Reliable diagnostics using wireless sensor networks

Jacques Bahi, Wiem Elghazel, Christophe Guyeux,
Mourad Hakem, Kamal Medjaher, and Noureddine Zerhouni

November 1, 2018

Abstract

Monitoring activities in industry may require the use of wireless sensor networks, for instance due to difficult access or hostile environment. But it is well known that this type of networks has various limitations like the amount of disposable energy. Indeed, once a sensor node exhausts its resources, it will be dropped from the network, stopping so to forward information about maybe relevant features towards the sink. This will result in broken links and data loss which impacts the diagnostic accuracy at the sink level. It is therefore important to keep the network's monitoring service as long as possible by preserving the energy held by the nodes. As packet transfer consumes the highest amount of energy comparing to other activities in the network, various topologies are usually implemented in wireless sensor networks to increase the network lifetime. In this paper, we emphasize that it is more difficult to perform a good diagnostic when data are gathered by a wireless sensor network instead of a wired one, due to broken links and data loss on the one hand, and deployed network topologies on the other hand. Three strategies are considered to reduce packet transfers: (1) sensor nodes send directly their data to the sink, (2) nodes are divided by clusters, and the cluster heads send the average of their clusters directly to the sink, and (3) averaged data are sent from cluster heads to cluster heads in a hop-by-hop mode, leading to an avalanche of averages. Their impact on the diagnostic accuracy is then evaluated. We show that the use of random forests is relevant for diagnostics when data are aggregated through the network and when sensors stop to transmit their values when their batteries are emptied. This relevance is discussed qualitatively and evaluated numerically by comparing the random forests performance to state-of-the-art PHM approaches, namely: basic bagging of decision trees, support vector machine, multinomial naive Bayes, AdaBoost, and Gradient Boosting. Finally, a way to couple the two best methods, namely the random forests and the gradient boosting, is proposed by finding the best hyperparameters of the former by using the latter.

1 Introduction

During their life cycle, industrial systems are subjected to failures, which can be irreversible or have undesirable outcomes varying from minor to severe. From this context, it is important to monitor a system, assess its health, and plan maintenance activities. Over the past years, research in Prognostic and Health Management (PHM) field has gained a great deal of attention. PHM aims at defining a maintenance schedule and preventing system shutdown. Yet, if the prediction model and the provided measurements are not accurate, it is possible that the maintenance activity will be performed either too soon or too late.

Health assessment is a key step for Remaining Useful Life (RUL) estimation. Based on the analysis and the predefined thresholds, the machine/component's health state is identified. Sensory data is reported periodically to monitor critical components. This data corresponds to measurements of monitoring parameters and is useful to assess the machine/component's condition. Each monitoring parameter has a threshold; once reached, the system is considered to be in the corresponding state. Reliable health state estimations depend on accurate measurements and fast data processing. The information in question is often gathered by means of individual sensor nodes or via a wired network of sensors. Nevertheless, for some applications, the use of a Wireless Sensor Network (WSN) can be a requirement rather than a choice. For example, due to accessibility or extra weight issues, connecting the sensors through physical wires is not feasible. WSNs are designed for an efficient event detection. They consist of a large number of sensor nodes deployed in a surveillance area to detect the occurrence of possible events. Such an activity necessitates efficiency, which is hard to achieve with the constraints of WSNs [9].

Available energy is a big limitation to WSN capabilities. In fact, sensor nodes are small sized devices, resulting in tiny and non-refillable batteries as energy supply [5]. Therefore, to keep the network running for as long as possible, we need to preserve the available energy. As reducing packet transfer distance and frequency helps consume less energy, a possible solution would be combining data into one packet and forward all the information at once to the base station: this is called data aggregation.

Data gathering in WSNs can be either periodic or event-driven. In periodic applications, data is gathered periodically while in event-driven applications gathering depends on the occurrence of some events. In both cases, the goal from aggregation is reducing energy dissipation by holding packets for as long as possible in intermediate nodes. All packets will be combined together then forwarded in the network. It is obvious to see that a decrease in energy consumption leads to an increase in the overall delay, and vice versa. A reliable solution would aim at finding an acceptable tradeoff between energy consumption and delay in WSNs [23].

Packet transfer consumes the highest amount of energy in the network. The higher the distance of transfer gets, the more energy is consumed. It is therefore preferable that the sensors communicate within the shortest radio range possible. Several solutions to preserve the network's energy have been investigated,

and they include the study of the topology. In this paper, we compare several network topologies and study their impact on the quality of health assessment.

The machine/component's health state goes through different classes varying from healthy to degraded. Health assessment consists in identifying the class corresponding to the current health state. In this article, the use of random forests (RF) is proposed for industrial functioning health assessment, particularly in the context of devices being monitored using a WSN. A prerequisite in prognostics and health management (PHM) is to consider that data provided by sensors is either flawless or simply noisy. WSNs monitoring is somehow unique in the sense that sensors too are subjected to failures or energy exhaustion, leading to a change in the network topology. Thus, the monitoring quality is variable too and it depends on both time and location on the device. To say this differently, to extend the life of WSN nodes will increase the monitoring duration, but it may decrease the diagnostic performance due to strategies deployed in the network (aggregation, scheduling, etc.) that enlarges noise in a certain way. Our aim is to show the effects of such strategies on the compromise between monitoring duration and quality, and to propose a diagnostic approach that is compatible with such strategies.

Indeed, various strategies can be deployed on the network to achieve fault tolerance or to extend the WSN's lifetime, like nodes scheduling or data aggregation. However, the diagnostic processes must be compatible with these strategies, and with a coverage of a changing quality [1, 11]. The objective of this research work is to show that RFs achieve a good compromise in that situation, being compatible with a number of sensors which may be variable over time, some of them being susceptible to errors. More precisely, we will explain why random methods are relevant to achieve accurate diagnostics of an industrial device being monitored using a WSN. Algorithms will be provided, and an illustration on a simulated WSN will finally be detailed.

The contributions of this article can be summarized as follows. The functioning of RF is recalled and applied in the monitoring context, when data are gathered by a wireless sensor network instead of a wired one. We show that diagnostic is more difficult in such networks, due to broken links, data loss, and deployed network topologies. To do so, three aggregation strategies to reduce packet transfers are considered, and their impact on the diagnostic accuracy is discussed qualitatively. It is evaluated numerically by comparing the random forests performance to state-of-the-art PHM approaches. Finally, a hybridation of the two best methods (random forests and gradient boosting) is proposed, to achieve the best RF hyperparameters selection by using gradient boosting.

The remainder of this paper is organized as follows. In Section 2 we give the state of the art. Section 3 presents the proposed algorithm for WSN based diagnostics, namely the random forests. Its performance on various sensor topologies is shown in the next section, while the RF-based diagnostic is compared to other machine learning methods in Section 5. This article ends by a conclusion section, in which the contribution is summarized and intended future work is outlined.

2 State-of-the-art review

To perform a periodic data gathering in a wireless sensor network, data aggregation is achieved through organizing the network according to a logical structure, mainly a tree or a clustering [28]. When a tree is used, aggregators are the internal nodes in the tree routed at the sink. With clustering structures, aggregators are the Cluster Heads (CH). In [14, 19], the authors prove that clustering methods provide better results for data aggregation, as they consume less energy. Defining a specific cluster and choosing the CH (aggregator node in the cluster) have an important impact on aggregation quality and energy consumption. Besides, structured approaches incur high maintenance overhead in event based applications. In fact, the source nodes change when a new event occurs. In other words, when the network starts running, the structure is fixed based on the positions of nodes sensing the event (source nodes). For the next round, the event may occur somewhere different in the network, which results in a change in source nodes. Consequently, the fixed structure will perform poorly [34].

Reference	**Context**	**Routing protocol**	**WSN drawback**
Kait *et al.*[21]	Paddy growth	Multi-hop rooting to nearest neighbor	Inefficient energy protocol
Yoo *et al.* [33]	Growing process of melon and cabbage	Parent-child tree	Single point failure
Yang *et al.* [32]	Irrigation	Through (widely separated) clusters	Inefficient energy protocol
Chiti *et al.* [7]	Agro-food production	Dynamic flooding	Inefficient energy protocol
Kabashi [20]	Agriculture	Shortest path graph	Sensing holes
CNS [17]	Agriculture	Tree structure	Single point failure

Table 1: Comparison of WSN based monitoring. Note that none of them provide a monitoring impact measurement of the WSN embedded protocol.

Several WSN topologies were used in existing monitoring applications, see Table 1. In [21], Kait *et al.* propose a WSN-based paddy growth monitoring system. Sensor nodes gather and send field data, such as temperature, periodically to the Base Station (BS). This is done by using multi-hop routing which is not considered energy efficient. Sensor nodes transmit data through the nearest neighbor which might lead to the longest path. Moreover, this routing protocol does not consider the energy level of the sensor nodes to generate transmission path. Another interesting study by Yoo et al. [33] proposes a precision and intelligence agricultural system referred to as the Automated Agriculture System. The goal of this system is to monitor and control the growing process of melon and cabbage in a greenhouse. In the system, sensor nodes are organized in a parent-child tree structure. The nodes join the network by broadcasting a parent search packet. Furthermore, the nodes transmit data to the BS using

three gateway nodes. However, the tree structure has a single point of failure. Yang et al. [32] developed an intensive WSN-based irrigation monitoring system. Sensor nodes are placed by this system in widely separated clusters. Thus, sensor nodes consume much energy for transmitting data to remote nodes in other clusters.

Chiti et al. [7] propose next generation firm for Agro-food productions. This system uses Ambient Intelligence and WSNs. The proposed system provides feedback and adaptability to increase productions in Agro-food. However, the deployed WSN uses a dynamic flooding inefficient-energy routing protocol. This is due to the fact that a large number of messages are broadcasted. Village eScience for Life [20] is a WSN-based agriculture project. It is implemented in developing regions in Africa and uses dynamic zone-based topology. This project initially deploys sensor nodes into zones in such a way that each sensor node remains within the transmission range of the nodes of at least two zones and each node belonging to a zone elects nodes in neighboring zones to which it can connect with a minimum transceiver power. Hence, several graphs are generated and the graph requiring minimum transmission power is selected for routing. However, this routing protocol does not guarantee to eliminate sensing holes. COMMONSense Net (CNS) [17] is another WSN-based agriculture monitoring project developed for semiarid regions in developing countries. The routing protocol of CNS uses tree structure which is not reliable since a link failure or sensor node failure can make other nodes unreachable to BS. Unlike the earlier works that focus mainly on the WSN-based monitoring applications, recent research [6] has significantly considered studying the actual structure of WSN through graph theory. In particular, geometric graphs are used in WSNs [22] to model the relationship between a sensor node and its neighboring sensor nodes [13, 24]. To sum up, each of the state-of-the-art algorithms contains WSN drawbacks, and none of them provide a monitoring impact measurement of the WSN embedded protocol.

Before studying WSN network dependability, we focused on finding an algorithm that is able to produce good diagnostics with incomplete monitoring data [2]. As summarized in Figure 1, maintenance strategies evolved through time and became predictive and condition based.

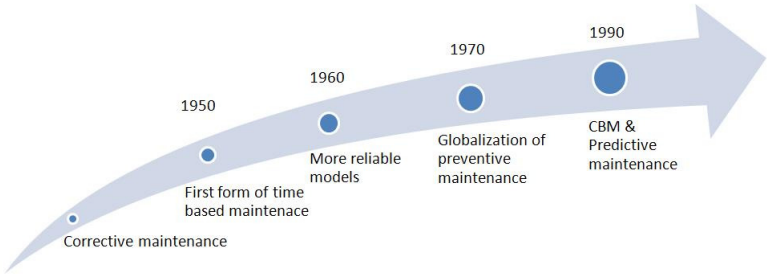


Figure 1: History of maintenance strategies.

Condition-based maintenance (CBM) is a proactive process for maintenance scheduling, based on real-time observations. It aims at assessing machine's health through condition measurements. As any maintenance strategy, CBM aims at increasing the system reliability and availability. The benefits of this particular strategy include avoiding unnecessary maintenance tasks and costs, as well as not interrupting normal machine operations [15]. In order to be efficient, a CBM program needs to go through the following steps [18], as illustrated in Figure 2.

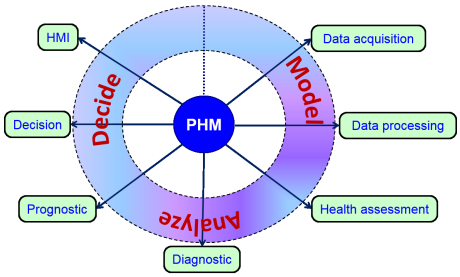


Figure 2: CBM Flowchart.

In this study, we limit our work to diagnostics. Sensory data are reported periodically to monitor critical components. These data correspond to measurements of parameters (pressure, temperature, moisture...), and are useful to assess the machine's condition. Thresholds related to the monitored parameters are fixed. Once a threshold is reached, the system is considered to be in the corresponding state. In Figure 3, the successive steps of a diagnostic process are illustrated.

Figure 3: Diagnostic's different steps.

- Fault detection is used to report an anomaly in the system behavior.
- Fault isolation is charged of determining and locating the cause (or source) of the problem. It identifies exactly which component is responsible of the failure.
- Fault identification aims at determining the current failure mode and how fast it can spread.

The diagnostics of a system's state of health is the equivalent of a classification problem. In machine learning, classification refers to identifying the class to

which a new observation belongs, on the basis of a training set and quantifiable observations, known as properties. In ensemble learning, the classifiers are combined to solve a particular computational intelligence problem. Many research papers encourage adapting this solution to improve the performance of a model, or reduce the likelihood of selecting a weak classifier. For instance, Dietterich argued that averaging the classifiers' outputs guarantees a better performance than the worst classifier [8]. This claim was theoretically proven correct by Fumera and Roli [12]. In addition to this, and under particular hypotheses, the fusion of multiple classifiers can improve the performance of the best individual classifier [30].

Two of the early examples of ensemble classifiers are Boosting and Bagging. In Boosting algorithm [26], the distribution of the training set changes adaptively based on the errors generated by the previous classifiers. In fact, at each step, a higher degree of importance is accorded to the misclassified instances. At the end of the training, a weight is accorded to each classifier, regarding its individual performance, indicating its importance in the voting process. As for Bagging [3], the distribution of the training set changes stochastically and equal votes are accorded to the classifiers. For both classifiers, the error rate decreases when the size of the committee increases.

In a comparison made by Tsymbal and Puuronen [29], it is shown that Bagging is more consistent but unable to take into account the heterogeneity of the instance space. In the highlight of this conclusion, the authors emphasize the importance of classifiers' integration. Combining various techniques can provide more accurate results as different classifiers will not behave in the same manner faced to some particularities in the training set. Nevertheless, if the classifiers give different results, a confusion may be induced. It is not easy to ensure reasonable results while combining the classifiers. In this context, the use of random methods could be beneficial. Instead of combining different classifiers, a random method uses the same classifier over different distributions of the training set. A majority vote is then employed to identify the class.

In this article, the use of random forests (RF) is proposed for industrial functioning diagnostics, particularly in the context of devices being monitored using a WSN. Up to now, a prerequisite in diagnostics is to consider that data provided by sensors are either flawless or simply noisy. This prerequisite must be relaxed in case where sensed data come from a wireless sensor network, as data aggregation, node scheduling, and other energy optimization strategies in possibly hostile environments lead to incomplete or totally erroneous sensed values. We will show that RF, detailed in the next section, can get around these problems, leading to an accurate diagnostics even in WSN harsh conditions, and even without feature selection.

Finally, as the other ensemble learning methods, RF can indicate the importance weights of predictors, which is a significant advantages of such approaches in the determination of the failure origin.

3 Proposed techniques

3.1 The research framework

As mentioned earlier in this article, the objective is to study the possibility of using random forests for prognostic and health management purposes. The latter have several advantages that make their use interesting in this context, such as their compatibility with time-varying feature vectors (which can happen, for example, when the sensors are on battery power: some batteries run out over time, and the associated feature therefore disappears when the battery is empty).

Our framework therefore consists of an industrial device on which predictive maintenance is deployed based on a wireless sensor network. Each sensor sends, as long as it can communicate (i.e., as long as it still has battery), its measurement periodically to the sink. These measurements are potentially noisy: typically, we want to deploy many sensors, therefore of poor quality, and in a potentially hostile environment (high or very low temperature, etc.) And since batteries can be drained or scheduling devices can be put in place to extend the life of the network, we therefore potentially have features missing over time. Finally, WSN-based PHM usually deploys data aggregation techniques, always in order to extend the network's lifetime, and this operation corresponds to feature aggregation.

Feature selection techniques are obviously to be implemented at the sink level in the case where the industrial system is large, leading to a large network of sensors (and therefore to a large number of features). This selection can be done in various ways, e.g. univariate feature selection or by using the spareness associated with ℓ_1 norms to preprocess the features. However, improving this feature selection step is not the objective of this article, and a great deal of work has already been produced on this theme.

Finally, based on a pre-established basis of knowledge, our framework consists in deploying random forests at the sink level, in order to be able to predict the RUL of the device under surveillance. This RF-based prediction is then compared to other tools traditionally used in PHM, and includes a phase of discovery of the best hyperparameters of each technique. These algorithms are the Suport Vector Machines (SVM), the Classification And Regression Trees (CART), AdaBoost, Gradient Boosting, and multinomial Naive Bayes.

3.2 The proposal

The RF algorithm is mainly the combination of Bagging [3] and random subspace [16] algorithms, and was defined by Leo Breiman as a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [4]. This method resulted from a number of improvements in tree classifiers' accuracy.

This classifier maximizes the variance by injecting randomness in variable selection, and minimizes the bias by growing the tree to a maximum depth (no

pruning). For the sake of completeness, the steps of constructing the forest are recalled in Algorithm 1.

Algorithm 1 Random forest algorithm

Input: Labeled training set S, Number of trees T, Number of features F.
Output: Learned random forest RF.

```
initialize RF as empty
for i in 1..T do
  S'_i ← bootstrap (S)
  initialize the root of tree i
  repeat
    if current node is terminal then
      affect a class
      go to the next unvisited node if any
    else
      select the best feature f* among F
      sub-tree ← split(S'_i, f*)
      add (leftChild, rightChild) to tree i
    end if
  until all nodes are visited
  add tree i to the forest
end for
```

In a RF, the root of a tree i contains the instances from the training subset S'_i, sorted by their corresponding classes. A node is terminal if it contains instances of one single class, or if the number of instances representing each class is equal. In the alternative case, it needs to be further developed (no pruning). For this purpose, at each node, the feature that guarantees the best split is selected as follows.

The information acquired by choosing a feature can be computed through either the well-known entropy of Shannon, which measures the quantity of information, or the reputed Gini index, which measures the dispersion in a population. The best split is then chosen by computing the gain of information from growing the tree at given position, corresponding to each feature as follows:

$$Gain(p,t) = f(p) - \sum_{j=1}^{n} P_j \times f(p_j) \tag{1}$$

where p corresponds to the position in the tree, t denotes the test at branch n, P_j is the proportion of elements at position p and that go to position p_j, $f(p)$ corresponds to either $Entropy(p)$ or $Gini(p)$. The feature that provides the higher Gain is selected to split the node.

4 Experimental results

4.1 Proposed protocol

In order to illustrate the impact of topologies on the quality of health estimations, we consider 90 sensor nodes; 30 nodes for each of the monitoring parameters: temperature, pressure, and humidity. The sensors are randomly placed in the simulation window, and are equipped with batteries of $100j$. The sink is also placed randomly. With every data transfer, the energy of a sender is reduced regarding its distance from the recipient.

Data simulation

- Under normal conditions, temperature sensors follow a Gaussian law of parameter $(20 \times (1 + 0.005t), 1)$, while these parameters are mapped to $(35, 1)$ in case of a malfunction of the industrial device. These sensors return the value 0 when they break down.
- The pressure sensors produce data following a Gaussian law of parameter $(5 \times (1 + 0.01t), 0.3)$ when they are sensing a well-functioning area. The parameters changed to $(20, 2.5)$ in case of area failure in the location where the sensor is placed, as long as the pressure sensors return 1 when they are broken down.
- The Gaussian parameters are $(52.5 \times (1+0.001t), 12.5)$ when both the area and the humidity sensors are in normal conditions. These parameters are set to $(80, 10)$ in case of area failure in the range of this sensor, whereas malfunctioning humidity sensors produce the value 3.

The probability that a failure occurs at time t follows an exponential distribution of parameter $1 \div 100$.

In other words, the predictors are constituted by 30 temperature variables, 30 pressure variables, and 30 humidity ones, they are all numerical. The dependent Y variable, for its part, is the number of failures. Note that the predictors are correlated (their Gaussian parameter depends on t), and that the reduced number of features does not require a selection. Although low, this number of features will still allow us to demonstrate the good performance of our approach in relation to the state of the art.

Data is generated as follows.

Each sensor received 100 units of battery, and 2000 units for each aggregator. This energy decreases over time, proportionally to the transmission distance (for both sensors and aggregators), and proportionally to the times spent to periodically collect a new data (i.e., computing a new random value according to the probabilistic model) and for aggregating (averaging) a collection of data. This duration is computed thanks to a call to the time function before and after the operation. The final number of packets corresponds to what have

Algorithm 2 Data generation

```
for each time unit t = 1..200 during the industrial device monitoring do
  for each category c (temperature, pressure, humidity) of sensors do
    for each sensor s belonging to category c do
      if s has not yet detected a device failure then
        s picks a new data, according to the Gaussian law corresponding to
        a well-functioning device, which depends on both t and c
        a random draw from the exponential law detailed previously is real-
        ized, to determine if a breakdown occurs on the location where s is
        placed
      else
        s picks a new datum according to the Bernoulli distribution of a
        category c sensor observing a malfunctioning device
      end if
    end for
  end for
end for
```

been definitively received at the sink level when all nodes have emptied their batteries.

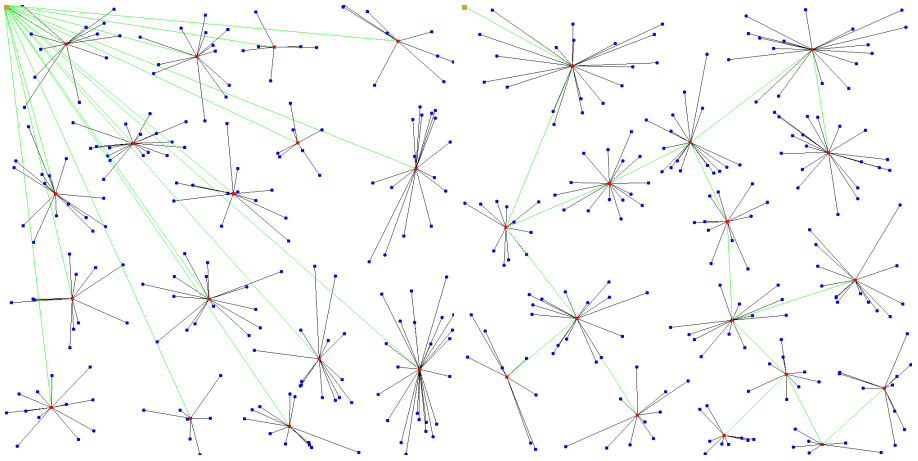
Considered topologies We have considered 3 different topologies during these simulations.

In the first scenario, we consider a default topology. When a node senses new data, it forwards it directly to the BS. At the end of each round, the sink will receive 30 different measures of temperature, pressure, and humidity each. The sink will only keep one value of each parameter. This is guaranteed by computing an average using a Gaussian distribution.

In the second scenario depicted in Figure 4a, 9 sensors are added to the topology. These sensors will be the aggregators (3 per parameter). Therefore, the topology now presents 9 clusters and in each, nodes send the sensed data to the CH. The CH aggregates the data packets from each round and sends the computed value of the relative parameter to the sink node. It should be noted that at this step, the CHs are placed randomly and their distance to their cluster members is not optimized.

In the third and last topology, we also considered 9 clusters. This time after all the sensors (CHs and regular nodes) are placed, each regular nodes finds the closest CH to it by using the K-mean algorithm, and adapts the same type (i.e., parameter). The aggregated data are then routed from CHs to CHs in direction to the sink, to reduce the communication cost. This topology is depicted in Figure 4b.

Let us notice that the first situation corresponds to what is usually considered in PHM. Conversely, the two other cases are related to data collected within a wireless sensor network, which thus embeds various strategies to increase the



(a) Situation 2 (b) Situation 3

Figure 4: Different strategies to route aggregated data to the sink

network lifetime, namely the data aggregation in our considered scenarios. Obviously, such aggregation may impact diagnostics, and usual machine learning algorithms are not designed to face such data manipulations.

4.2 Obtained results

We collect data in the network using the topologies described in Section 4.1. After data collection step, health assessment is performed through the RF algorithm described in Section 3. Nodes that capture new data packets forward the information (according to the corresponding network topology) towards the sink for processing. The data is then fed to the RF algorithm to assess the health of the monitored device.

We varied the number of trees in the forest from 1 to 100, and obtained in total 18 different forests. For each forest, we repeated the simulation 10 times. During the simulation, the sensors communicate the data generated following the laws described in Section 4.1. The simulations are timed, i.e., the simulation does not end when the system fails, but when the simulation time is reached. The decision for each tree is averaged over the 10 simulations, and the final decision is averaged over all the decisions given by each tree in the forest. In the following, we show the average number of errors in health estimation for each of the 3 proposed topologies.

In Figure 5 we plotted the average number of errors in health estimation, when all nodes can communicate with the BS. The error rate was maintained below 50% at all times. With the number of trees increasing in the forest, the error rate decreases and gets close to 0%. When the number of trees in the forest is more than 9, the error rate becomes almost constant.

Figure 6 shows the average number of errors in health estimation, when data

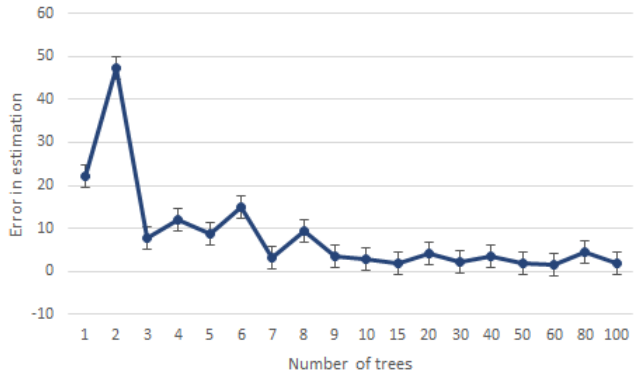


Figure 5: Error in health estimation for the star topology.

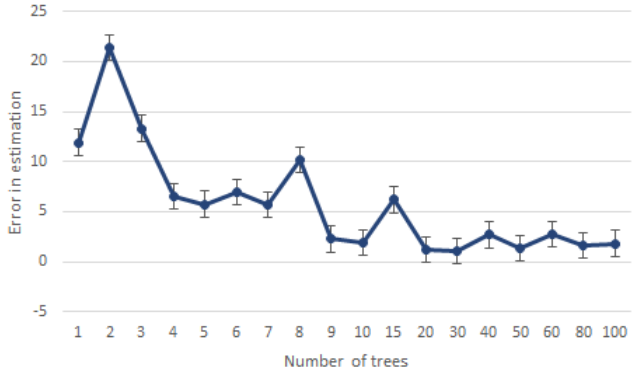


Figure 6: Error in health estimation for cluster topology.

is aggregated before being sent to the BS (as described in Section 4.1). The error rate, compared to the previous simulation, was reduced by half, and was stabilized when number of trees is greater than 20. Aggregating data reduces the frequency of transferring packets in the network; CHs will receive data from nodes within their range, combine them together and send them as one packet. As a result, the overall activity of sensors will be reduced, and consequently they will consume less energy. This means that sensors can live longer (comparing to the previous topology) to ensure transferring relevant data to the BS for health assessment. We can therefore conclude that reducing the number of packets in the network helps improve the quality of diagnostics.

In Figure 7 we plotted the average number of errors in health estimation, when nodes forward their data to the nearest aggregator. Error rate was reduced by almost a half when the distance of transfer is reduced, and reached 0% when the number of trees is greater than 80. Transferring data over a short distance requires less energy from the sender. This helps preserve energy for a longer

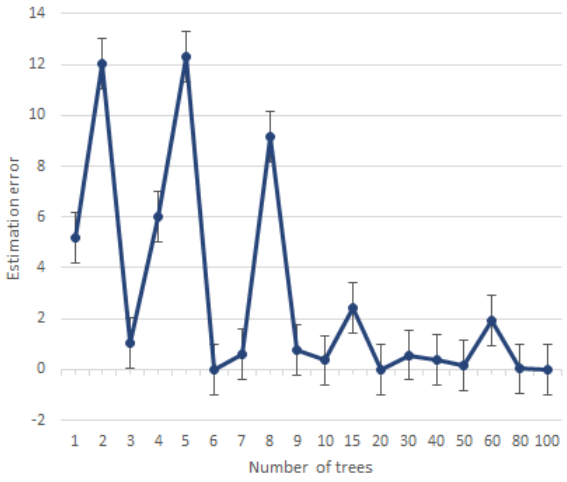


Figure 7: Error in health estimation for cluster topology with closest aggregator.

period and ensures that data needed for health assessment can be delivered to the BS over that time period.

To summarize, aggregating data packets ensures that nodes degrade gracefully (rather than abruptly) and results in more accurate estimations, which would have not been the case when using a common machine learning algorithm usually implemented for PHM (in wired case). Also, having nodes transfer their data over a short distance helps to preserve the available energy in the network. The point from which the error rate is stabilized can be considered as the optimal (or minimum) number of trees needed in the forest.

Figure 8 presents the delay between the time the system enters a failure mode and the time of its detection. This is done in the absence of correlations between the different features. The 0 time value of delay, the negative values, and positive values refer to in-time predictions, early predictions and late predictions of failures, respectively. The plotted values are the average result per number of simulations which varies from 1 to 100. With time, sensor nodes start to fail in order to simulate missing data packets. As a result, the RF algorithm was able to detect 54 % of the failures either in time or before their occurrence.

For each of the 100 performed simulations, we calculated the average number of errors in fault detection, produced by the trees in the forest. Figure 9 shows that this error rate remained below 15 % through the simulation. This error rate includes both "too early" and "too late" detections. When certain sensor nodes stop functioning, this leads to a lack on information, which has an impact on the quality of predictions; this explains a sudden increase in the error rate with time. We can conclude from the low error rate in the absence of some data packets that increasing the number of trees in the RF helps improve the quality and accuracy of predictions.

As described in Section 4.1, a correlation was introduced between the fea-

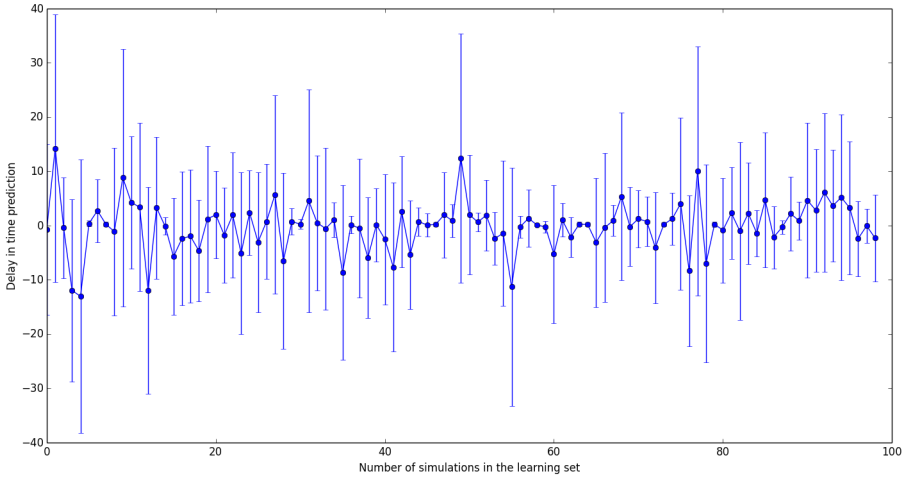


Figure 8: Delay in failure detection with respect to the number of simulations. X value represents the size of the learning set, while Y value is the averaged error between real and predicted RULs. Standard deviations are provided too.

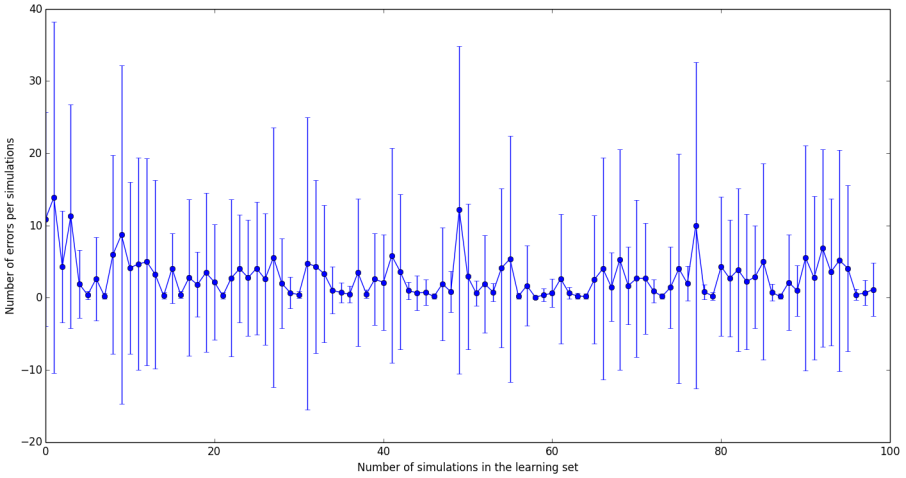


Figure 9: Error rate in health assessment with respect to the number of simulations. X is again the size of the learning set, while Y value measures the too-early vs. too-late detection. A value of five, for instance, means that there were 5 more too-early detection than too-late ones, for the considered learning size.

tures. Figure 10 shows the number of successful fault detection when the number of tree estimators in the forest changes. As shown in this figure, the RF method guarantees a 60 % success rate when the number of trees is limited to 5. As this

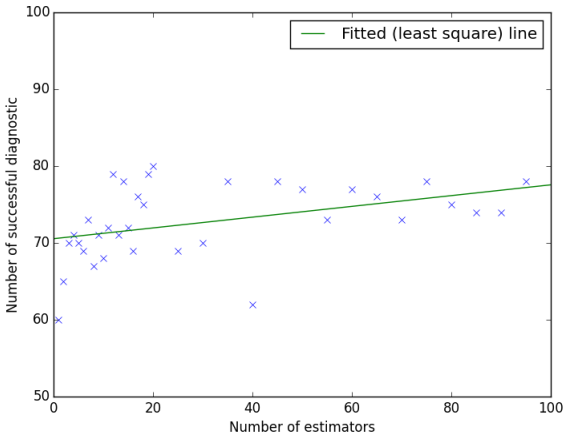


Figure 10: Number of successful health assessments with respect to the number of trees: the accuracy increases with the forest.

number grows, the accuracy of the method increases to reach 80 % when the number of trees is around 100. Comparing to the previous results, the correlation between the features helps decrease the uncertainties in health assessment when the number of trees increases. The algorithm is able to understand the relationship between two features. Thus, when some values describing a feature are missing, the algorithm can deduct them from the available information about the remained features.

5 Discussions

For the sake of discussion, we will evaluate in this section the RF-based diagnostic compared to other machine learning methods.

5.1 General comparison

Finding the optimal training of a classification problem is most of the times a real difficult problem. Tree ensembles have the advantage of running the algorithm from different starting points, and this can better approximate the near-optimal classifier. In his paper, Leo Breiman discusses the accuracy of Random Forests. In particular, he gave proof that the generalized error, although different from one application to another, always has an upper bound and so random forests converge [4].

The injected randomness can improve accuracy if it minimizes correlation while maintaining strength. The tree ensembles investigated by Breiman use either randomly selected inputs or a combination of inputs at each node to grow the tree. These methods have interesting characteristics as:

- Their accuracy is at least as good as Adaboost;

- They are relatively robust to outliers and noise;
- They are faster than bagging or boosting;
- They give internal estimates of error, strength, correlation, and variable importance;
- They are simple and the trees can be grown in parallel.

There are four different levels of diversity which were defined in [27], level 1 being the best and level 4 the worst.

- **Level** 1: no more than one classifier is wrong for each pattern.
- **Level** 2: the majority voting is always correct.
- **Level** 3: at least one classifier is correct for each pattern.
- **Level** 4: all classifiers are wrong for some pattern.

RF can guarantee that at least level two is reached. In fact, a trained tree is only selected to contribute in the voting if it does better than random, i.e., the error rate generated by the corresponding tree has to be less than 0.5, or the tree will be dropped from the forest [4]. Finally, in [31], Verikas *et al.* argue that the most popular classifiers like Support Vector Machine provide too little insight about the variable importance to the derived algorithm. They compared each of these methodologies to the random forest algorithm to find that in most cases RF outperform other techniques by a large margin.

This general discussion emphasizes that Random Forests should be considered in the context of PHM based on wireless sensor networks data [10], and that, due to their robustness and accuracy, they are real alternatives to state-of-the-art PHM algorithms. To illustrate this point by an experimental comparison between random forests and algorithms usually used for diagnosis such as Adaboost and SVM, a new series of simulations will be conducted in the section below.

5.2 Experimental comparison

Once again, we consider that data are gathered by the mean of a wireless sensor network in which sensor nodes have a limited lifetime, and strategies are deployed to optimize the network's lifetime like data aggregation and hop-by-hop routing. Data have been generated by our simulator as detailed in Section 4.1. As we take place in a WSN context, we considered that some nodes of the network are specifically designed to aggregate data from their neighboring sensor. 200 terminal nodes have been deployed, and 16 aggregators have been added. They have been linked to the closest terminal nodes according to the K-mean method. At each time an aggregator receives 3 values, it computes their average and transmits it towards the sink.

Situations 2 and 3 of Section 4.1 have been tested, depending on whether each aggregator sends its averaged values directly to the sink, or to the nearest aggregator that is closer to the sink. Note that this last situation reduces the transmission cost (thus enlarging the networks' lifetime), but data arrived to the sink are more averaged. In addition, all the sensors have limited batteries that are drained over time due to data transmission; they are spread randomly, if we except that the aggregators are well positioned thanks to the use of K-means. As a consequence, the sensors die one after the other as time goes by, impacting the evolution of the number of sensors having detected a failure.

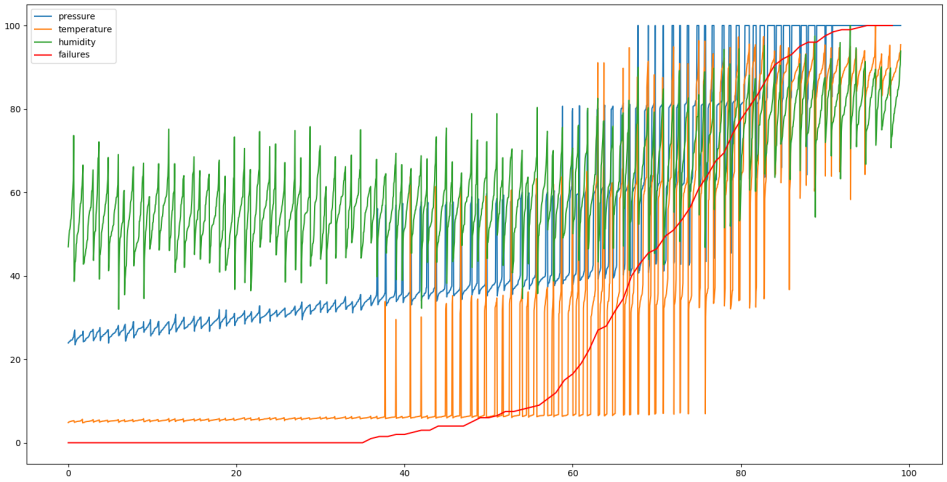


Figure 11: Failures at sink level when data are directly sent from each aggregator to the sink

A failure is randomly simulated according to a Poisson law. This failure disrupts the industrial system from close to close, and thus the number of sensors detecting aberrant values increases over time. In spite of the aggregation process, this increase is clearly observed at the sink level when the aggregators send their averages directly (Situation 2, see Figure 11), but tends to be less apparent when the averages are again aggregated during cluster routing (Situation 3, see Figure 12).

Various experiments have finally been conducted to compare the ability of Random Forests to accurately predict a failure to other machine learning approaches proposed in the PHM literature. The following regressors have been selected in this set of experiments, because they are frequently considered for prognostics and health management: a simple bagging of decision trees, the support vector machine, AdaBoost, Gradient Boosting, and multinomial Naive Bayes. Scikit-Learn [25] library has been used to implement the machine learning algorithms on data provided by our WSN simulator. No modification of the hyperparameter default values has been performed, due to the "meaningful default values" conception of this library: Scikit-Learn provides reasonable de-

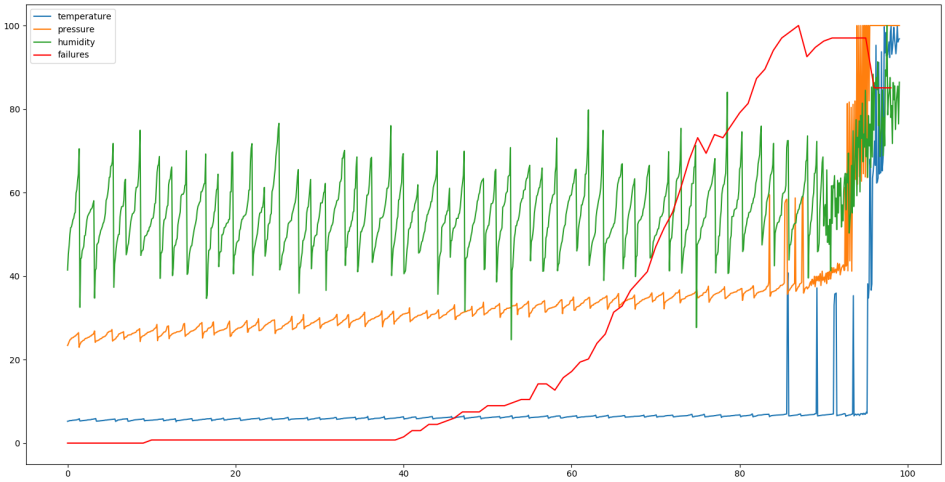


Figure 12: Failures at sink level when aggregated data are sent from aggregators to aggregators towards the sink

fault values for most parameters, making it easy and fast to create a basic and operational machine learning system. This is also the case for Random Forests, for which no hyperparameter optimization has been performed here, allowing an unbiased comparison of the various approaches (see the next section for a measure of performance increase when improving the hyperparameter selection).

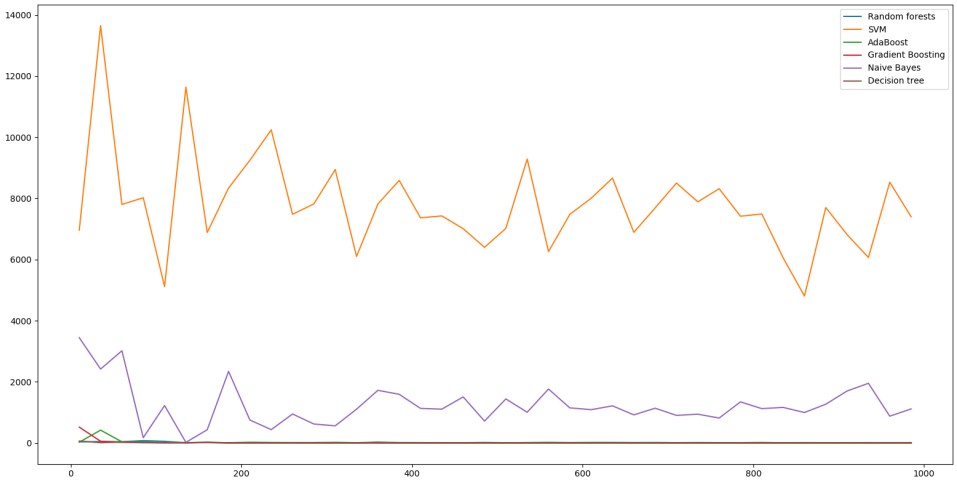


Figure 13: Comparison of mean absolute regression error in testing phase, for various machine learning algorithms in Situation 3.

At each time, the objective was to predict the number of sensors that de-

tected a failure from the aggregated data received at the sink. In doing so, we obtain a regressor capable of evaluating the severity of a failure, and we can easily make a classifier by looking at whether this number is strictly positive (there is a breakdown) or zero (there is no breakdown). The simulator has been launched several time, and N values collected at the sink level have been randomly picked from this basis of knowledge. The number of times the Poisson law has returned a new failure within sensors has been stored too as the objective function: the explanatory variables are the physical data captured and aggregated, and the variable to be explained is the number of failures. We tested forty N values equally distributed in the interval [0, 1000], to see if the regression error decreases when the basis of knowledge increases. Finally, 80% of these variables have been used for training, and the 20% remained values for evaluation during the testing stage.

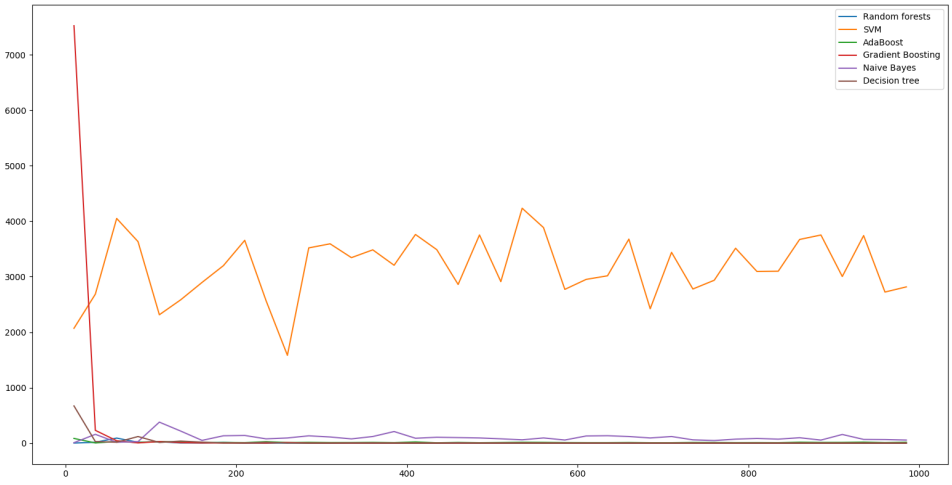


Figure 14: Comparison of mean absolute regression error in testing phase, for various machine learning algorithms in Situation 2.

As can be seen in Figure 13, both the Naive Bayes and SVM fail to reduce the regression error in Situation 3, even with the largest basis of knowledge. The same statement holds for SVM even in the simpler case of Situation 2, as can be seen in Figure 14. Obviously, the support vector machine fails to learn how to predict the severity of the failure, due to the fact that data have been averaged on some nodes in the network, and the same conclusion can be drawn, to a lesser extent, for the Naive Bayes method. In other words, the use of these methods for prognostic and health management must be seriously discussed in case the data are acquired via a wireless sensor network: energy saving strategies usually deployed in such networks can strongly impact their ability to make good predictions.

The four other machine learning algorithms reach good prediction scores in testing phase when a single aggregation stage is performed, as shown in

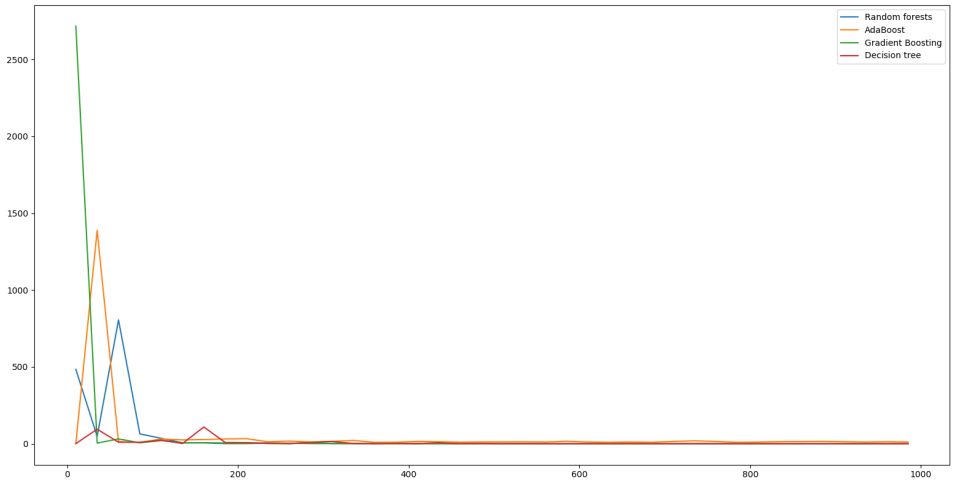


Figure 15: Comparison of mean absolute regression error in testing phase, for the best machine learning algorithms in Situation 2.

Figure 15. However, AdaBoost predictions are worse when several aggregation layers are made in the network, and the bagging of decision trees loses stability, as illustrated in Figure 16. To sum up, only Gradient Boosting was able to perform as well as Random Forests, in the context of a diagnostic on data gathered by a wireless sensor network embedding aggregation layers.

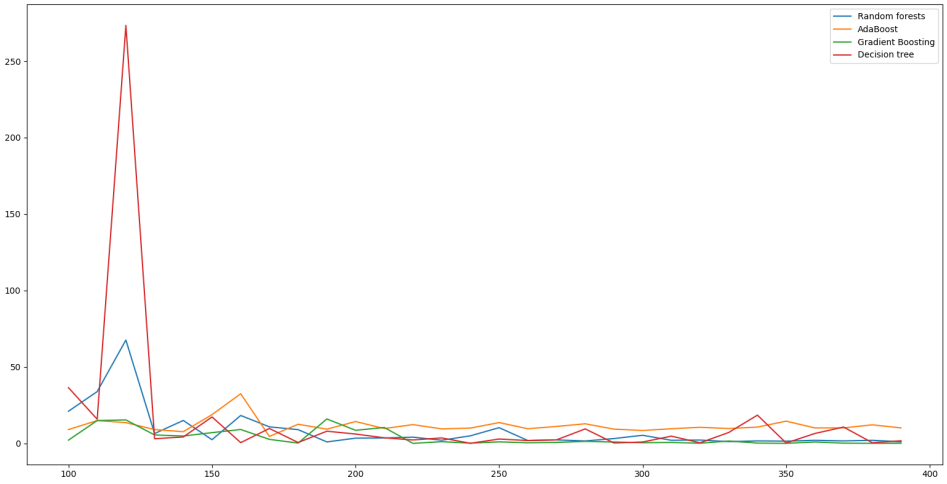


Figure 16: Comparison of mean absolute regression error in testing phase, for the best machine learning algorithms in Situation 3.

5.3 Hyperparameter optimization

The number of trees is not the only parameter to optimize in RF, and the regression error can be greatly reduced by playing on its many parameters. To illustrate this fact in a PHM scenario, we have considered the following parameters:

- *max depth*: the maximum depth of the tree.
- *max features*: the number of features to consider when looking for the best split.
- *min samples split*: the minimum number of samples required to split an internal node.
- *min samples leaf*: the minimum number of samples required to be at a leaf node.

The integer search interval has been defined as follows: between 1 and 10 for the *max depth* hyperparameter, between 1 and the total number of features for *max features*, between 2 and 1000 for *min samples split*, and finally between 1 and 100 for *min samples leaf*. The same dataset as in the previous section has been considered, and it has been separated again as learning and testing sets (80% and 20% respectively).

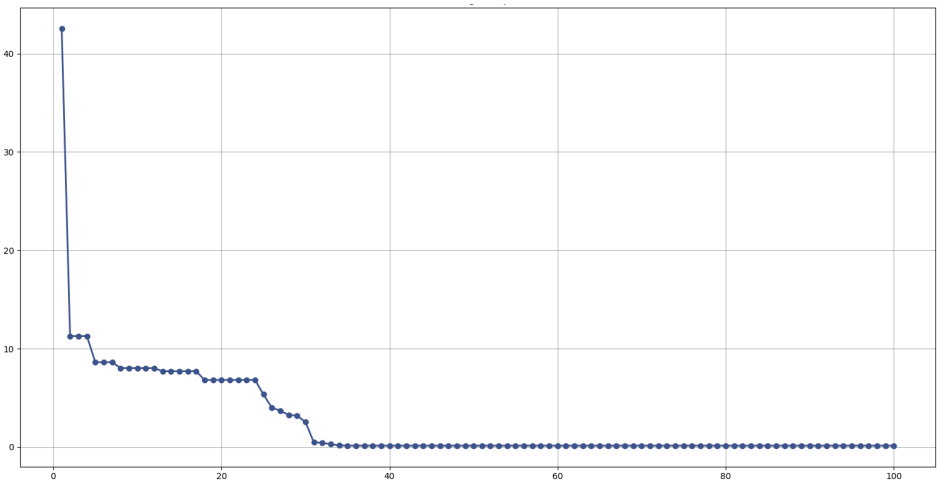


Figure 17: Convergence of mean absolute error function during RF's parameters optimization (number of calls in abscissa)

Various strategies are possible to achieve a hyperparameter optimization of the regression error in random forests. As GB and RF proved to be both finalists in the previous evaluation, we have considered here a mix of the two methods: gradient boosted regression trees have been used for RF hyperparameter selection, in which the model is improved by sequentially evaluating the

Regressor	RMSE	MAE	MAPE
Gradient Boosting	0.6602	0.1282	0.0091
AdaBoost	0.7885	0.3153	0.0091
CART	2.5494	1.8418	0.2693
SVM	59.7767	39.6415	2.6860
Random forests	0.2604	0.1218	0.0089

Table 2: Comparison of best Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) after hyperparameter optimization

score function at the next best point, thereby finding the optimum with as few evaluations as possible. The sequential optimization has been called 100 times and the optimum has been reached in 35 iterations, leading to a minimum of the mean absolute error between real and predicted number of failures equal to 0.1406. Best parameters are respectively equal to 10 (max depth), 4 (max features), 2 (min samples split), and 1 (min samples leaf). Obtained convergence curve is depicted in Figure 17, leading to a real improvement of RF performance to achieve reliable diagnostics on data collected within a WSN.

For the sake of completeness and fairness, this hyperparameter optimization has been performed too in the case of SVM (penalty parameter C of the error term), AdaBoost (learning rate and maximum number of estimators at which boosting is terminated), CART (max depth of the tree, minimum number of samples required to split an internal node, and minimum number of samples required to be at a leaf node), and gradient boosting (max depth, learning rate, number of boosting stages to perform, and minimum number of samples required to split an internal node). The optimization has been performed via a Bayesian optimization using Gaussian Processes, with a relevant search space depending on the considered regressor, and 100 iterations. Obtained results are compared in terms of Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE); they are provided in Table 2. As can be seen, the ensemble-based regressors can be really optimized, in such a way that they outperform the SVM. Note that random Forests and Gradient boosting have obtained in average the best results.

To put it in a nutshell, in the case where prognostic and health management is based on data gathered through a wireless sensor network, the prediction of the RUL should be based on Random Forest regression. This is for the following reasons. First of all, many regression algorithms are incompatible with the type of features produced by such networks. Indeed, data aggregation and scheduling policies, and the depletion of sensor batteries, cause feature vectors to have variable sizes over time. However, most machine learning techniques (SVM, neural networks...) are incompatible with these variable feature vector sizes. On the other hand, our simulations have shown that, even if these feature vectors remain fixed in size, the performance of random forests is better than that of the usual fault prediction techniques, for the various metrics considered,

and whether or not there is hyperparameter optimization.

6 Conclusions

In this paper, we proposed the random forests algorithm for diagnostics when the industrial device is monitored by a wireless sensor network. When the gathered data is incomplete, the algorithm adapts quickly to the change and continues to deliver reliable diagnostics. We also illustrated the impact of network topology on the quality of information at the sink level, by comparing two cluster topologies to the star one. We showed that organizing the network in clusters helps preserve the overall energy but reduces the quality of data used for diagnostics. We also showed that reducing the distance of packet transfer may impact the results. The relevance of random forests in such situations is explained and RF is compared to state-of-the-art PHM algorithms. Numerical experiments show that some of the latter have an obvious loss of accuracy when data are provided by a WSN, which is the case for instance of the support vector machines.

This good performance of the random forests for diagnostics in a wireless sensor network context has however been obtained only through simulations and qualitative discussion, which is a limitation of this research work. A real implementation of this algorithm in a deployed WSN should be operated, to reinforce the confidence put in RF for diagnostics in such kind of networks. Another limitation of this study is that only diagnostics aspects of PHM have been considered. This is why, in future work, we intend to develop a prognostic approach taking into consideration all the constraints discussed in this paper. We also intend to study the dependability of wireless sensor networks to improve both energy consumption and the quality of data at the sink level. The effects of an accurate feature selection on the performance of the aforementioned algorithms will be finally investigated deeply.

With the financial support of the EU (Feder) and the Swiss Confederation within the framework of the Interreg France-Switzerland programme, and the Labex ACTION one (contract ANR-11-LABX-01-01).

References

[1] Jacques M. Bahi, Wiem Elghazel, Christophe Guyeux, Mohammed Haddad, Mourad Hakem, Kamal Medjaher, and Noureddine Zerhouni. Resiliency in distributed sensor networks for prognostics and health management of the monitoring targets. The Computer Journal, 59(2), 2016.

[2] Jacques M. Bahi, Christophe Guyeux, Abdallah Makhoul, and Congduc Pham. Low cost monitoring and intruders detection using wireless video sensor networks. International Journal of Distributed Sensor Networks, 2012, 2012.

[3] Leo Breiman. Bagging predictors. Machine Learning, 24:123–140, 1996.

[4] Leo Breiman. Random forests. Machine Learning, 45:5–32, 2001.

[5] David W. Carman, Peter S. Kuus, and Brian J. Matt. Constraints and approaches for distributed sensor network security. Technical report, NAI Labs, The Security Research Division, Network Associates, Inc. Glenwood, September 2000.

[6] E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, H. Tejeda, J., and Urrutia. Half-space proximal: a new local test for extracting a bounded dilation spanner. In the International Conference On Principles of Distributed Systems, page 235–245, Pisa, Italy, 2006.

[7] F. Chiti, A. De Cristofaro, R. Fantacci, D. Tarchi, G. Collodo, G. Giorgett, and A. Manes. Energy efficient routing algorithms for application to agro-food wireless sensor networks. In the IEEE International Conference on Communication (ICC), page 3063–3067, Seoul, Korea, 2005.

[8] Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, 40:139–157, 2000.

[9] Wiem Elghazel, Jacques Bahi, Christophe Guyeux, Mourad Hakem, Kamel Medjaher, and Noureddine Zerhouni. Dependability of wireless sensor networks for industrial prognostics and health management. Computers in Industry, 68:1–15, 2015.

[10] Wiem Elghazel, Kamal Medjaher, Noureddine Zerhouni, Jacques Bahi, Ahmad Farhat, Christophe Guyeux, and Mourad Hakem. Random forests for industrial device functioning diagnostics using wireless sensor networks. In 2015 IEEE Aerospace conference, pages 1–9. IEEE, 2015.

[11] Ahmad Farhat, Christophe Guyeux, Abdallah Makhoul, Ali Jaber, and Rami Tawil. On the coverage effects in wireless sensor networks based prognostic and health management. International Journal of Sensor Networks (IJSNET), 28(2):125–138, 2018.

[12] Giorgio Fumera and Fabio Roli. A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6):942–956, 2005.

[13] K. Gabriel and R. Sokal. A new statistical approach to geographic variation analysis. Systematic Zoology, 18:259–278, 1969.

[14] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishna. Energy-efficient communication protocol for wireless sensor networks. In IEEE Proceedings of the Hawaii International Conference on System Sciences, January 4-7 2000.

[15] Aiwina Heng, Sheng Zhang, Andy C.C. Tan, and Joseph Mathew. Rotating machinery prognostics: State of the art, challenges and opportunities. Mechanical Systems and Signal Processing, 23:724–739, 2009.

[16] Tin Kam Ho. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8):832–844, 1998.

[17] P. Jacques, R. Seshagiri, TV. Prabhakar, H. Jean-Pierre, and HS. Jamadagni. Commonsense net: a wireless sensor network for resource-poor agriculture in the semiarid areas of developing countries. International Journal of Information Tecchnology, 4(1):51–67, 2007.

[18] Andrew K.S. Jardine, Daming Lin, and Dragan Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing, 20:1483–1510, 2006.

[19] Yan Jina, Ling Wanga, Yoohwan Kimb, and Xiaozong Yanga. Eemc. An energyefficient multi-level clustering algorithm for large-scale wireless sensor networks. Computer Networks, 52:542–562, 2008.

[20] AH. Kabashi and JMH. Elmirghani. A technical framework for designing wireless sensor networks for agricultural monitoring in developing countries. In the International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST), page 395–401, 2008.

[21] LK. Kait, CZ. Kai, R. Khoshdelniat, SM. Lim, and EH. Tat. Paddy growth monitoring with wireless sensor networks. In International Conference on Intelligent and Advanced Systems (ICIAS), page 966–970, Kuala Lumpur, Malaysia, 2007.

[22] W. Ke, W. Liqiang, C. Shiyu, and Q. Song. An energy-saving algorithm of wsn based on gabriel graph. In 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), page 1–4, Beijing, China, 2009.

[23] Soonmok Kwon, Jae Hoon Ko, Jeongkyu Kim, and Cheeha Kim. Dinamic timeout for data aggregation in wireless sensor netwoks. Computer Networks, 55:650–664, 2011.

[24] D. Matula and R. Sokal. Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane. Geographical Analysis, 12(3):205–222, 1980.

[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. Journal of machine learning research, 12(Oct):2825–2830, 2011.

[26] Robert E. Schapire. A brief introduction to boosting. In Proceedings of the sixteenth International Joint Conference on Artificial Intelligence, 1999.

[27] A. Sharkey and N. Sharkey. Combining diverse neural nets. The Knowledge Egineering Review, 12(3):231–247, 1997.

[28] Ignacio Solis and Katia Obraczka. The impact of timing in data aggregation for wireless sensor networks. In Proceedings of the IEEE International Conference on Communications, page 3640–3645, 2004.

[29] Alexey Tsymbal and Seppo Puuronen. Bagging and boosting with dynamic integration of classifiers. In The 4th European Conference on Principles and Practice of Knowledge Discovery in Data Bases PKDD, pages 116–125, 2000.

[30] K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. Connection Science, 8:385–404, 1996.

[31] A. Verikas, A. Gelzinis, and M. Bacauskiene. Mining data with random forests: A survey and results of new tests. Pattern Recognition, 44:330–349, 2011.

[32] W. Yang, H. Liusheng, W. Junmin, and X. Hongli. Wireless sensor networks for intensive irrigated agriculture. In the Consumer Communications and Networking Conference (CCNC), page 197–201, Las Vegas, NV, USA, 2007.

[33] S. Yoo, J. Kim, T. Kim, S. Ahn, J. Sung, and D. Kim. A2s: automated agriculture system based on wsn. In the IEEE International Symposium on Consumer Electronics (ISCE), page 1–5, Dallas, TX, USA,, 2007.

[34] Hamed Yousefi, Mohammad Hossein Yeganeh, Naser Alinaghipour, and Ali Movaghar. Structure-free real-time data aggregation in wireless sensor networks. Computer Communications, 35(9):1132–1140, May 15th 2012.