Predictive maintenance from event logs using wavelet-based features: an industrial application

Stéphane Bonnevay, Jairo Cugliari, Victoria Granger

Predictive maintenance from event logs using wavelet-based features: an industrial application

Stéphane Bonnevay[1] Jairo Cugliari[1] Victoria Granger[2]

[1]ERIC EA3083, Université de Lyon, 5 av. Pierre Mendès France, 69676 Bron Cedex, France
[2] ENEDIS, 124 boulevard Marius Vivier Merle, Lyon, France

August 10, 2018

Abstract

In industrial context, event logging is a widely accepted concept supported by most applications, services, network devices, and other IT systems. Event logs usually provide important information about security incidents, system faults or performance issues. In this way, the analysis of data from event logs is essential to extract key informations in order to highlight features and patterns to understand and identify reasons of failures or faults. The objective is to help anticipate equipment failures to allow for advance scheduling of corrective maintenance. In this paper, we address the problem of fault detection from event logs in the electrical industry. We propose a supervised approach to predict faults from an event log data using wavelets features as input of a random forest which is an ensemble learning method. This work was carried out in collaboration with ENEDIS, the distribution operator of the electrical system in France.

1 Introduction

Smart electric devices automatically monitor information about energy consumption or production, they are defined by the ability to connect to a network and to operate remotely. They report meaningful and appropriate information to relevant parties (consumers, energy distribution system operators or energy providers) and their systems. Modern electric smart devices produce enormous amount of data. The first one is the inherently primary data associated to the devices' main activity and implemented features. Its exploration and use involve privacy issues which have been largely debated and are beyond the scope of this work. In addition to this, the second category of transmitted information is about events, a relatively new category of data, the value of which has yet not been assessed. Event is basically a notification that originates from a electrical device and contains the information regarding the object, action or process to which the event is related. Events are issued while monitoring different aspects of the system and give an overview about equipment communications, devices' secondary non-core functionalities, network intrusions or activity on the grid.

We believe that event logs could be processed and analysed to unveil useful information, in addition to devices' primary data. More precisely, we assume that these data can be useful to inform about the device's operative state and eventually to predict device failure. However, event logs concern a wide range of uses and the difficulty comes from the volume and variety of logs received. Log events are continuously recorded composing a data streamflow related with high volumes, as being generated not only for irregular functional conditions, but also for normal operative states. The main challenge is to analyse this data and extract useful knowledge from the unremitting flow of notifications. The issue therefore is to identify appropriate events containing helpful information. Furthermore, it is essential to detect a shift or an alteration in the patterns of these specific events which could alert users about a fault occurrence.

In literature, patterns from event logs are defined in various ways, for example as partial orders of a process[1,2,3], or considered as Petri nets[4]. Also as repeated sequences that capture process models from event logs in order to improve their detection[5]. From these definitions, authors develop some specific pattern detection approaches mainly based on unsupervised or supervised learning

techniques. Unsupervised pattern detection approaches take an event log as input and generate patterns based on statistical properties[2,3,5]. In unsupervised learning, clustering techniques are widely used[6,7]. Supervised pattern detection approaches take patterns and logs as input and detect pattern instances as results[4]. Combination of these two approaches into semi-supervised techniques have been also studied[8]. From another point of view, visualization and interactive tools have been developed to help user observe and analyse both patterns and event sequences, as EventFlow[9]. Event logs are frequently composed of event codes and their associated text messages. In that case, the use of text parsing or natural language processing techniques is necessary[6,10].

Moreover, some specific works dealing with predictive maintenance based on event logs have also been tackled. Let us mention a general classification-based failure prediction method which has been tested on real ATM run-time event logs data[11], or event logs data extracted from medical equipments used to treat a multi-instance learning task[12]. Also, a Cox proportional hazard model has been used to provide a prediction of system failures based on the time-to-failure data extracted from the event sequences[13].

In this work, we consider the event distribution over time as a function of time. Our first objective is to extract characteristic features from the time series, which will then be presented to a learning algorithm. In order to make this step as automatic as possible, we decided to perform the Discrete Wavelet Transform (DWT) which is an appropriate tool for noise filtering, data reduction, and singularity detection, and thus it a good choice for time series and signal processing. The decomposition coefficients obtained from the DWT are then used as input of a supervised learning algorithm. A variety of task can be successfully tackled using this approach[14,15,16,17]. In our case, we use a random forest both to predict and to measure variable importance in order to select the best features.

In this paper, we propose a supervised approach to predict faults from event log data using wavelets features. The goal is first to use the Discrete Wavelet Transform to detect and characterize features of our electric event logs. Then, we use these features as an input of a random forest model to predict faults. Next section introduces the information we use from event logs and how we transform them into time series trajectories or time functions. To cope with the temporal dependence and functional structure of these objects, we introduce in Section 3 the wavelets transform. The section also includes an overview of random forest. Section 4 describes the experiences and presents the results. The work concludes with a discussion on both industrial and modelling aspects in Section 5.

2 From event to time functions

Our study is based on events monitored on electrical devices installed on ENEDIS network, the French Distribution System Operator. Each electrical device records and transmits real-time event data to a centralized information system. We extract and deal with 3 available attributes: the event code related to a time-stamp and the id of the source device. An example of our logs is displayed in Table 1. We define an additional feature, a group code representing an hierarchical level of event codes. These values were agreed upon with domain experts into 13 groups. None of these notifications have any level of criticality or priority.

timestamp	deviceId	eventCode	groupCode
2014-01-24 17:49:44.537	001	A3	A
2014-01-24 15:09:35.970	001	A23	A
2014-01-25 03:55:56.872	002	A3	A
2014-01-27 00:14:42.463	002	B8	B
2014-01-27 08:10:25.470	002	A23	A
...	...	...	...

Table 1: Event logs data.

Events were aggregated to a daily basis. Figure 1 shows the A23 event distributions for two electrical devices from the beginning of the study to the last day of observations. The observation period starts on May 01 2013 and stops on November 02 2014 and the total number of events is about 1.25 millions recorded on 2623 devices.

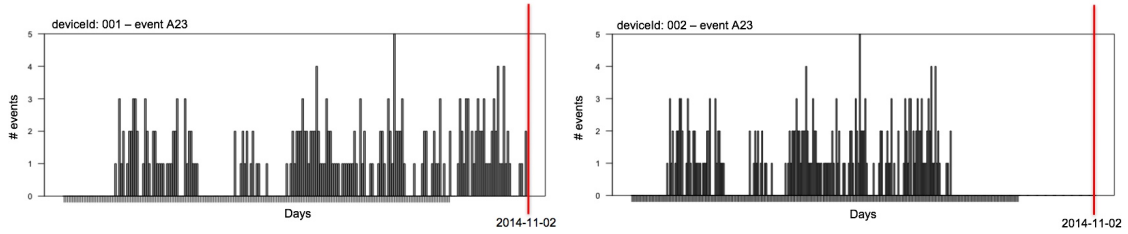

Figure 1: Example of A23 daily event distributions for two electrical devices (`dev001` and `dev002`) from 2013-05-01 to 2014-11-02. The red line shows the end of the observation period.

Devices were monitored over a considerable period of time and presented similar settings and technical specifications during this period. The devices main activity is monitored throughout their lifespan. A fault occurrence is considered when the device fails to provide its main function.

Among all, 1858 devices were properly functional and had a functional status over the observation period, providing their primary function. All of the device were brought to service *a priori* to the beginning of observation period (see `dev003`, Figure 2) and were selected as being operative *a posteriori* to the observation period, over a significant interval of time, to ensure their normal functioning.

A part of devices developed a fault before the end of the observation period, with the lost of their primary functions. These devices were brought to service either before or after the beginning of the observation time. The devices were withdrawn from the field and a technical diagnosis confirmed failure on these equipments. Devices failing to provide their main activity for which technical diagnosis did not confirm the failure were not considered in the study. 765 faulty devices were considered in this study (see `dev002` or `dev001`, Figure 2).

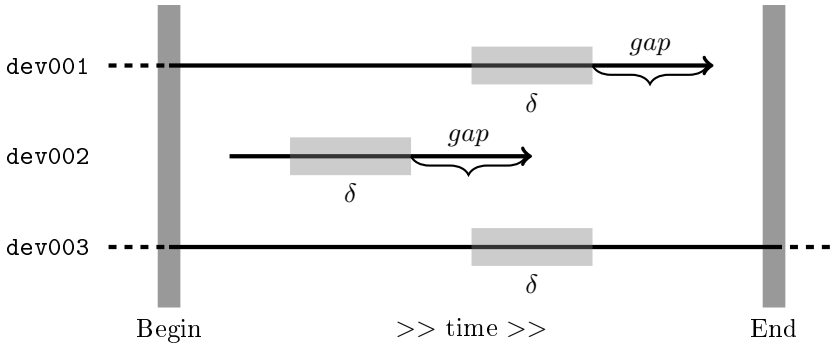

Figure 2: Examples of 3 devices throughout the observation period. Device `dev001` and `dev002` present a fault occurrence. *gap* represents the number of days in advance the fault occurrence is recorded, and therefore predicted. Device `dev003` does not present any fault occurrence. δ is the temporal period on which wavelet features decomposition is applied.

As stated above, both devices with positive and negative fault occurrences present event profiles throughout their lifespan. The purpose of this study is to compare temporal event profile of faulty and working devices in order to identify useful events for predictive maintenance. Hence, the first goal is to capture events frequencies and dynamics of both devices' health status. The second goal is to predict fault occurrence using these summarized temporal profiles while identifying meaningful events. In operative conditions, we wish to detect a failure with a delay which needs to be sufficiently long to allow the attendance of alarms on devices. In this study we considered a predictive gap ranging from zero up to 15 days.

From the available data we only use the absolute frequency of events and the event code classification. We focus then on the number of logs effectively observed over a reasonable period of length δ using a given time resolution (e.g. hours, days) for each type of event. We then consider

$$(N(t_1), \ldots, N(t_\delta)), \tag{1}$$

where $N(t_j)$ is the number of events at time t_j. To fix ideas, say that δ may span over two weeks

and using a daily resolution the vector of counts would have length equal to 14. Figure 3 plots four cases of these trajectories.

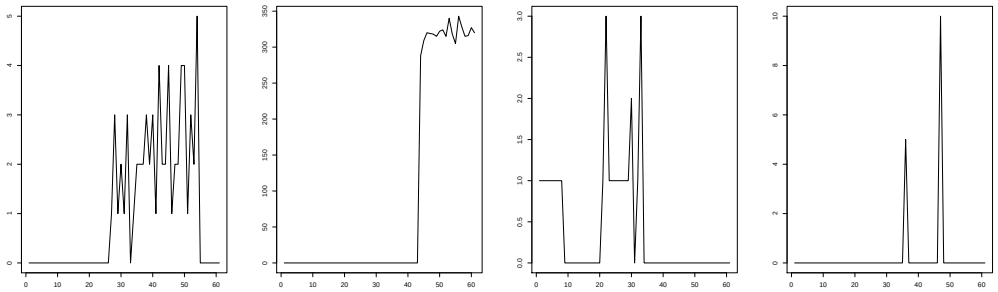

Figure 3: Examples of trajectories from event logs data. Tracking is done daily over 64 days. Cases (a) and (b) are from working devices; and cases (c) and (d) are from faulty ones.

This vector constitutes the building block of our approach since we create instances of this vector for both normal and abnormal regimes (*cf.* Section 3). Actually, each instance is the tracking of a device along a period of length δ. Then the couple device $\times$ time should not be view as tracking over contemporaneous instants but as snapshots of the life time of the devices.

If we now considerer that K different type of events exists, then we have

$$(N_k(t_1), \ldots, N_k(t_\delta)), k = 1, ..., K,$$

that is each device $\times$ time is a set of K counting vectors. From the mathematical point of view, we may look at vectors (1) as time series trajectories. And since there exists K of them, we have a multivariate time series where there may be some dependence structure between components of the vectors as well as time dependence within components.

One way to cope with time dependence is to see each trajectory as a discrete sampling, eventually with some noise, of the time function $z_k(t), t \in [0, \delta], k = 1, \ldots, K$. Notice again that time should be consider as relative to period δ and not as an absolute quantity.

To fix ideas, let us introduce a graphical representations of the some of the data up to this point. We follow the construction detailed above considering only events from one code to construct a sample of trajectories containing both faulty and working devices. Then, we use a simple metric between trajectories based on the euclidean distance on standardized versions of the trajectories. The associated distance matrix is then used as input of a multidimensional scaling in order to get the a simple planar representation of the observations, represented in Figure 4.

Here, each point is a trajectory and its coordinates are chosen to preserve, as well as possible, the distances between trajectories. Notice that since no information about the class is used this technique is essentially unsupervised. However we add a colour reference (grey: working devices, red: faulty devices) to the scatter plot in order to visualize eventual differences. Even if the sample is very unbalanced, a clear distinction between both classes is appreciated. Distances between working devices are relatively small with reference to distances between faulty devices. Other conclusion we can draw is that a (eventually non linear) reduction of the dimension may suffice to extract the useful information on the signals. Taking into account the time dependent structure of the functions is necessary to obtain an appropriate construction that yields on a dimension reduction.

3 Methods

We describe here the methods we use to construct our solution. They are related first on how to represent the multivariate time series with a handy set of interpretable features. Then we build a 2-class discriminant classifier where we assume that each class represents a logging regime. The first one is the normal log regime where the working device has a working status. This regime, should be the most frequent one. The second regime, more rare by nature, represent a working device that is close to a failure status. The first problem is solved using the wavelet transform while the classifier we use is random forest.

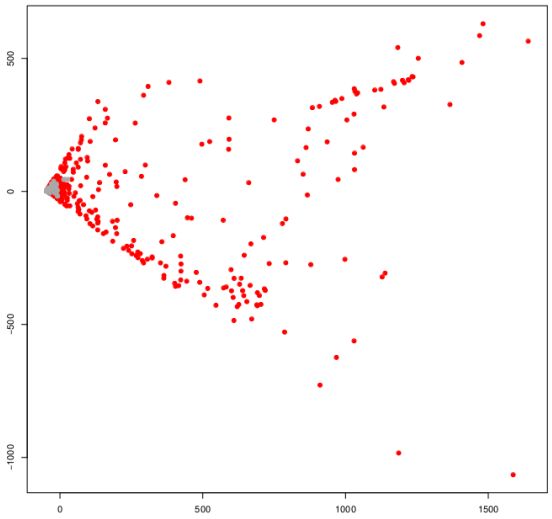

Figure 4: Multi dimensional scaling of trajectories from one event code. Each point represents a trajectory from a working device (in gray) or a faulty device (in red).

3.1 Wavelets transform

Wavelets are a domain transform technique that allows one to represent time domain signals into a bivariate domain location-scale[18]. While location in the new domain is connected to the original time domain, scales can be associated to Fourier frequencies and both with good localization properties. That is, the transform will give information on locations connected to only a time span (not the global time) and scales connected to only some frequencies (and not all of them). This is in difference with a time domain analysis that has no localization on frequencies or a frequency domain analysis that has no localization on time.

Moreover we use the Discrete Wavelet Transform (DWT) which is provides an orthonormal basis of the space, allowing us to encode all the available information on a signal without any loss of information[19]. In what follow we explain the necessary material to understand our approach.

Consider the signal $z(t)$ which is an univariate function defined on the time domain $\mathcal{T}$, for example $\mathcal{T} = [0, 1]$. The DWT will provide two terms: a global approximation of the signal $\mathcal{S}(t)$ and the ensemble of details $\mathcal{D}(t)$ well localized both in time and frequency. If $z \in L_2([0, 1])$, then the DWT provides us with a basis of the functional space. The basis is created by simple transformations of a scaling function $\phi(t)$ and a wavelet mother $\psi(t)$ which are associated to the orthogonal multi resolution analysis of $L_2([0, 1])$. Indeed, we consider the family $\{\phi_{j,k}(x) = 2^{-j}\phi(2^{-j}x - k)\}_{j,k}$ which is obtained by dilatations of a factor 2^j and by integer translations on the new scale. Similar operations are done to get the family $\{\psi_{j,k}(x)\}_{j,k}$. Then, a finite energy signal z can be expressed as

$$z(t) = \underbrace{\sum_{k=0}^{2^{j_0}-1} c_{j_0,k}\phi_{j_0,k}(t)}_{\mathcal{S}_{j_0}(t)} + \underbrace{\sum_{j=0}^{J-1}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(t)}_{\mathcal{D}(t)}, \tag{2}$$

where $c_{j,k} =< z, \phi_{j,k} >, d_{j,k} =< z, \phi_{j,k} >$ are the scale coefficients and wavelet coefficients respectively. The scale j_0 separates the two terms. The first one, gives a smooth approximation at resolution 2^{j_0}. The second one, keeps all the details of the curves on a hierarchical structure depending on scales and locations. The approximation coefficients $c_{j_0,k}$ retains the information of the local (at location k) mean level of the curve, while the detail coefficients $d_{j,k}$ codes the information of discontinuities and other singularities.

With finite data $\{z(t_i), i = 1, \ldots, N\}$, the signal $z(t)$ can only be approximated by a truncation at some maximum scale level $J = \log_2(N)$, that is we approximate (2) by

$$z_J(t) = c_0\phi_{0,0}(t) + \sum_{j=0}^{J-1}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(t). \tag{3}$$

Notice that we have also fixed the approximation part at the coarsest resolution $j_0 = 0$ which means that only one scaling coefficient is used to approximate this term. For convenience we choose the number of sampling points per curve, N, to be a power of 2. The maximum number of scales J is then an integer. With this, we are in conditions to use the highly efficient Mallat's pyramidal algorithm[18] to obtain both the scaling and wavelet coefficients. If the sampling grid $\{i/N, i = 1, \ldots, N\}$ is not regular or N is not a power of 2, then one can choose a finer regular grid and use any interpolation scheme to meet our choices.

Haar wavelet leads to a easy and clear intuition on the wavelets coefficients. The only scaling coefficients we retain is proportional to mean level of the whole signal. The approximation term is then a constant function $\mathcal{S}_0(t) = c_{0,0}\psi(t)$ proportional to the mean function of the signal.

If we increase the resolution of the approximation to the next scale, then the approximation part will be a ladder function, that is a piecewise constant function with a jump in the middle point of the sampling grid. Aside the jump, the signal is approximated by the mean level of each side. A similar reasoning applies to the next scales, at each time cutting into halves and approximating each half by a constant function equal to the mean level of the observations on the half.

The detail coefficients are the difference on the constant approximations between two juxtaposed halves. We interpret them as the change observed at some resolution (related to the scale j) and at some time (related to the location k).

In what follows we will need to reduce the number of coefficients we use in order to keep the calculations into a reasonable time. With this, we are further truncating the approximation on (3) into smaller values of J. Since finer approximations may capture only the signal's noise, the changes on these scales would reflect random fluctuations not necessarily connected to the structure of the signal. For this, one should only retain coarsest scales. and the detail coefficients $d_{0,0}, d_{0,1}$.

In what follows we set ψ to be

$$\psi(t) = \begin{cases} 1 & \text{if} \quad 0 \leq x < 1/2 \\ -1 & \text{if} \quad 1/2 \leq x < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

which is known as the Haar wavelet. The corresponding scaling function is $\psi(x) = 1$ if $0 \leq x \leq 1$ and 0 otherwise.

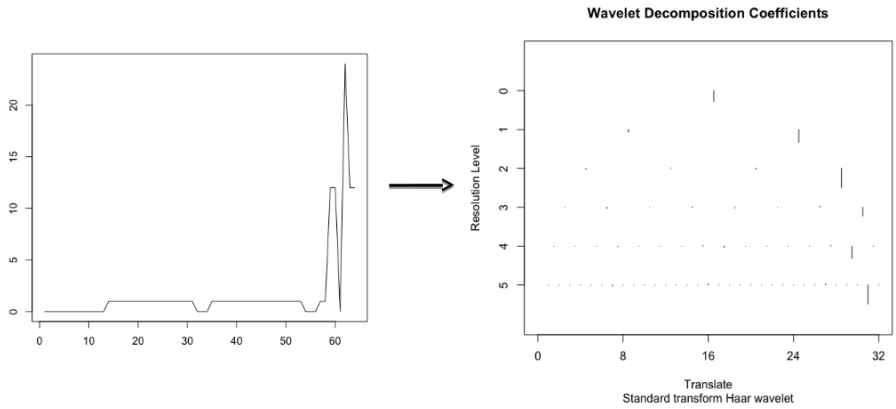

Figure 5: Tracking of one event over time (right) and its DWT (left).

To illustrate the kind of results we obtain with the wavelet transform, we show in right hand side of Figure 5 the detail coefficients of the signal represented on the right hand side. The signal shows a low-activity steady state during almost all the tracking period with an important raise on the number of events at the end. The wavelets coefficients (on the left of the figure) show an alternative picture of the same phenomena. Ranged by scales, the coefficients are small in absolute value almost everywhere but at those location near the end of period. What is near, depends on the resolution level at which we look at the signal, for scales close to $j = 0$ the analysing functions are global, while at scale $j = 5$, the 32 resulting coefficients gives very localized information. Note that this level of detail can be misleading if considered only at one individual scale. For instance the last coefficient at scale 5 is large and negative because the last number of logs is lower than the precedent one. Moreover, noise is also more important at these high frequencies. One may rely

on shrinkage methods to choose which of the estimated coefficients are significantly different from zero.

Our approach is slight different, we choose to work only with the scale coefficient $c_{0,0}$ and the detail coefficients $d_{0,0}, d_{0,1}$. With this, the number of coefficients retained in what follows is kept into a reasonable size when multiplied by the number of event codes. Intuitively, these coefficients allows one to reconstruct the trajectories with the approximation $\mathcal{S}_1(t)$ which is exactly the mean level of the function, given by $c_{0,0}$, and the detail coefficients $d_{0,0}, d_{0,1}$. Notice that this reconstruction is the best linear approximation one can do with three coefficients. In what follows we are using the estimated coefficients as features of a random forest predictor.

3.2 Random forest

Very popular in statistical machine learning, random forests (RF) are an ensemble method[20]. It builds up on specific versions of CART (Classification And Regression Trees)[21], which is an algorithm that constructs binary tree-based predictors. With respect to individual predictors, the aggregate one aims to augment robustness, variance reduction and improve prediction performance.

For this, RF add two layers of randomness. First, each tree-based predictors is trained only on a different bootstrap sample from the data. Second, only a strictly subset of variables are randomly chosen as candidates at each split of the trees' construction. Note that the trees are constructed up to its maximal size and they are not pruned. While using a stopping criterion and pruning are usual in CART, these versions of tree-predictors sacrifices generalization power by a better in-sample fit – at least on each bootstraped sample – and introduces bias by considering only partial information from available variables. With this, individual trees tend to be less dependent between them which is useful under an aggregation scheme. RF is then the resulting predictor obtained by some aggregation rule of the individual prediction of the so described trees. Usual choices of the aggregation rule are majority vote for classification and mean average for regression.

We use two intrinsic features of random forest to help the interpretation of the results : a measure of variable importance and a notion of proximity between observations.

Variable importance measure. Different approaches can be used to determine the importance of a feature for the construction of the forest[22,23]. In this work, a variable is considered more important if it participates more to the decrease of some impurity notion (e.g. the Gini index). Then, we can track over the individual trees where each variable participates on each node split and record the decrease on the Gini coefficient. Then a plot like the one in Figure 7 where the variables are represented in lines sorted decreasingly on the mean Gini reduction. Most important variables on the construction of the classifier are on top of the plot.

Observations proximity Two observations are closer if they are classified within the same terminal node by more and more individuals trees. Then, the proximity is normalized to be between 0 and 1. If we call $p_{ii'}$ the proximity between observation i and i', then we obtain a dissimilarity measure $1 - p_{ii'}$. While the change is trivial, it allows us now to perform a multidimensional scaling on the proximity matrix associated to the proximity measurements. This yields on a natural representation of the individuals that analogously to discriminant analysis represents in a low dimensional space how the classifier 'sees' the individuals.

4 Experiments

In all our experiments we use the open source `R` software. DWT is performed using `wavethresh` and `randomForest` is used to learn random forests. In the experiments to follow we use the default options of `randomForest` to construct the predictors, *i.e.* the number of trees `ntree` is set at 500 and the number of variables `mtry` chosen randomly at each split is roughly the square root of the total number of variables.

For each gap before fault, we create a dataset of positive and negative failure occurrences as shown in 2). From faulty devices, δ time points (albeit the gap period) before fault occurrence were considered to compute K event vectors. Among working devices, a period of length δ is drawn randomly per device to compute K event vectors per device. Notice that each device at

some point of the time is described as a number of 39 features, that is 3 wavelets coefficients per group of events' code, with a total of 13 event codes.

4.1 Predictive performance

We apply a random forest classifier for each of 16 datasets composed of 39 wavelets coefficients. Two week event profile (for each device) is characterized by 3 coefficients for each of the 13 groups of events. We compute both false negative rate (FNR) representing the percentage of faulty devices classified as working devices and false positive rate (FPR) as the percentage of working devices predicted as prone to failure. We also compute the global model error, summarizing the percentage of observations which are classified wrong and resuming model global accuracy.

The performance scores of random forest models are displayed in Figure 6, results are presented in relation to the predictive gap before failure occurrence. Global model accuracy ranges from 79%, when the predictive gap equals to 15 days, up to 89% when detecting fault the day of occurrence. This performance, that at first sight appears rather inaccurate in an industrial context, displays evidence of meaningful information in the event logs.

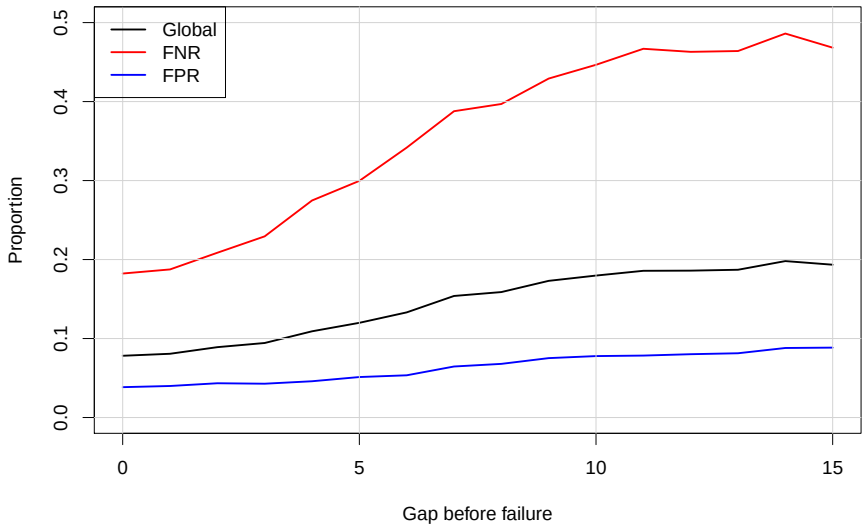

Figure 6: Predictions performance at various gaps, in days, to failure. Red curve shows classification results of failed devices, blue curve shows classification results of working devices and black curve global error.

Overall, the predicting error rate is higher for faulty devices and it is easier to decide on a working status on the basis of resumed event profile of a device, independently of the prediction gap (see Figure 6, the red curve corresponding to the FNR stays above the blue line representing the FPR). The error rate is lower when classifying working devices, as observations of negative fault occurrence dominate the learning error. The result is consistent with the fact that random forests tend to maximize the model global accuracy, keeping a low error rate on larger classes (working devices) while allowing the smaller classes have a larger error rate.

The smaller the temporal gap is, the more precise it is to predict both fault occurrence or devices' normal regime by event data, for example the FPR for working regimes being equal to 3.74% for 1 day-ahead prediction and 7.15% for a 10-day-ahead horizon, see Figure 6.

4.2 Variable importance

Figure 7 shows importance ranking of attributes in classification for a 0 days predictive gap, variable importance displays similar results for all of the 16 models for different values of the predictive gap (results not shown). 3 groups of events appear relevant when predicting fault occurrence : A, B and J. First and third wavelet components of B group appear to be the red flag for an abnormal regime

leading to a failure. A different level of these events for a device and an alteration of the number of received events can be seen as an alarm for failure occurrence. Overall, we observe the same pattern for all of three groups of events: the faulty devices' average level of events is generally higher than working devices' event frequencies. Moreover, there is a substantial gap between the event regime 7 days before failure occurrence and the week before that. This is particularly interesting, as events are related to low level communication on the grid. We suppose that failure affects the ability of devices to interact with other devices on the network.

More helpful, Figure 7 shows that a considerable amount of information received and processed by the system are not relevant for revealing devices' operative status. Independently of their number or frequency, events of group C, G, K, D and F, seem to carry very little information about a possible failure of an equipment. This is to be expected as these events monitor different software activity of various devices of the grid. In a predictive maintenance framework, the monitoring and processing of these categories presents no interest, events have no correlation with the fault occurrence.

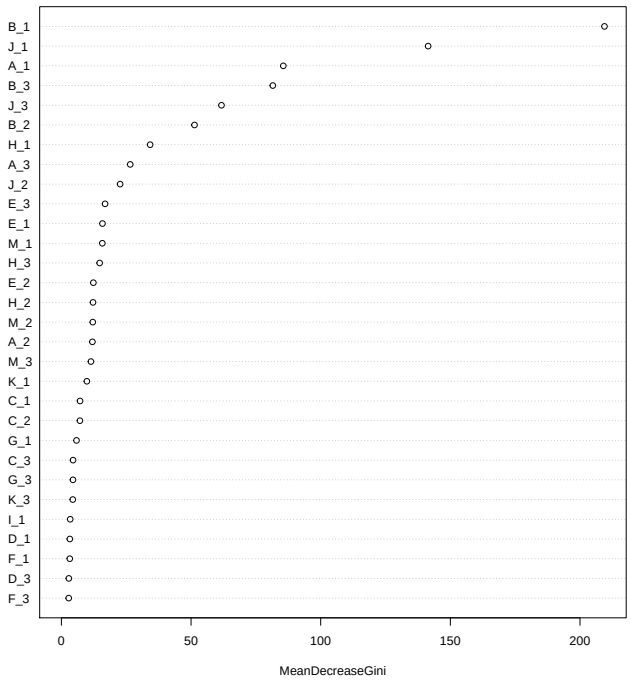

Figure 7: Random forest variable importance output for a 0 day temporal gap

4.3 Observations proximity

As in Figure 4, we use a multidimensional scaling to represent observation proximity in Figure 8. Recall that now the distances on plot are the ones implicitly learned by the classifier so it is effectively using the information on the labels (coded in colours on the plot). Two important differences are to be highlighted. First, the classes are now better clearly separated even if with some overlap. The class of faulty devices (smaller, in red) forms now a compact group that aligned along a straight pattern. Second, the class of normal operation, that is without fault, presents a two arm structure. This means that while connected by some elements that are close to both arms in the middle of the plan, the structure suggest that this class is actually formed by two subclasses which are homogeneous for each of them. From a technical point of view, this result also indicates that working devices present two distinctive event profiles, which shift to a single highly abnormal regime when failure emerges. This outcome is of a particular interest, as experts do not have any *a priori* knowledge about this singularity. Additional investigations may reveal different manufacturer implementations or material configurations having no impact on devices primary functionalities.

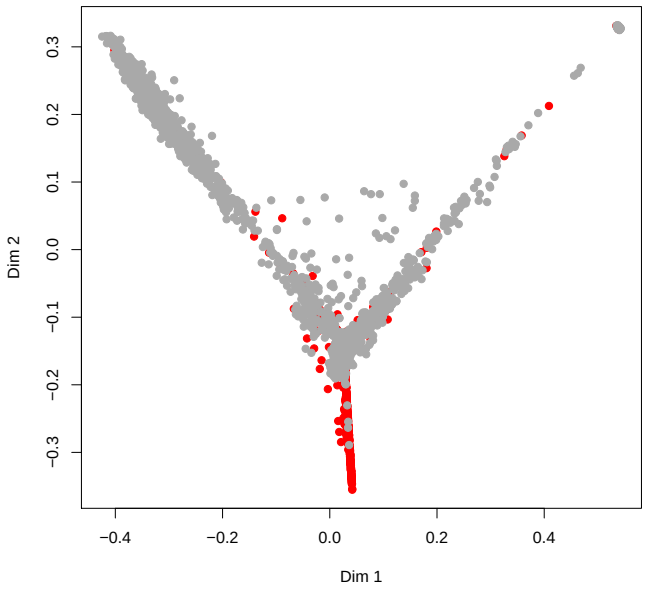

Figure 8: MDS from RF. Each point represents a trajectory from a working device (in gray) or a faulty device (in red).

4.4 Additional experiments

Events of category B, J and A were the most relevant to describe device operative state and to detect a device in a abnormal regime leading to a failure. Using exclusively events of these three groups, we performed an additional set of experiments by including non grouped events as features and applying same methodological approach as described above on individual events. We treated 17 individual code events, therefore 51 wavelets attributes were computed and introduced as features to random forests classifier. All other parameters remained unchanged. Performances scores of 16 models based on 51 features perform similarly to the global approach (results are not shown) which is using all grouped events.

Essentially, error rates of non grouped B, J and A events models are lower than error of models using all code events.

5 Discussion and conclusion

From the modeling point of view, the use of wavelets and random forests gave several benefits. First, the proposed approach is general in the sense that it is not specific for predictive maintenance. Actually, it may be used on different kind of anomaly detection from event logs such as intrusion detection, outage occurrence, *etc.*, as long as one disposes with a way to construct the two class learning data set. Second, wavelets allows an important dimension reduction while keeping discriminatory power. With this, up-scaling the procedure is feasible since the processing needed to pass from functions to wavelet coefficients may be done independently (and so using parallel or distributed computing schemes) for each device. Last, random forest gives interest insights through feature selection and observations proximity. The former benefits from localized coefficients that gives nice interpretation properties to the DWT. The latter can be used together with graphical displays to make emerge patterns in data that are otherwise difficult to unveil.

A natural question that may arise is about the particular choice of the functional basis (i.e. wavelets) and the classifier (i.e. random forest). One may naturally argue that other combination like for instance principal components and logistic regression would be equally reasonable. Our choices are guided for both interpretability and performance. Besides of having a low error level classifier we look for tools that allow the practitioner to better understand the underlying problem. The first three principal components would extract a mean behaviour of the curves where what we

look for is the specific behaviour of each frequency curve that best explains its evolution. Location properties of wavelets, discussed in Section 3, induce the nice interpretability we look for while compressing the information by a handy number of features. Random Forest also contribute to provide with insights on the fitted model as it was discussed in Section 4.2 and 4.3.

In our case, when processing log events we found evidence of untapped information on both fault occurrence and the normal devices' regime when monitoring electrical devices information flow. When predicting the fault occurrence the FNR ranges from 18.72% when confirming the fault by analysing event profile the day of fault occurrence, and goes up to 46.38% to 15 days predictive gap. In regard to these results, there a number of points that we should comment on.

First, the increase of FNR inversely proportional to temporal gap implies that, in some extent, at least two weeks before failure occurrence a part of the devices have a similar event profile as working devices and their event regime undergoes a gradual daily alteration until failure. The degradation seems to accelerate 7 days before failure occurrence, the FNR equals to 40.99% and we gain several points of precision each day. We suppose that failure firstly affects non essential functionalities, from which the event logs are issued, and only secondarily it leads to the cession of the main activity. This progressive shift underlines the fact that a fault occurrence does not necessarily imply a full and immediate standstill of a device as it continues to provide their primary function. In regard to these elements, the use of these notifications in a predictive maintenance tool is of a particular interest to track future devices fault.

Second, a part of faulty devices are misclassified and display similar log event profile as working devices until failure occurrence (FNR equals to 18.44% for a 1 day predictive gap). See also Figure 8, a part of red observations (faulty devices) are situated among grey observations (working devices). It is likely that the category of tracked failure is not similar to the previously described case, which affects primary and secondary functionalities differently. For these devices, functional features related to the main activity should probably be measured as log event profile do not change priori to the failure.

One last point of a particular interest in the results are the early signs of breakdown affecting the devices more than two weeks in advance (FNR equals to 46.24% 2 weeks priori to failure). This result is supported by the high variable importance of the first wavelet component for all three groups of events, even for a high prediction gap (see 4.2). A really moderate usury of hardware related to external factors or network overload could affect a part of equipment. To a different degree, we could also suppose that these devices show an abnormal event profile as soon as they are installed and a latent defect affects their non-core functionalities.

To sum up, classification results show that abnormal dynamics in specific events, can be considered, to a certain extent, forerunner of a future fault. For a long term preventive strategy, there is an obvious need to cross the profiles of identified group of events with other sources of informations to increase model accuracy. Geographical situation the grid, power demand, voltage quality, or environmental factors could affect gradually devices leading potentially to a failure. Primary data and the monitoring of information resulting of the implemented features could allow to enhance the predictive capacity of events. Information on other components of the grid could offer complementary perspectives on the network activity leading to devices usury. Even if the predictive performance does not allow to develop an operative tool, this model allows to identify a high risk population to failure. In a supervision context, the daily processing of ongoing events could allow for these devices to be prioritized and then acted upon with necessary actions.

Acknowledgements

We would like to acknowledge the ENEDIS company for this collaboration and we especially thank Pierre Achaichia, Paul Mersy and Thomas Pilaud from ENEDIS for rich discussions.

References

[1] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Discovering block-structured process models from event logs - a constructive approach. In *Application and Theory of Petri Nets and Concurrency*. Springer, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-38697-8_17.

[2] Sander J. J. Leemans and Wil M. P. van der Aalst. Discovery of frequent episodes in event logs. In *Data-Driven Process Discovery and Analysis*. Springer, Cham, 2015. doi: 10.1007/978-3-319-27243-6_1.

[3] Claudia Diamantini, Laura Genga, and Domenico Potena. Behavioral process mining for unstructured processes. *J. Intell. Inf. Syst.*, 47(1):5–32, August 2016. ISSN 0925-9902. doi: 10.1007/s10844-016-0394-7.

[4] F. Mannhardt, M. de Leoni, H. A. Reijers, Wil M. P. van der Aalst, and P. J. Toussaint. From low-level events to activities - a pattern-based approach. In *Business Process Management*. Springer, Cham, 2016. doi: 10.1007/978-3-319-45348-4_8.

[5] R. P. Jagadeesh Chandra Bose and Wil M. Aalst. Abstractions in process mining: A taxonomy of patterns. In *Proceedings of the 7th International Conference on Business Process Management*, BPM '09, pages 159–175. Springer-Verlag, 2009. ISBN 978-3-642-03847-1. doi: 10.1007/978-3-642-03848-8_12.

[6] Risto Vaarandi and Mauno Pihelgas. Logcluster - a data clustering and pattern mining algorithm for event logs. In *Proceedings of the 2015 11th International Conference on Network and Service Management (CNSM)*, CNSM '15, pages 1–7, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-3-9018-8277-7. doi: 10.1109/CNSM.2015.7367331. URL http://dx.doi.org/10.1109/CNSM.2015.7367331.

[7] Adetokunbo A.O. Makanju, A. Nur Zincir-Heywood, and Evangelos E. Milios. Clustering event logs using iterative partitioning. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 1255–1264, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557154. URL http://doi.acm.org/10.1145/1557019.1557154.

[8] Xixi Lu, Dirk Fahland, Robert Andrews, Suriadi Suriadi, Moe T. Wynn, Arthur H.M. ter Hofstede, and Wil M.P. van der Aalst. Semi-supervised log pattern detection and exploration using event concurrence and contextual information. In *25th International Conference on Cooperative Information System*, Rhodes, Greece, 2017. Springer Verlag. URL https://eprints.qut.edu.au/110716/.

[9] Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2227–2236, December 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.200.

[10] Wei Xu, Ling Huang, Armando Fox, David Patterson, and Michael Jordan. Using machine learning techniques in console log analysis. In *the 27th International Conference on Machine Learning*, ICML'10, 2010.

[11] J. Wang, C. Li, S. Han, S. Sarkar, and X. Zhou. Predictive maintenance based on event-log analysis: A case study. *IBM Journal of Research and Development*, 61(1):11:121–11:132, Jan 2017. ISSN 0018-8646. doi: 10.1147/JRD.2017.2648298.

[12] Ruben Sipos, Dmitriy Fradkin, Fabian Moerchen, and Zhuang Wang. Log-based predictive maintenance. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1867–1876. ACM, 2014. ISBN 978-1-4503-2956-9. doi: 10.1145/2623330.2623340.

[13] Zhiguo Li, Shiyu Zhou, Suresh Choubey, and Crispian Sievenpiper. Failure event prediction using the cox proportional hazard model driven by frequent failure signatures. *IIE Transactions*, 39(3):303–315, 2007. doi: 10.1080/07408170600847168.

[14] Skander Soltani. On the use of the wavelet decomposition for time series prediction. *Neurocomputing*, 48(1):267 – 277, 2002. ISSN 0925-2312. doi: https://doi.org/10.1016/S0925-2312(01)00648-8. URL http://www.sciencedirect.com/science/article/pii/S0925231201006488.

[15] C. A. G. Santos, P. K. M. M. Freire, G. B. L. Silva, and R. M. Silva. Discrete wavelet transform coupled with ann for daily discharge forecasting into tres marias reservoir. In *Proceedings of the International Association of Hydrological Sciences*, volume 364, pages 100–105, 2014.

[16] D. Jothimani, R. Shankar, and S.S. Yadav. Discrete wavelet transform-based prediction of stock index: A study on national stock exchange fifty index. *Journal of Financial Management and Analysis*, 28(2):35–49, 2015.

[17] Tianhong Liu, Haikun Wei, Chi Zhang, and Kanjian Zhang. Time series forecasting based on wavelet decomposition and feature extraction. *Neural Computing and Applications*, 28: 183–195, 2017.

[18] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[19] Guy Nason. *Wavelet methods in statistics with R*. Springer Science & Business Media, 2010.

[20] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[21] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[22] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225 – 2236, 2010. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2010.03.014. URL http://www.sciencedirect.com/science/article/pii/S0167865510000954.

[23] Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre. Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics & Data Analysis*, 90:15–35, 2015.