H2020-MG-8.1a-2014

INFRALERT: Linear Infrastructure Efficiency Improvement by Automated Learning and Optimised Predictive Maintenance Techniques

Grant Agreement number: 636496

Deliverable D4.3

Methodologies and procedures for inferring three-dimensional alert-severity-intervention pattern space. Supervised and unsupervised approaches

Work Package	WP4. Alert management		
Task	T4.1 Implementing an asset's alert level estimator T4.2. Developing a supervised/unsupervised alert-time-probability/possibility pattern space T4.3 Developing a supervised/unsupervised alert-severity-interventions pattern space		
Revision	Final	**Due date**	31/01/2017
Revision date	31/01/2017	**Submission date**	31/01/2017
Deliverable type	R: Report		

Deliverable leader	University of Sevilla (USE)
Contributing partners	CEMOSA, Lulea Tekniska Universitet, Infraestruturas de Portugal

Dissemination Level		
PU	Public	**X**
CO	Confidential, only for members of the consortium (including the Commission Services)	
CI	Classified, as referred to in Commission Decision 2001/844/EC.	

Document status

Revision	Date	Description
1	09/01/2017	First Draft Version
2	27/1/2017	Version 2
3	31/1/2017	Final
Status	Final	

Executive Summary

The aim of the WP4 “Alert management” is to prioritise assets of the infrastructure, needed of maintenance interventions according to forecasted severity of degradation and failure of the assets themselves, and the know-how brought in by the information recorded in the historical maintenance interventions.

In a previous document, Deliverable D4.2, the activities developed in task 4.1 “Implementing an asset’s alert level estimator” and task 4.2 “Developing a supervised/unsupervised alert-time-probability/possibility pattern space”, were presented. The document also contained relevant preliminary information on the methodologies and procedures of task 4.3 “Developing a supervised/unsupervised alert-severity-interventions pattern space”.

Deliverable D4.3 reflects the activities developed in task 4.3, and includes the new advances and developments, affecting task T4.2, reached to the date of this document, which complement the information provided in Deliverable 4.2.

The work contained in this document presents the approach defined and being developed for predicting the required maintenance interventions, to be carried out, on the interested asset of a linear transport infrastructure in forecasted scenarios. Along the document, diverse methodologies, algorithms and techniques are described, detailed and demonstrated.

The first level of relevant developments are collected in Section 5, where the methodologies for estimating alerts based on: i) the information provided by the forecasted evolving behaviour of asset features and, ii) the information recorded in the historical database of conducted maintenance interventions, are used to determine the most probable intervention to be carried out to keep the asset in a safe state.

A second level of relevance is associated to Section 6, where a selection of results proves the capacity of the inferred models to make predictions.

A third level of general relevance is affiliated to Sections 7 and 8. They define probabilistic and deterministic criteria to set a level of severity to any forecasted alert and its recommended maintenance intervention.

Table of content

List of tables

List of figures

Abbreviations and acronyms

Abbreviation / Acronym	Description
ANN	Artificial Neural Networks
A-WO	Alert-Work Order
CT	Crocodile cracking
DT	Decision Trees
eIMs	Expert-based Infrastructure Management System
FAT	Functional-related Asset-Tree
IRI	International Roughness Index
KNN	K-Nearest Neighbours
MAB	Maintenance Administration Body
RUT	Transversal Unevenness
SVM	Support Vector Machines
TSL	Technical Severity Level
WO	Work Order
WP	Work Package

1 Background

INFRALERT is a European Horizon 2020 project whose aim is to develop an expert-based information system to support and automate linear asset infrastructure management from measurement to maintenance. This enfolds the collection, storage and analysis of inspection data, the deduction of interventions to keep the performance of transport networks in optimal condition, and the optimal planning of maintenance interventions. The results also facilitate the assessment of new construction strategic decisions.

For this purpose, one of the goals of the project focusses on the development of an intelligent alert management system, with the purpose to analyse asset condition and operational information, to provide alerts whenever the infrastructure reaches, or is close to reaching, a critical level in the present time or in a near future scenario. It combines the current and predicted asset condition with operational and historical maintenance data, to get information about the needed maintenance tasks to avoid later severe degradation or mismatching of safety and/or comfort conditions. By means of data analytics and machine learning methodologies, the system generates a prioritised listing (ranked on severity level) corresponding to the alerts generated by all assets of a linear transport infrastructure.

The concept of maintenance alerts is linked to any asset (e.g. part, element, component, subsystem, system) when its functional condition is in danger or will be jeopardized in a future scenario. An alert is generated when the condition of an infrastructure asset crosses a threshold limit value defined by a standard in a specific forecasted scenario and/or through machine learning techniques using the recorded information from previous maintenance interventions. The evolving condition, soundness/unsoundness, of an asset in forecasted scenarios depends on the technical characteristics of the asset itself, the set of other assets implied in the infrastructure, the evolution of the stresses and environmental loads the infrastructure is undergoing, and many other characteristics/attributes denoted under the generic term “feature”.

Service loadings and their frequency acting on an asset are the main reason for a decrease in the reliability and occurrence of failures, which obligate to carry out maintenance interventions. The historical evolution of the asset features and the historical recording of maintenance operations are a valuable source of information to predict further maintenance interventions.

This document describes a methodology for making use of the historical records of features evolution and previous maintenance interventions, in order to grade the severity of the alerts triggered by the estimated condition of the involved assets and the know-how reflected in the maintenance histories.

2 Objectives

The triggering of alerts regarding the health condition of transport infrastructure assets has been customarily based on surpassing deterministic fixed thresholds defined by technical standards prescribed by the corresponding infrastructure administration/regulator. These thresholds are grounded on the accumulated knowledge acquired during a prolonged period of time regarding the adequacy of the assets and respond to a conservative envelope which guarantees the safety, integrity and right performance of the asset itself as a part of the system it works for. The alerts may be triggered by either the appearance of corrective failures, faults or malfunctioning; the preventive programmed actions to be executed to avoid further corrective breakdowns; or the predictive activities to weave future troubles.

The existing knowledge of historical maintenance works, and their corresponding asset condition encountering during the repair, have been used in a non-structured manner to gather a repository of knowledge on severity of failures and the according maintenance interventions carried out, besides the resources demanded, to bring the infrastructure back into service.

A structured database of historical maintenance interventions, founded on quantifying the information available on the said repository of knowledge, paves the way for using data analytic techniques in order to infer tools for decision making, be either fully or semi-automatic.

There follow the procedures and codes to ascertain the reliability of positive alert levels from false positives and false negatives based on existing recorded knowledge of previous maintenance interventions and activities. The use of machine learning, data analytics, data inference and statistic techniques, allows addressing this goal compiling and processing the prior available pieces of information regarding maintenance interventions and operations.

This report presents the methodologies, approaches and models for triggering alerts associated to assets of transport linear infrastructures needed of maintenance interventions, be corrective, preventive or predictive. The estimated alerts are assessed according to the information provided by a decision making tool based on the forecasted evolving state of physical explanatory features relevant to the state condition of interested assets, and the historical maintenance intervention database. The output of the said tool will tag each alert with a hierarchical list of maintenance interventions and their associated probabilities. The report is complemented by a previous document focused on defining the preliminary basic structure, algorithms and techniques of WP4 Alert Management toolkit.

The document is organized in five main parts. Section 4 describes the approach developed for enlarging the historical database of available measurements for the road pilot case, in order to have a larger sample set available for training the developed machine learning models. Section 5 is devoted to a detailed description of the methodologies for estimating alerts based on two information sources: i) the forecasted state condition of those interested assets referred to prospective time scenarios, and ii) the available information recorded in the historical database of conducted maintenance interventions. Explanation of the specific algorithms used and diverse models implemented are also included. Section 6 summarises some of the most important outcomes obtaining during the optimising process of the models, and their assessment analysis. Section 7 details the statistical treatment of the data relevant to alert predictions: i) forecasted asset state conditions provided by WP3, and ii) recorded historical maintenance intervention. Section 8 summarises the defined criteria to infer the severity level of any forecasted alert; these criteria are based on prescribed threshold limits, for the explanatory features relative to those monitored assets.

The results and methodologies presented may undergo modifications, updates and additions according to the advances reached during the development of the project.

3 General overview of methodologies of WP4

Figure 1 shows the different modules and submodules of WP4 and their interaction, as well as the output of each one. The modules were described in deliverable D4.2 (INFRALERT, 2016), but some advances have been carried out in order to improve the modularity and the flexibility of the system. In next sections of this document, these advances are described in depth and all inputs and outputs explained.

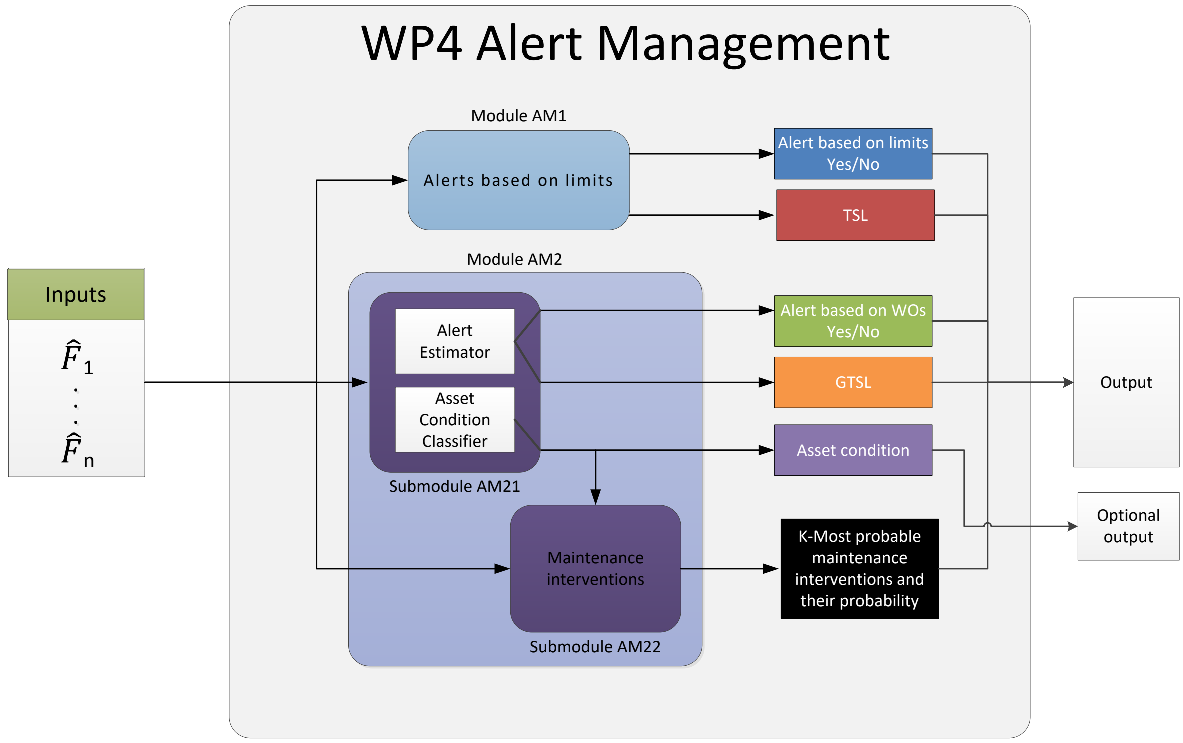
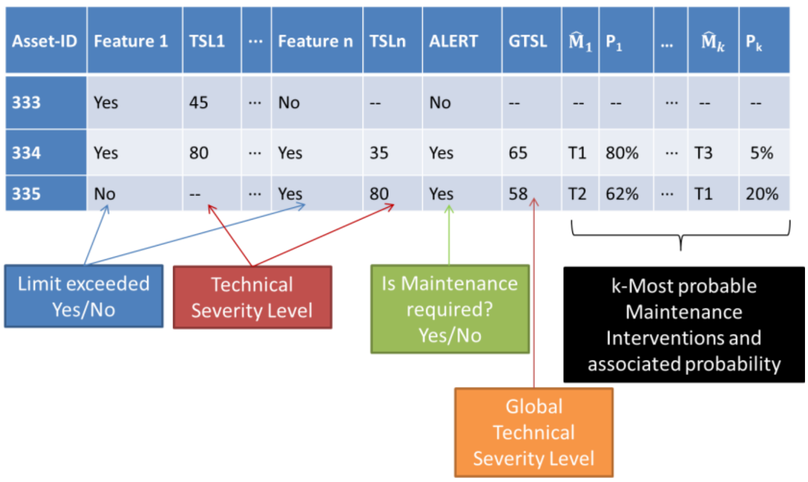


FIGURE 1. WP4 ALERT MANAGEMENT

Figure 2 displays an example of WP4 outcomes wherein the different outputs from each module/submodule are identified with a colour.



Asset-ID	Feature 1	TSL1	...	Feature n	TSLn	ALERT	GTSL	$\hat{M}_1$	P_1	...	$\hat{M}_k$	P_k
333	Yes	45	...	No	--	No	--	--	--	...	--	--
334	Yes	80	...	Yes	35	Yes	65	T1	80%	...	T3	5%
335	No	--	...	Yes	80	Yes	58	T2	62%	...	T1	20%

FIGURE 2. OUTPUT EXAMPLE

4 Advances in the data simulator

Section 5.1 in deliverable 4.2 (INFRALERT, 2016) describes the implementation of a simulated database used to test the different automatic machine learning models designed. The records of this database contain the parameters commonly used for support decision making in asset management. In practice, these parameters about asset condition will come from measurement campaigns but, for the purpose of WP4 development, have been randomly created by a simulation tool.

As explained in deliverable 4.2, for parameter simulating, the tool simply selects one item from the range of feasible values, using a probability density function as the probability of selection. The histogram of the selected data should roughly approximate the shape of a curve of the probability distribution.

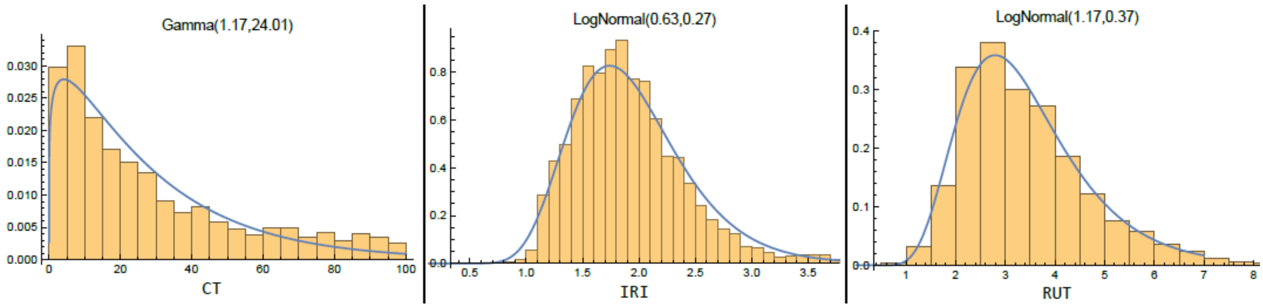
New advances have been achieved related to the inclusion of new probability density functions based on empirical distributions. Formerly, the generation of the simulated database was focused on obtaining samples in the range of each feature for each road type. The previous methodology used a truncated normal distribution for each range and their statistics (i.e. mean and variance) were established to generate the sample set values within the feasible range.

With the improvement introduced, the randomly simulated records generated with the new methodology represent better the real condition of the assets currently measured in the road case study. For this purpose the histograms of each measured feature, for all road types, have been analysed and new probability density functions have been fitted. The procedure has been applied to the road case study and is explained hereinafter.

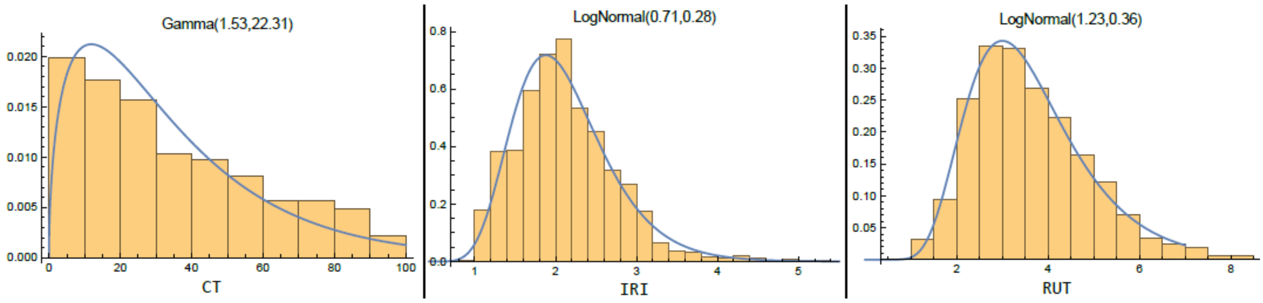
Each feature is analysed individually. At a first step, a histogram is generated considering the values of a specific measurement campaign conducted on a given road type. Several kinds of distribution functions are fitted to every histogram. The distribution function that globally fits better to all histograms is chosen as the statistical distribution associated to the feature. Once the specific distribution function type of a feature is selected, a calibration of the probability density function parameters is performed independently for each road type.

Figure 3 shows the final result where a probability density function was obtained for each feature and road type. In particular, the CT feature has been fitted using the gamma distribution and both IRI and RUT features have been fitted using the lognormal distribution.

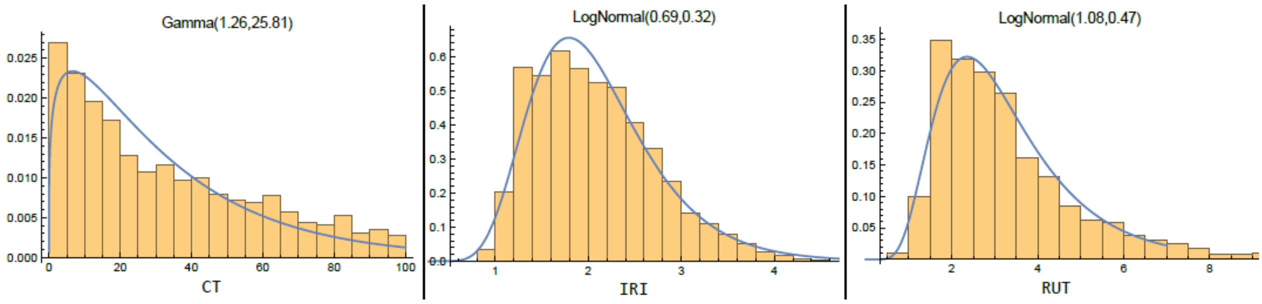
At this point it is important to underline that the described simulating approach developed has the purpose of generating a larger number of measurement samples inferred from the original limited real data set, in sucha way that the statistical properties of the simulated sample coincide with those of real data set, and the physical quantitative values are also inside the ranges defined by the real data. This way of proceeding ensures that the simulated data set used, for developing and testing the methodologies, algorithms and models in WP4, conforms to the data nature of the pilot road case and similar to other road cases.



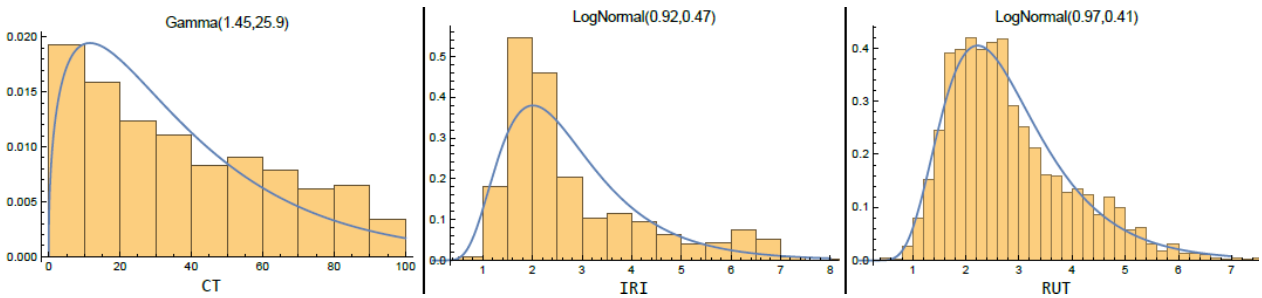
(A)



(B)



(C)



(D)

Figure 3. Calibrated probability density functions. (A) network class 1 (2501), (B) network class 2 (2503), (C) network class 3 (2521), (D) network class 4 (2522)

5 Advances in Module AM2 (alerts based on WOs)

This module triggers the alerts using the recorded information from previous maintenance interventions through a machine learning processing. Moreover, a set of most probable maintenance interventions associated to the alerts is inferred by this module using the same historical maintenance interventions database.

The basic operation of this module was presented in the previous deliverable (INFRALERT, 2016). However, some improvements have been carried out in order to have a more modular and flexible piece of software. In the previous release, the alerts and their associated most probable maintenance intervention were predicted in a simultaneous way by the same single module. This was possible because the asset condition was correlated with the most probable maintenance through pre-defined know-how guidelines provided by the maintenance manager (see Table 1 by IP where maintenance types are described in Table 2); this approach forces the machine learning models correlate a group of feature's values with just one maintenance type. However, the correlation described by pre-defined maintenance intervention guidelines (Table 1), despite being a good initial approximation, does not reflect a generic and realistic situation because in most cases maintenance types are not directly associated with asset conditions but influenced by other factors; two assets under the same state condition could be fixed with different maintenance types depending on external factors (e.g. budget, available machinery, merging criteria to fix different sections belonging to the same asset, maintenance policies, time opportunity). It is for that reason that module AM2, in the new release, has been divided in two submodules, AM21 and AM22. Submodule AM21 triggers alerts depending on the values of features and classifies the asset condition depending on the level of each feature. On the other hand, submodule AM22 provides a set of most probable maintenance interventions (and their associated probabilities) according to historical data.

Asset condition ID	Feature Level			Type Maintenance			
	CT	IRI	RUT	Network class 2501	Network class 2521	Network class 2522	Network class 2503
A1	H	H	H	T4	T4	T3.1	T3.1
A2	H	H	L	T4	T4	T3.1	T3.1
A3	H	L	H	T4	T4	T3.1	T3.1
A4	H	L	L	T3.1	T3.1	T2	T2
A5	M	H	H	T4	T3.1	T3	T3
A6	M	H	L	T3	T3	T3	T3
A7	M	L	H	T4	T3.1	T3	T3
A8	M	L	L	T3	T2	T1	T1

TABLE 1. ALERT IDENTIFICATION AND ASSOCIATED TYPE OF MAINTENANCE ACCORDING TO NETWORK CLASS

Type of Maintenance	Description
T1	Do nothing
T2	Microsurfacing, Surface dressing
T3	Thin Hot-Mix Asphalt overlay (thickness less or equal to 5 centimeters)
T3.1	Surface milling with Thin Hot-Mix Asphalt overlay (thickness less or equal to 5 centimeters)
T4	Thick Hot-Mix Asphalt overlay (thickness greater than 5 centimeters) combined or not with milling

TABLE 2. DESCRIPTION OF TYPES OF MAINTENANCE

5.1 ALERTS

In this section, submodule AM21 is described, whose aim is to determine and trigger alerts based on the values reached by explanatory features. The submodule methodology is built on supervised machine learning techniques, which are deeply described in the previous deliverable 4.2 (INFRALERT, 2016). In addition, this submodule provides a classification of the asset state condition quantified on different levels reached by either individual features, combined features or a global one. This last piece of information is used (as an option) by submodule AM22.

To start the process, submodule AM21 has to be initialised. The initialisation is carried out with recorded information from previous maintenance interventions. A basic sketch of the flow diagram is depicted in Figure 4.

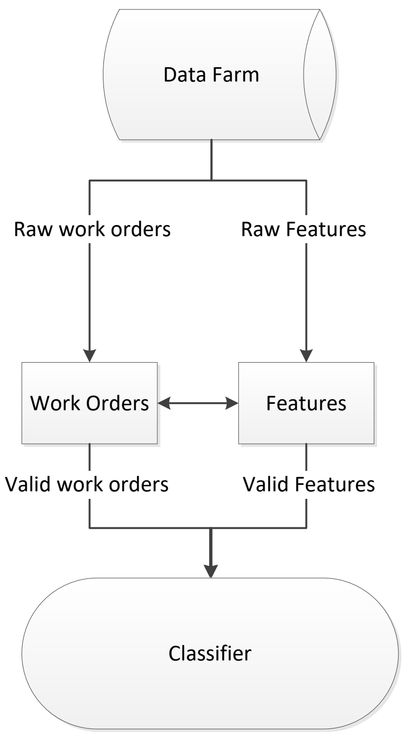


FIGURE 4. INITIALISATION OF THE CLASSIFIER

The Data farm, a repository containing the historical maintenance recording of the monitored infrastructure, provides all work orders ("raw work orders") which, after being retrieved are filtered to sieve those valid record according to the type of maintenance; this process is also conducted to those physical explanatory features associated to the work orders.

These pieces of information determine the inputs to a machine learning frame (INFRALERT, 2016) constituted by two initial classifiers, one of them triggers alerts and the other classifies the asset condition.

Once both classifiers are defined, submodule AM21 is prepared to predict alerts using the forecasted value for each feature provided by WP3.

Then, to set this submodule operative as an estimating tool, it has to undergo two stages: a learning process stage (section 5.1.1) and a predicting stage (sections 5.1.2 and 5.1.3) which provides the outcome to be transferred to further steps of WP4 Alert Management.

5.1.1 Training of machine learning models

The first stage of a machine learning model implies a training process with the adequate information, available in the historical maintenance database, in order to extract the know-how included in it. Three main set pieces of information are exploitable: measurements carried out in the analysed linear asset section/segment, associated to physical explanatory features; endogenous and exogenous characteristics/variables related to state condition evolution (this set can also be integrated in the previous set); and the performed maintenance interventions. All that information can be correlated to infer relevant outcome data for predicting purposes.

More specifically, the data inputs for the training stage can be listed herein below:

- Asset identification.
- Values of the considered features (F_i).
- Subjective evaluations associated to any individual feature (A_i).
- Subjective evaluations associated to a combination of features (C_i).
- A global subjective evaluation associated to the condition of the asset as a whole (G).
- Requirement for maintenance (Yes/No).
- Maintenance intervention executed (Yes/No). NOTE: It is possible to have a maintenance executed with the value of the previous field stated as "No". This case appears when the maintenance performed is either preventive or predictive.
- Classification of the maintenance executed (CM): Corrective/preventive/predictive.

All these inputs are retrieved from the historical maintenance database (Table 3 shows a sample record set).

Asset-ID	F_1	…	F_n	A_1	…	A_m	C_1	…	C_p	G	M	CM

Table 3. Inputs for machine models' training

As mentioned before, this submodule AM21 is made of two parallel predicting blocks. Block AM21AE (Alert Estimator) is used to trigger the alerts using a classifier. The classifier just correlates the value of the available measures with the requirement of maintenance. Then, the required inputs for the training are the value of the features, and the target variables are the field "Requirement for maintenance" (Figure 5). The learning system consists of an automatic classification in two labels (Yes/No) and therefore any automatic binary classification machine learning can be used (e.g. Decision tree, ANN, KNN, SVM…).

FIGURE 5. TRAINING OF THE CLASSIFIER "ALERT ESTIMATOR", BLOCK AM21AE

On the other hand, the second block AM21ACC (Asset Condition Classifier) correlates feature measures with different subjective evaluations of the asset condition provided by the maintenance manager team. This way, the system will "learn" from the Maintenance Manager know-how and, when a new measure is introduced in the system, it predicts the asset condition without the intervention of the maintenance manager (there is no need to make use of a specific maintenance team to evaluate the asset state condition). According to this, the inputs for the training are the value of the features and the target variables will be the subjective evaluations (Figure 6).

FIGURE 6. TRAINING OF THE CLASSIFIER "ASSET CONDITION CLASSIFIER", BLOCK AM21ACC

Depending on the information provided by the historical maintenance database, it is possible to training the model just with either the subjective evaluations associated to an individual feature (A_i), the subjective evaluations associated to a combination of features (C_i) or with the global subjective evaluation associated to the global asset condition (G). This global asset condition can be seen as the overall performance of the asset regarding the simultaneous contribution of all feature effects as a whole.

Following the same philosophy, the Asset Condition Classifier (AM21ACC) can also be divided into smaller and more accurate classifiers. As an example, in case the Maintenance Manager is only interested in the explanatory contribution of one feature (e.g. F_1), the system could be simplified as shown in Figure 7.

FIGURE 7. TRAINING OF THE CLASSIFIER “ASSET CONDITION CLASSIFIER” JUST FOR FEATURE F_1, BLOCK AM21ACC

5.1.2 PREDICTION OF ALERTS

Once the system is initialised and the machine learning models are trained, the objective is to detect asset segments where maintenance will be required in the future. To do this, the “Alert Estimator” (AM21AE) has to be applied to the forecasted value of the features ($\hat{F}_i$) provided by WP3. This will be the only information available and therefore the only input to the system. The forecasted value of each feature is used as input to the system, which provides just an output. This output will be the requirement for maintenance; in case an intervention is needed, an alert will be triggered (Figure 8).

FIGURE 8. PREDICTION OF AN ALERT, BLOCK AM21AE

The relevant inputs for the alert estimation are the following (Table 4):

- Asset ID.
- Forecasted value of features ($\hat{F}_i$).

Asset-ID	$\hat{F}_1$	...	$\hat{F}_n$

TABLE 4. INPUTS FOR ALERTS’ PREDICTION, BLOCK AM21AE

The outputs are the alerts (Table 5).

Asset-ID	ALERT
	Yes/NO

TABLE 5. OUTPUT FOR ALERTS’ PREDICTION, BLOCK AM21AE

5.1.3 Prediction of Asset Condition

As previously mentioned, the goal of block AM21AC is to predict the asset condition without the intervention of the maintenance manager; that is, to estimate the different subjective evaluations of the asset condition that would be provided by a hypothetical maintenance inspection. In this way, there is no need to make use of a maintenance team to evaluate the state of the asset by physical inspections.

For each subjective evaluation ($\hat{A}_i, \hat{C}_i, \hat{G}$) a classifier is defined and, in general, this may be a not binary classifier. This rule has as exception the ANN, because all subjective evaluations can be predicted with just a single model.

As in the previous block AM21AE, the inputs will be the forecasted values of features ($\hat{F}_i$), but in this case the output will be the predicted values of the subjective evaluations (Table 6).

Asset-ID	$\hat{A}_1$	…	$\hat{A}_m$	$\hat{C}_1$	…	$\hat{C}_p$	$\hat{G}$

TABLE 6. OUTPUT FOR ASSET CONDITION'S PREDICTION, BLOCK MA21AC

Figure 9 shows a generic case in which the assessment of the global asset condition is predicted as a function of all features. In this example, just three levels have been considered: Low, Medium and High (L/M/H, respectively).

FIGURE 9. PREDICTION OF A SUBJECTIVE EVALUATION, BLOCK MA21AC

5.2 Maintenance interventions and alert associated probabilities

In AM21 submodule, the alerts triggering is based on asset's features. Now it is necessary to define the most probable interventions associated to those alerts in order to ease the decision making by the decision support tool (WP6). To do this, unsupervised and supervised machine learning techniques are used. All the following methodologies will have the same inputs and outputs but the accuracy of the results depends on the particular distribution of the data included in the historical maintenance database. The manager can use all of them to compare results and act accordingly.

Relevant inputs for defining the models are:

- Asset identification.

- Values of the considered features (F_i).
- Type of maintenance performed (T) (defined in Table 2).
- Output from submodule AM21 ($\hat{A}_i, \hat{C}_i, \hat{G}$).

Once the models are defined, submodule AM22 will be applied using the forecasted value of the features ($\hat{F}_i$) as inputs, and the output will be the prediction of the *K* most probable maintenance interventions that can be applied, according to historical interventions carried out in similar conditions ($\hat{M}_1,\ldots,\hat{M}_k$) and their probability (P_1, …, P_k) (Table 7).

Asset-ID	$\hat{M}_1$	P_1	…	$\hat{M}_k$	P_k

TABLE 7. OUTPUT OF THE SUBMODULE AM22

In the following sections, three different models used to define these maintenance interventions are presented.

5.2.1 CLUSTERING OF HISTORICAL DATA

The applied methodology consists of an unsupervised machine learning algorithm module in which the available data is grouped into different clusters. The application of this approach depends on the data because it requires the existing clusters are more or less well-defined. Assuming this hypothesis, the historical data will define *L* clusters so that samples with similar condition (similar values for all features) are grouped in the same cluster. The clustering is carried out taking into account just the asset condition (value of features F_i) but it does not use the real conducted maintenance intervention. In this way, the same cluster could contain different types of maintenance as it can be seen in the example presented in Figure 10. In this figure only two features have been considered, but several maintenance types are involved, identified by different plotting shapes (i.e. squares, triangles and circles) identifying distinct types of maintenance interventions.

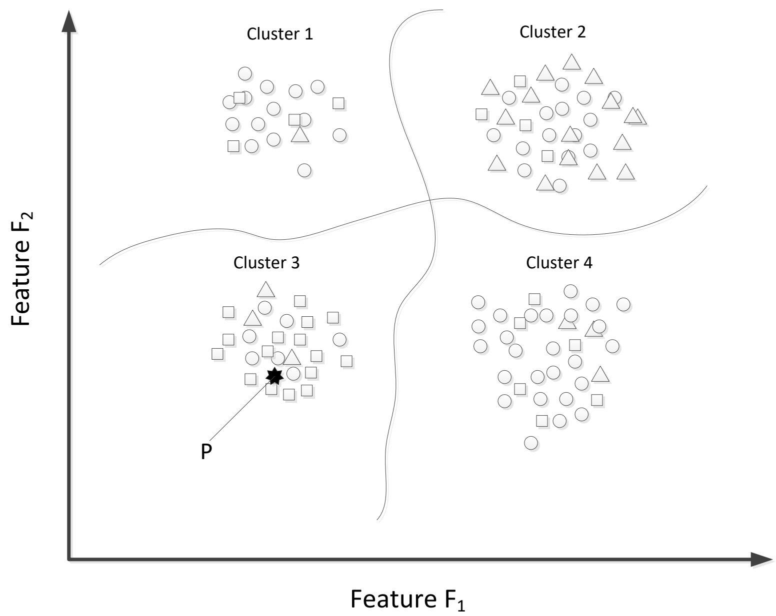


FIGURE 10. EXAMPLE OF CLUSTERING. EACH MAINTENANCE TYPE IS REPRESENTED BY A DIFFERENT SYMBOL

In order to determine the most probable types of maintenance associated to an alert, the forecasted values of the features ($\hat{F}_i$) that triggered the alert are introduced in the model. Their feature values are plotted in the N-dimensional space (depending on the number of features, N) and the corresponding sample *P* is identified (Figure 10). This sample *P* will belong to a specific cluster *k* (based on the distance to the centroid). The set of maintenance interventions that appear in this cluster defines the possible maintenance types associated to the alert, being the probability of each intervention a function of the empiric occurrence of each one in such a cluster *k-th*.

To take into account the Maintenance Manager's know-how, collected through the subjective evaluations, the empiric occurrence of each maintenance type can be weighted by scores giving more importance to those samples whose evaluation equals to the predicted evaluation of sample *P* ($\hat{A}_i, \hat{C}_i, \hat{G}$) given by submodule AM21. According to this example, Figure 11 shows a zoom on the cluster sample *P* belongs to, together with the subjective evaluation of the features for the rest of samples contained in the cluster. According to the forecasted value of features for sample *P,* submodule AM21 predicts a condition H (High) for the first feature and M (Medium) for the second one; those samples with the same labels (H/M) will be scored with a higher weight than the rest. In this way, an unsupervised machine learning approach is complemented with the information provided by the Maintenance Manager.

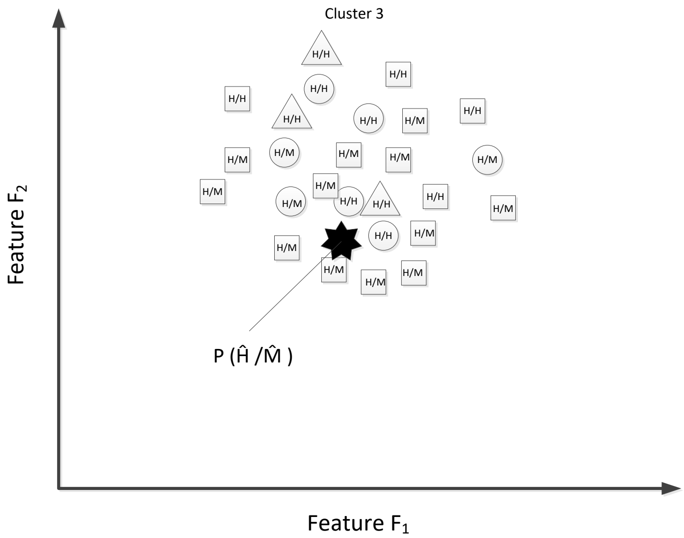


Figure 11. Example cluster 3

Using this clustering technique, a huge number of assets with similar condition can be used for estimating possible maintenance strategies, circumventing recurrent problems such as bias.

5.2.2 K-Nearest Neighbours algorithm (KNN)

The methodology described in the previous section has serious limitations when data do not show a clear grouping. This problem can be solved using the *k*-Nearest Neighbours Algorithm which was described in a previous document (INFRALERT, 2016). In this case, the *k* nearest samples to a given sample *P* are chosen to define the most probable types of maintenance associated to an alert (Figure 12). From now on, the methodology is similar to the previous one considering just these *k* samples. This means that the selected samples and sample *P* belong to the same "cluster" in order to apply the previous technique.

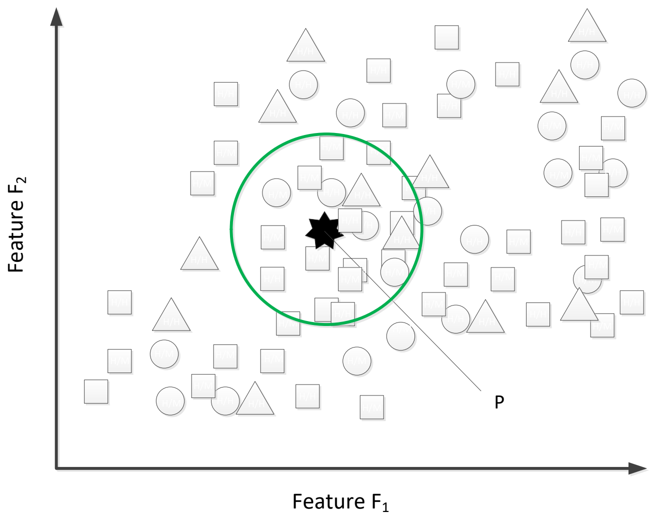


Figure 12. Example KNN

Following the same approach, the Maintenance Manager's know-how can be taken into account by giving more importance to those samples with the same evaluation than the predicted one for sample *P* ($\hat{A}_i, \hat{C}_i, \hat{G}$) given by submodule AM21 (Figure 13). In the example, the occurrence of each sample (of the k-nearest one) is weighted by a score that depends on its evaluation; higher for those equals to the predicted evaluation of the considered sample *P* than for the rest.

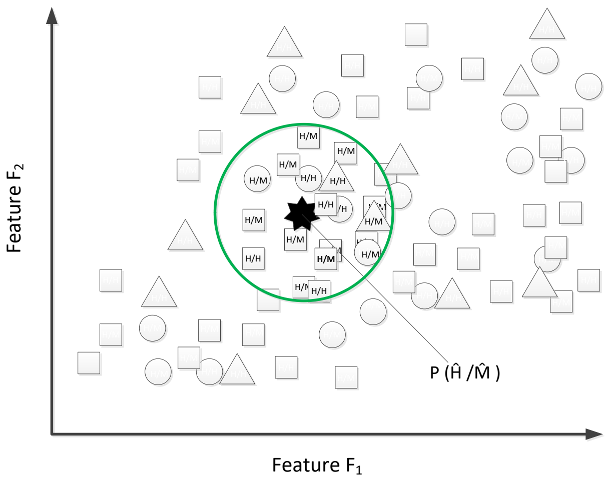


Figure 13. Example using simultaneous KNN plus Maintenance Manager know-how approach

This methodology is less sensitive to the distribution of data than the clustering but, in general, it uses fewer samples to define the most probable maintenance types and some information can be screened out depending on the value of parameter *k*.

5.2.3 Fusion Model (KNN+ANN).

To take into account the advantages of the two single methodologies developed (KNN and ANN), a third one obtained by integrating both of them has been constructed. The steps to implement this fusion model are summarised in:

1. To generate the cluster model proposed (section 5.2.1).
2. To compute the Euclidean distance between sample *P* and each centroid of the *L* clusters (d_{CE-1}, …, d_{CE-L}).
3. To choose the *k* nearest samples to *P* using the distance *Z* between sample *P* and a generic sample *Q* belonging to cluster *J*.

$$z_{P-Q} = d_{P-Q} \cdot d_{CE-J}$$

where d_{P-Q} stands for the Euclidian distance between *P* and *Q.*

In this way samples belonging to different clusters, than *P's,* are penalised.

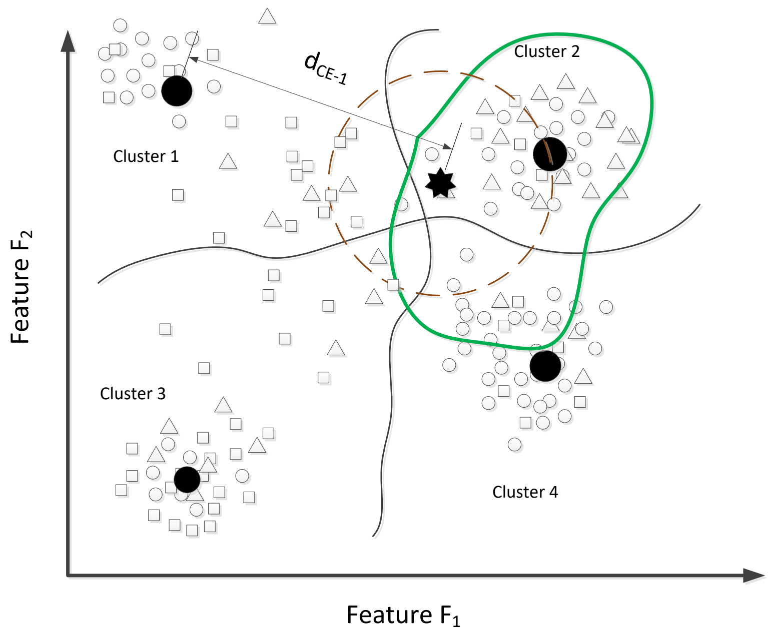


Figure 14. Example of Fusion Model

Figure 14 shows an example of the difference between applying the KNN model (brown dotted line) and the fusion model (green solid line). As it can be seen, this last fusion model is not so dependent on the existence of a clear data grouping (cluster definition) and enables to increment the value of *k* in a guided way, regarding those samples belonging to the same cluster than *P*.

As in those previous single models, Maintenance Manager's know-how can be taken into account by giving more importance to those samples with the same evaluation than the predicted one for sample *P* ($\hat{A}_i, \hat{C}_i, \hat{G}$) given by submodule AM21.

6 Optimisation of machine learning models

This section extends former sections 4.2 and 5 included in deliverable 4.2 (INFRALERT, 2016). Section 4.2 introduces the automatic classification concept where maintenance interventions are predicted by means of asset forecasted conditions values. Section 5 describes the procedure used to train the models and performs a capability analysis of forecasting maintenance intervention types using diverse machine learning techniques. These techniques are not fully optimized and only preliminary results are shown.

This section describes the calibration process of the generated machine learning models, which are implemented by means of Matlab code (Matlab, 2015; Martinez and Martinez, 2002). In particular, the models are based on Decision Trees (DT), k-Nearest Neighbours (KNN), Support Vector Machines (SVM) and Artificial Neural Networks (ANN). The main objective is to build models with good generalization capabilities to perform well on new data (i.e. test data for which the model has not been trained). The model performance is strongly related to the concept of over-fitting, which impedes a reliable prediction. Another objective is to determine the sample size required for each model to be able of generalizing.

As described in a previous document (INFRALERT, 2016), a simulated dataset is used to explore the fundamental parameters of the machine learning models implemented. This simulation is developed under the assumption that the different faults that can occur in the monitored infrastructures are known. Furthermore, the measurable features that warn of these faults are known. Once the system is full operative, machine learning models will be fed and trained with real maintenance interventions; at present, the models are trained with a simulated dataset where feature ranges are known. With this simulated dataset, the relationships between training set size, model complexity, and prediction error, have been analysed.

There are in fact two separate goals addressed:

- **Model selection:** estimating the performance of different models in order to choose the best one. This action includes the calibration of parameters for each model.
- **Model assessment:** once a final model is chosen, estimating its prediction error (generalization error) on new data. Assessment of this performance is extremely important in practice, since it guides the choice of learning method or model, and gives a measure of the quality of the ultimately chosen model.

As the sample set used is a data-rich situation, since the inputs come from a simulated dataset, the dataset is randomly divided into three parts: a training set, a validation set, and a test set. The training set is used to fit the models; the validation set is used to estimate prediction errors for model selection; and finally, the test set is used for assessing the generalization error of the final chosen model (Hastie et al., 2009). To estimate the prediction error of the final model and the unknown tuning parameters, a cross-validation method is used; the cross-validation function trains a model using a supplied formula and modeling function, then the model performance is tested on a held-out test set. The training set will be sampled from the data available for training. This method divides the simulated dataset into k disjoint subsamples (or folds), randomly chosen but with roughly equal size.

6.1 MODEL SELECTION

At this stage the complexity of models are tuned. For small training sets, there is less opportunity for averaging out the noise, and unintended correlations might influence the predicting capacity of the model. In large training sets, the random noise tends to be averaged out, so the underlying patterns are clearer. Therefore, a large sample size is almost compulsory to be used.

As case of use (INFRALERT, 2016), the alerts and work order (A-WO) simulator has been applied to the road case using the information provided by the Road Administrator based on its accumulated expertise. However, the procedure can be also applied to the railway case as the physical nature of the problem is similar to the road one: features coming from measurements may suggest specific maintenance interventions are to be carried out.

To perform the optimization of models (adjusting their complexity), a sample of 2000 simulated maintenance interventions is used. The selection of this sample size is based on experience based on the number of model inputs and possible alerts (outputs). This sample size allows obtaining models without under-fitting behaviour and which generalize well. The figures describing the optimization for each model are shown in Figure 15 .

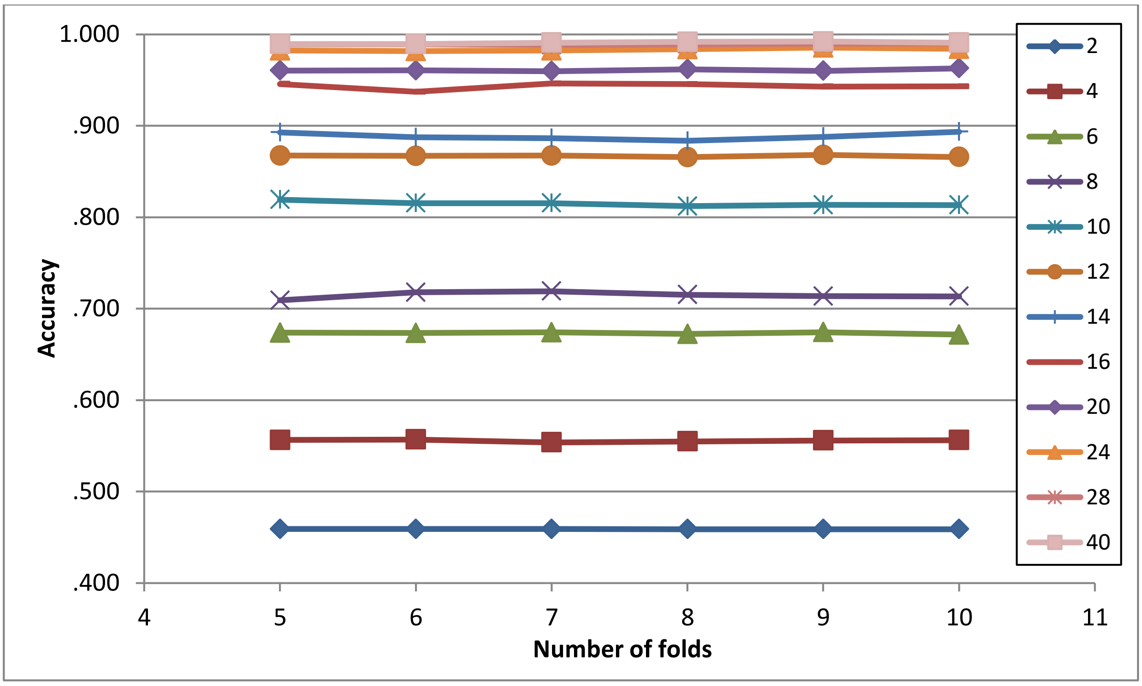


FIGURE 15. ACCURACY OF DECISION TREE METHOD MODIFYING THE NUMBER OF SPLITS

The main parameter to be calibrated in decision tree (DT) methods is the number of splits or branches. Figure 15 shows 12 series of simulations where the number of splits ranges from 2 to 40. The horizontal axis represents different simulations of cross-validating tests with different fold size. The first and last simulations divide the sample in 5 and 10 folds, each fold consisting of either 20% or 10% of the sample set, respectively. Based on the results, it is checked that the size of the folds does not modify the accuracy. As shown in Figure 15, a DT model with 40 splits achieves the same results as a 28 splits DT model. Other parameters have been calibrated but their influence is not very noticeable.

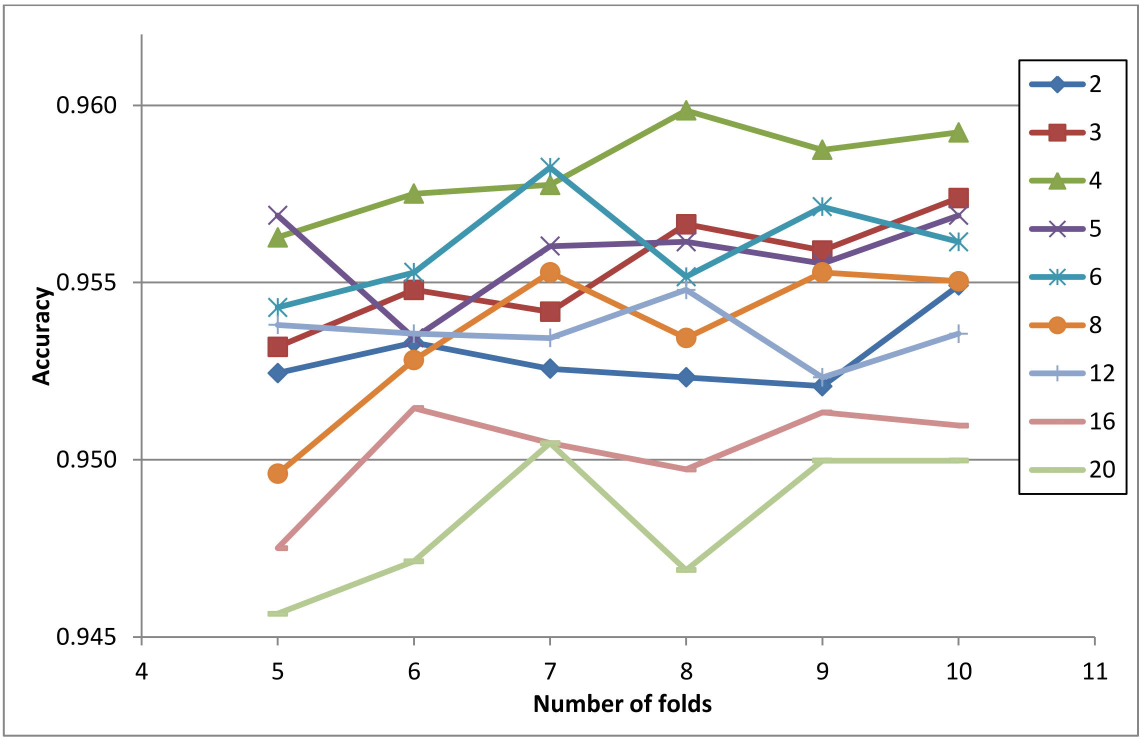


FIGURE 16. ACCURACY OF K-NEAREST NEIGHBOURS METHOD MODIFYING THE NUMBER OF NEIGHBOURS

In k-Nearest Neighbours classification method, an object is classified by a majority vote of its neighbours. Figure 16 shows a total of 9 methods that ranges from 2 to 20 neighbours. As in Figure 15, the horizontal axis represents different simulations of cross-validation method with different fold size. In this case, the method with 4 neighbours gets the best performance although there is only a difference of 1.5% between the best and worst method. It is observed that the choice of the number of folds in which the sample is divided may improve the final accuracy of the method although up to a maximum of 0.4%. In both methods, DT and KNN, the selection of the number of folds is not determinant; there are no bias or variance issues. This is because the size of the sample is large enough and all possible cases are collected in both test and train sets. Therefore, any value can be selected.

The final calibration of this method includes the choice of additional parameters such as distance metric and distance weighting function. This step is intensive since there are many options for these two parameters. The optimal choice has been the standardised Euclidean distance metric and squared inverse distance weighting function. Using the standardised Euclidean distance, each coordinate difference between rows is scaled by dividing by the corresponding element of the standard deviation S (though it is possible to specify a different value for S). The influence of each neighbour on the model decreases with distance (this case in particular as the squared inverse distance weighting function has been selected).

Kernel function	Accuracy
linear	0,889
quadratic	0,960
polynomial of degree 3	0,969
polynomial of degree 4	0,965
Gaussian	0,948

TABLE 8. ACCURACY OF SUPPORT VECTOR MACHINE METHOD MODIFYING THE KERNEL FUNCTION AND KFOLD

It has been previously concluded that the number of folds, in which the sample is divided, is not an influential parameter. In next two methods only a cross-validation testing that divides the sample in 7 folds is performed. Therefore the results of Support Vector Machine (SVM) method can be summarized in table format.

The SVM method is mainly characterized by its kernel function. In Table 8 it is shown that a 3-degree polynomial kernel function is the best option (case with kfold=7). The kernel scale has also been calibrated manually but the obtained results do not improve those obtained by the default setting in Matlab.

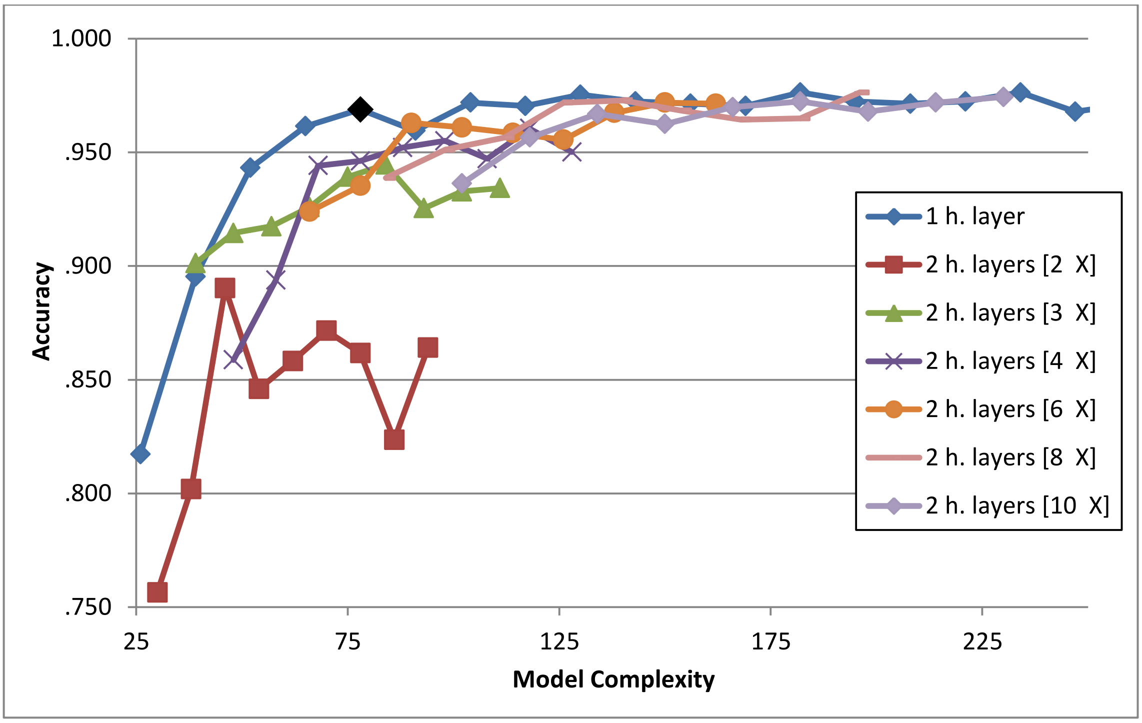


FIGURE 17. ACCURACY OF ARTIFICIAL NEURAL NETWORK METHOD MODIFYING THE NUMBER OF NEURONS

The Artificial Neural Network (ANN) is a more complicated algorithm from the calibrating point of view due to the number of parameters to be configured. In this deliverable the design of the number of hidden layers and neurons is just addressed. A pattern neural model, with Levenberg-Marquardt backpropagation training function and mean squared normalized error performance function, has finally been selected. The validation set ratio is set to 15%, and the convergence parameters of the training method are kept at default values.

Again a unique cross-validation methodology with kfold parameter set to 7 is carried out. Figure 17 shows seven series of different ANN designs, whose description follows:

- 1 h. layer: Only a hidden layer. The number of neurons ranges from 2 to 20.
- 2 h. layers [2 X]: Two hidden layers. First hidden layer has 2 neurons and second hidden layer neuron number ranges from 2 to 10.
- 2 h. layers [3 X]: Two hidden layers. First hidden layer has 3 neurons and second hidden layer neurons from 2 to 10.
- …

In order to compare the different proposed ANN designs, the model complexity parameter is defined as the sum of the weights to be trained in each model. Model complexity increases as the number of neurons and hidden layers increase.

Figure 17 displays that models with only one hidden layer (1 h. layer) achieve better performance. For complexity greater than 110 there are several models that reach high accuracy. The optimal model is one that achieves high accuracy with less complexity. The smaller the complexity, the smaller the sample needed to train the model. The ANN model chosen has one hidden layer with only 6 neurons (black sample in Figure 17); the complexity is 78, unlike the model with 8 neurons that reaches a similar accuracy but increases complexity up to 104.

Once the models have been selected using a large sample set, Table 9 presents and compares the accuracy achieved. In this case, the DT model reaches higher accuracy.

Model	Accuracy
DT	0.989
KNN	0.960
SVM	0.969
ANN	0.969

TABLE 9. ACCURACY OF SELECTED MODELS

Finally, it is worth highlighting a last remark regarding the tests. As the model becomes more and more complex a greater adaptability to more complicated underlying structures is reached. However, the generalization error increases due to the model trying to fit all data. Hence a decrease in bias and an increase in variance are obtained. There is some intermediate model complexity that gives minimum expected test error (Hastie et al., 2009). This behaviour is not shown in the previous results, since Matlab algorithms performs an internal regularization avoiding variance models, no over-fitting are reported.

6.2 MODEL ASSESSMENT

As already mentioned, model performance assessment is extremely important in practice, since it guides the choice of learning method or algorithm, and provides a measure of the quality of the ultimately chosen model.

In this section, the expected test error of each estimated model is calculated using learning curves. These plots illustrate the important issue in assessing the ability of a learning method to generalize. In these curves, both the training error and the test error are shown; x-axis represents different training set sizes and y-axis the error.

The conducted tests of each selected model follows the sequence: i) a sample set of 1000 simulations is generated; ii) for a fixed training set size (50, 100, 150,…,700), several random extractions from the sample set are performed; iii) following a bootstrap method, each extraction is trained and tested, and a train and test error to this set is obtained. In addition, for each extraction, the 20th and 80th percentiles are computed.

The learning curves of selected models are shown in Figure 18 to Figure 21, where a total of 6 lines are included. The two main lines represent the average value of the train and test set (thick lines); the rest of lines are generated with the 20th and 80th percentiles of train and test set and provide an insight of the variance of the predictions (thin lines).

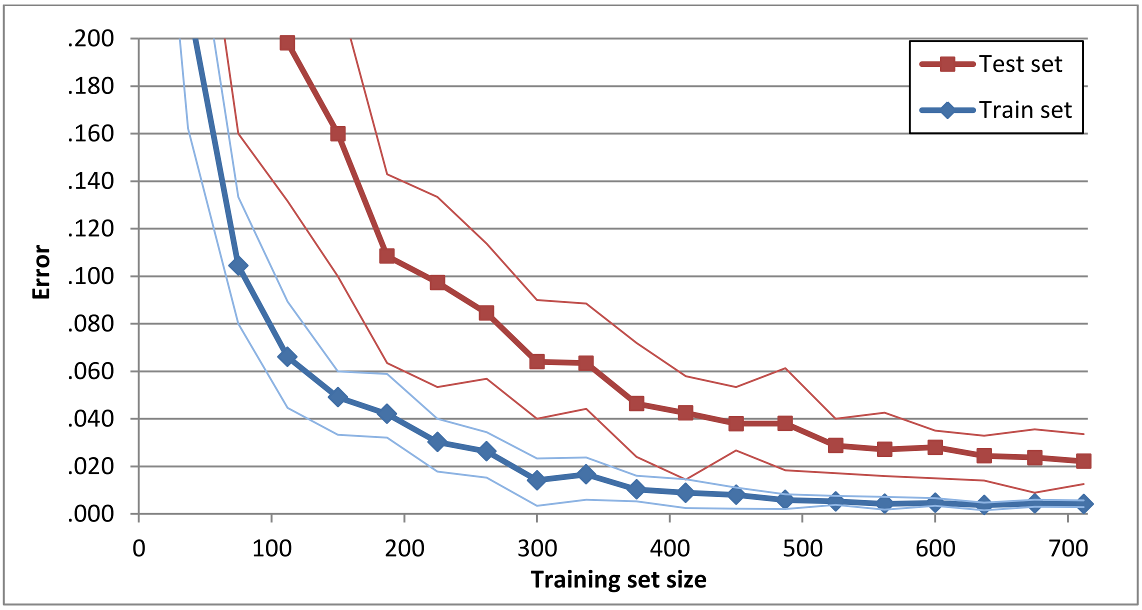


FIGURE 18. LEARNING CURVE OF DECISION TREE MODEL

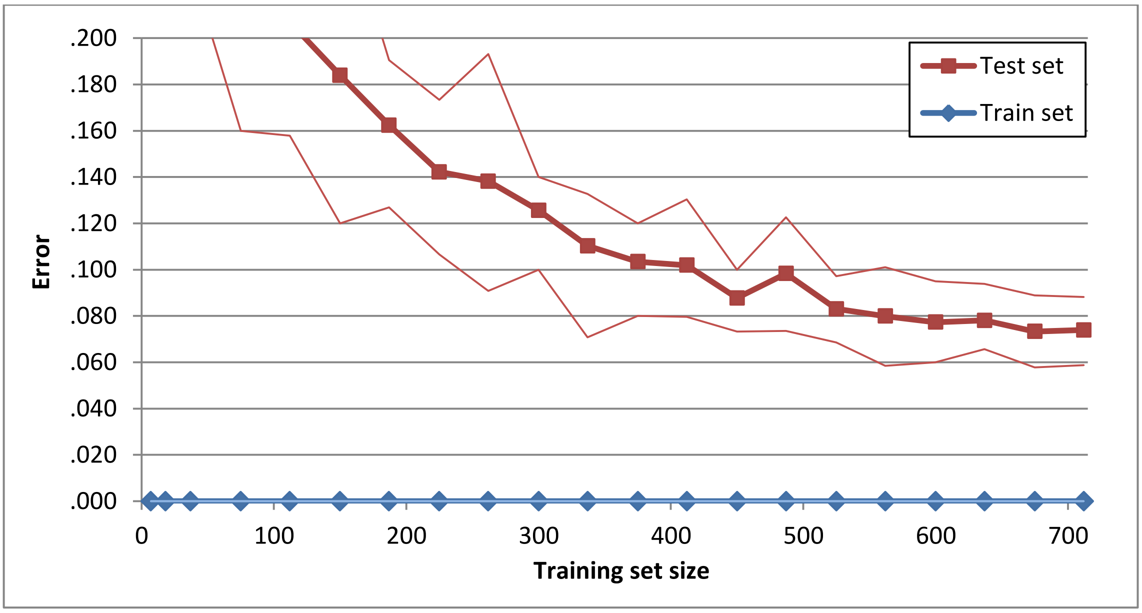


FIGURE 19. LEARNING CURVE OF K-NEAREST NEIGHBOUR MODEL

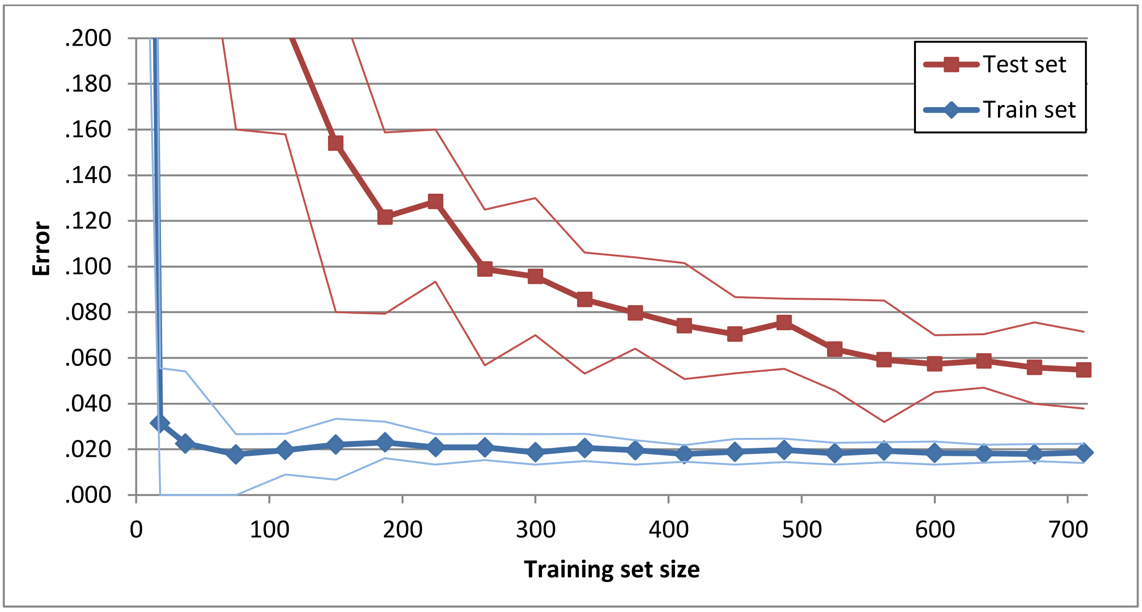


FIGURE 20. LEARNING CURVE OF SUPPORT VECTOR MACHINE MODEL

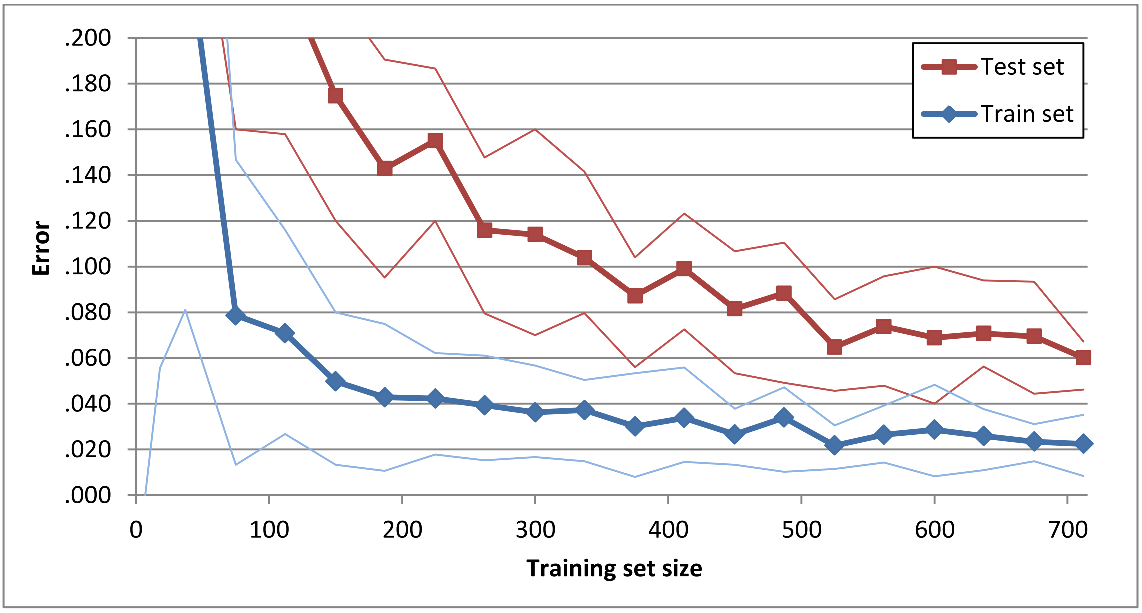


FIGURE 21. LEARNING CURVE OF ARTIFICIAL NEURAL NETWORK MODEL

As the training set sizes get larger, these curves converge toward a threshold representing the amount of irreducible error in the data. These plots were generated using a simulated dataset where the magnitude of the irreducible error is precisely known. In this case the inclusion of a percentage of erroneous samples has not been considered and therefore a good model should achieve a 0% error.

The error of the train set prediction, in the k-Nearest Neighbour model, is always 0%. This is because the prediction of a record is calculated from a neighbourhood in which that record is included.

The most relevant data of the learning curves shown are tabulated in Table 10.

	DT	KNN	SVM	ANN
Final Test set average error	2.21%	7.39%	5.48%	6.01%
Final difference between test and train average error	1.8%	7.39%	3.62%	3.76%
Size of training set to reach a test error < 5%	375	-	-	-

TABLE 10. SUMMARY OF RESULTS

DT model achieves the smallest error. SVM and ANN models yielded similar results and neither of them reduces the test set average error below 5%. Only DT model has reached an average test set error of less than 5% using a training set size of 375 records.

By attending to the evolution of the test set error curves, the error values are still decreasing over a training set size of 700. Therefore, if larger training sets are used, smaller errors would be expected. Although it is clearly noticeable the curve trend is almost horizontal, pursuing a smaller error decrease might need a larger number of records to be available.

It is worth to underline that the results presented corresponds to a simulated database, which has been used to design the models. This is a data-rich case. In real practice, once the models are designed, opposite problems arise. As a general rule, the historical maintenance database provides a limited valid sample set due to various causes: a) few data on maintenance interventions are recorded with sufficient rigor; b) the information stored lacks of chronology consistencies (i.e. real maintenance interventions are not described with the same rigor along time, the condition state before and after maintenance interventions are not consistently assessed by maintenance personnel); c) not sufficient physical explanatory features are quantified. Therefore, it is important the selected models achieve a low average test set error with limited number of records. In this case, DT model is the best option, as it needs a smaller number of simulations to generalize the problem.

7 Statistical treatment of input data

A relevant input for the WP4 Alert Management is the asset condition provided by WP3 Asset Condition. WP4 is interested not just in the actual state condition of any asset of the transport infrastructure but also in the forecasted state conditions of further temporal scenarios. Forecasted state conditions will be provided by WP3 and, as usual, the predictions will have associated uncertainties. There are different ways to quantify the features in each scenario:

- By a probability distribution,
- by a value (mean) and a standard deviation,
- by a confidence interval,
- non-probabilistic theories and tools such as possibility and fuzzy approaches, among other more recent though less disseminated/accepted ones.

Restricting to the three first, all of those forms have more or less the same statistical treatment from the point of view of WP4 operational framework. In this section, a brief description of the WP4 processing of these inputs is exposed.

7.1 Asset nowcast and forecasted condition

In this section asset nowcast and forecasted state conditions are to be clarified. This issue was deeply discussed in deliverable 4.2 (INFRALERT, 2016) and a brief summary is displayed.

Focusing on a particular asset-segment, *a-i th*, the current and the estimated evolving conditions will be taken as features themselves, defined by $X_p\big|_{a-i}$, and may be referred to one or several other independent features of the assets, denoted by $X_j\big|_{a-i}$ where subscript *j* stands for the *j-th* feature. The assessed condition $X_p\big|_{a-i}$ can be either a design parameter, a feature monitored by (periodic or random) launched inspections and auscultations, or a new feature created for this purpose. In both cases these asset condition features will constitute new records to be stored as new features in the system database, in order to be retrieved when needed.

For the sake of illustration Figure 22 depicts the estimated asset condition $X_p\big|_{a-i}$, function of the independent variable X_t (e.g. time), showing a sample of three values corresponding to the evolution of the feature, in future scenarios; the most probable value is identified by the bold broken line in the middle of two extreme border lines corresponding to lower/higher probabilities according to some statistical reliability (e.g. 0.955, 0.783). The vertical cross-section lines stand for the values of the independent variable X_{tm+1} $X_{tm+2,}$ X_{tm+3} at three future scenarios; the nowcast scenario X_{tm} is pinpointed by the square dot. The horizontal thresholds shown define the Normal Limit (L_N) and Exceptional Limit (L_E), denoted by RT_i and RT_{i+1} respectively, of the asset condition criteria set by the relevant standard on design parameters. Other thresholds affecting the track geometry quality can also be considered, as it is the case of the Alert Limit (AL), Intervention Limit (IL) and the Immediate Action Limit (IAL) indicators. According to the criteria defined by the applicable standards (e.g. European Standard, EU Member State Standard, Rail Administration Standard), the asset condition will be quantified according to the proximity the forecasted values show respect to the specified limits. Besides, the previous cited probabilities of the estimated condition values are also a very valuable piece of information to assess the severity of each scenario.

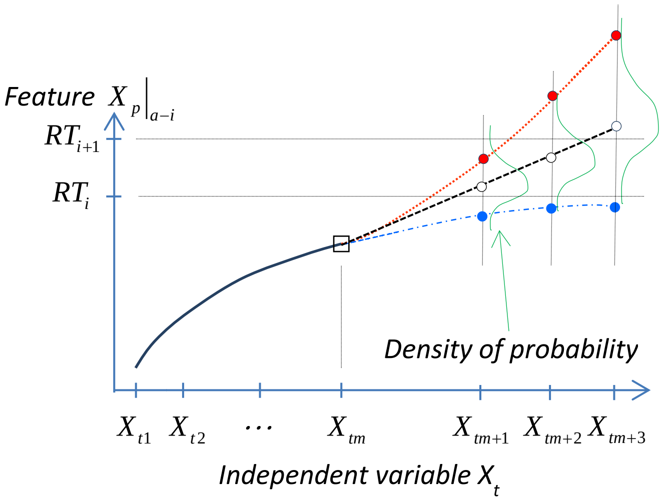


Figure 22. Asset condition prediction

Asset nowcast and forecast conditions will be stored in the Data Farm and can be retrieved at convenience as an input to WP4. WP3 will be the provider of these data when the INFRALERT system is fully implemented.

This explanation (and the following ones) is general for forecasted features with a known probability distribution. If the predicted feature is given in other manner (e.g. by the mean and standard deviation, by confidence intervals), WP4 transforms it to a probability distribution.

7.2 Processing of WP4 modules

An alert is generated when the condition of an infrastructure asset crosses a threshold limit value defined by a standard, in a specific forecasted scenario, and/or through machine learning techniques using the recorded information from previous maintenance interventions. WP4 has a module for each alert type and applies specific processes of the aleatory variables in each of them.

7.2.1 Module AM1: Alerts based on limits

As mentioned in Deliverable 4.2 (INFRALERT, 2016), there are different limits that are able to trigger an alert. Those limits can be for instance:

- Normal Limit (L_N) and Exceptional Limit (L_E) of the asset condition criteria set by either relevant Standards on Design Parameters, or more constraint limits set by the competent Maintenance Administration Body (MAB).

- Alert Indicators regulating the transport infrastructure quality, as for example the Alert Limit (AL), Intervention Limit (IL) and the Immediate Action Limit (IAL). In most cases these indicators are based on know-how developed by the competent body responsible of the infrastructure exploitation.

Then, the goal of this module is to compare the value of the predicted feature with the corresponding limit and act in consequence. As it is earlier mentioned, the forecasted state condition

of the asset will have associated a degree/level of uncertainty. The comparison procedure with the limit is not straightforward.

The alert estimator module (AM1) uses the evolution of the mean values of each feature and its density distribution function to predict a future state condition of the infrastructure asset. For every value of the independent variable $X_{,t}$ a distribution probability curve of the feature X_p can be predicted by WP3. The mean curve joins the expected mean values of these probability distributions of the feature X_p conditioned to the X_t values, X_p (X_t). The Υ percentile curve joins the points with a probability Υ of finding X_p (X_t) values above these points. The inclusion of the probability of failure allows users to specify the desired level of reliability for each feature. The reliability level of the asset design could be based on the general consequence of reaching the terminal (end-of-life) state condition earlier than the predefined design life.

The expected mean value of a feature corresponds to a probability of failure equal to 50%. The maintenance policy may set that an alert is triggered only when a feature exceeds a prescribed threshold limit with a γ probability of failure, or correspondingly when its reliability does not reach the 1-γ limit. An example is shown in Figure 23 for the IRI feature in the road case.

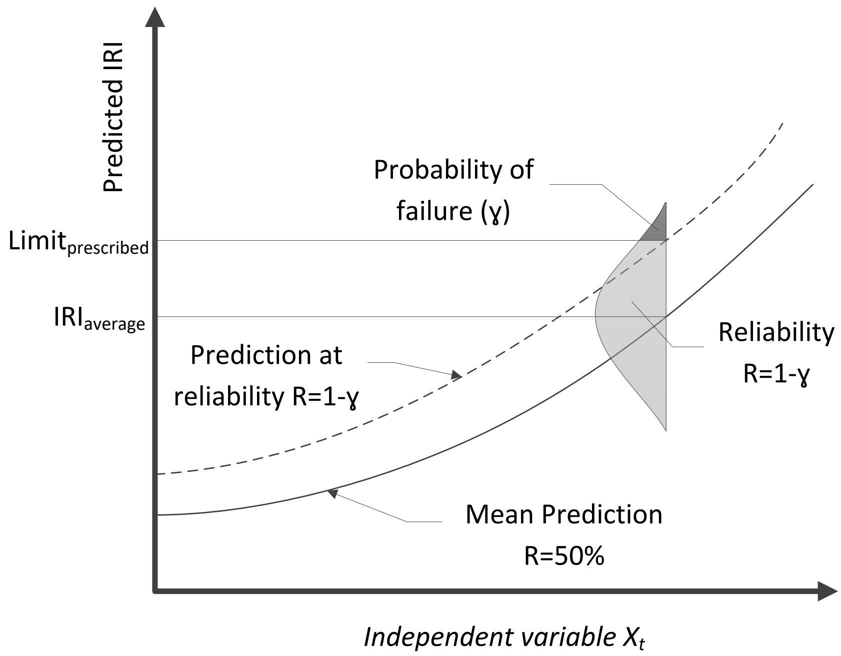


FIGURE 23. PROBABILITY DENSITY FUNCTION OF AN ASSET STATE CONDITION MODEL

In this example, the alert will be triggered according to equation (1). Depending on the value of γ, an alert may be either considered or neglected even when the mean value of the feature is the same.

$$P(IRI > Limit_{prescribed}) \geq \gamma \rightarrow ALERT \quad (1)$$

The value γ is a means to demand a higher reliability for those features that are considered more relevant from the maintenance activity point of view. The assignment, by the maintenance planner, of a low value of γ to a particular feature means that the feature is very influential and a low possibility of failure is allowed. In this case 1-γ quantifies the reliability.

Looking at Figure 23 it is straightforward to rewrite the equation (1) as:

$$\text{Prediction (at realibility 1-}\gamma) \geq Limit_{prescribed} \rightarrow ALERT \quad (2)$$

In many cases, based on the experience and know-how of the user, it may be appropriate to handle some additional features to monitor, control and summarise the asset state condition. As a general rule, these new features will be a combination of some of the original ones.

Taking the road case as an example, it is possible to define a quality index as a function of different measured features. In this case the new combined feature, denoted as pavement quality index (IQ), is defined by the original features IRI, RUT (R) and CT (C) through equation (3).

$$IQ_t = 5 \cdot e^{0.0002030 \cdot IRI_t} - 0.002129 \cdot R_t^{\ 2} - 0.03 \cdot C_t^{\ 0.5} \quad (3)$$

This new feature has associated some limits that trigger the corresponding alerts with a determined severity level as in the case of individual features. Following the example, the limits of the IQ feature are prescribed by four ranges shown in Table 11.

Asset condition	Limit
Bad	IQ≤1.5
Poor	1.5<IQ≤2.5
Fair	2.5 < IQ ≤ 3.5
Good	IQ>3.5

TABLE 11. LIMITS FOR THE FEATURE IQ

As it can be seen, a combined feature can be treated in the same way as the individual ones due to the fact that the combined feature has associated its own limits. However in this case, the probability distribution is unknown. To deal with this issue, Module AM1 uses an empirical distribution function associated with samples of the individual features. So, it chooses a large number of samples of the individual features and with them the combined one is calculated. The obtained values are sorted in ascending order and the empirical distribution function is given by (4):

$$F(x) = \frac{\text{Number of elements in the sample} \leq x}{n} = \frac{1}{n}\sum_{i=1}^{n} 1\{x_i \leq x\} \quad (4)$$

Where $1\{x_i \leq x\}$ is the indicator of event $\{x_i \leq x\}$ and n is the number of samples. For a fixed x, the indicator $1\{x_i \leq x\}$ is a Bernoulli random variable with parameter $p=F(x)$, hence $n \cdot F(x)$ is a binomial random variable with mean $n \cdot F(x)$ and variance $n \cdot F(x) \cdot (1-F(x))$. This implies that $F(x)$ is an unbiased estimator for $F(x)$.

So far, the alert is triggered when the probability of failure exceeds the limit γ. This definition of alert can be generalised through the value of the asset condition technical severity level (TSL). So when this TSL is greater than a predefined parameter α, an alert is triggered (a more detailed explanation is provided in section 8.1).

Returning to the example of Figure 23, a possible definition for the TSL_γ could be (5):

$$\begin{aligned} &TSL_\gamma = \text{Prediction (at realibility 1-}\gamma) - Limit_{prescribed} \\ &If\ TSL_\gamma \geq \alpha \rightarrow ALERT \end{aligned} \tag{5}$$

Doing α=0:

$$\begin{aligned} &0 \leq TSL_\gamma = \text{Prediction (at realibility 1-}\gamma) - Limit_{prescribed} \\ &Limit_{prescribed} \leq \text{Prediction (at realibility 1-}\gamma) \rightarrow ALERT \end{aligned} \tag{6}$$

the alert is exactly the same as defined previously by equation (1).

This example shows that, with generic values of TLS and α, it is possible to define a general alert and Module AM1 can be easily modified to take into account different strategies. In section 8.1 a non-exhaustive list of definitions for TSL is provided.

7.2.2 Module AM2: Alerts based on Work Orders

In this module, alerts are not based on limits but triggered using the recorded information from previous maintenance interventions through machine learning approaches (INFRALERT, 2016).

To explain the way inputs are statistically addressed, it is stated for granted that the machine learning module (Module AM2) has been previously trained (see INFRALERT, 2016 for a detailed explanation on the training) and an initial classifier is previously defined.

The next step of the procedure continues by requesting, through the Alert Generation toolkit of WP4 Alert Management, for estimated alert-maintenance interventions in future scenarios of interest by the user (e.g. maintenance manager, planner). At the same time, WP4 interrogates WP3 for forecasted values of all features for each interested monitored asset (e.g. all assets of the analysed infrastructure, set of assets of a particular section) in the scenario of concern (see Figure 24). These features will have associated uncertainties that quantify and assess the probabilities of the estimated values.

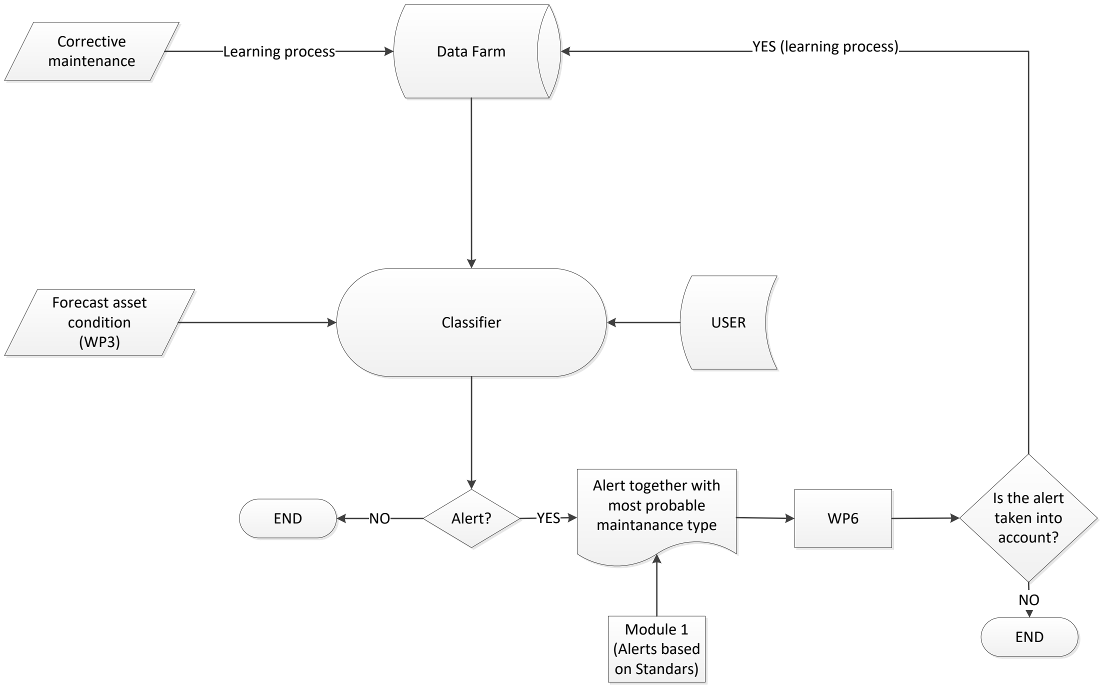


FIGURE 24. WP4 MODULE AM2 WORKFLOW

As mentioned before, this module has two different functional submodules. The first one (AM21) is devoted to triggering alerts and the second one (AM22) to defining the most probable maintenance interventions that have to be conducted. The techniques and procedures are different for both submodules because alerts have to be identified taking into account the probability of failure stating the specification of the desired level of alert reliability. However, in order to specify the maintenance intervention, the most probable asset condition should be handled instead of the condition provided by the desired level of reliability; this is so because, in general, forecasted asset state conditions will be a more realistic and reliable piece of information. In this way, the alert is triggered under the condition of only permitting a pre-specified probability of failure γ (similar to the case of just one feature described previously), but the forecasted maintenance interventions associated to this alert will be calculated using the most probable asset condition (e.g. the expected value, said mean, of the forecasted feature distribution). To clarify this issue, Figure 25 shows an example, involving just one feature with an evolving behaviour according to two different probability distributions ("dist 1" and "dist 2"). The solid line stands for the feature value "Expected value" with a 0.5 probability (in both distributions); the blue broken line exemplifies the feature evolving values under a hypothesis of $R=1-\gamma$ reliability under probability distribution function "dis 1"; the orange broken line represents the presumed feature values with under the hypothesis of $R=1-\gamma$ reliability and "dis 2" probability distribution. As shown, distribution "dis 2" (orange) implies that the alert is triggered before case of distribution "dis 1" (grey), for identical reliability level *R*, because the value used to define the alert is higher. This can be explained by the fact that the dispersion of distribution "dis 2" is higher and it is more probable that higher feature values come into place.

On the other hand, WP6 designs maintenance plans based on the most probable maintenance interventions taking place, instead of the most severe one and therefore, that is the reason, the expected value is used regardless of the probability distribution.

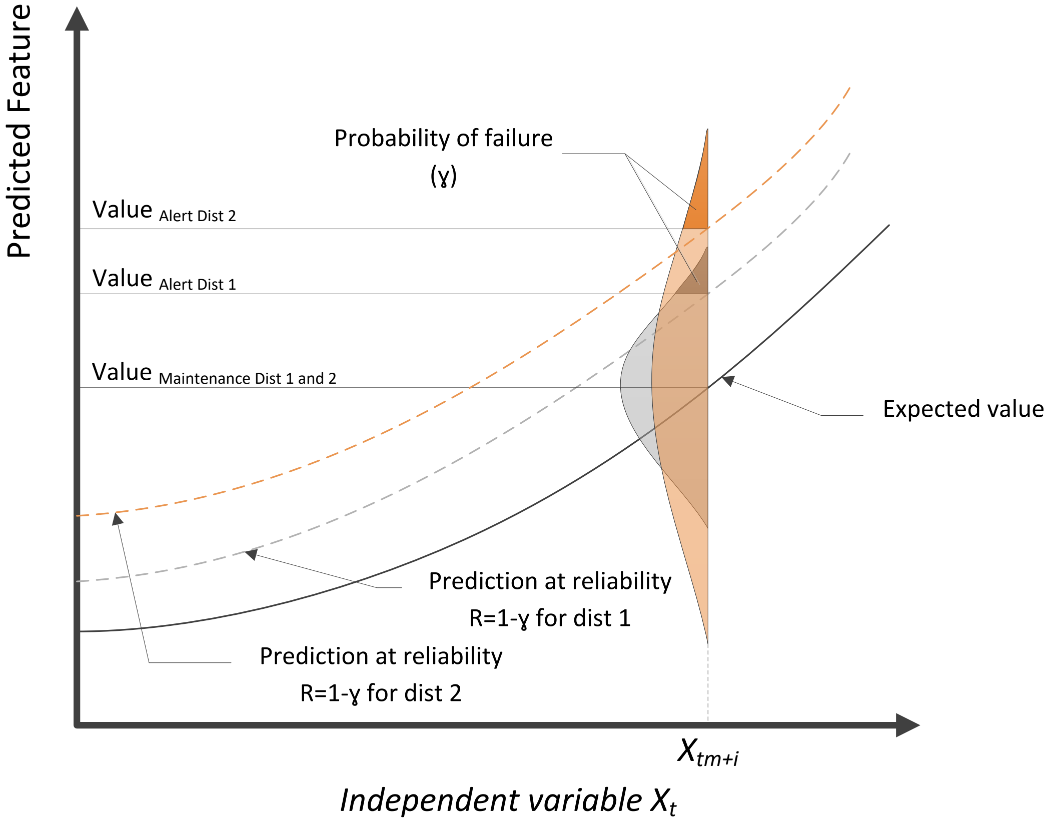


FIGURE 25. EXAMPLE OF DEALING WITH ONE FEATURE UNDER THE HYPOTHESIS OF TWO DIFFERENT PROBABILITY DISTRIBUTIONS

As a summary, taking into account the uncertainty of the forecasting yielded by WP3, the information data associated to the features to be used for predicting alerts based on Work Orders, and their associated maintenance interventions, will present a the record listing in a tabular schematic as shown in Table **12**.

	Alerts	Asset condition	Most probable interventions
Value of the feature	Prediction at reliability 1-γ	Expected value	Expected value

TABLE 12. TABULAR SCHEMATIC OF FEATURE VALUES FOR PREDICTING BASED ON WORK ORDERS

8 Alert severity levels

Two different ways are defined for inferring a severity level depending on the specific module of WP4.

8.1 Severity of Alerts based on limits (Module AM1)

Basically, the severity of this type of alert assessment is given, for a particular asset, by the deviation of the predicted condition from either the Standards or, more generally, from a specific prescribed limit. This severity can be estimated according to various criteria.

For the sake of illustration, Figure 22 depicts the estimated asset condition $X_p\big|_{a-i}$, evolving respect to the independent variable X_t (e.g. time, accumulated load), corresponding to three scenarios identified by the values X_{tm+1} $X_{tm+2,}$ X_{tm+3} of the independent variable. The nowcasting scenario X_{tm} is identified by the square dot. The probabilities of the estimated state condition values should be also provided for each scenario, represented by the bell-function.

The horizontal lines stand for Reference Thresholds (*RT*) which are frontiers defining quality stripes. In Figure 22, only two border lines are depicted and termed as RT_i and RT_{i+1}, delineating three stripes (lower, medium and upper), which can be identified to any of the *RTs*, with the only peculiarity of being $RT_i < RT_{i+1}$. A larger number of thresholds and subsequent stripes can be defined, but for the sake of simplicity the example is restricted to a two-threshold instance.

A first approach assessment of the technical severity of the state condition for a particular asset can be based on the absolute or relative divergence between the deterministic/probabilistic quantified state condition and the *RTs*. In this case the asset condition technical severity level (*TSL*) can be defined in either a deterministic (upper script *d*) or probabilistic (upper script *p*) way. Below a brief description of several criteria is included.

- Case 1. Deterministic criterion (Figure 26).

  This case applies when: i) the state condition derives from a direct measurement/observation, and no stochastic information is available; ii) the state condition, corresponding to a forecasted scenario, is limited to providing an estimate with no probability regarding its reliability.

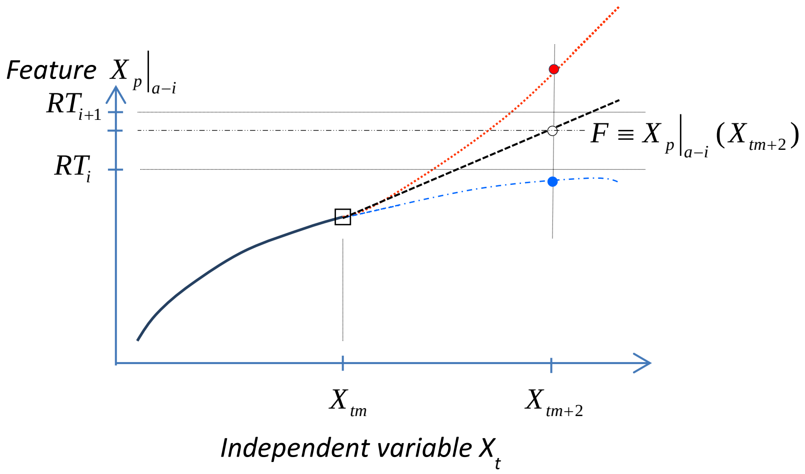


FIGURE 26. ALERT DETERMINISTIC CRITERION

– Subcase 1.1. The state condition technical severity is referenced to the Reference Threshold RT_{i+1}, upper bound of strip (RT_i, RT_{i+1}).

  - Based on the absolute distance to threshold RT_{i+1}:

$$TSL_1^{(d)} = F - RT_{i+1} : \begin{cases} > 0 & threshold\ exceeded \\ \leq 0 & threshold\ not\ exceeded \end{cases}$$
$$\geq \alpha_1^{(d)} \Rightarrow Technical\ Alert$$

where $\alpha_\cdot^0$ stands for the criterion prescribed by the MAB (according to prior know-how, quality policies, and other requirements).

  - Based on the relative distance to threshold RT_{i+1}, respect to the stripe:

$$TSL_2^{(d)}(\%) = 100 \cdot \frac{F - RT_{i+1}}{RT_{i+1} - RT_i} : \begin{cases} \geq 0 & threshold\ exceeded \\ < 0 & threshold\ not\ exceeded \end{cases}$$
$$\geq \alpha_2^{(d)} \Rightarrow Technical\ Alert$$

– Subcase 1.2. The state condition technical severity is referenced to the Reference Threshold RT_i, lower bound of strip (RT_i, RT_{i+1}).

  - Based on the absolute distance to threshold RT_i:

$$TSL_3^{(d)} = F - RT_i : \begin{cases} \geq 0 & threshold\ exceeded \\ < 0 & threshold\ not\ exceeded \end{cases}$$
$$\geq \alpha_3^{(d)} \Rightarrow Technical\ Alert$$

- Based on the relative distance to threshold RT_i, respect to the stripe:

$$TSL_4^{(d)}(\%) = 100 \cdot \frac{F - RT_i}{RT_{i+1} - RT_i} : \begin{cases} \geq 0 & threshold\ exceeded \\ < 0 & threshold\ not\ exceeded \end{cases}$$
$$\geq \alpha_4^{(d)} \Rightarrow Technical\ Alert$$

- Case 2. Probabilistic criterion.

  This case applies when the estimated state condition, corresponding to a forecasted scenario, provides probability regarding its reliability.

  - Subcase 2.1. The state condition technical severity is referenced to the Reference Threshold RT_{i+1}, upper bound of strip (RT_i, RT_{i+1}).
    - Based on the probability of exceeding threshold RT_{i+1} (Figure 27):

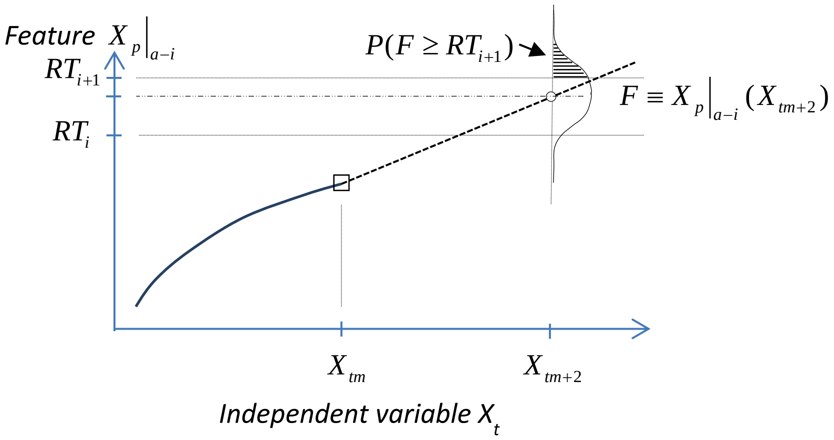


FIGURE 27. ALERT PROBABILISTIC CRITERION

$$TSL_1^{(p)} = P(RT_{i+1} \leq F) \geq \alpha_1^{(p)} \Rightarrow Technical\ Alert$$

where $\gamma_\cdot^0$ stands for the criterion prescribed by the MAB (according to prior knowhow, quality policies, and other requirements). A technical alert will be triggered when the TSL exceed the prescribed value γ .

- Based on the probability to be found between the estimated average value F (50 % normal probability) and the upper threshold RT_{i+1} (Figure 28):

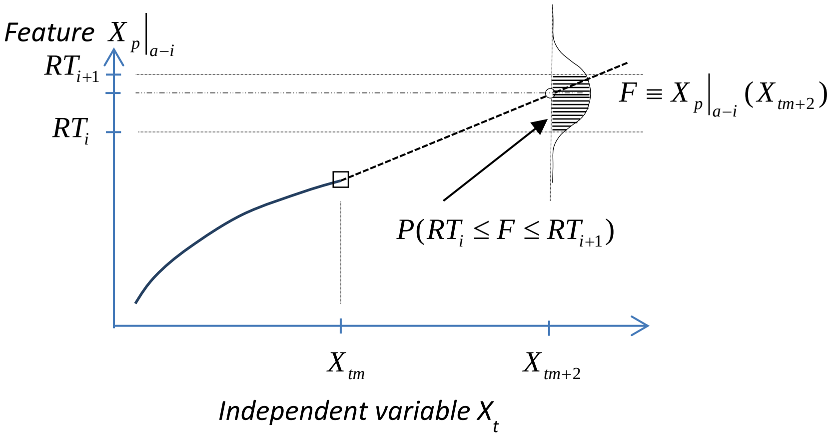


FIGURE 28. ALERT PROBABILISTIC CRITERION

$$TSL_2^{(p)} = 0.5 - \underbrace{P(RT_{i+1} \le F)}_{TSL_1^{(p)}} \ge \alpha_2^{(p)} \Rightarrow Technical\ Alert$$

- Based on the probability to be found between the estimated average value *F* (50 % normal probability) and the upper threshold RT_{i+1}, conditioned to the probability to be in strip (RT_i, RT_{i+1}) (Figure 28):

$$TSL_3^{(p)} = \frac{0.5 - P(RT_{i+1} \le F)}{P(RT_i \le F \le RT_{i+1})} \ge \alpha_3^{(p)} \Rightarrow Technical\ Alert$$

- Subcase 2.2. The state condition technical severity is referenced to the Reference Threshold RT_i, lower bound of strip (RT_i, RT_{i+1}).
  - Based on the probability of exceeding threshold RT_i (Figure 29):

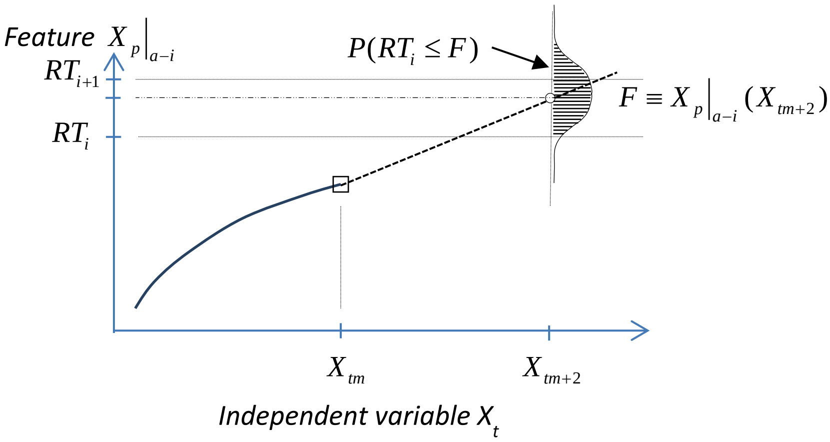


FIGURE 29. ALERT PROBABILISTIC CRITERION

$$TSL_4^{(p)} = P(RT_{i+1} \le F) \ge \alpha_4^{(p)} \Rightarrow Technical\ Alert$$

- Based on the probability of exceeding RT_i, conditioned to the probability to be in strip (RT_i, RT_{i+1}):

$$TSL_5^{(p)} = \frac{P(RT_i \leq F)_i}{P(RT_i \leq F \leq RT_{i+1})} \geq \alpha_5^{(p)} \Rightarrow Technical\ Alert$$

In former expressions, the following notations apply:

d: deterministic; *p*: probabilistic;
F: value of feature $X_p\big|_{a-i}$ at scenario X_{tm+2};
lower: reference threshold RT_i; *upper*: reference threshold RT_{i+1};
P(): probability conditioned to ();
α_k : criterion prescribed by the MAB.

8.2 Severity of Alerts based on Work Orders (Module AM2)

In this case, all relevant features are taken into account simultaneously and there is not limit associated to the general asset condition. However, this global asset condition is connected, in any way, with the value of each individual feature. That is the reason the proposed methodology generates a global severity level based on the severity level of each individual feature weighted according to the Maintenance Manager know-how.

To do this, a technical severity level for each feature has to be calculated but, in this case, no limits to trigger the alert are prescribed; the *TSL* is computed in an absolute way. After that, all individual *TSLs* have to be normalised in order to refer all features under the same scale. Finally, all normalised *TSLs* are added but, in order to take into account the Maintenance Manager criteria, the normalised *TSLs* are weighted to give more importance to some features than others.

The approach steps follow:

- Calculate the Technical Severity Level of each individual feature (TSL_1, …, TSL_n). As mentioned before, *TSLs* are calculated in absolute way. This implies that each *TSL* is referred to the value of the corresponding feature *F* (see Figure 30).
- Normalise the previous Technical Severity Level to refer all values to the same scale ($TSLN_1,\ldots,TSLN_n$).
- Give different weights to each normalised Technical Severity Level ($\alpha_1 \cdot TSLN_1,\ldots,\alpha_n \cdot TSLN_n$). Where $\alpha_1 + \ldots + \alpha_n = 1$.
- Compute the Normalised Global Technical Severity Level (*GTSLN*):

$$GTSLN = \alpha_1 \cdot TSLN_1 + \cdots + \alpha_n \cdot TSLN_n$$

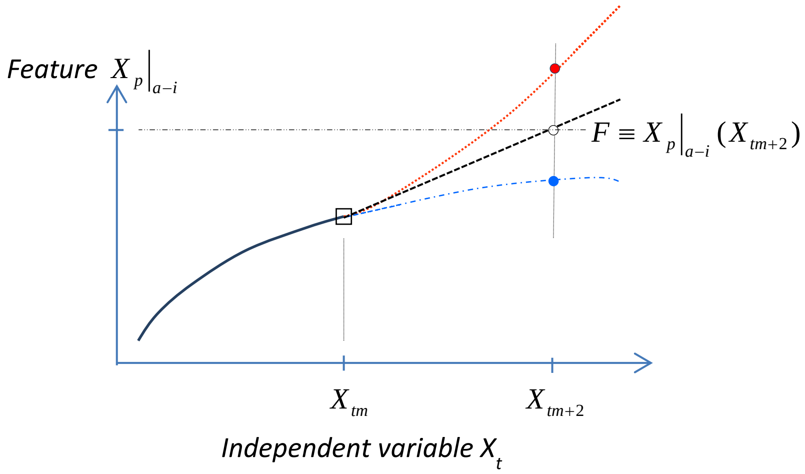


FIGURE 30. TSL IN AN ABSOLUTE WAY

The value *GTSLN* is associated to the asset condition; an asset with a higher *GTSLN* will be in a worse condition than one with a lower level, this process prioritises the alerts generated by Module AM2.

9 Glossary of terms

Asset The physical transportation infrastructure (e.g. travel way, structures, etc.); more generally can include the full range of resources capable of producing value-added for an agency (e.g. human resources, equipment, materials, financial capacity, real state, corporate information, etc.).

Asset condition Measure of an asset's physical state as affected by deterioration and past maintenance and repair; can be expressed in terms of damage present (e.g. amount or percentage of cracking), an agency defined or standard scale (e.g. condition states 1 through 5; or good, fair, poor); often used in conjunction with 'performance' when described in the context of performance-based processes.

Crocodile cracking Crocodile cracking (CT), also called fatigue cracking or alligator cracking, is a common type of distress in asphalt pavement. Crocodile cracking is characterised by interconnecting or interlaced cracking in the asphalt layer resembling the hide of a crocodile. Crocodile cracking is generally a loading failure, but numerous factors can contribute to it. It is often a sign of sub-base failure, poor drainage, or repeated over-loadings. Most load related cracking of this type begins as a single longitudinal, discontinuous crack within the wheel path that progresses with time and loads to a more branched pattern that begins to interconnect. The stage at which several discontinuous longitudinal cracks begin to interconnect is regarded as crocodile cracking. It is important to prevent crocodile cracking, and repair as soon as possible, as advanced cases can be very costly to repair and can lead to formation of potholes or premature pavement failure (Kay, 1992).

International Roughness Index The international roughness index (IRI) was developed by the World Bank in the 1980s (Sayers et al., 1986). It is a scale for roughness based on the response of a standardised motor vehicle to the road surface. The IRI simulates response to the surface profile, and also considers the effect of vehicle suspension. Roughness or ride quality is important as numerous studies have shown that there are strong correlations between motorists' subjective ratings of ride quality and the ratings derived from measurement of IRI. The road user's view of satisfactory or unsatisfactory road condition is primarily influenced by roughness or ride quality. There is also significant correlation between IRI and the maximum speed at which a road user is comfortable while driving on the road in question (Feighan et al., 2015).

RUT depth Rutting in the wheel path is a structural distress induced by heavy vehicle traffic. Rutting is identified as a permanent deformation of the pavement, creating channels in the wheel paths. It can be caused by the consolidation of material under repeated traffic loading, inadequate compaction of the pavement layers during construction or inadequate thickness of pavement layers. (Feighan et al., 2015).

10 References

Feighan, K., McGowan, R., Casey, T., & O'Sullivan, (2015). A. Recent developments in pavement management on Irish national roads. 9th international conference on managing pavement assets, Washington, DC. Blacksburg: VirginiaTech Transportation Institute.

Hastie, T., Tibshirani, R., Friedman, J., (2009). The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Second edition, Springer.

INFRALERT, (2016). INFRALERT: Linear Infrastructure Efficiency Improvement by Automated Learning and Optimised Predictive Maintenance Techniques. Deliverable D4.2. Asset alert level estimator.

Kay, K.R, (1992). Pavement Surface Condition Rating Manual. Washington State Department of Transportation.

Martinez, W. L. and A. R. Martinez. 2002. Computational Statistics Handbook with MATLAB, Boca Raton: CRC Press.

Matlab (2015). Statistics and Machine Learning Toolbox™. The MathWorks Inc.

Sayers, M.W., Gillespie, T. D., Queiroz, C.A.V., (1986). The International Road Roughness Experiment: Establishing Correlation and a Calibration Standard for Measurements, World Bank Technical Paper No. 45, The World Bank, Washington DC.