Energy-Efficiency and Coverage Quality Management for Reliable Diagnostics in Wireless Sensor Networks

Ahmad Farhat[1], Christophe Guyeux[1], Mohammed Haddad[2], and Mourad Hakem[1]

[1]Disc Laboratory, Femto-ST Institute, UMR 6174 CNRS, Université de Bourgogne
Franche-Comté, France
[2]LIRIS Laboratory, UMR CNRS 5205, Université de Lyon 1, F-69622, France

Abstract

The processing of data and signals provided by sensors aims at extracting relevant features which can be used to assess and diagnose the health state of the monitored targets. Nevertheless, Wireless Sensor Networks (WSNs) present a number of shortcomings that have an impact on the quality of the gathered data at the sink level, leading to imprecise diagnostics of the observed targets. To improve data accuracy, two main critical and related issues, namely the energy consumption and coverage quality, need to be considered. In this paper, we present a distributed algorithm based on a theory of domination in graphs, and we study its impact on diagnostics by using six machine learning algorithms. First, we give the correctness proofs and next we assess its behavior through simulations. Obtained results show that the proposed algorithm exhibits good performances despite its lower complexity.

Keywords: sensor networks; distributed algorithms; lifetime optimization; coverage; diagnostics; domination in graphs.

1 Introduction

Wireless sensor networks are widely used in diverse fields of applications such as smart spaces, health care, tracking, machine failure diagnosis, battlefield surveillance, home security, and so on. They are constituted of many low cost and low powered sensor nodes that collaborate with each other to monitor, gather, process, and forward information using wireless communications. Each sensor node is a small sized device with a limited energy supply which, in most cases, is non refillable especially in remote and hostile environments.

Due to energy constraints and the inherent nature of WSN, resource failures may frequently occur and have an adverse effect on the monitoring application. This leads to inaccurate data and therefore to inefficient diagnostics of the observed targets. Consequently, there is an increasing need for developing techniques to achieve reliability in WSNs. The network should be able to continuously deliver a correct service and accurate data for a long time period. To this end, sensing coverage with minimal energy consumption needs to be considered. Unfortunately this problem is known to be **NP-hard** in its general form and heuristics are required to achieve sub-optimal solutions but in polynomial time complexity.

By and large, coverage and energy saving are the most fundamental and related issues in WSN. They reflect how well the observed targets are monitored, and serve as a basis for QoS of the surveillance activity. Although both target coverage and energy saving have been studied and reported in the literature, to the best of our knowledge, none of the existing research works has considered their impact on the quality of the sensor readings. Indeed, when it comes to diagnostics, it is usually assumed that the available data at the sink level are accurate and complete. These assumptions are far from being true and the resulting diagnostics cannot provide reliable understanding of the relationship between what we observe and what happened before in real-world applications. Thus, it is crucial to consider a diagnostic-oriented reliability in WSNs.

In this paper, we focus on data accuracy under coverage and energy constraints, which is an important matter due to its direct influence on the diagnostics of the observed targets. The presented algorithm is a hybridization of three variants of domination in the sensors-targets graph, namely the simple, double, and total ones, in a single algorithm. Major achievements include a low time complexity and a good compromise between energy and coverage performance related objectives, leading to more accurate and efficient diagnostics for a long time period. In the following, we summarize the main contributions and novelties of the presented work.

- We orchestrate the trade-offs relationship between energy saving and coverage optimization and their impact on the accuracy of the provided outcomes at the sink level.

- Unlike earlier works, we use a new concept of domination in graphs [9], to achieve significant energy/coverage enhancement and therefore to increase the reliability of the diagnostic process.

- We provide provable guarantees for the election process of the minimal dominating sensor-targets. Our algorithm can start from any initial configuration with arbitrary number of awaken nodes, and reach a legitimate stable situation in a finite number of steps.

- To assess the algorithm's efficiency, six machine learning algorithms (Support Vector Machine (SVM) [8], Naive Bayes (NB) [25, 34], Random Forests (RF) [3, 7], Gradient Tree Boosting (GTB) [4], Tree-Based Feature Selection (TBFS) [27], and Nearest Neighbors (NN) [23]) have been used to diagnose the state of the monitored targets.

The remainder of this article is organized as follows. Section 2 is devoted to the review of related work. In Section 3, we present in details the proposed algorithm and give the correctness proofs. We report in Section 4 series of numerical results that assess the behavior of the presented algorithm. Section 5 summarizes the results of this work and draws conclusions.

2 State of the art

2.1 Literature review

In the literature, the problem of target coverage with minimal energy consumption has been formulated and studied in various ways [1, 16, 19–22, 28, 30, 31, 33, 36]. To improve the energy preservation problem in WSNs, the authors in [35] studied three different approaches, all maintaining the initial coverage QoS. The first approach focuses on optimizing coverage deployment strategy, while the second one consists of planning a scheduling of active sensors that enables other sensors to switch to a sleep mode. Finally, the third approach is to adjust the sensing range of sensors for energy preservation.

A distributed node scheduling mechanism is presented in [29]. The possibility of one node to become a redundant one can be analyzed according to the relationship of the positions of different nodes. If a node is a redundant one, it moves to sleep. The algorithm assesses the redundant nodes by considering only the neighbors in its sensing range, while there are still other redundant nodes in active mode. Thus, the performance of this mechanism needs to be enhanced.

A centralized k-decision algorithm is provided in [14,15]. It only needs to detect whether arbitrary nodes in the perimeter of the sensing disk can be covered by other k nodes. Then, the mechanism can determine if the entire region is k-covered. Unfortunately, the algorithm is centralized with high complexity, in particular for large-scale WSN applications.

The mechanism presented in [32] is a k-coverage protocol, which ensures that if any point within the intersection area of sensing disks can be k-covered, then the entire region also can be k-covered. However, the mechanism does not consider the contribution value of the network, when a node changes from the sleep state to the active one. This leads to have a network coverage of low efficiency and nodes of larger redundancy degree.

In [5, 6], the authors proposed a distributed connected coverage algorithm. The method aims to minimize the number of active nodes in order to save energy and extend the network lifetime.

The basic idea of this algorithm is to construct a dominating set of network. Through the periodic reconstruction of the dominating set, the mechanism can effectively extend the network lifetime.

Authors of [26] construct continuous disjoint sets. At each time step, sensor nodes that can cover as many targets as possible are selected to form the cover sets. Although the mechanism in [26] can ensure that all targets are covered in the network, the mechanism has a very stringent constraint: at each time, the constructed sets must be disjoint.

The authors in [17] discuss how to maximize the network lifetime when the deployment environment of the network is safe and controllable, and they guarantee that all the network targets have been covered and that the network is connected. The work presented in [12] assumes that the network is homogeneous, *i.e.*, the sensing radius of the nodes in the network are all equal, and the communication radius of nodes is equal to the sensing radius. Then, the paper proposes a polynomial approximate algorithm, based on minimum spanning tree, which improves the performance of the network.

In [24], the k-target coverage problem in wireless sensor networks has been studied, and an algorithm that does not require location information of the nodes has been presented. Based on a mathematical model presented in [24] and on the ratio of node's sensing radius to monitor a given area, the algorithm is able to know the number of sensor nodes that are needed to cover the monitoring area. The algorithm could greatly reduce the hardware cost in the network, and it can improve the QoS of the network.

The authors in [18] leverage prediction to prolong the network lifetime, by exploiting temporal-spatial correlations among the data sensed by different sensor nodes. Based on Gaussian Process, the authors formulate the issue as a minimum weight sub-modular set cover problem and propose a centralized and a distributed truncated greedy algorithms (TGA and DTGA). They prove that these algorithms obtain the same cover set.

In [11], the authors design a polynomial-time distributed algorithm for maximizing the lifetime of the network. They proved that the lifetime attained by their algorithm approximates the maximum possible lifetime within a logarithmic approximation factor.

Lifetime optimization using knowledge about the dynamics of stochastic events has been studied in [10]. The authors presented the interactions between periodic scheduling and coordinated sleep for both synchronous and asynchronous dense static sensor network. They show that the event dynamics can be exploited for significant energy savings, by putting the sensors on a periodic on/off schedule.

Finally, in [13], a general coverage algorithm, which also considers the network connectivity, has been presented. The proposal, called Probabilistic Coverage Protocol (PCP), works for the common disk sensing model as well as for the probabilistic sensing one. To support the latter, authors have introduced the notion of probabilistic coverage of a target area with a given threshold θ, which means that an area is considered to be covered if the probability of sensing an event occurring at any point in the area is at least θ. They prove the correctness of the protocol and provide bounds on its convergence time and message complexity.

As stated above, to our knowledge, none of the aforementioned research works has studied the impact of coverage and energy related objectives on the quality of the provided data by a WSN. This is why, in this paper, we present a distributed algorithm to achieve reliable sensor reading without scarifying the global energy and coverage QoS of the deployed network.

In order to compare our algorithm to PCP [13] protocol, which is the closest work to the one presented in this paper, we briefly outline hereafter the main features of this algorithm.

2.2 Probabilistic Coverage Protocol

The idea of PCP is to activate a subset of deployed sensors to form an approximate triangular lattice over the area to be covered. PCP works in rounds of R seconds each. R is chosen to be much smaller than the average lifetime of sensors. In the beginning of each round, all nodes start running PCP independent of each other, then a number of messages will be exchanged between nodes to determine active/sleep nodes. PCP refers to the distance between the vertices of the triangular lattice as the maximum separation between working nodes, and it is denoted by s. The value of s is computed

from the sensing range r_s of sensor nodes. In the disk sensing model, the maximum separation is set to $\sqrt{3}r_s$ as it has been shown in [2].

Under the exponential sensing model, to ensure that the probability of sensing at the least-covered point is at least the coverage threshold parameter θ, the authors compute the maximum separation, s, as: $\sqrt{3}\left(r_s - \frac{\ln\left(1-\sqrt[3]{1-\theta}\right)}{\alpha}\right)$, where α is a factor that describes how fast the sensing capacity decays with distance. From this equation, it is clear that if we set $\alpha = \infty$, then the exponential sensing model reduces to the disk sensing model.

Unlike our algorithm, PCP protocol assumes nodes to have positioning informations. This helps to construct a triangular lattice allowing high performance for the coverage issue. Conversely, the algorithm we develop is completely distributed and does not need to know the positions of any sensor node in the network.

3 The proposed algorithm

3.1 Definitions and algorithm

Before getting into details of our algorithm, we introduce some definitions and notations: let $G = (V; E)$ be the graph modeling the network, where V is the vertex set and E is the edge set, such that $|V| = n$ and $|E| = m$. In our study, we have $V = S \cup T$ where S is the set of sensor nodes and T is the set of targets.

Our algorithm will construct either, a dominating set, a total dominating set or a double dominating set according to the following definitions :

- A Sensor-Target dominating set is a subset D of S such that every element of $V = S \cup T$ is either in D or is adjacent to at least one element of D.

- A Sensor-Target double dominating set is a subset D of S such that every element of $V = S \cup T$ is either in D or is adjacent to at least two element of D.

- A Sensor-Target total dominating set is a subset D of S such that every element of $V = S \cup T$ is adjacent to at least one element of D.

The algorithm will aim to provide minimal constructions of such sets.

In our algorithm, we assume nodes' identifiers to be unique. We recall that node identifier is unique if and only if $i.Id \neq j.Id$ holds for each $i, j \in V\,(i \neq j)$. A sensor node can be in one of these three states: IN, OUT, or WAIT, while a target node can only be in OUT state.

The algorithm is presented in an Event-driven programming way. It is composed of a set of rules, each handling a particular event that could happen on target or sensor nodes.

3.2 Correctness and convergence proofs

In this section we first prove the correctness of the proposed algorithm, that is: once the algorithm converges, the system will be in a legitimate configuration. This is proven in the three first following lemmas, after what the convergence will be considered. The complexity is computed in rounds, *i.e.*, executions are observed in rounds, where a round is defined as follows. This is the smallest sequence of an execution, in which every node that was eligible at the beginning of the round either executes one of its enabled rule, or is disabled by the execution of some rules by neighbors. Observe that the last case could only happen if the system is asynchronous. In a synchronous distributed algorithm, enabled nodes operate in synchronous rounds and execute their rules simultaneously.

Lemma 1 *Once the algorithm terminated, there is no node in the WAIT state.*

PROOF If we suppose the opposite, there will be at least one node activated either for rule [R3] or for rule [R5]. A contradiction with algorithm's termination.

Algorithm 1: Distributed minimal parametric domination Algorithm

---

Nodes: v is the current node, u is a neighbor of v

Constants: Double: a boolean that indicates whether domination is double or simple.
                   Total: a boolean that indicates whether domination is total or not.

Variables: v.Need = Lacking, Level, Over, indicates if a node has enough dominators in
                                   its neighborhood;
              v.State = OUT, WAIT, IN, indicates whether the node is IN or OUT the
                               dominating set. WAIT is meant to show
                               neighbors that will have to leave the set;
              v.Leave = pointer to a node u or NULL, indicates which neighbor with WAIT
                               state to authorize to leave the dominating set;

Macros: N(v) is the set of neighbors of v;
          Count(v) the number of neighbors having state IN or WAIT;
          Need(v) decides if v has sufficient number of dominators in its
                 neighborhood and returns either Lacking, Level, or Over value;
          Leave(v) returns a pointer to a neighbor that v authorizes to leave the
                 dominating set;
          NeighborInNeed(v) returns a boolean indicating whether v has a neighbor
                 lacking of dominators in its neighborhood;

Count(v) $\equiv$ return $|\{u \in N(v) : \text{u.state} = \text{IN or WAIT}\}|$

$$\text{Need(v)} \equiv \begin{cases} \textbf{if } \text{Double } \textbf{then } \text{NbDom} = 2 \textbf{ else } \text{NbDom} = 1 \\ \textbf{if } \text{Count(v)} < \text{NbDom } \textbf{then } \text{return Lacking} \\ \textbf{else if } \text{Count(v)} = \text{NbDom } \textbf{then } \text{return Level } \textbf{else } \text{return Over} \end{cases}$$

$$\text{Leave(v)} \equiv \begin{cases} \textbf{if } (\text{Total or state} = \text{OUT}) \text{ and need } != \text{Over } \textbf{then } \text{return NULL} \\ \textbf{else } \text{return} \min\{u \in N[v] : \text{u.state} = \text{WAIT}\} \end{cases}$$

NeighborInNeed(v) $\equiv$ return $\exists u \in N(v) : (u.state = OUT \text{ or } TOTAL) \text{ and } u.need = Lacking$

Rule executed only on target nodes:

[R0]: Ensures target nodes are outside the dominating set
if $v.state \neq OUT$ **then** $v.state := OUT$

Rule executed only on both target and sensor nodes:

[R1]: Updates Need and Leave variables
if $v.Need \neq Need(v) \vee v.Leave \neq Leave(v)$ **then** $v.Need := Need(v); v.Leave := Leave(v);$

Rules executed only on sensor nodes:

[R2]: Makes node v join the dominating set
if $v.state = OUT \wedge NeighborInNeed(v)$ **then** $v.state := IN;$

[R3]: Cancels v's will to leave the set
if $v.state = WAIT \wedge Need(v) = Lacking \vee NeighborInNeed(v)$ **then** $v.state := IN;$

[R4]: Indicates that v considers to leave the set
if $v.state = IN \wedge Need(v)! = Lacking \wedge \forall u \in N(v) : (u.state = OUT \vee TOTAL) \wedge u.need = Over$
then $v.state := WAIT;$

[R5]: Makes v leave the dominating set
if $v.state = WAIT \wedge Need(v)! = Lacking \wedge \forall u \in N[v] : u.leave = v$ **then** $v.state := OUT;$

Lemma 2 *Once the algorithm terminated, the set formed by nodes with state = IN forms a (total/double/simple) dominating set according to the specification.*

PROOF If it is not the case, there will be at least one node activated for rule [R1]. A contradiction.

Lemma 3 *Once the algorithm terminated, the set formed by nodes with state = IN is minimal.*

PROOF If it is not the case, there will be at least one node activated for rule [R4] because it is an unnecessary dominator. A contradiction.

Lemma 4 *After the two first rounds, all nodes will have their variables set to correct values according to the algorithm.*

PROOF Rule [R1] is responsible to update *Need* and *Leave* variables, all other rules update the *state* variable. After two execution rounds, all activated nodes will have the possibility to execute [R1] (if activated) and another rule modifying the *state* variable.

Lemma 5 *After the third round, all nodes will be dominated.*

PROOF By previous lemma, all not dominated nodes will have their *Need* variables set to *Lacking*. Thus all neighbors of a not dominated node will be activated to join the dominated set by rule [R2]. Thus, after this third round no node will have its *Need* variable set to *Lacking*.

Lemma 6 *After the fourth round, at least one unnecessary dominator node will leave the dominating set every two rounds.*

PROOF At the fourth round, all unnecessary dominators will have their *state* variables set to *WAIT* by executing [R4]. Then, to leave the dominating set, a *WAIT* node will have to be elected by its neighborhood. To do so, neighbors will elect the node to leave the set with their *Leave* pointers. This process will need one round. Once election finished, at least the node with minimum ID and having its state set to *WAIT* will be activated for rule [R5] and leave the dominating set. Thus two rounds will be needed for each node to leave the dominating set.

Theorem 1 *The algorithm constructs a minimal (total/double/simple) dominating set within $\mathcal{O}(n)$ distributed execution rounds.*

PROOF A direct consequence of all previous lemmas. The exact count of execution rounds is $2n + 4$.

4 Experimental results

4.1 Experimental protocol

Let us first notice that, for the sake of simplicity, we will use throughout this section the terms "area" and "target" indifferently. Indeed, we can transform the area coverage into the target coverage problem, by dividing the monitored area into a number of monitored fields where each field can be treated as a target. We now intend to show how the monitoring process is affected by the behavior of the proposed algorithm in WSNs. We focus on studying the diagnostics of the monitored area to examine its state.We used three types of sensing measures: temperature, pressure, and humidity, in order to simulate a WSN monitoring a given area. Each sensor type captures specific data depending on the operating age t, and we consider that there is no correlation between the different features:

- Under normal conditions, temperature sensors follow a Gaussian law of parameter $(20 \times (1 + 0.005t), 1)$. In case of a malfunction of the monitored target, these parameters are mapped to $(350, 20)$. Finally, these sensors return the value 2 when they break down.

- The pressure sensors produce data following a Gaussian law of parameter $(5 \times (1 + 0.01t), 0.3)$ when the monitored target is in normal conditions. The parameters changed to $(20, 2.5)$ in case of an abnormality of the monitored target, as long as the pressure sensors return 1 when sensor nodes fails.

- The Gaussian parameters are $(52.5 \times (1 + 0.001t), 12.5)$ when both the monitored target and the humidity sensors are in normal conditions. These parameters are set to $(80, 10)$ in case of abnormality, whereas malfunctioning humidity sensors produce the value 3.

Each sensor follows a Poisson process (Pp) of parameter $(200 \times (1 - 0.01t) + 0.01)$, to determine if an abnormal event occurs in the location where the sensor is placed. Each category of sensors has its own constant threshold, depending on the abnormality of the sensed data. If the data captured by the sensor in a specific category exceeds the prescribed threshold, this indicates that an abnormality has been detected. The threshold values are considered as follows: 26 degrees for temperature, 7 bars in pressure, and 80 percents of humidity. The region to be monitored is a rectangle of area $A = L \times W$, where L and W are the length and the width of the region of interest respectively. The area of the coverage range of a sensor is mostly related to the area of the monitored region. Therefore we consider that the coverage area is set to be equal to 1% of the total area of the region. Subsequently, the coverage radius will be $R_c = 1/10 \times \sqrt{A/\pi}$. And we consider that the radio radius $R_r = 2R_c$.

As stated previously, we use six machine learning algorithms to diagnose the monitored system. These algorithms are: Support Vector Machine (SVM) [8], Naive Bayes (NB) [25, 34], Random Forests (RF) [3, 7], Gradient Tree Boosting (GTB) [4], Tree-Based Feature Selection (TBFS) [27], and Nearest Neighbors (NN) [23].

Let us recall that, 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. Machine learning displays a detailed study about the system and from it, an algorithm is built. These algorithms can be operated by building a model from example inputs, in order to be able to make predictions or decisions. We thus need a large and reliable data set in order to train these algorithms. And so, we can later diagnose the system's health state from the new data that will be provided by WSN. To this end, we take data consisting of N lines, each line is composed by T temperature data, P data of pressure, and H data of humidity to train these algorithms. All of these data are generated in the way mentioned above (same type of data that will be captured by WSN during area monitoring).

In the following sections, we assess the impact of the proposed algorithm on the quality of data provided by the WSN over time, and therefore its consequence on the accuracy of diagnostic. We first remark that density is a very important factor in such networks, which has an effective impact on both lifetime and coverage. Indeed, the average distance between sensors decreases when density increases. This affects the packet transfer distance on the one hand, and the number of neighbors for each sensor on the other hand. Thus, to take this fact under consideration, we performed simulations using two densities of WSNs: 300 and 600 sensors respectively.

Figure 1 shows the variation of error rate over time for the six considered algorithms, in the case where the proposal is applied in simple domination mode. Each point in the figures represents the average of error rates on 20 simulations. As shown in Figure 1a (with 300 sensors), during the time-steps $t = 1..120$, each algorithm exhibits a specific but quite constant error rate. However, after the time-step $t = 120$, the error rate increases significantly for each algorithm. Indeed, the number of sensor nodes that "die" due to an empty battery (or fail-stop failure) becomes non negligible, leading to uncovered targets in the area (i.e., coverage holes) and therefore to incomplete data for diagnostics. More precisely, after $t = 120$, errors increase as time goes up to finally reach a rate of 91 % at $t = 200$ in the support vector machine, 90 % for NB, 89 % for RF, 90 % for GTB, 89 % for TBFS, and 91 % for NN. Finally, at time-step 200, almost all the network is inactive. This phenomenon is less sensitive in the case of 600 sensors, as depicted in Figure 1b, even if a similar observation can be outlined about the importance of density in wireless sensor networks.

Figure 2 also presents the variation of error rate for the six considered algorithms, but now in the case where the proposal is in double domination mode. When considering 300 sensors, we can see in
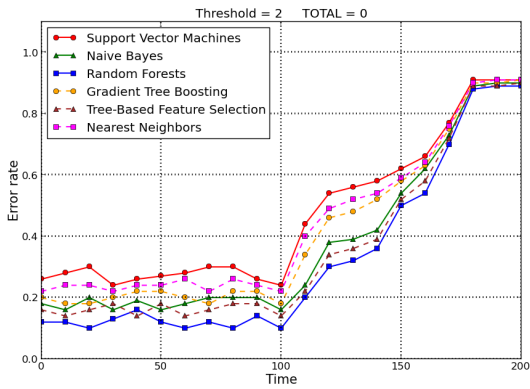
<div align="center">
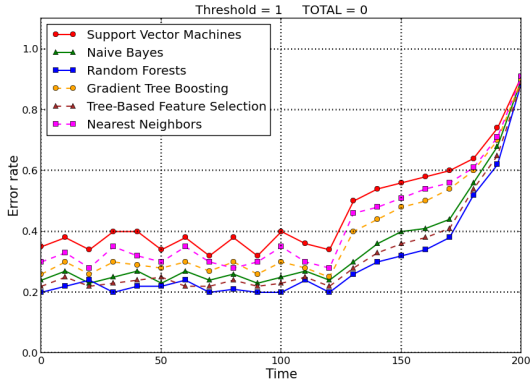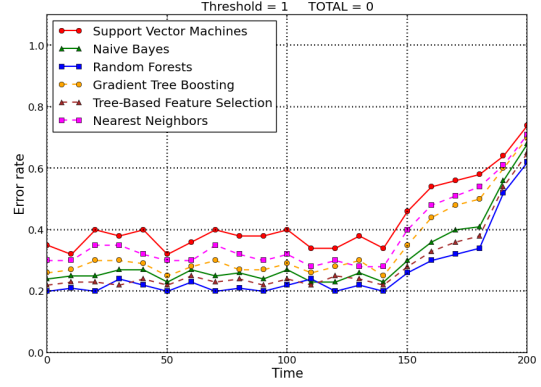
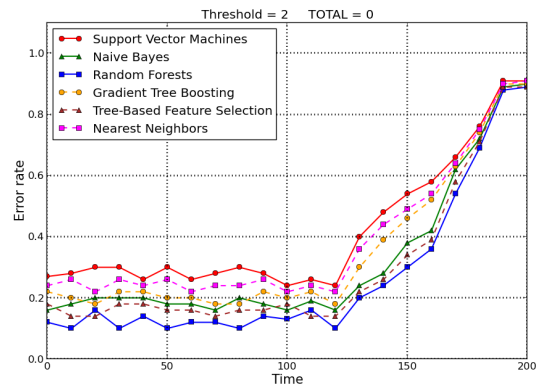(a) With 300 sensors   (b) With 600 sensors

Figure 1: Diagnostics error rate over time with the simple domination algorithm

</div>

<div align="center">

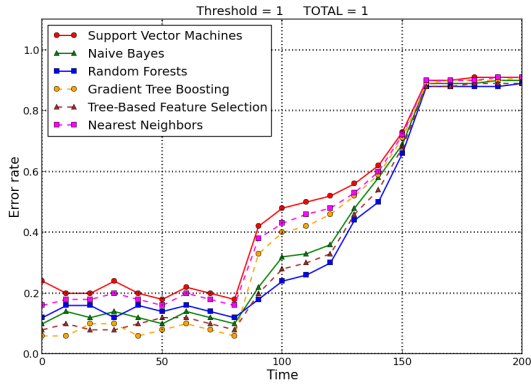(a) With 300 sensors   (b) With 600 sensors

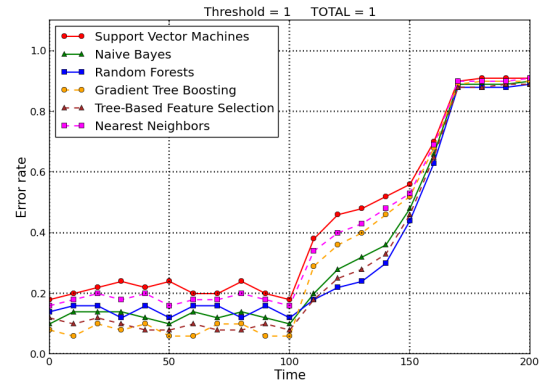Figure 2: Diagnostics error rate over time with the double domination algorithm

</div>

Figure 2a that each algorithm has a specific error interval during the time-steps $t = 1..100$, which is a bit better than in the previous simple domination case. However, the increasing of this rate due to empty batteries starts earlier than in Figure 1, and the networks shut down 10% before than in the simple domination case. A similar observation stands for, *mutatis mutandis*, the network with 600 sensor nodes, as shown in Figure 2b.

Considering the total domination mode, Figure 3 reveals that the changes are in the same direction than for the double domination rate, but they are more pronounced. For instance, in Figure 3a, the error rate is approximately equal to 10% during the half part of the simulation time-step, while it was in average equal to 30% and 20%, for simple and double domination respectively. However, this good performance on error rates impacts the network lifetime, and nodes start to die faster after the time-step $t = 80$ (120 and 100 time-steps for simple and double domination resp.). The time where nodes start to die can however be shifted when the density of the network is increased.

In all simulations, we can observe that Support Vector Machines and Nearest Neighbors have the worst behavior over time. Random Forests have the lowest error rate in general, except in the total domination case, where its error rate is not improved when compared with the double domination mode. The best results have been obtained in general using the Gradient Tree Boosting, no matter the number of nodes. However, this good performance has a cost, and the network is a bit more impacted than RF, TBFS, or NB, by the death of nodes.
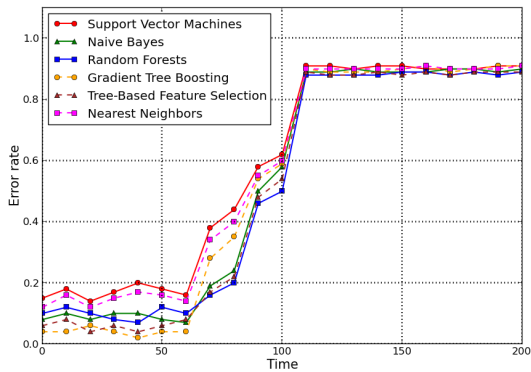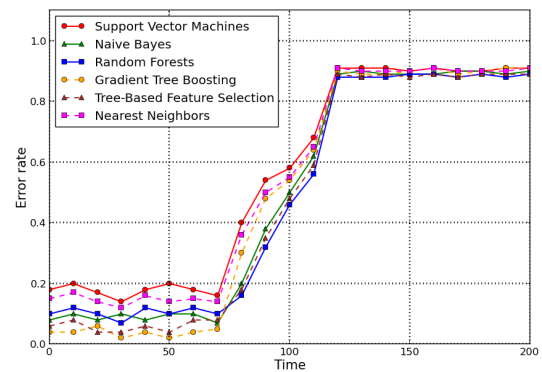
(a) With 300 sensors.

(b) With 600 sensors.

Figure 3: Diagnostics error rate over time with the total domination algorithm.

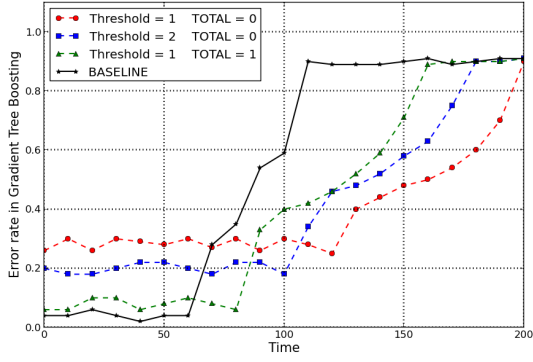(a) With 300 sensors.

(b) With 600 sensors.

Figure 4: Diagnostics error rate over time using *BaseLine* on a WSN.

## 4.2 Comparison with the optimistic *BaseLine* solution

The *BaseLine* considered in the experimental study is defined as the fault free protocol assuming that all sensors in WSN are active at the same time (coverage and energy consumption are maximal). In other words, the *BaseLine* is the case when the coverage and energy consumption are maximal.

Obtained results are shown in Figure 4. As expected, no matter the considered machine learning algorithm: diagnostic error rates start to increase faster after time-steps $t = 60$ or $t = 70$, depending on the number of sensors, if we compare it with our proposal. Furthermore, the increase is faster, the whole network becoming inactive at time-steps $t = 110$ and $t = 120$ respectively. This is a consequence of the large number of active sensors in the network. Focusing on the error rate value, and during $t = 0..60$ and $t = 0..70$ (where the whole network is active with 300 and 600 sensor respectively), the *BaseLine* performs slightly as well as our proposal, namely in the total domination mode. But we achieve better performances by increasing the lifetime service of the network up to 33%. This allows sensor networks to gracefully degrade in performance instead of failing unpredictably.

We now compare our proposed algorithm with the optimistic *BaseLine* case, in order to evaluate its efficiency in WSN at both the coverage level and the network lifetime, and therefore on diagnostics. We use the Gradient Tree Boosting for the sake of readability and its performance compared to other machine learning algorithms as shown above. The best diagnostic accuracies are obtained with the *BaseLine* and the total domination as pointed in Figure 5. The first one is slightly better than our proposal during $t = 0..60$. However, the slight improvement of this rate has a cost in terms of network lifetime, if we compare again with the total domination: our proposal shuts down almost
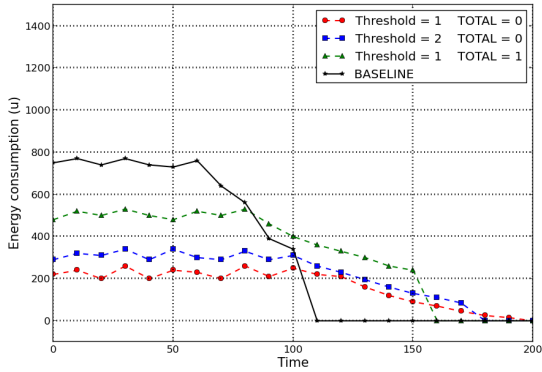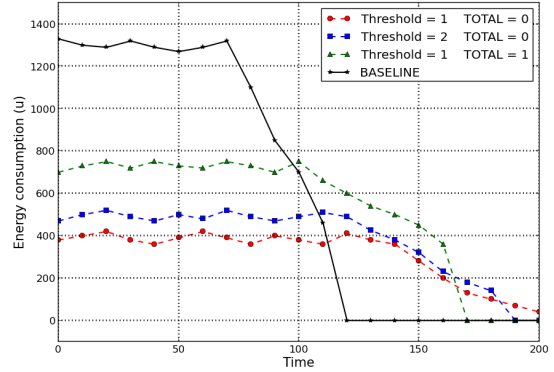
(a) With 300 sensors.
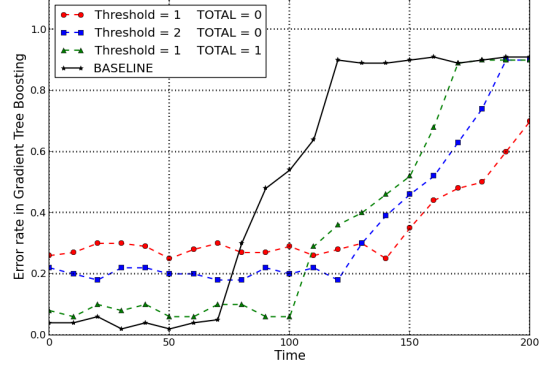
(b) With 600 sensors.

Figure 5: Error rate using the Gradient Tree Boosting with our three proposals and with *BaseLine*.
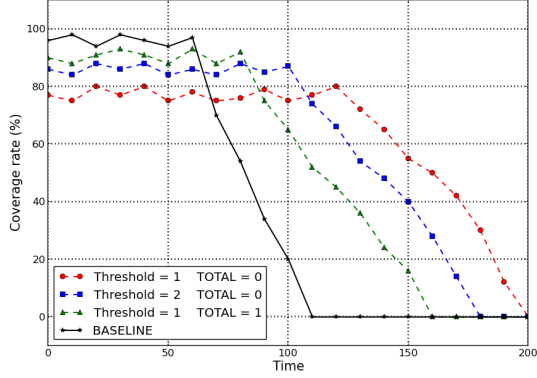

(a) With 300 sensors.

(b) With 600 sensors.

Figure 6: Energy consumption of sensors in battery unit u, for our three proposals and with *BaseLine*.

50% later than the *BaseLine*. Generally speaking, the WSN shuts down faster when using the *BaseLine*, for instance before the performance decrease for the single domination mode. This latter succeeds the most to prolong the lifetime of the WSN. The double domination, for its part, achieves a good compromise performance between network lifetime and diagnostic accuracy. This is due to the reasonable minimal number of necessary active sensors in the network at the same time.

Similar results have been obtained for both energy consumption and coverage, as respectively depicted in Figures 6 and 7, which are not surprising. The slight improvement of diagnostic accuracy in the *BaseLine* case is paid a high price in terms of energy consumption: this latter being larger in the *BaseLine*, each sensor empties its battery more rapidly, and therefore reducing the network lifetime. This reduction has a direct impact on the coverage, as can be verified in Figure 7. Another remark to point out is the decreasing speed of the energy, which is very large in the *BaseLine* case, no matter the number of sensors. In comparison, this decrease is more regular and less pronounced for our proposal, as can be seen in Figure 6. Indeed, Figure 6 explains the results obtained in Figure 7. The network having the highest energy consumption has a larger coverage rate, and this coverage rate varies over time according to the network state.

Comparisons between our proposed algorithm with its three modes on the one hand, and the *BaseLine* on the other hand, illustrate the good performance of our proposal, as we are able to reduce the energy consumption while preserving at the same time a good coverage rate. This has a direct impact on diagnostics accuracy. To put it in a nutshell, as both the network lifetime and
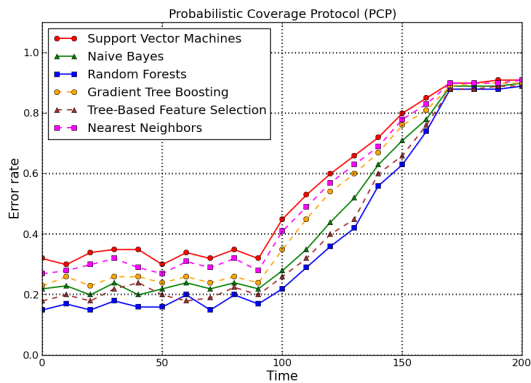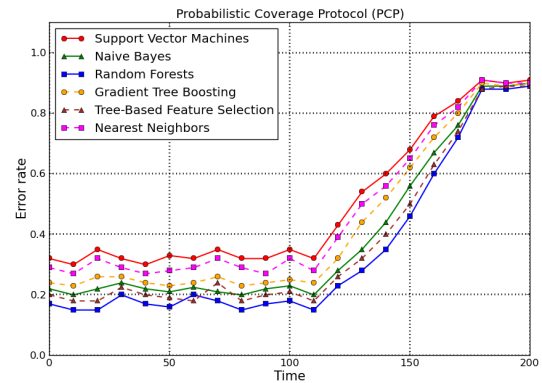
(a) With 300 sensors.
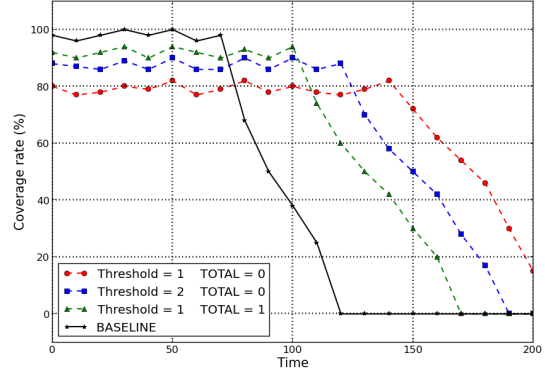
(b) With 600 sensors.

Figure 7: Coverage rate of sensors (%) with time in WSN with the three cases in the proposed algorithm and with *BaseLine*.

(a) With 300 sensors.

(b) With 600 sensors.

Figure 8: Diagnostic error rate with Probabilistic Coverage Protocol on a WSN.

coverage are improved, a low diagnostic error rate has been obtained for a longer duration.

4.3 Comparison with the PCP algorithm

Let us now compare the proposal with the Probabilistic Coverage Protocol [13] presented in Section 2.2, in terms of coverage rate, energy consumption, and then on diagnostics. The same simulation protocol used in the previous section has been respected.

Figure 8 presents the error rate variation when considering the PCP with the six machine learning algorithms studied in this article. Again, two stages can be emphasized: a normal running stage of the WSN that leads to an acceptable error rate for the diagnosis, and a second stage where nodes begin to empty their batteries. Like in our proposal and for the *BaseLine* case, we obtain a good performance for the random forests and the worst case for the support vector machine. The gradient tree boosting, for its part, achieves a median performance compared to the five other machine learning algorithms.

During the first half part of the experiments, the diagnostic using data provided by the network that embeds the PCP algorithm has presented an error rate in average equal to 22%, no matter the machine learning algorithm and the network density. This error rate is better than our proposal in the single domination mode, but it is a little less than the double domination one. This is explained by the fact that the increase of the error rates starts earlier with the PCP, and the network is totally quiet earlier too. Furthermore, both the *BaseLine* and the total domination outperform largely the
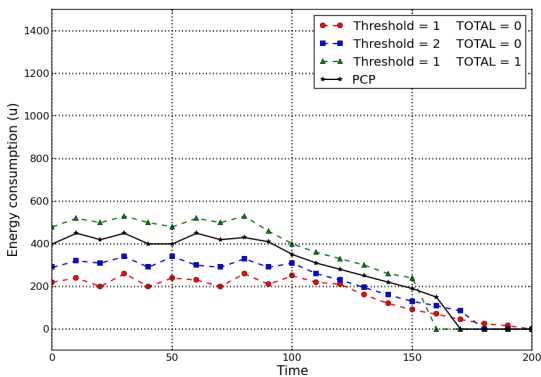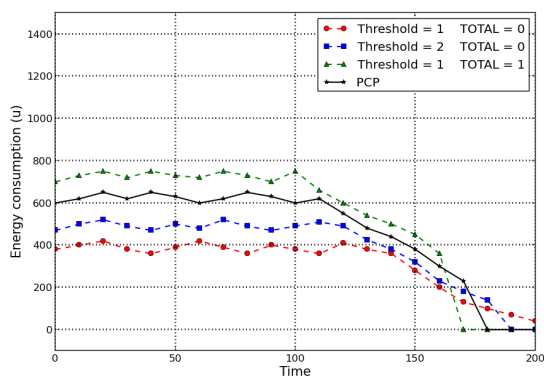
11
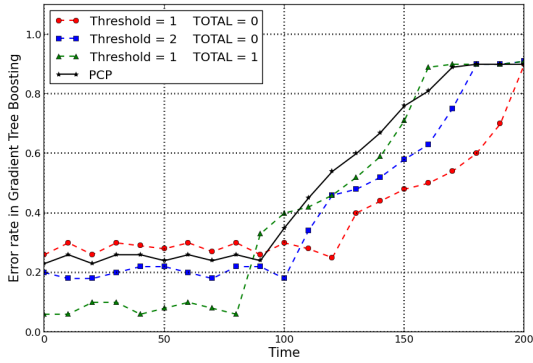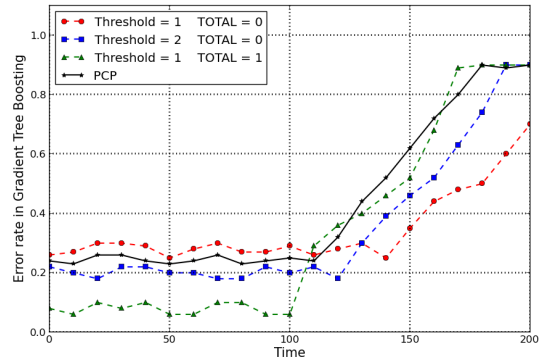
(a) With 300 sensors.

(b) With 600 sensors.

Figure 9: Error rate using the Gradient Tree Boosting, with the PCP and our proposal.

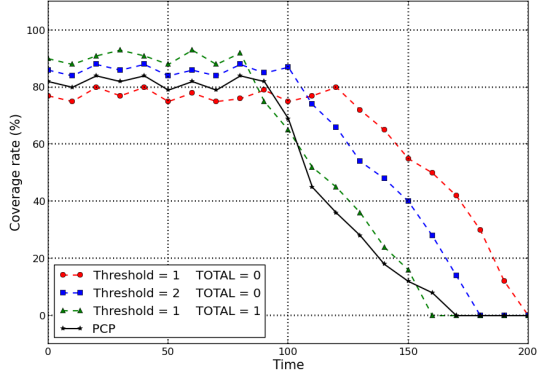(a) With 300 sensors.

(b) With 600 sensors.

Figure 10: Energy consumption of sensors (in battery unit) between PCP and our proposal.

PCP algorithm, if we focus only on the error rate during the normal running stage of the network. The duration of this normal running stage is finally almost the same between PCP and our total domination mode, while it is lower for the *BaseLine* (and larger for both the single and the double domination modes). To sum up, the probabilistic coverage protocol is poorer in average than our proposal, according to this first set of experiments that considers the six aforementioned machine learning algorithms.

Like in the previous section, we decided to compare more deeply the behaviors of the PCP and of our proposal, in terms of error rates, energy consumption, and coverage rate. The gradient tree boosting machine learning has been selected again, for the sake of its performance. The evolution of error rates over time are provided in Figure 9, in which we can see that the PCP performance is not as good as our proposal. Indeed, if we except the single domination, error rates in diagnostics are larger for the PCP. Furthermore, the nodes obviously emptied their batteries earlier in PCP than for our proposal, no matter the density of the network.

The energy consumption of sensors when the probabilistic coverage protocol is deployed in the WSN is really better than when the *BaseLine* is considered, see Figure 10. The PCP algorithm has in general a similar energy consumption compared to our algorithm, being between the total domination mode (the most consuming one) and the two other modes. This is the case too for the coverage criterion, in which the two algorithms have a similar behavior in the normal working stage, while our proposal tends to preserve more the coverage over time, when the effect of battery energy depletion can be emphasized. This is indeed illustrated in Figure 11b.

We can conclude this study by the good performance of our proposal when compared with
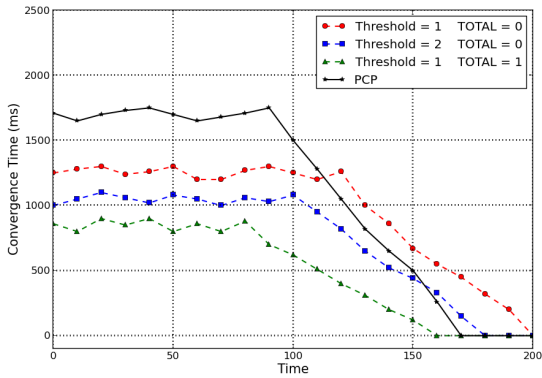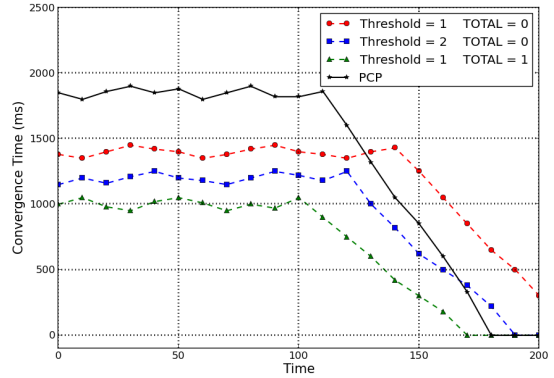
(a) With 300 sensors.
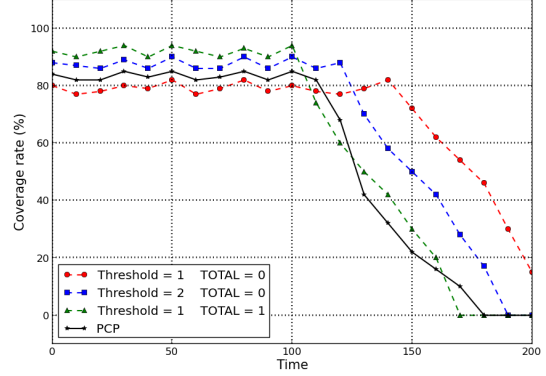
(b) With 600 sensors.

Figure 11: Coverage rate of sensors between our proposal and the PCP.

(a) With 300 sensors.

(b) With 600 sensors.

Figure 12: Convergence time in the three cases in the proposed algorithm and with PCP.

the state-of-the-art PCP algorithm, as the simple domination is able to greatly reduce the energy consumption of the network, while the total domination has a larger coverage rate if compared with the PCP. The double domination, for its part, achieves a better compromise between these two criteria. Thus, data captured from the network will be more reliable for a longer time, leading therefore to a more accurate diagnostics.

Let us finally perform a last numerical simulation, namely on global convergence time. This latter is the time needed by the algorithm to determine active/sleep nodes, and it is desired to be shortened [13]. In order to optimize the convergence speed, sensor nodes should switch to *sleep* or *active* state as quickly as possible, leading to significant energy savings. Indeed, as the number of sensors in the wait state decreases, the convergence time will be faster. Therefore, the algorithm has to avoid this state as possible. To study this metric is then very important to show the efficiency of the proposed algorithm. This latter has been compared with PCP, and obtained results are shown in Figure 12. We can see that this convergence time is in general larger for the PCP than for our proposal. This is explained by the low time complexity of the proposed algorithm to compute the minimal dominating sensor-targets as proven in Section 3.2.

5 Conclusion

In this research work, we proposed and evaluated a fully distributed algorithm for reliable diagnostics of the monitored targets under energy and coverage constraints. The proposed algorithm is a

13

hybridization of three variants of domination in graph theory, namely the simple, double, and total ones, in a single algorithm with a distributed computation. We theoretically studied and numerically simulated it, to show for each variant its impact on the effectiveness of the WSN over time, and on the diagnostics provided by six machine learning methods. Our proposal has been compared with both the optimistic solution (*BaseLine*) and its direct competitor (the Probabilistic Coverage Protocol PCP). Obtained results have shown that the proposed algorithm exhibits better performances in terms of energy consumption, coverage, and diagnostics error rate.

To summarize and taken as a whole, it can be concluded that the findings of our research highlight that when dealing with WSN monitoring, the achieved tarde-off between target coverage and energy saving related issues play an important role on the quality of the sensor readings and therefore on the resulting diagnostics of the monitored targets.

References

[1] Sanay Abdollahzadeh and Nima Jafari Navimipour. Deployment strategies in the wireless sensor network: A comprehensive review. *Computer Communications*, 91-92:1–16, 2016.

[2] Xiaole Bai, Santosh Kumar, Dong Xuan, Ziqiu Yun, and Ten-Hwang Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *MobiHoc*, pages 131–142, 2006.

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

[4] Peter Bühlmann and Torsten Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, pages 477–505, 2007.

[5] Jean Carle, Antoine Gallais, and David Simplot-Ryl. Preserving area coverage in wireless sensor networks by using surface coverage relay dominating sets. In *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, pages 347–352. IEEE, 2005.

[6] Jean Carle and David Simplot-Ryl. Energy-efficient area monitoring for sensor networks. *Computer*, 37(2):40–46, 2004.

[7] Wiem Elghazel, Jacques Bahi, Ahmad Farhat, Christophe Guyeux, Mourad Hakem, Kamal Medjaher, and Noureddine Zerhouni. Random forests for industrial device functioning diagnostics using wireless sensor networks. In *IEEE AEROSPACE CONFERENCE, 2015.*, pages 1–9, Big Sky, Montana, USA, mar 2015.

[8] D Galar, U Kumar, J Lee, and W Zhao. Remaining useful life estimation using time trajectory tracking and support vector machines. In *Journal of Physics: Conference Series*, volume 364, page 012063. IOP Publishing, 2012.

[9] T.W. Haynes, S. Hedetniemi, and P. Slater. *Fundamentals of Domination in Graphs*. Chapman & Hall/CRC Pure and Applied Mathematics. Taylor & Francis, 1998.

[10] Shibo He, Jiming Chen, Xu Li, Xuemin (Sherman) Shen, and Youxian Sun. Leveraging prediction to improve the coverage of wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 23(4):701–712, 2012.

[11] Shibo He, Jiming Chen, David K. Y. Yau, Huanyu Shao, and Youxian Sun. Energy-efficient capture of stochastic events under periodic network coverage and coordinated sleep. *IEEE Trans. Parallel Distrib. Syst.*, 23(6):1090–1102, 2012.

[12] TC He, WM Cao, and WX Xie. Coverage analyses of plane target in sensor networks based on clifford algebra. *Acta Electron. Sin*, 37(8):1681–1685, 2009.

[13] Mohamed Hefeeda and Hossein Ahmadi. Energy-efficient protocol for deterministic and probabilistic coverage in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(5):579–593, 2010.

[14] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10(4):519–528, 2005.

[15] Chi-Fu Huang, Yu-Chee Tseng, and Li-Chu Lo. The coverage problem in three-dimensional wireless sensor networks. In *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, volume 5, pages 3182–3186. IEEE, 2004.

[16] Yiannis Kantaros and Michael M Zavlanos. Distributed communication-aware coverage control by mobile sensor networks. *Automatica*, 63:209–220, 2016.

[17] Koushik Kar and Suman Banerjee. Node placement for connected coverage in sensor networks. In *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 2–pages, 2003.

[18] Gaurav S. Kasbekar, Yigal Bejerano, and Saswati Sarkar. Lifetime and coverage guarantees through distributed coordinate-free sensor activation. *IEEE/ACM Trans. Netw.*, 19(2):470–483, 2011.

[19] Chuanfeng Li, Zeyu Sun, Huihui Wang, and Houbing Song. A novel energy-efficient k-coverage algorithm based on probability driven mechanism of wireless sensor networks. *IJDSN*, 12(4), 2016.

[20] Feng Li, Jun Luo, Wenping Wang, and Ying He. Autonomous deployment for load balancing k-surface coverage in sensor networks. *IEEE Trans. Wireless Communications*, 14(1):279–293, 2015.

[21] Lei Li, Baoxian Zhang, and Jun Zheng. A study on one-dimensional k-coverage problem in wireless sensor networks. *Wireless Communications and Mobile Computing*, 13(1):1–11, 2013.

[22] Junbin Liang, Ming Liu, and Xiaoyan Kui. A survey of coverage problems in wireless sensor networks. *Sensors & Transducers (1726-5479)*, 163(1), 2014.

[23] Ronald E McRoberts. Estimating forest attribute parameters for small areas using nearest neighbors techniques. *Forest Ecology and Management*, 272:3–12, 2012.

[24] Liu Ming, Cao Jian-Nong, Zheng Yuan, Chen Li-Jun, and Xie Li. Analysis for multi-coverage problem in wireless sensor networks. In *Journal of Software*. Citeseer, 2007.

[25] Selina SY Ng, Yinjiao Xing, and Kwok L Tsui. A naive bayes model for robust remaining useful life prediction of lithium-ion battery. *Applied Energy*, 118:114–123, 2014.

[26] Sasha Slijepcevic and Miodrag Potkonjak. Power efficient organization of wireless sensor networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 2, pages 472–476. IEEE, 2001.

[27] V Sugumaran, V Muralidharan, and KI Ramachandran. Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. *Mechanical systems and signal processing*, 21(2):930–942, 2007.

[28] Zeyu Sun, Longxing Li, Xuelun Li, Xiaofei Xing, and Yuanbo Li. Optimization coverage conserving protocol with authentication in wireless sensor networks. *IJDSN*, 13(3), 2017.

[29] Di Tian and Nicolas D Georganas. A node scheduling scheme for energy conservation in large wireless sensor networks. *Wireless Communications and Mobile Computing*, 3(2):271–290, 2003.

[30] Jie Tian, Xiaoyuan Liang, and Guiling Wang. Deployment and reallocation in mobile survivability-heterogeneous wireless sensor networks for barrier coverage. *Ad Hoc Networks*, 36:321–331, 2016.

[31] Javad Akbari Torkestani. An adaptive energy-efficient area coverage algorithm for wireless sensor networks. *Ad hoc networks*, 11(6):1655–1666, 2013.

[32] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 28–39. ACM, 2003.

[33] Hafsa Zannat, Taslima Akter, Mashrura Tasnim, and Ashikur Rahman. The coverage problem in visual sensor networks: A target oriented approach. *J. Network and Computer Applications*, 75:1–15, 2016.

[34] Harry Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004.

[35] Chuan Zhu, Chunlin Zheng, Lei Shu, and Guangjie Han. A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, 35(2):619–632, 2012.

[36] Dimitrios Zorbas and Tahiry Razafindralambo. Prolonging network lifetime under probabilistic target coverage in wireless mobile sensor networks. *Computer Communications*, 36(9):1039–1053, 2013.