Trainmining

October 2, 2012

Chapter 1

Objectives and context definition

1.1 Introduction

Maintenance is one of the most important tasks to assure the quality and correct operation of any kind of system. Even the highest quality systems, built by the best engineers to operate for long periods with the least possible human assistance, will eventually be exposed to damage or malfunction. In order to avoid the negative effects that system malfunction can produce, a significant amount of resources and effort is usually needed to be put on maintenance tasks. However, putting resources and effort on maintenance procedures might still not be enough if the procedures and strategies are not adequate and efficient.

Traditionally, we have discerned between two types of maintenance procedures:

Corrective maintenance is the most common approach, although it has very important limitations. With this approach, elements of our system are repaired or replaced once they have failed or worn out, to bring them back to operation. This usually means a high downtime in operation, as no actions are taken until our system is already malfunctioning.

Preventive maintenance focuses on preventing these failures. Elements can be periodically examined and analysed in order to control their operation and perform simpler procedures to adjust them before reaching malfunction and downtime. This approach means much higher costs, as a significantly bigger amount of time is needed to monitor the elements on our system and correct them. However, as downtime means business losses in almost all cases, these higher costs usually pay back in terms of loss reduction.

A balance can be easily achieved by spending on preventive maintenance not more than the losses we would suffer from downtime if we were using a corrective approach.

However, the costs of preventive maintenance can be drastically reduced by optimising procedures and using the adequate techniques. For example, we can reduce the amount of variables and magnitudes we are monitoring (and which cost us money to monitor) if we know which ones give the better insight on the status on our systems. The same can be done with corrective maintenance. If we can somehow foresee which systems are going to fail, we can be prepared and reduce impact on our business even if we cannot do anything to prevent its failure.

In both cases, *prediction* can be a key element for maintenance optimisation. Either we know which are the indicators of a system deterioration which we can repair, or we know which systems are going to fail and when to be prepared and optimise corrective procedures. We can even speak of a new type of maintenance - *predictive maintenance* - which embraces several techniques to try and obtain this knowledge of future events.

1.2 Project overview

Trainmining aims to implement predictive maintenance techniques on already-existing maintenance stations of a railway network. These maintenance stations monitor different elements and subsystems over a railway line and raises alarms whenever a line element fails or requires human intervention. Additionally, maintenance workers perform different preventive maintenance procedures, gathering information about several parameters on each element and performing the appropriate actions when needed. Acquisition of values and determination of necessary actions is however not automatised within the maintenance stations, and workers have to manually perform these tasks.

In order to design *predictive procedures* for the railway network, we have a big amount of event logs gathered by the maintenance stations, as well as registries filled by maintenance workers when performing preventive tasks. We will therefore try to extract, from that large amount of data, knowledge on how to predict future events from current observations

In this directon, *Data Mining* techniques can be extremely useful in order to find relations between patterns in environment variables and the occurrence of events, or even relations between events themselves. These relations, which may at first not be apparent for the human mind, can be obtained through different automated learning processes, and thus infer markers which will act as indicators of when and how failures can happen. In order to extract this data we will need to count on a significantly high amount of event logs, gathered during previous years, on which we will apply said techniques.

1.3 Objectives

1.3.1 Alarm-based prediction

Association sequential rules

The main objective of the project is acquiring knowledge on how events are related to each others (in terms of occurrence). For example, some events may be directly triggered by previous ones, having a direct occurrence relation; or might be caused by the same environmental conditions, being most likely for them to happen along the same time periods. As a result, even in cases where event might seem completely unrelated, the occurrence of one of them can give us information on the chances of others happening within a defined time span.

Our objective is to find and analyse these relations and use them to build useful prediction rules. Depending on the parameters we use for our knowledge discovery procedures, we might obtain different types of rules. For instance, varying the temporal resolution of our analysis, we might obtain rules to predict events in terms of months, days or hours. Depending on the timespan we work with, our prediction rules may be useful to prevent failures, to be prepared to fix them, or be completely useless if there is not enough anticipation.

Station type differentiation

In the railway network we are working with, there are different maintenance stations in different railway lines. Neither the maintenance stations or the lines are equal troughout the whole network, and therefore we will have to follow different procedures for each of them. Initially, we will treat every station (and the set of elements under its management) independently, even though we already know their classification and the similarities between them. Unless generalisation is evident and clearly convenient, we will always maintain this separation and obtain a different set of rules for each of them.

Predictive rules evaluation

Due to the characteristics and large size of the available data, we are likely to find a vast amount of frequent sequences and association rules from which not all of them will be useful for maintenance purposes. Different metrics can be applied to evaluate the *importance* of a rule, such as its confidence (its probability to be true on a given situation), the severity of the predicted events, or its support (absolute frequency of the sequence happening).

A comprehensive analysis is necessary to extract the most useful association rules from the set and discard the others, in order to obtain the most efficient set possible. Additionally, the different metrics can even allow predictions to be filtered in real time, according to the available resources or the desired results.

1.3.2 Measuring-based prediction

In order to predict events happening on our railway network, the most natural approach seems trying to observe the environment variations which may be the direct causes of system failures. For example, we might want to analysis which variations of temperature lead to system breakdown, or how slight variations on power supply voltage can indicate a future failure. This is indeed a good approach, and a lot of useful information is very likely to be obtained using this information.

However, automatic acquisition of this kind of data is not a trivial issue. With hundreds of elements along the railway lines, and maybe up to dozens of variables to monitor in each of them, it would require an efficient way to obtain this large amount of data.

Currently, the only source of this kind of information we have is the logs maintenance workers fill when performing preventive maintenance procedures. Although we might be able to actually extract useful knowledge from this kind of data, its implementation would not be immediate or easy, as said automatic data acquisition systems would be needed in order for our rules to be applied.

For these reasons, we will relegate this kind of predictions to a second place, and focus on alarm-based predictions as mentioned in section 1.3.1.

1.4 Database description

As mentioned in section 1.1, we will count on two different types of databases. First of all, a database gathering a timeline of alarms (failures and other events which require assistance) and a database gathering environment variables which can act as indicators for the alarms. Both types of databases will be present for each maintenance station present in the system.

Chapter 2

Data analysis

2.1 Alarm database

The alarm database has is structured as follows:

Table ER_ERRORS

Contains every alarm received by the Maintenance Station. Has the following fields:

- DVNI_ERRORNUMBER - Alarm identifier
- DVNS_ERRORTIME - Timestamp for the alarm
- DVNI_INSTALLATIONCODE - Code of the installation in which the alarm was raised
- DVNI_SENDERINSTALLATIONCODE - Code of the installation from which the alarm was sent (might be different from the one which raised it)

Table IG_INSTALLATIONGENERAL

This table contains information on all the installations. Has the following fields:

- DVNI_INSTALLATIONCODE - Installation identifier
- DVNI_SYSTEMCODE - Type of system, as defined in the "SG_SYSTEMSGENERAL" table
- DVNI_VERSION - System version
- DVAC_SHORTNAME - Short name of the installation
- DVAC_INSTALLATIONNAME - Name of the installation
- DVAC_LOCATION - Location for the installation
- CHK_IS_NODE - Whether it is a node (doesn't directly send alarms, only raise them) or not

Table IG_NODO_INSTALLATION

This table gathers additional information on installations which are nodes. This is, installations that can raise alarms but need a Parent installation to send them. Has the following fields:

- IG_NODO_INSTALLATION - Identifier of the installation which is a node
- DVNI_FATHER_INSTALLATION - Identifier of the parent installation

Alarm information tables ERH_ERRORS_HSL1 or ERS_ERRORS_SAM_ENCE

Both these tables record information on the alarms. Either one or other table is filled depending on which version of the system is installed in the station. However, in terms of information, both contain the following fields:

- DVNI_ERRORNUMBER - Alarm identifier
- MESSAGE_ID - Unique alarm identifier
- MESSAGE_TYPE - Type of alarm, always set as "notification" (not relevant)
- INVOKE_TYPE - Tells whether the alarm has generated itself due to a connection or disconnection (if type is "node") or is generated by a diagnosis system ("saml") or energy system ("energy")
- INVOKE_NAME - Irrelevant, always set to "diagnosis"
- EVENT_TYPE - Defines the type of alarm which has been generated. Its possible values will be described afterwards.
- ADDITIONAL_TEXT - Alarm code
- ADDITIONAL_INFOS - Additional parameters to be shown in error message
- DVNI_ERRORCATEGORY - Alarm severity. Values from 1 to 5 indicating importance of the alarm, or -1 if the alarm indicates recovery from a previous failure.

The "ERH_ERRORS_HSL1" table, has one additional field:

- CLAZZ - Shows the type of system which has sent the alarm

The field "EVENT_TYPE" can have one of the following values:

- fieldElementAlarm - Alarm related to a field element
- fieldElementFailure - Failure in a field element
- operatorInformation - Information to the operator
- imCpuAndCommunications - Related to IM CPU or IM communications
- internalDiagnosis - Internal diagnosis of a system

- operationsDiagnosisCommunications - Communication error in Operation and Diagnosis systems
- ImFecVersions - IM or FEC version
- internalTraces - Internal traces of a system
- operatorCommandAnswer - Answer to an operator command
- CommProblem - Undefined communication problem
- Information - Information message: versions, etc.
- CommunicationsAlarm - Procedures and processes to carry information from one point to other
- QualityOfServiceAlarm - Loss of quality of service
- ProcessingErrorAlarm - SW or processing error
- EquipmentAlarm - Equipment failure
- EnvironmentAlarm - Related to the environment where the system is located
- other - Other

2.2 Statistic analysis

2.2.1 Alarm classification

In order to have a better insight of the provided databases and the mentioned descriptions, a preliminary insight was made, quantitatively analysing some of the parameters which seemed more relevant for alarm definition. Specifically, the chosen parameters are the following:

- EVENT_TYPE
- INVOKE_TYPE
- DVNI_ERRORCATEGORY (Error Category)

The proportion of each kind of alarms in each of the provided databases (Antequera, Camas, Segovia and Sevilla) is as follows:

Hourly timeline

In order to make a first approach to data analysis, we decided to analyse the alarms on a hourly distribution, checking which types of alarms are more likely to happen in different hours during the day. The result is the following:

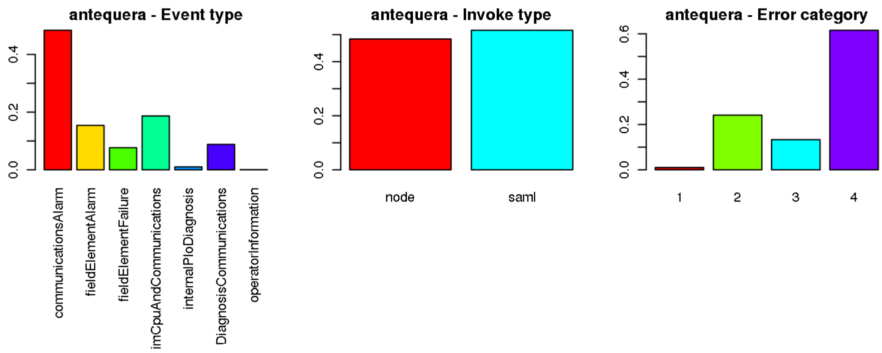

Figure 2.1: Alarm information for Antequera

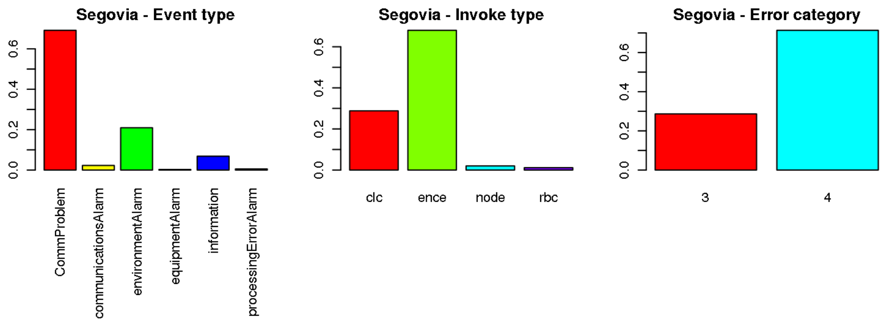

Figure 2.2: Alarm information for Segovia

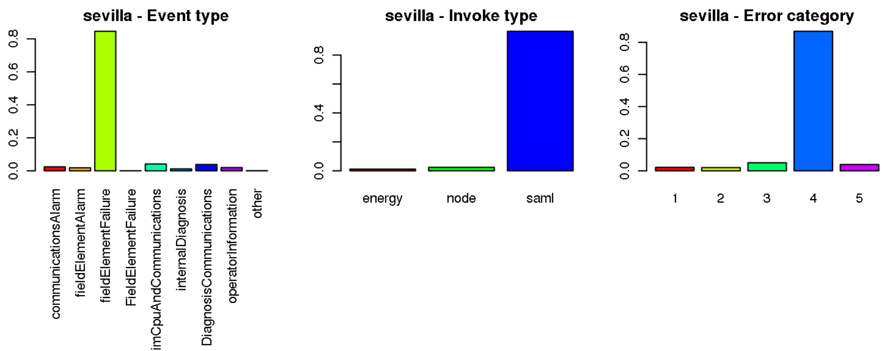

Figure 2.3: Alarm information for Sevilla

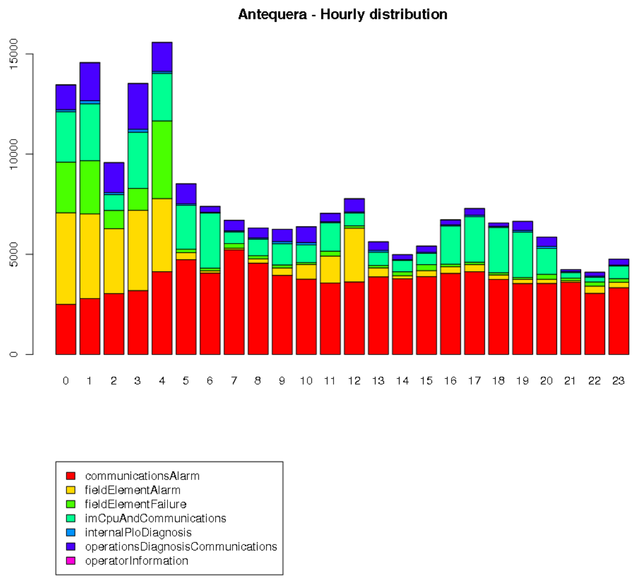

Figure 2.4: Hourly distribution for Antequera (stacked)

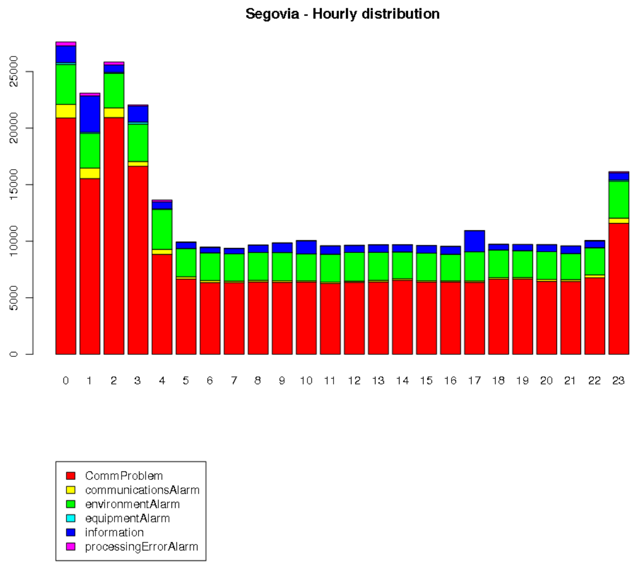

Figure 2.5: Hourly distribution for Segovia (stacked)

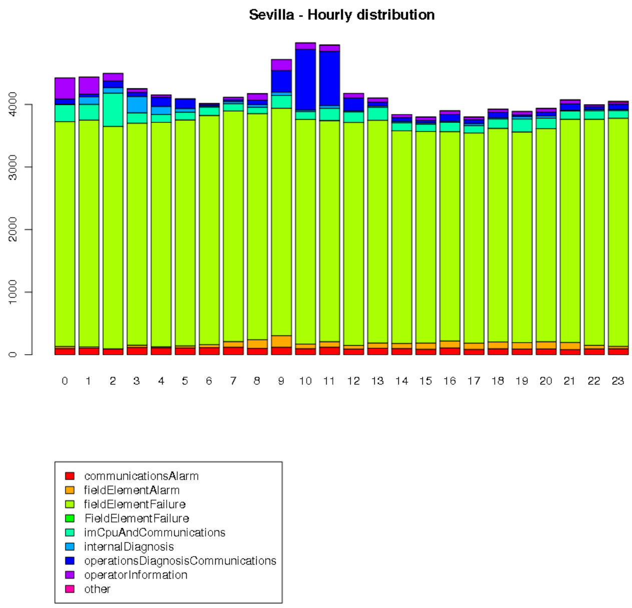

Figure 2.6: Hourly distribution for Sevilla (stacked)

Daily correlation

We have also generated graphics for correlation between number of alarms of each type during the day, and occurrences of other types of alarms. The result is as follows:

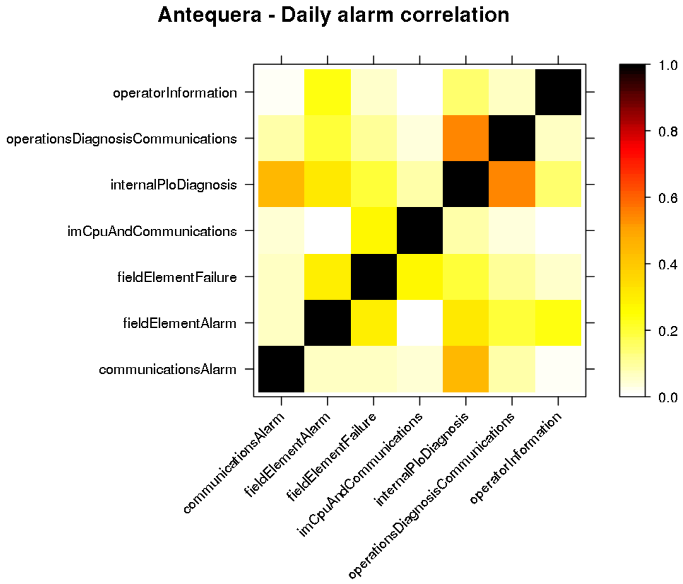

Figure 2.7: Daily correlation for Antequera

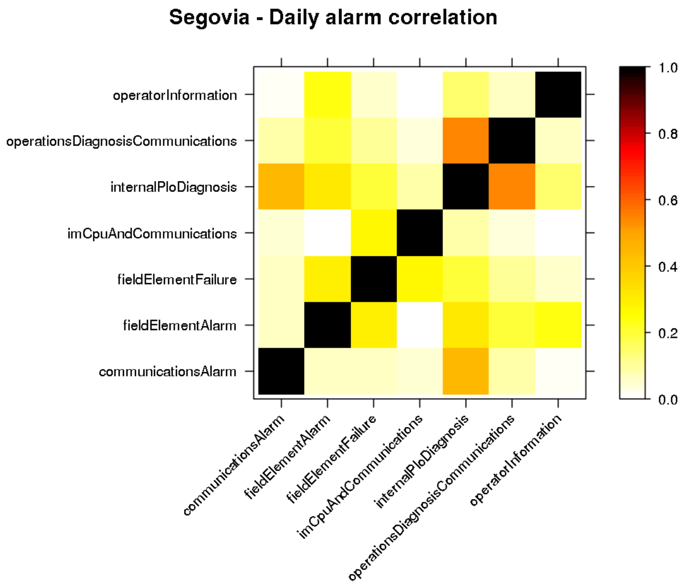

Figure 2.8: Daily correlation for Segovia

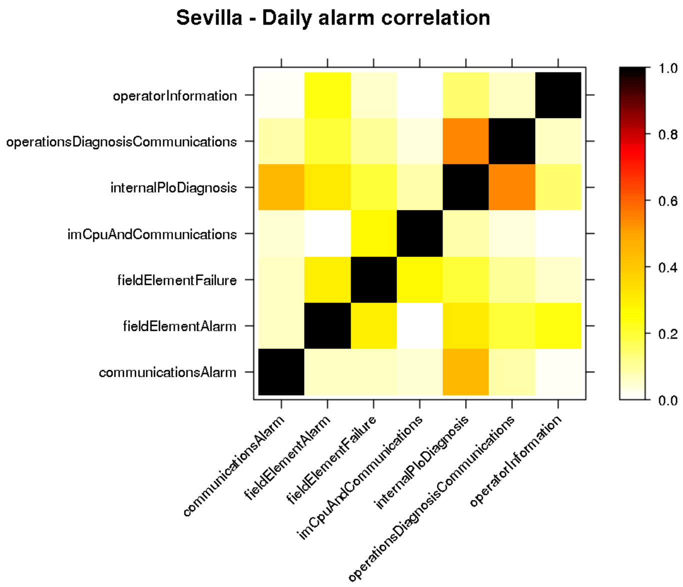

Figure 2.9: Daily correlation for Sevilla

In this first approach, we see that this correlation is not consistent between the different databases. Therefore, information cannot be directly inferred from these results, and additional analysis will be necessary.