps of this test, which we apply on all target events, that is, Tilt and Traction "Driver Action High" events. For each target event, the algorithm returns the list $\mathcal{L}$ of A events that have validated H_1.

Algorithm 4 Pseudo-code of the hypothesis test for the equality of two proportions, performed to search for events that can potentially predict a target event T

Inputs: Target event T, Data sequence S, Significance degree α^*, list of events$\mathcal{L} = \phi$

```
# Step 1 : Find sequences preceding the each existence of $T$ and randomly selected sequences
- Determine $W_T$ and $W_R$
# Step 2 : Test $H_0$ against $H_1$
for all Distinct events $A$ occurring in $W_T$ do
    $A_{test} = A$
    Calculate $X_i$ according to 5.24
    Calculate $Y_j$ according to 5.25
    Calculate $\overline{X}$, $\overline{Y}$ according to 5.22
    According to 5.23 ,
    if $z > \Phi^{-1}(1 - \alpha^*)$ then
        $\mathcal{L} \longleftarrow A_{test}$
    end if
end for
```

Output: List $\mathcal{L}$ of events that have validated H_1

The list of events $\mathcal{L}$ validating the alternative hypothesis H_1, considered as potentially predictor events with respect to the target event, constitutes our reduced list of selected attributes. Thus, there is a list of selected attributes for each target event.

5.5 Experimental Results

In this section, we present the results of the experimental study that has been performed on the TrainTracer data sequences. First, the choice of measures used to assess the performance of the classifiers is explained, followed by a study we have executed to tune the value of the scanning window w in order to transform the data sequences into observation matrix (dataset) in the most optimal way for our problematic. Finally, the results of the various experiments are presented and a comparative study between the performance of the four classifiers is established.

5.5.1 Choice of performance measures

The measures which we will use to evaluate the performance of the approaches presented earlier in this chapter are the Correct classification rate, Recall and Precision.

- Correct classification rate (CCR)

The correct classification rate is calculated by the following:

$$\text{Correct classification rate} = \frac{\text{Number of observations correctly classified}}{\text{Total number of observations}} \qquad (5.26)$$

Considering a two class classification problem, the confusion matrix can be presented in table 5.1 below:

	Positive decision	Negative decision
Positive label	True positive (TP)	False Negative (FN)
Negative label	False positive (FP)	True negative (TN)

Table 5.1: Confusion matrix of a 2-class classification problem

The equation 5.26 can thus be written as:

$$\text{Correct classification rate} = \frac{\text{TP + TN}}{\text{TP + FP + FN + TN}} \tag{5.27}$$

We should note that the correct classification rate is a global general indicator that cannot take into account the repartition of the classes of the problem. For this reason, and to reflect a more precise image of the obtained results, we compute two complementary indicators which are the Recall and Precision.

- Recall and Precision

When the classes of the classification problem are not well balanced, the performance of the classification can be evaluated using two interestingness measures: recall and precision. These measures are calculated independently for each class, where:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{5.28}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{5.29}$$

For a given class, the recall indicates the percentage of samples of this class that have been successfully labelled while precision indicates the percentage of samples attributed to their real class.

5.5.2 Choice of scanning window w

In order to transform the data sequence into observation data matrix, the scanning window w needs to be set. For this reason, we have applied the four classifiers K-Nearest Neighbours (K-NN), Naive bayes, SVM and Artifical Neural Networks (ANN) on observation data matrices obtained with different values of w in order to decide which value is the most optimal for our case study.

The evaluation of the results of all classifiers was performed using a cross validation technique, consisting of randomizing the data set before dividing it into training and testing datasets. The number of randomizations was set to be 100 and thus the validation technique consisted of 100 folds. We used 70% of the observations for training and the remaining 30% for testing. We have also varied some key parameters of certain classifiers such as the number of hidden neurons and the function type for the neural networks as well as the type of kernel for SVM and the number of neighbours for the K-Nearest Neighbour approach.

We consider one particular type of Tilting Pantograph Failures as our target event. Due to confidentialy reasons, this event will not be revealed and will simple be denoted by T_1. Target event T_1 has 302 occurrences in the data sequences, hence there are 302 windows leading to this target event and hence are labelled with 1 following the methodology we proposed in Section 5.3 to transform data sequence into an observation data matrix. We have then selected 906 random windows from the sequence that do not contain nor are directly succeeded by the target event and labelled them 0 (we have chosen that the number of 0-labelled windows is to be triple the number of 1-labelled windows). Thus the observation data matrix consists of 1208 observations and is of dimension 1208 × 436 with all attributes selected and 1208 × d with selected attributes considered only, where d is the number of attributes. Table 5.2 shows the number of selected attributes discovered following the approach we previously proposed in Section 5.4 for various values of w. Table 5.3 shows the optimal results of the four classifiers on the four different datasets.

w (hours)	Selected Attributes	Dataset dimensions
2	56	1208 × 56
6	58	1208 × 58
10	55	1208 × 55
20	36	1208 × 36

Table 5.2: The number of selected attributes for different values of scanning window w

	w = 2 hours			**w = 6 hours**		
	CCR	Precision	Recall	CCR	Precision	Recall
K-NN	91.6	92.7	84.5	90.9	89.6	85.3
NB	85.3	80.5	80	83.1	78.4	74.1
SVM	**94.7**	**95.1**	**90.5**	**92.6**	**93.7**	**86.2**
ANN	90.9	90.5	84	90.5	90.3	83.6
	w = 10 hours			**w = 20 hours**		
	CCR	Precision	Recall	CCR	Precision	Recall
K-NN	87.8	89.3	77.4	**87.5**	**86.5**	**78.4**
NB	83	79.3	72.5	82.7	78.6	71.8
SVM	**89.8**	**91.5**	**81**	86.4	87.4	75
ANN	87.6	85.5	78.8	87.2	83.4	79.4

Table 5.3: Synthesis of the best correct classification rates (CCR) obtained by four classifiers: K-NN, Naive Bayes (NB), Support Vector Machines (SVM) and Artifical Neural Networks (ANN) with different values of w leading to different datasets with different selected attributes for target event T_1

From the table 5.3 above, having varied w from 2 to 20 hours, we can observe that the performance of all classifiers decreases with the increase of the value of w. Inspite the fact that transforming the data sequence to an observation dataset with $w = 2$ hours have led to the best results for target event T_1, however, if we consider the case of a target event that might be caused by a long sequence of events extending beyond 2 hours, a lot of information might be lost. For this reason, we consider a trade-off between the performance and the abundance of information and we set the value of w for the following experiments to 6 hours.

5.5.3 Performance of algorithms

In what follows, we give the results for two tilt and traction “Driver Action High” events, considered as target: T_1 corresponding to a particular Tilting Pantograph failure and T_2 corresponding to a particular Train Tilt System defect. Knowing that the total number of attributes is 436, the attribute selection approach proposed in 5.4, with a scanning window size w defined to 6 hours, cuts the number of attributes for target event T_1 to 58 and that of the target event T_2 to only 6. In order to evaluate the efficiency of the attribute selection process, we compare the classification results obtained by all classifiers with the selected attributes to the results with all attributes considered.

Tables 5.4 and 5.5 show the results of the K-Nearest Neighbours classifier and Naive Bayes respectively for target event T_1. The value of k for the K-NN was varied from 1 to 11. The best classification results obtained were 88.1% on raw data with all attributes with $k = 9$ neighbours and 90.9 % with the 58 selected attributes and $k = 1$ neighbour. Results show that the performance of the K-NN with selected attributes outperforms that with all attributes for whatever number of neighbours used, which proves the efficiency of the attribute selection approach proposed. On the other hand, Naive Bayes have led to less important results than K-NN, with 84.6% on centered data with all attributes and 86.7% on centered data with the 58 selected attributes.

	All attributes		Selected attributes (58 attributes)	
K-NN	Raw data	Centered data	Raw data	Centered data
1	87.9 ± 1.3	87.9 ± 1.5	**90.9 ± 1.3**	**90.9± 1.2**
3	87.5 ± 1.3	87.5± 1.3	89.7 ± 1.5	89.9± 1.3
5	86.3 ± 1.3	86.3 ± 1.4	89.2 ± 1.3	89.2 ± 1.4
7	85.9 ± 1.2	85.8 ± 1.4	88.7 ± 1.4	88.8± 1.3
9	**88.1 ± 1.2**	**88 ± 1.2**	88.3 ± 1.8	88.5 ± 1.5
11	84.6 ± 1.3	84.4± 1.2	88.2 ± 1.3	88.1± 1.2

Table 5.4: Correct classification rates obtained by the K-Nearest Neighbour algorithm on both raw and centered data with all and selected attributes for target event T_1

	All attributes		Selected attributes (58 attributes)	
	Raw data	Centered data	Raw data	Centered data
Naive Bayes	81 ± 1.3	84.6 ± 1.5	83.1 ± 1.6	86.7 ± 1.7

Table 5.5: Correct classification rates obtained by the Naive Bayes algorithm on both raw and centered data with all and selected attributes for target event T_1

Similarly, the results of SVM using 2 kernels are shown in 5.6. The two kernels used were Polynomial and Radial Basis Function (RBF). Following a grid search to find the optimal parameter value, the degree of the polynomial kernel was set to 1 whereas the $log_2(C)$ and $log_2(\gamma)$ values of the RBF kernel were set to 9 and -15 respectively. The RBF kernel has slightly outperformed the polynomial kernel on both data with all attributes and selected attributes, with a maximal correct classification rate of 91.4% and 92.6% respectively.

	All attributes		Selected attributes (58 attributes)	
SVM Kernel type	Raw data	Centered data	Raw data	Centered data
Polynomial (Degree 1)	91.3 ± 1.2	91.4 ± 1.3	92 ± 1.2	91.9 ± 1.2
RBF ($log_2C = 9$, $log_2(\gamma) = -15$)	**91.4 ± 1.2**	**91.3 ± 0.9**	**92.6 ± 1.2**	**92.5 ± 1.1**

Table 5.6: Correct classification rates of SVM with two kernels on both raw and centered data for target event T_1. Parameters were defined following an extensive grid search

Tables 5.7 and 5.8 below present results obtained by Artificial Neural Networks on target event T_1. We define a neural network of 2 layers. We vary the number of hidden neurons as well as the function used.

	Raw data		Centered data	
Hidden neurons	tansig	logsig	tansig	logsig
3	80.1 ± 15	82.2 ± 12.9	79.8 ± 16	81.7 ± 13
5	85.1 ± 7.9	85 ± 9.7	82.3 ± 15	86 ± 5
7	87.2 ± 3.6	86.5 ± 4.3	86.1 ± 7.4	86.2 ± 7.2
10	86.5 ± 3.8	86.5 ± 3.9	86.6 ± 4.1	87.1 ± 3.2
15	**87.6 ± 3.8**	**87.2 ± 3.1**	**87.2 ± 2.8**	**87.2 ± 3.3**
20	87.4 ± 3	87.2 ± 3.4	87.1 ± 3.3	87 ± 3.3

Table 5.7: Correct classification rates of Artificial Neural Networks on raw and centered data for target event T_1 with all attributes

	Raw data		Centered data	
Hidden neurons	tansig	logsig	tansig	logsig
3	86 ± 8.9	86 ± 10.5	84.5 ± 14.8	88.3 ± 5.5
5	87.7 ± 7.9	89.2 ± 4.4	89.1 ± 4.6	89 ± 4.5
7	88.3 ± 10.2	88.9 ± 4.5	89.2 ± 4.8	89.3 ± 4.5
10	88.5 ± 7.8	89.2 ± 4.4	88.9 ± 6.9	89.1 ± 4.5
15	**90.5 ± 2.4**	**89.8 ± 4.1**	**90.6 ± 2.4**	**89.6 ± 4.4**
20	89.9 ± 3.7	89.7 ± 3.5	89.4 ± 4.4	88.9 ± 4.7

Table 5.8: Correct classification rates of Artifical Neural Networks on raw and centered data for target event T_1 with selected attributes (58 instead of 436)

A network with 15 hidden neurons leads to a slightly better performance than those with a different number of neurons. The hyperbolic tangent sigmoid transfer function (tansig) and the log-sigmoid transfer function (logsig) have approximately led to the same order of results. The best performance was 90.5 % on raw data with selected attributes and 90.6% on centered data with selected attributes as well, both using the tansig function.

Now that the optimal parameters for each classifier are known, we establish a comparative study aiming to define the approach with the best performance. The results of this study are exhibited in tables 5.9 and 5.10. In addition to the correct classification rate, we have calculated Recall and Precision to assess results in a better way. The four pattern recognition approaches have led to good results with high correct classification rates as well as Recall and Precision values. The best performance was obtained by the SVM approach with a Radial Basis Function kernel (with $log_2(C)$ and $log_2(\gamma)$ values equal to 9 and -15 respectively) when applied on data with all attributes as well as with selected attributes. The Precision values varied between 78.4% with Naive Bayes on centered data with all attributes up to 93.7% with SVM on raw data with selected attributes. Similary, the Recall value varied between 64.2% with Naive Bayes on raw data with all attributes to 86.2% with SVM with selected attributes.

	Raw data			Centered data		
	CCR	Precision	Recall	CCR	Precision	Recall
K-NN	88.1	87.3	72.2	88	87.2	72.2
NB	81	81.2	64.2	84.6	82.8	73.5
SVM	**91.4**	**92.2**	**84.8**	**91.3**	**91.9**	**84.6**
ANN	87.6	85.5	78.1	87.2	83.6	77.5

Table 5.9: Synthesis of the best correct classification rates obtained by four pattern recognition classifiers on both raw and centered data with all attributes for target event T_1

	Raw data			Centered data		
	CCR	Precision	Recall	CCR	Precision	Recall
K-NN	90.9	89.6	85.3	90.9	89.2	84.8
NB	83.1	78.4	74.1	86.7	82.1	82.8
SVM	**92.6**	**93.7**	**86.2**	**92.5**	**93.6**	**86.2**
ANN	90.5	90.3	83.6	90.6	90.7	83.6

Table 5.10: Synthesis of the best correct classification rates obtained by each algorithm on both raw and centered data with selected attributes (58 instead of 436) for target event T_1

From all of the above results, we can notice that the performance of the classifiers on raw and centered data with the selected attributes was better than that with all attributes. Thus we can deduce that the attribute selection method that we have proposed was indeed effective in enhancing the results of the classification process by increasing the performance of all the classifiers used: K-Nearest Neighbour, Naive Bayes, SVM and Neural Networks. This also brings to our attention furthermore that many of the initial attributes were misleading the classification process and can be considered as noise which needed to be pruned out to sharpen the classification process.

In the following, we apply the classifiers on target event T_2, corresponding to a train tilt system defect, while considering the selected attributes only. As mentioned before, the attribute selection approach we proposed has cut down the number of attributes for event T_2 from 436 to only 6. Knowing that there are 6226 occurrences of event T_2 in the data sequence, the transformation of the data sequence following the approach we proposed in Section 5.3 led to an observation matrix of dimensions 24904 x 6, consisting of 6226 observations labelled 1 and 18678 labelled 0.

Results in tables 5.11 and 5.12 show that the K-Nearest Neighbour classifier has led to a correct classification rate of 82.6% with 11 neighbours on raw data with the 6 selected attributes. Naive Bayes on the other hand has led to a slightly inferior correct classification rate of 81.9%.

	Selected attributes (6 attributes)	
K-NN	Raw data	Centered data
1	76.3 ± 14	78.9±8.9
3	79.9 ± 8.9	79±10
5	81.7 ± 1.1	81.2±5.3
7	82.2 ± 0.8	82.1±1
9	82.4 ± 0.6	82.3±0.9
11	**82.6 ± 0.6**	**82.3±0.9**

Table 5.11: Correct classification rates obtained by the K-Nearest Neighbour classifier on both raw and centered data with selected attributes for target event T_2

	Selected attributes (6 attributes)	
	Raw data	Centered data
Naive Bayes	81.9± 0.4	81.9± 0.3

Table 5.12: Correct classification rates obtained by the Naive Bayes algorithm on both raw and centered data with selected attributes for target event T_2

Similarly, the results of SVM using 2 kernels, Polynomial and Radial Basis Function (RBF), are given in 5.13. Following a grid search to find the optimal parameter value, the degree of the polynomial kernel was set to 1 whereas the $log_2(C)$ and $log_2(\gamma)$ values of the RBF kernel were set to 8 and -14 respectively. The RBF kernel outperforms the Polynomial kernel with a correct classification rate of 83.2% on both raw and centered data.

	Selected attributes (6 attributes)	
SVM Kernel type	Raw data	Centered data
Polynomial (Degree 1)	80.6 ± 0.3	80.6 ± 0.3
RBF ($log_2C = 8$, $log_2\gamma = -14$)	**83.2 ± 0.3**	**83.2 ± 0.3**

Table 5.13: Correct classification rates of SVM with two kernels on both raw and centered data for target event T_2. Parameters were defined following an extensive grid search

Applying Artifical Neural Network classifier on the data while varying the number of hidden neurons, the best results (shown in Table 5.14) were obtained with a network consisting of 7 hidden neurons, with a slight difference in results with respect to a different number of neurons. The hyperbolic tangent sigmoid transfer function (tansig) slightly outperforms the log-sigmoid transfer function (logsig). The correct classification rate has reached 81.7 % on raw data.

	Raw data		Centered data	
Hidden neurons	tansig	logsig	tansig	logsig
3	81.3±1.6	81.3±1.9	81.3 ± 1.7	81.2±1.9
5	81.3 ± 1.7	81.1±2.2	81.7 ± 1.1	81.4± 1.7
7	**81.7 ± 1**	**81.2± 2**	**81.7 ± 0.8**	**81.6 ± 1.3**
10	80.9 ± 1.4	81.2±1.9	81.5 ± 1.5	81.3 ± 1.9
15	81.6 ± 1	81.2 ± 1.7	81.5 ± 1.6	81.5 ± 1
20	80.9 ± 5.9	80.9±2	81.3 ± 1.7	81.2 ± 1.7

Table 5.14: Correct classification rates of Artificial Neural Networks on both raw and centered data with selected attributes for target event T_2

Table 5.15 recapitulates the best results obtained by each classifier. Similar to before, in addition to the correct classification rate, we have calculated Recall and Precision to acquire a better vision of results. The four pattern recognition approaches have led to good results with high correct classification rates as well as Recall and Precision values. The best performance was obtained by the SVM approach with a Radial Basis Function kernel (with $log_2(C)$ and $log_2(\gamma)$ values equal to 8 and -14 respectively) as well as by the K-Nearest Neighbour classifier with 11 neighbours, both slightly outperfoming the Artificial Neural Networks and Naive Bayes classifiers. The Precision values varied from 76.3% with Naive Bayes up to 78.4% with SVM. Similary, the Recall value varied from 66.6% with SVM up to 72.9% with Naive Bayes.

	Raw data			Centered data		
	CCR	Precision	Recall	CCR	Precision	Recall
K-NN	82.6	77.8	72.7	82.3	77.4	72.7
NB	81.9	76.3	72.9	81.9	76.3	72.9
SVM	**83.2**	**78.4**	**66.6**	**83.2**	**78.4**	**66.6**
ANN	81.7	77	70.3	81.7	76.9	69.8

Table 5.15: Synthesis of the best correct classification rates obtained by each classifier on both raw and centered data with selected attributes (6 instead of 436) for target event T_2

The four classifiers have maintained a good performance despite the large decrease of attributes from 436 to only 6. This validates the effectiveness of the attribute selection process we have proposed and highlights the amplitude of the attributes misleading the classification process and the importance of pruning them out to sharpen the performance.

5.6 Conclusion

In this chapter, we have adopted a non-temporal approach to tackle the problem of the prediction of infrequent target events in sequences based on pattern recognition techniques. First, we propose a methodology to transform the data sequence into a set of observation matrix constituting the dataset upon which four classification methods were applied: K-Nearest Neighbours, Naive Bayes, Support Vector Machines and Artificial Neural Networks. A hypothesis testing approach was proposed to decrease the number of attributes by selecting those which are most significant and contributive to the classification performance. The obtained results have shown that the attribute selection method was indeed effective and has led to an increase in the performance of the four classifiers, which highlighted also the fact that many of the initial attributes were misleading the classification process and can be considered as noise that needed to be pruned out to sharpen the performance.

After tuning the value of the scanning window width w following a thorough comparative study, the performance of the four approaches was evaluated using three measures: Correct Classification rate, Recall and Precision. Although the four algorithms have led to good results that have reached 92.6% with Support Vector Machines using the RBF kernel as well as high Recall and Precision values, some major points regarding inconveniences should be highlighted. The main disadvantage of using classification approaches is that they do not take into account the temporal evolution of data and sequential nature of events within the window preceding the occurrence of the target event, and thus cannot indicate the order of events leading to a target event as well as predict exactly when that event will occur. Furthermore, each target event is handled and predicted separately since the sequence has to be transformed into a dataset that is specific for each target event and thus the mining process is complex and consists of multiple preprocessing steps. In addition to that, the scanning window that we have fixed to 6 hours following a trade-off might be limiting in case where the target event is caused by an accumulation of events over a duration exceeding 6 hours. Another arising problem would be setting the parameters for each dataset and target event, which might not be a one-time task knowing that the observation dataset should be updated for every new occurrence of the target event. Performing a grid search for the best parameters following each update would be computationally expensive and time consuming.

The pattern recognition approach has shown to be indeed effective for the prediction of infrequent target events in temporal sequences, once these sequences are transformed properly into an observation data matrix, and an effective attribute selection process is established.

Chapter 6

Conclusion and Perspectives

6.1 Conclusion

Within the persisting efforts to enhance availability and reliability of railway rolling stock, railway manufacturers and operators have adopted an automatic diagnosis approach aiming to establish an effective predictive maintenance process. Commercial trains are being equipped with state-of-the-art on-board intelligent sensors providing real-time flow of data consisting of georeferenced events, along with their spatial and temporal coordinates. Once ordered with respect to time, these events can be considered as long temporal sequences which can be mined for possible relationships. This has created a necessity for sequential data mining techniques in order to derive meaningful association rules or classification models from these data. Once discovered, these rules and models can then be used to perform an on-line analysis of the incoming event stream in order to predict the occurrence of target events, i.e, severe failures that require immediate corrective maintenance actions.

The work in this thesis tackles the above mentioned data mining task. We aimed to investigate and develop various methodologies to discover association rules and classification models which can help predict rare failures in sequences. The investigated techniques constituted two major axis: **1- Association analysis**, which is temporal and aimed to discover association and episode rules, and **2- Classification techniques**, which is not temporal. The main challenges confronting the data mining task and increasing its complexity were mainly the rarity of the target events to be predicted in addition to the heavy redundancy of some events and the frequent occurrence of data bursts.

Axis 1: Association Analysis

In this axis, constituting Chapters 3 and 4 of this thesis, we analyzed the temporal floating data sequences for association and episode rules leading to target events. We worked on two completely different approaches.

First in chapter 3, we proposed two methodologies based on existing significance-testing algorithms that have been adapted to our problem, T-Patterns and Null models in order to derive length-2 association rules of the form $A \longrightarrow B$ where B is a failure event. Although co-occurrences are decided by means of a hypothesis test in both methods, the approaches used to calculate the p-value used by the test are completely different. The first, null models, consists of randomization techniques followed by the calculation of various co-occurrence scores. The second, T-Patterns, exploits the temporal dimension by investigating the statistical dependence between inter-arrival times of couples of events in order to highlight possible relationships and temporal dependencies. We then proposed Double Null Models (DNM) as an extension to null models, a bipolar approach for discovering significant couples in a way that best assesses recall and precision and renders the mining process more resistant to spuriousness, hence decreasing the false positive rate.

In chapter 4, we adopted a different approach based on Weighted Association Rule Mining (WARM), which we adapted to temporal sequences in order to derive longer rules. We proposed Weighted WINEPI, an episode rule mining algorithm based on the fusion of both the WINEPI frequent episode mining algorithm consisting of a sliding window transforming a temporal data sequence into a series of overlapped windows as well as the valency model proposed in (Koh et al., 2010) that we especially adapted for the problem of data sequences instead of transaction data. The Weighted WINEPI algorithm was modified to be able to integrate more efficiently infrequent events into the mining process and a measure called cruciality was proposed as a preprocessing filter to conserve events that are useful for the target event to be predicted. We also proposed a constraint-based approach derived from Weighted WINEPI that we called Oriented Weighted WINEPI to focus the mining process on mining rules leading to a specific target event. Several tests on synthetic data have showed that Weighted WINEPI outperforms the classical WINEPI algorithm in detecting patterns between infrequent events.

Axis 2: Classification techniques

In this axis, constituting Chapter 5, we adopted a non-temporal approach to tackle the problem of the prediction of infrequent target events in sequences based on pattern recognition techniques. We attempted to train classification models instead of discovering rules. First, we proposed a methodology to transform the long temporal data sequence into a set of labelled observation matrix constituting the dataset. Following that, an attribute selection approach based on hypothesis-testing was also proposed to reduce the dimensionality of the dataset by selecting attributes that are most significant and contributive to the classification performance. Four different pattern recognition classifiers were then applied on the dataset: K-Nearest Neighbours, Naive Bayes, Support Vector Machines and Artificial Neural Networks and a comparative study has been established.

Association analysis vs. Classification techniques

Mining temporal sequences for associations between infrequent events is a recent problem that has not received much attention yet. Most existing sequence mining techniques are frequency-based and are not suitable for our problem.

The association analysis approaches we have adapted in Axis 1 have led to very good results on synthetic data which proved their efficiency and capability to detect associations between infrequent events. The association rules and episode rules that have been obtained on real data by all of the above approaches were evaluated using precision and recall measures, which we believe are well capable to evaluate the directionality aspect of the rule which should be leading to target events and not vice versa. However, the complex nature of the real floating train data extract under disposal lying in the heavy presence of redundancies and bursts have disrupted the precise assessement of obtained rules by these measures and altered their values negatively. No rules have been discovered having both a recall and precision values exceeding 50 %. In addition to that, due to the lack of ground truth knowledge on whether rules actually exist in the data or not, it was not possible to verify whether the results and rules obtained were the best we can find in the data extract under disposal or not. The main advantage lies in the exploitation of the temporal aspect. T-Patterns algorithm depends on the inter-event time between events while Null models and Weighted Winepi depend on a temporal scanning window w. The temporal aspect was also integrated in the calculation of the recall and precision values.

As for the pattern recognition approach constituting Axis 2. The obtained results have shown the approach to be indeed effective for the prediction of infrequent target events in temporal sequences, once these sequences are transformed properly into an observation data matrix and an effective attribute selection process in established. However, although the four algorithms have led to good results that have reached 92.6% with Support Vector Machines using the RBF kernel as well as high Recall and Precision values, some major points regarding inconveniences should be highlighted. The main disadvantage of using classification approaches is that they do not take into account the temporal evolution of data and sequential nature of events within the window preceding the occurrence of the target event, and thus cannot indicate the order of events leading to a target event as well as predict exactly when that event will occur. Although, in our approach to transform the data sequence into observation dataset, the temporal aspect was indeed taken into account when defining the scanning window value w.

Furthermore, each target event is handled and predicted separately since the sequence has to be transformed into a dataset that is specific for each target event and thus the mining process is complex and consists of multiple preprocessing steps. Another arising problem would be setting the parameters for each dataset and target event, which might not be a one-time task knowing that the observation dataset should be updated for every new occurrence of the target event. Performing a grid search for the best parameters following each update would be computationally expensive and time consuming.

If we consider target event T_1, corresponding to a Tilting Pantograph failure, as an example. The approaches of Axis 1 have not discovered rules leading to this event with recall and precision values exceeding 50%, and hence according to the obtained results, it is not possible to predict this event with high accuracy. Moreover, due to the absence of ground truth, we cannot be certain whether there are actually better rules or not. However, adapting the blackbox-like classification approaches of Axis 2 (where prediction occurs without rules), event T_1 was successfully predicted by the four classifiers combined with the attribute selection process we proposed with a correct classification rate of at least 83.1% and up to 92.6%, with a precision and recall values of 93.7% and 86.2% respectively. although, as mentioned before, the temporal aspect was not taken into account.

6.2 Future Research Directions

This thesis tackled the challenging problem of the analysis of Floating Train Data (FTD) obtained from intelligent sensors mounted on commercial trains. This automatic diagnosis-enforcing process aims to predict infrequent target events in temporal sequences. The subject of this work is a recent and novel problem both applicative wise and scientific wise. The results obtained by the two axis of approaches show the interest of data mining algorithms applied on Floating Train Data for the prediction of rare events and encourage the pursuit of the work achieved so far.

Several future research directions and open issues can be derived from our work. One interesting extension of this thesis would be to merge both axis association rule mining and classification. For example, the association analysis approaches can be used as an attribute selection process for pattern recognition approaches. That is, events constituting the discovered rules can be adapted as selected attributes for pattern recognition classifiers, since these events have temporal dependencies and relationships with the target event to be predicted. This way, the temporal aspect is integrated indirectly in the classification process. Another extension would be to develop an automatic method which aims to find the optimal value of the scanning window size w for each target event. This same approach can also be used to find the optimal warning time value between the observation (scanning) window and the target event by sliding the window away from the target event and searching for the optimal window size-warning time combination that leads to the best correct classification rate.

A second track would be to extend the Null models to discover longer rules (length-3 and more) in addition to the extension of the Oriented Weighted Winepi algorithm by integrating the attribute selection technique proposed in Chapter 5. The algorithm would attempt to discover rules leading to a specific target event by taking into account only events that were selected by the attribute selection technique instead of those with the highest cruciality value and establish a comparative study.

Furthermore, it would be interesting to test the performance of all approaches in an online-prediction environment using untested floating train data sequences. It is also important to highlight the vital need to establish data cleaning protocols and techniques to decrease event bursts and redundancies in the data. This would no doubt reflect very positively on the performance of association analysis algorithms as well as on the statistical assessment of obtained results thru a more accurate calculation of interestingness measures.

Appendix A

A.1 Expression of the critical interval of the test of equality of proportions

In a test for equality of proportions :

$$\begin{cases} H_0 : & p_1 & = & p_2 \\ H_1 : & p_1 & > & p_2 \end{cases} \tag{1}$$

The unknown proportions p_1 et p_2 are thus estimated by the frequencies $\overline{X}$ et $\overline{Y}$:

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \quad \text{et} \quad \overline{Y} = \frac{1}{m}\sum_{i=1}^{m} Y_i \tag{2}$$

The classical solution relies on the difference $\overline{X} - \overline{Y}$. The hypothesis H_0 defined in 1 is rejected with a fixed error probability $\alpha = \alpha^*$ when this difference becomes significant. Supposing H_0 to be true, i.e, $p_1 = p_2 = p$ and considering that the sample sizes are sufficiently large, the Central Limit Theorem (see 6) enables the approximation of a sum of Bernoulli distributions by a normal distribution. Hence,

$$\begin{cases} \overline{X} & \overset{app}{\sim} & \mathcal{N}\left(p, \frac{p(1-p)}{n}\right) \\ \overline{Y} & \overset{app}{\sim} & \mathcal{N}\left(p, \frac{p(1-p)}{m}\right) \end{cases} \Longrightarrow \overline{X} - \overline{Y} \overset{app}{\sim} \mathcal{N}\left(0, p(1-p)\left(\tfrac{1}{n} + \tfrac{1}{m}\right)\right)$$

therefore,

$$\frac{\overline{X} - \overline{Y}}{\sqrt{p(1-p)\left(\frac{1}{n} + \frac{1}{m}\right)}} \overset{app}{\sim} \mathcal{N}(0,1) \tag{3}$$

The proportion p, being known, is substituted under H_0 by its estimator asymptotically without bias $\frac{\overline{X}+\overline{Y}}{n+m}$. But since:

$$\begin{aligned}
\overline{X}-\overline{Y} &\xrightarrow{L} \mathcal{N}\left(0, p(1-p)\left(\tfrac{1}{n}+\tfrac{1}{m}\right)\right)\\
\frac{1}{\sqrt{\frac{\overline{X}+\overline{Y}}{n+m}\left(1-\frac{\overline{X}+\overline{Y}}{n+m}\right)\left(\frac{1}{n}+\frac{1}{m}\right)}} &\xrightarrow{\mathbb{P}} \frac{1}{\sqrt{p(1-p)\left(\frac{1}{n}+\frac{1}{m}\right)}}
\end{aligned}$$

then using the Theorem of Slutsky (See 7), wee can infer:

$$\frac{\overline{X}-\overline{Y}}{\sqrt{\frac{\overline{X}+\overline{Y}}{n+m}\left(1-\frac{\overline{X}+\overline{Y}}{n+m}\right)\left(\frac{1}{n}+\frac{1}{m}\right)}} \xrightarrow{L} \frac{\overline{X}-\overline{Y}}{\sqrt{p(1-p)\left(\frac{1}{n}+\frac{1}{m}\right)}}$$

Finally, under H_0:

$$Z = \frac{\overline{X}-\overline{Y}}{\sqrt{\frac{\overline{X}+\overline{Y}}{n+m}\left(1-\frac{\overline{X}+\overline{Y}}{n+m}\right)\left(\frac{1}{n}+\frac{1}{m}\right)}} \overset{app}{\sim} \mathcal{N}(0,1) \tag{4}$$

In order to determine the critical region W, which gives a reply to this decision problem, we should precise what is to be meant by *significant difference*. We hence search for the scalar k such that:

$$\mathbb{P}\left(Z>k\right) \quad = \quad \alpha^*$$

developing the above equation by replacing Z by its value:

$$\begin{aligned}
\mathbb{P}\left(\frac{\overline{X}-\overline{Y}}{\sqrt{\frac{\overline{X}+\overline{Y}}{n+m}\left(1-\frac{\overline{X}+\overline{Y}}{n+m}\right)\left(\frac{1}{n}+\frac{1}{m}\right)}}>k\right) &= \alpha^*\\
\Rightarrow \quad 1-\mathbb{P}\left(\frac{\overline{X}-\overline{Y}}{\sqrt{\frac{\overline{X}+\overline{Y}}{n+m}\left(1-\frac{\overline{X}+\overline{Y}}{n+m}\right)\left(\frac{1}{n}+\frac{1}{m}\right)}}\leq k\right) &= \alpha^*\\
\Rightarrow \quad 1-\Phi\left(k\right) &= \alpha^*
\end{aligned}$$

where $\Phi(.)$ corresponds to the distribution function of the standard normal distribution. La value of k corresponding to $\Phi^{-1}\left(1-\alpha^*\right)$ can be immediately deduced using statistical tables, hence enabling us to propose the following critical region as follows:

$$W = \left\{(x;y) \mid z = \frac{\overline{x}-\overline{y}}{\sqrt{\frac{\overline{x}+\overline{y}}{n+m}\left(1-\frac{\overline{x}+\overline{y}}{n+m}\right)\left(\frac{1}{n}+\frac{1}{m}\right)}} > \Phi^{-1}\left(1-\alpha^*\right)\right\} \tag{5}$$

For $z > \Phi^{-1}\left(1-\alpha^*\right)$, H_0 is rejected: the alternative hypothesis $p_1 > p_2$ is adopted in view of the observed data.

A.2 Central Limit Theorem and Slutsky's Theorem

Theorem 1. (Central Limit Theorem (CLT)) Let (X_n) be a sequence of an independent and identically distributed random variable, with an expected value μ and variance σ^2. We consider $(\overline{X}_n)$ to be the sequence of general term $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. Hence, asymptotically:

$$\frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}} \xrightarrow{L} \mathcal{N}(0,1) \tag{6}$$

Theorem 2. (Slutsky's Theorem) Let (X_n) and (Y_n) be two sequences of real random variables which converge in distribution to a random variable X and a constant c respectively, then, the sequence (X_nY_n) converges in distribution to cX. More generally:

$$\begin{cases} X_n & \xrightarrow{L} & X \\ Y_n & \xrightarrow{P} & c \end{cases} \Longrightarrow \begin{cases} X_n + Y_n & \xrightarrow{L} & X + c \\ X_nY_n & \xrightarrow{L} & cX \\ \frac{X_n}{Y_n} & \xrightarrow{L} & \frac{X}{c},\ c \neq 0 \end{cases} \tag{7}$$

Bibliography

R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad. Depth First Generation of Long Patterns. pages 108–118. ACM Press, 2000a.

R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. *Journal of Parallel and Distributed Computing*, 61: 350–371, 2000b.

R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*, pages 487–499, 1994.

R. Agrawal and R. Srikant. Mining Sequential Patterns. pages 3–14, 1995.

R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. *Proceedings of ACM SIGMOD '93*, pages 207–216, 1993.

R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. Verkamo, et al. Fast discovery of association rules. *Advances in knowledge discovery and data mining*, 12:307–328, 1996.

A. R. Andrade and P. Teixeira. A Bayesian model to assess rail track geometry degradation through its life-cycle. *Research in Transportation Economics*, 36(1):1 – 8, 2012. Selected papers from the 12th {WCTR} Topic Area Transport Economics and Finance.

P. P. Angelov, D. P. Filev, and N. K. Kasabov. Evolving intelligent systems: methodology and applications. 2010. ISBN 978-0-470-28719-4.

L. Angstenberger. *Dynamic Fuzzy Pattern Recognition.* Phd thesis, Fakultat fur Wirtschaftswissenschaften der Rheinisch-Westfalischen Technischen Hochschule, Aachen, Germany, 2000.

J. Antoni and R. B. Randall. The spectral kurtosis: application to the vibratory surveillance and diagnostics of rotating machines. *Mechanical Systems and Signal Processing*, 20(2):308 – 331, 2006.

P. Baraldi, M. Compare, G. Rossetti, A. Despujols, and E. Zio. A modelling framework to assess maintenance policy performance in electrical production plants. *in: Andrews, Berenguer, Jackson (Eds.), Maintenance Modelling and Applications, ESREDA-ESRA Project Group Report*, pages 263–282, 2011.

S. D. Bay and M. J. Pazzani. Detecting Change in Categorical Data: Mining Contrast Sets. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pages 302–306. ACM Press, 1999.

R. J. Bayardo. Mining the Most Interesting Rules. pages 145–154, 1999.

R. J. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-Based Rule Mining in Large, Dense Databases. pages 188–197, 2000.

D. R. Bellwood, P. C. Wainwright, C. J. Fulton, and A. Hoey. Assembly rules and functional groups at global biogeographical scales. *Functional Ecology*, 16:557–562, 2002.

V. Belotti, F. Crenna, R. C. Michelini, and G. B. Rossi. Wheel-flat diagnostic tool via wavelet transform. *Mechanical Systems and Signal Processing*, 20(8):1953 – 1966, 2006.

J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA, 1981. ISBN 0306406713.

C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Inc., New York, NY, USA, 1995. ISBN 0198538642.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

M. Bocciolone, A. Cigada, A. Collina, D. Rocchi, and M. Belloli. Wheel and rail longitudinal wear measurements and diagnostics. *Techrail International Workship and Dem on High Technology Systems for Railway Transportation, Paris*, 2002.

M. Bocciolone, A. Caprioli, A. Cigada, and A. Collina. A measurement system for quick rail inspection and effective track maintenance strategy. *Mechanical Systems and Signal Processing*, 21(3):1242 – 1254, 2007.

B. Boser, I. Guyon, and Vapnik V. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

L. Bouillaut, O. Francois, and S. Dubois. A Bayesian network to evaluate underground rails maintenance strategies in an automation context. *Journal of Risk and Reliability*, 2013.

B. Bouqata, C. Carothers, B. Szymanski, and M. J. Zaki. VOGUE: A Novel Variable Order-Gap State Machine for Modeling Sequences, 2006.

H. A. Broadbent and J. Lucas. Neural Network Model of Serial Learning. In *Conference Proceedings on APL As a Tool of Thought*, APL '89, pages 54–61, New York, NY, USA, 1989. ACM. ISBN 0-89791-327-2.

S. Bruni, M. Carboni, D. Crivelli, M. Guagliano, and P. Rolek. A preliminary analysis about the application of acoustic emission and low frequency vibration methods to the structural health monitoring of railway axles. *Chemical Engineering Transactions*, 33:697–702, 2013.

C. H. Cai, A. W. Fu, C. H. Cheng, and W. W. Kwong. Mining association rules with weighted items. In *Database Engineering and Applications Symposium, 1998. Proceedings. IDEAS'98. International*, pages 68–77, 1998.

C. Capdessus, M. Sidahmed, and J. L. Lacoume. Cyclostationary Processes: Applications in Gear Faults Early Diagnonsis. *Mechanical Systems and Signal Processing*, 14(3):371 – 385, 2000.

G. Casas-Garriga. Discovering Unbounded Episodes in Sequential Data. In *Proceedings of 7th Pacific-Asic Conference on Knowledge and Data Mining (PAKDD '03)*, pages 83–94. 2003a.

G. Casas-Garriga. Discovering Unbounded Episodes in Sequential Data. In Nada Lavra, Dragan Gamberger, Ljupo Todorovski, and Hendrik Blockeel, editors, *Knowledge Discovery in Databases: PKDD 2003*, volume 2838 of *Lecture Notes in Computer Science*, pages 83–94. Springer Berlin Heidelberg, 2003b. ISBN 978-3-540-20085-7.

M. Cayrol, H. Farreny, and H. Prade. Fuzzy Pattern Matching. *Kybernetes*, 11(2):103 – 116, 1982.

F. Chamroukhi, A. Samé, G. Govaert, and P. Aknin. A hidden process regression model for functional data description. Application to curve discrimination. *Neurocomput.*, 73(7-9):1210–1221, March 2010.

J. Chen, C. Roberts, and P. Weston. Fault detection and diagnosis for railway track circuits using neuro-fuzzy systems. *Control Engineering Practice*, 16:585–596, 2008.

S. K. Chen, T. K. Ho, and B. H. Mao. Reliability evaluations of railway power supplies by fault-tree analysis. *Electric Power Applications, IET*, 1(2):161–172, 2007.

Y-L. Chen and T. Huang. A novel knowledge discovering model for mining fuzzy multi-level sequential patterns in sequence databases. *Data Knowl. Eng.*, 66(3):349–367, September 2008.

E. Cohen, M. Datar, and S. Fujiwara. Finding Interesting Associations without Support Pruning. In *Proceedings of the 16th International Conference on Data Engineering*, ICDE '00, pages 489–, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0506-6.

R. K. Colwell and D. W. Winkler. A null model for null models in biogeography. *Ecological Communities: Conceptual Issues and the Evidence (D.R. Strong, Jr., D. Simberloff, L.G. Abele, and A.B. Thistle, eds.). Princeton University Press, Princeton, NJ.*, 88:344–359, 1984.

E. F. Connor and D. Simberloff. Interspecific competition and species co-occurrence patterns on islands: null models and the evaluation of evidence. *Oikos 41*, 88:455–465, 1983.

C. Conte and A. Shobel. Identifying dependencies among delays. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, RailHannover 2007, 2007.

G. Cosulich, P. Firpo, and S. Savio. Power electronics reliability impact on service dependability for railway systems: a real case study. In *Industrial Electronics, 1996. ISIE '96., Proceedings of the IEEE International Symposium on*, volume 2, pages 996–1001 vol.2, 1996.

F. G. Cozman, I. Cohen, M. C. Cirelo, and E. Politcnica. Semi-Supervised Learning of Mixture Models. In *ICML-03, 20th International Conference on Machine Learning*, pages 99–106, 2003.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods.* Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-78019-5.

B. Cule, B. Goethals, S. Tassenoy, and S. Verboven. Mining Train Delays. *Proceedings of the 10th International Symposium on Intelligent Data Analysis, IDA 2011*, 2011.

G. Das, K. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule Discovery From Time Series. pages 16–22. AAAI Press, 1998.

B. V. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* IEEE Computer Society Press, Los Alamitos, CA, 1991.

H. Dassanayake. *Fault diagnosis for a new generation of intelligent train door systems.* PhD thesis, University of Birmingham, 2002.

H. Dassanayake, C. Roberts, C. J. Goodman, and A. M. Tobias. Use of parameter estimation for the detection and diagnosis of faults on electric train door systems. *Journal of Risk and Reliability*, 223:271–278, 2009.

S. De Fabris, G. Longo, and G. Medeossi. Automated analysis of train event recorder data to improve micro-simulation models. In *In: Allan, J., Arias, E., Brebbia, C.A., Goodman, C., Rumsey, A.F., Sciutto, G., Tomii, A. (Eds.), Computers in Railways XI. WIT Press, Southampton*, RailHannover 2007, pages 573–583, 2008.

W. E. Dietz, E. L. Kiech, and M. Ali. Classification of data patterns using and autoassociative neural network topology. In *IEA/AIE (2)*, pages 1028–1036, 1989.

D. Dubois and H. Prade. From Blanchés Hexagonal Organization of Concepts to Formal Concept Analysis and Possibility Theory,. *Logica Universalis*, 6(1-2):149–169, 2012.

M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen. Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20 –26, jan. 2008.

C. Farhan Ahmed, S. Khairuzzaman Tanbeer, B. Jeong, and H. Choi. A framework for mining interesting high utility patterns with a strong frequency affinity. *Information Sciences*, 181(21):4878 – 4894, 2011.

R. Faure. In Madrid Espana Colegio de Ingenieros de Caminos, Canales y Puertos, editor, *La traccion electrica en la alta velocidad ferroviaria (A.V.F.)*. 2004. ISBN 84-380-0274-9.

U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Advances in knowledge discovery and data mining. chapter From data mining to knowledge discovery: an overview, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996. ISBN 0-262-56097-6.

O. Fink, E. Zio, and U. Weidmann. Predicting component reliability and level of degradation with complex-valued neural networks. *Reliability Engineering and System Safety*, 121(0):198 – 206, 2014.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen.*, 7:179–188, 1936.

H. Flier, R. Gelashvili, T. Graffagnino, and M. Nunkesser. Mining Railway Delay Dependencies in Large-Scale Real-World Delay Data. In RavindraK. Ahuja, RolfH. Mhring, and ChristosD. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, volume 5868 of *Lecture Notes in Computer Science*, pages 354–368. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-05464-8.

L. S. Freedman and D. J. Spiegelhalter. Comparison of Bayesian with group sequential methods for monitoring clinical trials. *Controlled Clinical Trials*, 10(4):357 – 367, 1989.

H. Frigui and R. Krishnapuram. A Robust Algorithm for Automatic Extraction of an Unknown Number of Clusters from Noisy Data. *Pattern Recogn. Lett.*, 17(12): 1223–1232, October 1996.

H. Frigui and R. Krishnapuram. Clustering by Competitive Agglomeration. *Pattern Recogn.*, 30(7):1109–1119, July 1997.

B. Gabrys and A. Bargiela. General fuzzy min-max neural network for clustering and classification. *Neural Networks, IEEE Transactions on*, 11(3):769–783, 2000.

V. Gancho. Online Classification of Machine Operation Modes Based on Information Compression and Fuzzy Similarity Analysis, 2009.

M. Garofalakis, Rajeev Rastogi, and Kyuseok Shim. Mining sequential patterns with regular expression constraints. *IEEE Transactions on Knowledge and Data Engineering*, 14:530–552, 2002.

L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3), September 2006.

N. J. Gotelli and G. L. Entsminger. Swap and fill algorithms in null model analysis: rethinking the knights tour. *Oecologia*, 129:281–291, 2001.

N. J. Gotelli and G. R. Graves. Null models in ecology. *Smisonian Inst. Press.*, 1996.

R. M. P. Goverde. A delay propagation algorithm for large-scale railway traffic networks. *Transportation research part C*, pages 269–287, 2011.

G. Grahne, X. Wang, and L.V. Laksgmanan. Efficient mining of constrained correlated sets. *International Conference on Data Engineering*, page 512, 2000.

S. Grassie. Rail corrugation: advances in measurement, understanding and treatment. *Wear*, 258(78):1224 – 1234, 2005. Contact Mechanics and Wear of Rail/Wheel Systems.

N. Haiminen, H. Mannila, and E. Terzi. Determining significance of pairwise co-occurrences of events in bursty sequences. *BMC Bioinformatics*, 9(1), August 2008.

W. Hamalainen and M. Nykanen. Efficient Discovery of Statistically Significant Association Rules. In *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pages 203–212, 2008.

J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation, 1999.

J. Han, J. Pei, Y. Yin, and R. Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach, 2004.

J. Han, M. Kamber, and J. Pei. *Data mining: concepts and techniques.* Morgan kaufmann, 2006.

Y. Han, O. Francois, A. Same, L. Bouillaut, L. Oukhellou, P. Aknin, and G. Branger. Online predictive diagnosis of electrical train door systems. January 2013.

S. Hanhijrvi, M. Ojala, N. Vuokko, K. Puolamki, N. Tatti, and H. Mannila. Tell Me Something I Dont Know: Randomization Strategies for Iterative Data Mining. *In KDD 09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 379–388, 2009.

S. Hannenhalli and S. Levy. Predicting transcription factor synergism. *Nucleic Acids Res*, 30(19), 2002.

S. K. Harms, J. Deogun, J. Saquer, and T. Tadesse. Discovering Representative Episodal Association Rules from Event Sequences Using Frequent Closed Episode Sets and Event Constraints. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 603–606. IEEE Press, 2001a.

S. K. Harms, J. S. Deogun, J. Saquer, and T. Tadesse. Discovering Representative Episodal Association Rules from Event Sequences Using Frequent Closed Episode Sets and Event Constraints. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 603–606, Washington, DC, USA, 2001b. IEEE Computer Society. ISBN 0-7695-1119-8.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, August 2003. ISBN 0387952845.

S. Honda, K. I. Fukui, K. Moriyama, S. Kurihara, and M. Numao. Extracting Human Behaviors with Infrared Sensor Network. *Proceedings of the 4th International Conference on Networked Sensing Systems, INSS 2007*, pages 122–125, 2007.

M. Ignesti, A. Innocenti, E. Meli, L. Pugi, and A. Rindi. Development and validation of innovative weighing in motion systems. *Chemical Engineering Transactions*, 33: 763–768, 2013.

R. Insa, P. Salvador, and J. Inarejos. Predictive Railway Maintenance based on Statistical Analysis of Track Geometric Parameters. *in J. Pombo, (Editor), Proceedings of the First International Conference on Railway Technology: Research, Development and Maintenance, Civil-Comp Press, Stirlingshire, UK, Paper 48*, 2012.

S. Iwnicki. In Ed. Taylor and Reino Unido Francis Group. ISBN 978-0-8493-3321-7. London, editors, *Handbook of Railway Dynamics.* 2006.

S. Jia and M. Dhanasekar. Detection of rail wheel flat using wavelet approaches. *Structural Health Monitoring*, 6:121–131, 2007.

I. T. Jolliffe. *Principal Component Analysis.* Springer Verlag, 1986.

D. Kifer, S. Ben-David, and J. Gehrke. Detecting Change in Data Streams. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 180–191. VLDB Endowment, 2004. ISBN 0-12-088469-0.

H. Klein and M. Vingron. Using transcription factor binding site co-occurrence to predict regulatory regions. *Genome informatics. International Conference on Genome Informatics*, 18:109–118, 2007.

T. Kobayashi, Y. Naganuma, and H. Tsunashima. Condition monitoring of shinkansen tracks based on inverse analysis. *Chemical Engineering Transactions*, 33:703–708, 2013.

Y. Koh and N. Rountree. Finding Sporadic Rules Using Apriori-Inverse. In T. Ho, D. Cheung, and H. Liu, editors, *Advances in Knowledge Discovery and Data Mining*, volume 3518 of *Lecture Notes in Computer Science*, pages 97–106. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-26076-9.

Y. S. Koh, R. Pears, and W. Yeap. Valency Based Weighted Asssociation Rule Mining. *In Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS, vol. 6118, PP. 274–285. Springer, Heidelberg*, 2010.

T. Kojima, H. Tsunashima, and A. Matsumoto. Fault detection of railway track by multi-resolution analysis. *Proceedings of the IEEE International Conference on Wavelet Analysis and its Applications*, 2005.

T. Kojima, H. Tsunashima, and A. Matsumoto. Fault detection of railway track by multi-resolution analysis. *Computer in Railway X, WIT Press*, pages 955–964, 2006.

R. J. Kuo, C. M. Chao, and C. Y. Liu. Integration of K-means algorithm and Apriori-Some algorithm for fuzzy sequential pattern mining. *Applied Soft Computing*, 9(1): 85 – 93, 2009.

P-A. Laur, J-E. Symphor, R. Nock, and P. Poncelet. Statistical supports for mining sequential patterns and improving the incremental update process on data streams. *Intell. Data Anal.*, 11(1):29–47, January 2007.

S. Lavorel and E. Garnier. Predicting changes in community composition and ecosystem functioning from plant traits: revisiting the Holy Grail. *Functional Ecology*, 16:545–556, 2002.

N. Lehrasab. *A generic fault detection and isolation approach for single-throw mechanical equipment.* PhD thesis, University of Birmingham, 1999.

N. Lehrasab, H. P. B. Dassanayake, C. Roberts, S. Fararooy, and C. J. Goodman. Industrial fault diagnosis: pneumatic train door case study. *Journal of Rail and Rapid Transit*, 216:175–183, 2002.

V. Lehsten and Harmand P. Null models for species co-occurrence patterns: assessing bias and minimum iteration number for the sequential swap. *Ecography*, 29:786–792, 2006.

P. Lenca, P. Meyer, B. Vaillant, and S. Lallich. A multicriteria decision aid for interestingness measure selection. 2004.

S. Levy, S. Hannenhalli, and C. Workman. Enrichment of regulatory signals in conserved non-coding genomic sequence. *Bioinformatics*, 17(10):871–877, October 2001.

T. Li and X. Li. Novel alarm correlation analysis system based on association rules mining in telecommunication networks. *Inf. Sci.*, 180(16):2960–2978, August 2010.

T. Li, X. Li, and H. Xiao. An Effective Algorithm for Mining Weighted Association Rules in Telecommunication Networks. In *Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on*, pages 425–428, 2007.

B. Liang, S. Iwnicki, G. Feng, A. Ball, V. T. Tran, and R. Cattley. Railway wheel flat and rail surface defect detection by time-frequency analysis. *Chemical Engineering Transactions*, 33:745–750, 2013.

N. P. Lin, H-J. Chen, W-H. Hao, H-E. Chueh, and C-I. Chang. Mining strong positive and negative sequential patterns. *W. Trans. on Comp.*, 7(3):119–124, March 2008.

Z. Lin-Hai, W. Jian-Ping, and R. Yi-Kui. Fault diagnosis for track circuit using AOK-TFRs and {AGA}. *Control Engineering Practice*, 20(12):1270 – 1280, 2012.

Y. Liu, W. Xu, and H. Du. The method of test for state of railway tunnel lining based on association rules. pages 387–390, May 2011.

E. Lughofer and P. Angelov. Detecting and Reacting on Drifts and Shifts in On-Line Data Streams with Evolving Fuzzy Systems, 2009.

J. Luo and S. M. Bridges. Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. *International Journal of Intelligent Systems*, 15(8):687–703, 2000.

S. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behakvior Research Methods, Instruments, & Computers*, 32s:93–110, 2000.

H. Mannila and H. Toivonen. Discovering Generalized Episodes Using Minimal Occurrences. In *Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining*, pages 146–151. AAAI Press, 1996.

H. Mannila, H. Toivonen, and I. Verkamo. Efficient Algorithms for Discovering Association Rules. pages 181–192. AAAI Press, 1994.

H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of Frequent Episodes in Event Sequences. *Data mining and knowledge discovery*, 1:259–289, 1997.

M. Marseguerra and E. Zio. Basics of the Monte Carlo method with application to system reliability. *Hagen, Germany: LiLoLe-Verlag GmbH*, 2002.

A. Matsumoto, Y. Sato, H. Ono, M. Tanimoto, Y. Oka, and E. Miyauchi. Formation mechanism and countermeasures of rail corrugation on curved track. *Wear*, 253(12): 178 – 184, 2002.

N. Méger and C. Rigotti. Constraint-based mining of episode rules and optimal window sizes. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD '04, pages 313–324, New York, NY, USA, 2004. Springer-Verlag New York, Inc. ISBN 3-540-23108-0.

N. Méger, C. Leschi, N. Lucas, and C. Rigotti. Mining episode rules in STULONG dataset. In *In Proc. of ECML/PKDD04 Discovery Challenge - A Collaborative Effort in Knowledge Discovery. Prague: Univ. of Economics*, 2004.

E. Miguelanez, K. E. Brown, R. Lewis, C. Roberts, and D. M. Lane. Fault diagnosis of a train door system based on semantic knowledge representation. In *Railway Condition Monitoring*, pages 1–6, June 2008.

R. K. Mobley. *An Introduction to Predictive Maintenance, Second Edition.* Butterworth-Heinemann, Amsterdam, 2002.

P. Nada Lavrac, F. Peter, and Z. Blaz. Rule Evaluation Measures: A Unifying View. In *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99*, pages 174–185. Springer-Verlag, 1999.

R. Neapolitan et al. *Learning bayesian networks*, volume 1. Prentice Hall Upper Saddle River, 2004.

A. Ng and A. Fu. Mining frequent episodes for relating financial events and stock trends. In *Proceedings of the 7th Pacific-Asia conference on Advances in knowledge discovery and data mining*, PAKDD'03, pages 27–39, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-04760-3.

R. T. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained association rules. *SIGMOD*, 27(2):13–24, 1998.

L. Nie and I. Hansen. System analysis of train operations and track occupancy at . . ., 2005.

T. Nielsen and F. Jensen. *Bayesian networks and decision graphs.* Springer, 2009.

M. Ohsaki, S. Kitaguchi, K. Okamoto, H. Yokoi, and T. Yamaguchi. Evaluation of rule interestingness measures with a clinical dataset on hepatitis. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD '04, pages 362–373, New York, NY, USA, 2004. Springer-Verlag New York, Inc. ISBN 3-540-23108-0.

M. Ojala, N. Vuokko, A. Kallio, N. Haiminenn, and H. Mannila. Randomization Methods for Assessing Data Analysis Results on Real-Valued Matrices. *Statistical Analysis and Data Mining*, 2(4):209–230, 2009.

L. Oukhellou, A. Debiolles, T. Denux, and A. Patrice. Fault diagnosis in railway track circuits using Dempster-Shafer classifier fusion. *Eng. Appl. Artif. Intell.*, 23(1):117–128, February 2010.

J. S. Park, M. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. pages 175–186, 1995.

D. Patnaik, P. S. Sastry, and K. P. Unnikrishnan. Inferring neuronal network connectivity from spike data: A temporal data mining approach. *Scientific Programming*, 16(1):49–77, 2008.

D. Patnaik, M. Marwah, R. K. Sharma, and N. Ramakrishnan. Temporal Data Mining Approaches for Sustainable Chiller Management in Data Centers. *ACM Trans. Intell. Syst. Technol.*, 2(4):34:1–34:29, July 2011.

R. Pears, Y. S. Koh, G. Dobbie, and W. Yeap. Weighted association rule mining via a graph based connectivity model. *Inf. Sci.*, 218:61–84, January 2013.

J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang. H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases. pages 441–448, 2001a.

J. Pei, J. Han, B. Mortazavi-asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. pages 215–224, 2001b.

J. Pei, J. Han, and W. Wang. Constraint-based sequential pattern mining: the pattern-growth methods. *J. Intell. Inf. Syst.*, 28(2):133–160, April 2007.

P. Pennacchi, R. Ricci, S. Chatterton, and P. Borghesani. Effectiveness of MED for Fault Diagnosis in Roller Bearings. In Ji Nprstek, Jaromr Horek, Miloslav Okrouhlk, Bohdana Marvalov, Ferdinand Verhulst, and Jerzy T. Sawicki, editors, *Vibration Problems ICOVP 2011*, volume 139 of *Springer Proceedings in Physics*, pages 637–642. Springer Netherlands, 2011. ISBN 978-94-007-2068-8.

A. Pieringer and W. Kropp. A fast time-domain model for whee[barred l]rail interaction demonstrated for the case of impact forces caused by wheel flats. *The Journal of the Acoustical Society of America*, 123(5):3266–3266, 2008.

M. Qin and K. Hwang. Frequent Episode Rules for Internet Anomaly Detection. In *Proceedings of the Network Computing and Applications, Third IEEE International Symposium*, NCA '04, pages 161–168, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2242-4.

G. D. Ramkumar, R. Sanjay, and S. Tsur. Weighted Association Rules: Model and Algorithm. In *Proc. Fourth ACM Intl Conf. Knowledge Discovery and Data Mining*, 1998.

R. B. Randall and J. Antoni. Rolling element bearing diagnosticsA tutorial. *Mechanical Systems and Signal Processing*, 25(2):485 – 520, 2011.

A. N. Randriamanamihaga. *Masters Thesis, Outils de Fouille de Données pour lanalyse de séquences dalarmes issues de véhicules sondes.* Masters thesis, Université de Technologie de Compiègne, 2012.

B. Rhayma, Ph. Bressolette, P. Breul, M. Fogli, and G. Saussine. A probabilistic approach for estimating the behavior of railway tracks. *Engineering Structures*, 33 (7):2120 – 2133, 2011.

N. Rhayma, Ph. Bressolette, P. Breul, M. Fogli, and G. Saussine. Reliability analysis of maintenance operations for railway tracks. *Reliability Engineering and System Safety*, 114(0):12 – 25, 2013.

T. Richter. Systemetic analysis of train run deviations from the timetable. In *In: Ning, B., Brebbia, C.A., (Eds.), Computers in Railways XI. WIT Press, Southampton*, Computers in Railways XI. WIT Press, Southhamption, pages 651–662, 2010.

C. Roberts, H. P. B. Dassanayake, N. Lehrasab, and C. J. Goodman. Distributed quantitative and qualitative fault diagnosis: railway junction case study. *Control Engineering Practice*, 10(4):419–429, April 2002.

A. A. Salah, E. Pauwels, R. Tavenard, and T. Gevers. T-Patterns Revisited: Mining for Temporal Patterns in Sensor Data. *Sensors*, 10(8):7496–7513, 2010.

A. Samé, F. Chamroukhi, G. Govaert, and P. Aknin. Model-based clustering and segmentation of time series with changes in regime. *Advances in Data Analysis and Classification*, 5(4):301–321, 2011.

J. W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Trans. Comput.*, 18(5):401–409, May 1969.

A. Savasere, E. Omiecinski, and S. Navathe. An Efficient Algorithm for Mining Association Rules in Large Databases. pages 432–444, 1995.

L. Songfeng, H. Heping, and L. Fan. Mining weighted association rules. *Intelligent Data Analysis*, 5(3):211–225, 2001.

R. Srikant and R. Agrawal. Mining Generalized Association Rules. pages 407–419, 1995.

R. Srikant and R. Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, 1996. Springer-Verlag. ISBN 3-540-61057-X.

R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. *KDD*, pages 67–73, 1997.

Q. L. Stephen, S. Levinson, Y. Wu, and T. Huang. Interactive and Incremental Learning via a Mixture of Supervised and Unsupervised Learning Strategies. In *Proceedings of the Fifth Joint Conference on Information Sciences*, pages 555–558, 2000.

M. Su. Discovery and prevention of attack episodes by frequent episodes mining and finite state machines. *Journal of Network and Computer Applications*, 33(2):156 – 167, 2010.

K. Sun and Fengshan B. Mining Weighted Association Rules without Preassigned Weights. *Knowledge and Data Engineering, IEEE Transactions on*, 20(4):489–495, 2008.

P. Tan and V. Kumar. Selecting the Right Interestingness Measure for Association Patterns, 2002.

P. Tan, M. Steinbach, V. Kumar, C. Potter, S. Klooster, and A. Torregrosa. Finding spatio-temporal patterns in earth science data. *Proceedings of the KDD Workshop on Temporal Data Mining*, 2001.

P. N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321321367.

F. Tao, F. Murtagh, and M. Farid. Weighted Association Rule Mining using weighted support and significance framework. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 661–666, New York, NY, USA, 2003. ACM. ISBN 1-58113-737-0.

R. Tavenard, A. A. Salah, and E. J. Pauwels. Searching for temporal patterns in ami sensor data. *Constructing Ambient Intelligence*, pages 53–62, 2008.

C. W. Therrien. *Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics.* John Wiley & Sons, Inc., New York, NY, USA, 1989. ISBN 0-471-83102-6.

H. Tsunashima, T. Kojima, Y. Marumo, A. Matsumoto, and T. Mizuma. Condition monitoring of railway track and driver using in-service vehicle. In *Railway Condition Monitoring, 2008 4th IET International Conference on*, pages 1–6, 2008.

A. Tsymbal. The Problem of Concept Drift: Definitions and Related Work. Technical report, 2004.

O. Uncu and I. B. Turksen. A novel feature selection approach: Combining feature wrappers and filters. *Information Sciences*, 177(2):449 – 466, 2007.

C. Vale and S. M. Lurdes. Stochastic model for the geometrical rail track degradation process in the Portuguese railway Northern Line. *Reliability Engineering and System Safety*, 116(0):91 – 98, 2013.

V. Vapnik. Estimation of Dependences Based on Empirical Data [in Russian], Moscow: Nauka. 1979.

V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

G. Vettori, B. Allotta, M. Malvezzi, L. Pugi, A. Ridolfi, P. D'Adamio, F. Salotti, and L. Landi. Innovative management of wheel-rail adhesion conditions in localization algorithms for the automatic train protection. *Chemical Engineering Transactions*, 33:685–690, 2013.

B. Vo, F. Coenen, and B. Le. A new method for mining Frequent Weighted Itemsets based on WIT-trees. *Expert Systems with Applications*, 40(4):1256 – 1264, 2013.

M. F. Wang, Y. C. Wu, and M. F. Tsai. Exploiting Frequent Episodes in Weighted Suffix Tree to Improve Intrusion Detection System. In *AINA Workshops*, pages 1246–1252. IEEE Computer Society, 2008.

W. Wang, J. Yang, and P. S. Yu. Efficient Mining of Weighted Association Rules (WAR). In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 270–274. Press, 2000.

Y. Wang, B. Deng, H. J. Li, Z. J. Tian, and J. Ke. Reliability simulation calculation on the railway catenary system. In *Journal of Basic Science and Engineering*, volume 13 No.3, pages 307–312. Press, 2005.

G. I. Webb. Efficient Search for Association Rules, 2000.

V. A. Weeda and K. S. Hofstra. Performing analysis: improving the Dutch railway service. In *In: Allan, J., Arias, E., Brebbia, C.A., Goodman, C., Rumsey, A.F., Sciutto, G., Tomii, A. (Eds.), Computers in Railways XI. WIT Press, Southampton*, Computers in Railways XI. WIT Press, Southhamption, pages 463–471, 2008.

C. Wei, Q. Xin, W. Chung, S. Liu, H. Tam, and S. L. Ho. Real-Time Train Wheel Condition Monitoring by Fiber Bragg Grating Sensors. *International Journal of Distributed Sensor Networks*, 4:568–581, 2012.

P. F. Weston, C. Roberts, C. J. Goodman, and C. S. Ling. Condition Monitoring of Railway Track using In-Service Trains. In *Railway Condition Monitoring, 2006. The Institution of Engineering and Technology International Conference on*, pages 26–31, 2006.

P. F. Weston, C. S. Ling, C. J. Goodman, C. Roberts, P. Li, and R. M. Goodall. Monitoring lateral track irregularity from in-service railway vehicles. *Proceedings of the IMechE: Part F Journal of Rail and Rapid Transit*, 221(1):89–100, 2007.

T. X. Wu and D. J. Thompson. A Hybrid Model For the Noise Generation due to Railway Wheel Flats. *Journal of Sound and Vibration*, 251(1):115 – 139, 2002.

L. Yan and C. Li. Incorporating Pageview Weight into an Association-Rule-Based Web Recommendation System. In A. Sattar and B. Kang, editors, *AI 2006: Advances in Artificial Intelligence*, volume 4304 of *Lecture Notes in Computer Science*, pages 577–586. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-49787-5.

H. Yao, H. J. Hamilton, and C. J. Butz. A Foundational Approach to Mining Itemset Utilities from Databases. In *Proceedings of the Third SIAM International Conference on Data Mining*, pages 482–486, 2004.

N. Ye. *The Handbook of Data Mining.* Mahwah, NJ: Lawrence Erlbaum, 2003.

S. Yella, M. Dougherty, and N. K. Gupta. Condition monitoring of wooden railway sleepers. *Transportation Research Part C: Emerging Technologies*, 17(1):38 – 55, 2009.

U. Yun. A new framework for detecting weighted sequential patterns in large sequence databases. *Know.-Based Syst.*, 21(2):110–122, March 2008.

U. Yun and J. J. Leggett. WIP: Mining Weighted Interesting Patterns with a Strong Weight and/or Support Affinity. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, 2006.

M. J. Zaki. Scalable Algorithms for Association Mining. *IEEE Transactions on Knowledge and Data Engineering*, 12:372–390, 2000.

M. J. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning Journal*, 42:31–60, 2001.

M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New Algorithms for Fast Discovery of Association Rules. Technical report, 1997.

J. Zheng, J. Cheng, and Y. Yang. Generalized empirical mode decomposition and its applications to rolling element bearing fault diagnosis. *Mechanical Systems and Signal Processing*, 40(1):136 – 153, 2013.

E. Zio. Reliability engineering: Old problems and new challenges. *Reliability Engineering and System Safety*, 94(2):125 – 141, 2009.

E. Zio. *The Monte Carlo Simulation Method for System Reliability and Risk Analysis.* Springer Series in Reliability Engineering, 2013.

List of Figures

List of Tables

Glossary

ANN	Artificial Neural Networks
ARM	Association Rule Mining
CCR	Correct Classification Rate
CFM	Centralized Fleet Management
DNM	Double Null Models
FTD	Floating Train Data
KDD	Knowledge Discovery in Databases
K-NN	K-Nearest Neighbours
NB	Naive Bayes
SVM	Support Vector Machines
TASS	Tilt and Speed Supervision System
TP	T-Patterns
UTM	Unified Train Maintenance
WARM	Weighted Association Rule Mining
WW	Weighted WINEPI

List of publications

International Revue papers

- Sammouri, W., Côme, E., Oukhellou, L., Aknin, P. and Fonlladosa, Ch-E., *Pattern recognition approaches for the prediction of infrequent target events: Application on floating train data for preventive maintenance*, International Journal of Production Research (IJPR), Under preparation, 2014.
- Sammouri, W., Côme, E., Oukhellou, L., Aknin, P. and Fonlladosa, Ch-E., *Weighted Episode Rule Mining of floating train data for the prediction of infrequent failure events within a predictive maintenance framework*, IEEE Transactions on Intelligent Transportation Systems (ITS), Under preparation, 2014.

International Conference Papers

- Sammouri, W., Côme, E., Oukhellou, L., Aknin, P., Fonlladosa, Ch-E., Prendergast, K., *Temporal association rule mining for the preventive diagnosis of onboard subsystems within floating train data framework*, 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2012 , pp. 1351,1356, 16-19 Sept. 2012.
- Sammouri, W., Côme, E., Oukhellou, L. and Aknin, P., *Mining floating train data sequences for temporal association rules within a predictive maintenance framework*, Industrial conference on Data Mining (ICDM), 2013.
- Sammouri, W., Côme, E., Oukhellou, L., Aknin, P., Fonlladosa, Ch-E., *Floating Train Data Systems for Preventive Maintenance: A Data Mining Approach*, International Conference on Industial Engineering and Systems Managament (IESM), 2013.

Other contributions

- Sammouri, W., *Temporal association rule mining for the preventive diagnosis of onboard systems within floating train data framework*, Journée des doctorants IFSTTAR 2012 STIC et Métrologie, 2012.