# Using Analytics to Predict Data Dissemination Policy Under Energy and Coverage Constraints

Loïc Guégan
loic.guegan@uit.no
UiT The Arctic University of Norway
Tromsø, Norway

Issam Raïs
UiT The Arctic University of Norway
Tromsø, Norway
issam.rais@uit.no

Otto Anshus
UiT The Arctic University of Norway
Tromsø, Norway

## Abstract

Disseminating data in a wireless distributed system under limited resources, where nodes have constrained energy and communications capabilities, is challenging. In such contexts, common data dissemination technics cannot be used. Instead, simple device-to-device communication policies allows to mitigate the impact of communications on nodes energy consumption. However, depending on nodes configuration (up-times duration, wireless technology capabilities and energy consumption), choosing a suitable communication policy is challenging.

In this paper, we propose and study two approaches based on classification algorithms that aim at predicting the most suitable communication policy to use, for a given node configuration, to match a given coverage and energy consumption target. The first approach called *in situ* learning, trains the classification models during deployment. The second approach called *offline* learning, uses existing data that are collected from previous deployments or simulations. Results show that, for a resource constrained environment, common classification models can take several months to converge with *in situ* learning. Depending on the policy, *in situ* learning can have a significant energy consumption overhead. Results underline *offline* learning as an interesting alternative to *in situ* learning, but requires to collect data from previous deployments or simulations.

## CCS Concepts

• **Computing methodologies** → **Simulation evaluation**; **Classification and regression trees**.

## Keywords

energy consumption, data dissemination, energy constraint, distributed systems, CPS, IoT, WSN

**ACM Reference Format:**
Loïc Guégan, Issam Raïs, and Otto Anshus. 2018. Using Analytics to Predict Data Dissemination Policy Under Energy and Coverage Constraints. In *Proceedings of International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX


*PE-WASUN, Barcelona, Spain*

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

Environmental monitoring based on distributed systems is a crucial need for various applications. It allows to record the evolution of several phenomenons such as $CO_2$ concentration, water quality or earthquake detection over large areas [1]. These distributed systems must perform various tasks such as sensing, processing and disseminating data. They can be built with technologies from the Internet of Things (IoT), Wireless Sensor Networks (WSN) and more generally Cyber-Physical System (CPS). Depending on the context they are deployed in, performing environmental monitoring can be challenging.

The Arctic Tundra (AT) is a particularly harsh environment to monitor, with large isolated areas, where (i) nodes are expected to operate for several months, under a very limited energy budget; (ii) mobile networks provide little to no coverage on the monitored area, forcing nodes to rely on their own wireless technologies; (iii) nodes are not consistently reachable because of harsh weather conditions (heavy snow, rain, humidity etc.). Consequently, monitoring and disseminating data in this context is difficult to achieve.

To tackle this problem, different loosely-coupled communication policies are proposed in [2]. This related work studies four communication policies that can be used in the AT context. Two metrics are considered: 1) the energy consumption 2) the coverage (representing the number of nodes that received the data). This related work highlights that, in a given context, a policy can be better than another. Also, depending on the use case, full coverage is not always required, especially in scenarios with energy consumption constraints. A trade off between coverage and energy consumption must be found. According to the node configuration (up-times duration, wireless technology capabilities and energy consumption), this trade-off can change.

In this paper, we study the performance of two supervised learning classification algorithms, when predicting the policy to use according to (i) node configuration, (ii) the targeted coverage and (iii) the energy consumption budget. To train these models, two learning approaches are proposed. The first one, called *in situ* learning, consists in training the models during deployment. The second, called *offline*, consists in using the data collected from simulations or previous deployments. The contributions of this paper are:

- A study of the usage of classification models to predict the appropriate data dissemination policy, under energy consumption and coverage constraints, for systems deployed in scarce resource environments, like [2, 3].
- A study of *in situ* and *offline* training for the chosen analytical models, for resource constrained environments.

This paper is organized as follow. Section 2 presents the state of the art. Section 3 details the analytical process followed along with

the two approaches used to train the models. Then, Section 4 and 5 present the results of both approaches. Finally, Section 6 concludes this work.

## 2 State of The Art

Disseminating data in a distributed system that monitors the environment is crucial to back up data, coordinate nodes or provide feedback to the users. However, it is challenging when it is constrained in energy and communication capabilities.

Several works provide structure-based solutions to save energy during communications using Q-Learning algorithms [4, 5]. Similarly, clustering-based solutions are proposed to tackle these issues [6–8]. However, the proposed algorithms are not loosely-coupled, as they require additional communications to operate and coordinate nodes. Consequently, they cannot be used in scenarios like the AT with low nodes availability, communication bandwidth and energy resources.

To mitigate the usage of communications, Machine Learning (ML) based node reconfiguration can be used to learn from nodes communication history and existing data. In [9], authors use reinforcement learning algorithms to schedule Time Slotted Channel Hopping (TSCH) 802.15.4 radio communications. Similarly, the authors in [10] use random forest classifiers to determine the configuration parameters to use for 802.15.4, in a dynamic IoT network. However, these contributions are specific to the 802.15.4 and cannot be used with various wireless technologies.

In [11], authors propose a global approach based on Q-Learning. The model self-adjusts the duty-cycle of wireless nodes to reduce their energy consumption. Despite being a global approach to the problem, the work focuses on reducing the nodes energy consumption by acting on their orchestration rather than focusing on the algorithms to disseminate data.

In [12], authors propose a Q-Learning solution that can work with any wireless technology. It maximizes the nodes operation time by adjusting the duty-cycle periods, while taking into account energy harvesting. Similarly, authors in [13] propose an energy manager for wireless sensor networks, that takes into account energy harvesting. Despite being wireless technology agnostic, these solutions target the energy management of the nodes without taking into account the dissemination of data nor its cost in energy.

To the best of our knowledge, literature does not provide a solution to determine the network communication policy to use among several ones, when it can significantly impact the coverage and nodes energy consumption. In such a context, nodes configuration must be taken into account as it impacts both coverage and energy consumption. In this work, we propose to study how classification algorithms can provide a solution, for nodes in a constrained environment like the AT.

## 3 Analytics for Data dissemination policies

This section presents an overview of a distributed observatory in an environment with scarce resources (the Arctic Tundra) its data dissemination policies and the metrics used to evaluate our proposed classification approaches.

### 3.1 Data dissemination policies under constrained environment

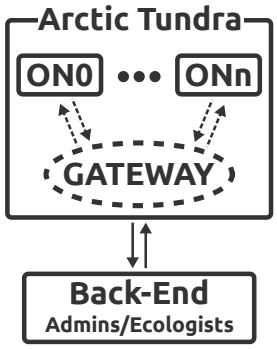



**Figure 1: Overview of the observation system architecture. The "Back-end" hosts a set of services, like [2]. Its connectivity to Observation Nodes (ON) deployed in the Arctic Tundra uses the gateway (when rarely available). This wireless gateway is used for 1:1 communications between ONs, forming a star topology.**

In related work [3], a distributed monitoring infrastructure for constrained scenarios is proposed. The architecture for network of nodes is depicted Figure 1. The Observation Nodes (ON) are in charge of monitoring the environment and communicate with other ON and the back-end, though a gateway. The ON communicates using one of the available policies: *Baseline*, *Hint*, *Extended* or *Hint+Extended* (c.f Figure 2):

***Baseline*** – Nodes wake up at a random time, each hour, for a duration called *up-time*. When an overlap between the sender and the receiver up-time happens, the sender starts transmitting data. If one of the nodes up-time ends, ongoing communications are aborted and the node turns off.

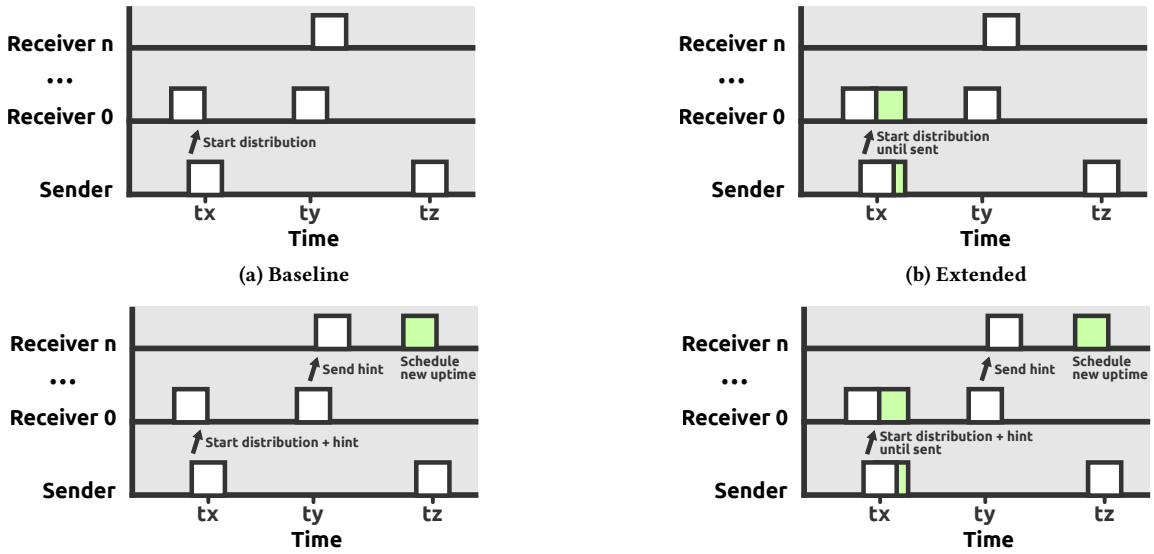
***Extended*** – Compared to *Baseline*, the *Extended* policy does not abort ongoing communications and nodes keep communicating until data is transmitted. It implies that up-time duration of nodes can be extended.

***Hint*** – The *Hint* policy is based on *Baseline*. The sender performs additional communications to send a timestamp to receivers. It informs the receiver about the sender's next up-time, to increase the likelihood of overlap between them. This timestamp can be gossiped between receivers.

***Hint+Extended*** – This policy combines the principles of the *Extended* and *Hint* policies, with the aim of combining their effects.

In [3], simulations were performed to study the impact of the policies on coverage and nodes energy consumption.

Since the arctic tundra is a particularly difficult and wide environment, it requires long range communication capabilities. Low Power Wide Area Network (LPWAN) wireless technologies allow to reduce the energy consumed during wireless communications [18]. To extend lifetime, nodes have short and sparse daily up-times. To match reality, the simulations from [3] use parameters presented Table 1. Each run simulates 24 hours of deployment for 13 nodes that operate and wake-up randomly each hour for an up-time duration




(a) Baseline (b) Extended

(c) Hints (d) Hint and extended

**Figure 2: Example of communication scenarios for each policy. Messages, up-times and added up-times are represented as arrows, gray and green rectangles, respectively.**

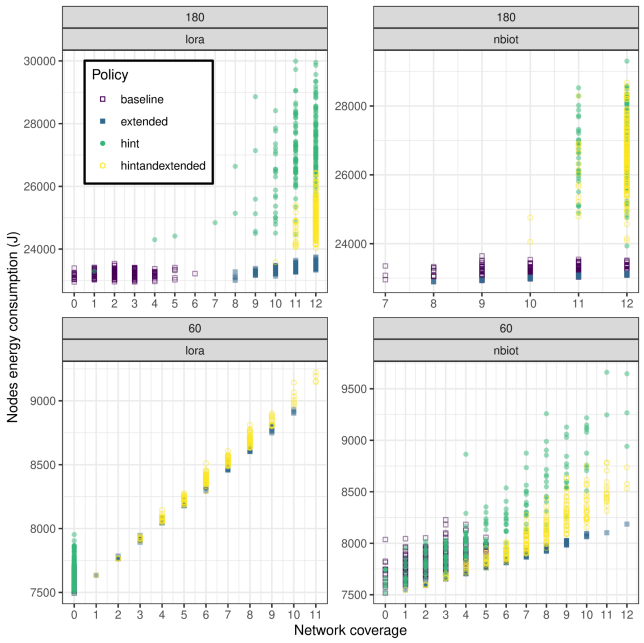



**Figure 3: Results from [3] showing the overall energy consumption of the nodes according to the coverage achieve by the nodes communication policy.**

of 1min or 3min. During each simulation, a sender node attempts to communicate with the 12 receivers to forward data. Nodes can either use LoRa or NbIoT as wireless technology, depending on the simulated scenario.

**Table 1: Simulation Parameters**

| Parameters | | Value | Citations |
|---|---|---|---|
| **Bandwidth (Ltnc)** | LoRa | 50kbps (0s) | [14, 15] |
| | NbIoT | 200kbps (0s) | [14] |
| **Energy states** | $P_{idle}$ | 0.4W | [16] |
| | LoRa | 0.16W or 32mA at 5V | [17] |
| | NbIoT | 0.65W or 130mA at 5V | [17] |
| **Up-time** | Long | 3 min/hour | |
| | Short | 1 min/hour | |
| **Data size** | | 1MB | |
| **# Receivers** | | 12 | |

Related work [3] shows that each policy has a different impact on the coverage and the energy consumption of nodes. Figure 3 summarises the results for each node configuration. This figure shows the energy consumption and coverage for each combination of policy, wireless technology and up-time duration. It shows a wide trade-off between different available parameters.

When the scenario requires to meet a certain energy consumption budget for a given coverage, a policy could answer one constraint while violating the other. To solve this problem, a model predicting the appropriate policy to use for a given coverage and energy budget must be introduced. This work proposes to study the feasibility of such predictions, using classification models.

## 3.2 Models and Approaches

Machine learning classification models allow to predict the class of an new instances based on their features. In our case, we want to predict the policy to use (class) based on the nodes configuration

(features), for a given coverage and energy budget (features). Thus, supervised learning classification algorithms is a perfect fit.

This work focuses on two commonly used supervised learning classification algorithms: K-Nearest Neighbours (KNN) and Classification and Regression Tree (CART). Two different learning methods are investigated in this work: *in situ* and *offline*. The aim of *in situ* learning is to perform the training after deployment. With this approach, no prior experiments are required to collect data and train the models. However, it requires to have a learning period for the models to converge. The *offline* learning approach uses data collected from simulations or previous deployments to make predictions. No learning period are required during the deployment.

The presented experiments use the *R* classification packages *class* (KNN), *rpart* (CART) and *MLMetrics* (classification performance metrics). The source code is available online [1] and the reproducibility is improved with the *Renv* package.

### 3.3 Performance Metrics

To measure how precise and accurate the predictions of the models are, classification performance metrics must be used. The first one measures the overall model accuracy and is defined as follow:

$$\text{OAcc} = \frac{\sum_{c \in C} \text{TP}_c}{\sum_{c \in C} \text{TP}_c + \text{TN}_c + \text{FP}_c + \text{FN}_c} \tag{1}$$

Where $C = \{Baseline, Hint, Extended, Hint+Extended\}$ is the set of predicted classes with $c \in C$ and $\text{TP}_c$, $\text{TN}_c$, $\text{FP}_c$, $\text{FN}_c$ the number of True Positive, True Negative, False Positive and False Negative, respectively. This metric measures the total percentage of correctly predicted classes, for each model.

Next, three common metrics for performance measurements of classification models are used [19]. First, the recall, for each class $c$:

$$\text{recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{2}$$

The recall measures the percentage of instances, from class $c$, that are correctly classified. Second, to account for False-Positive predictions, the precision is defined as:

$$\text{precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \tag{3}$$

Third, to synthesize both metrics ($\text{recall}_c$ and $\text{precision}_c$), the F1-Score is defined as the harmonic mean:

$$\text{F1}_c = 2 \times \frac{recall_c \times precision_c}{recall_c + precision_c} \tag{4}$$

Since F1-Score accounts for both $\text{recall}_c$ and $\text{precision}_c$, it provides enough information to state about the model performance. Thus, only $\text{F1}_c$ are reported in the results.

In our use-case, two other metrics are used. First, the overall nodes energy consumption, defined as:

$$E_{total} = \sum_{n \in N} E_n \tag{5}$$

With $N$ representing the set of nodes in the network and $E_n$, the energy consumption of node $n$.

[1] https://gitlab.com/manzerbredes/loosely-policies-analytics

Finally, the coverage achieved by the communication policies is measured. It represents the total number of nodes that received the data transmitted by the sender.

## 4 In situ Approach Analysis

This section presents the studied scenario and analyses the results of the *in situ* training approach.

### 4.1 Scenario and Hypothesis

As explained in Section 3, the *in situ* training approach consists in training the KNN or DT models on the deployed nodes. To train the models, the sender node performs communication attempts every day and monitors the required metrics. In our case, these metrics are $E_{total}$ and the coverage achieved, each day. These metrics are used as the model inputs. We study the training period in an ideal scenario, where at least $\text{OAcc} \geq 0.8$ and a $\text{F1}_c \geq 0.8$ threshold is reached for all class $c$, the same order as good classifiers [20, 21]. The following assumptions are set :

(1) All nodes use the same set of configuration parameters from Table 1
(2) All nodes use the same policy for a given day, changed in a round-robin scheme
(3) The sender has access to the monitored metrics from other nodes, with no communication overhead

These assumptions allow us to study the KNN and DT classifiers for the *in situ* approach.

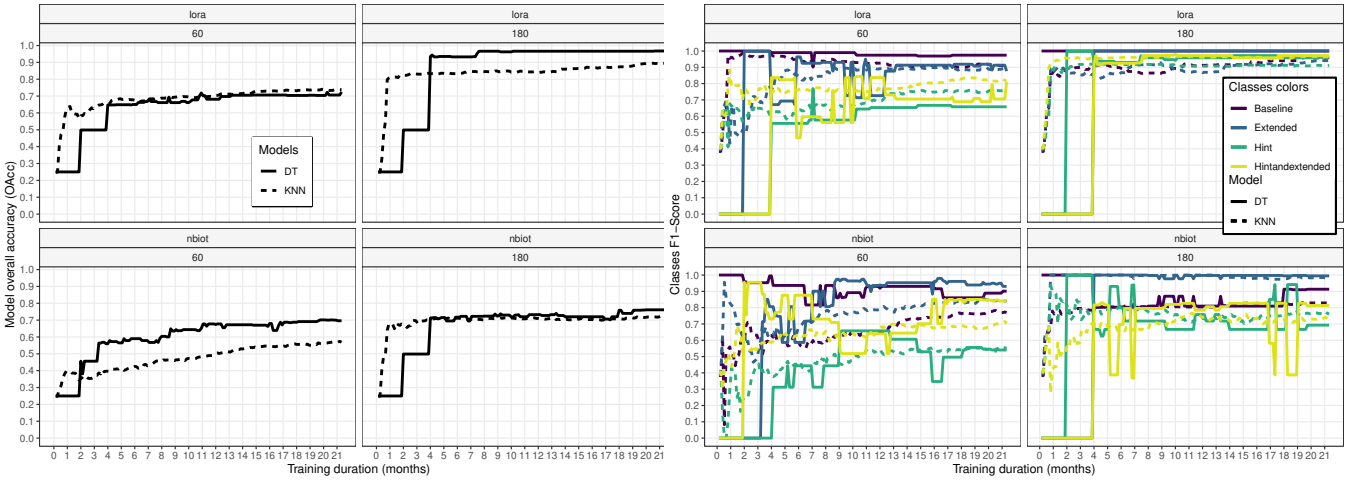
### 4.2 Analysis

Based on data from related work [3], we study how *in situ* training performed in terms of learning curve [22], $E_{total}$ and coverage compared to communication policies. The results of this analysis using KNN and DT classifiers are depicted on Figure 4.

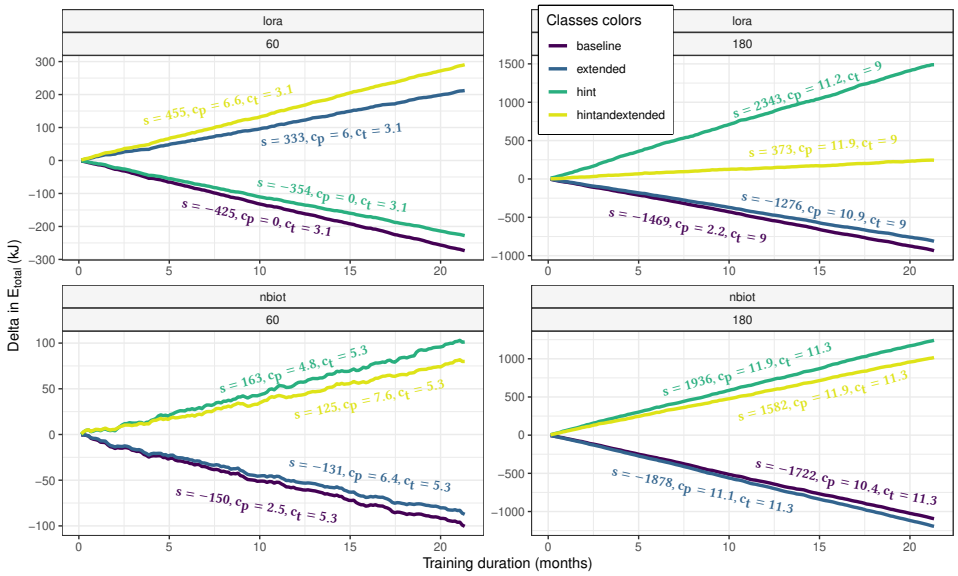
The Figure 4a shows the accuracy of a given model throughout the training duration. This figure shows that scenarios with 60*s* of up-time have a longer training period compared to scenarios with 180*s* up-time. In fact, with the DT model, it takes more than a year to reach 0.7 accuracy with 60*s* up-time where it is reached after 4 months with 180*s* up-time and the 0.8 threshold is not even reached. Similarly, the wireless technology has a significant impact on the models learning curve. Depending on the nodes configuration, the learning curves of both models are difficult to predict. For example, the KNN model reaches an accuracy of 0.5 after 13 months with NbIoT and 60*s* up-time, whereas with 180*s* up-time, it reaches more than 0.7 accuracy after four months. This range of uncertainty makes it difficult to determine the earliest time by which the models can be used on the nodes.

Figure 4b shows the evolution of the F1-Score, for each model and class. These results show no specific trend for the classes under the KNN model. However, using the DT model, *Baseline* reaches a high F1-Score (more than 0.8) in less than four months. As shown on Figure 3, it is explained by the fact that, *Baseline* is easily predicted as it is linearly separable from *Extended* and *Hint+Extended*.

Figure 4c shows the evolution of the difference in $E_{total}$ between a fixed class for all nodes and the *in situ* approach, through time. Negative values means that *in situ* training consumes more energy than using the given class and positive values means the opposite.




**(a) Accuracy** OAcc **(b) F1-Scores**




**(c) Difference in $E_{total}$ between *in situ* training and defined policies**

**Figure 4: In situ training results generated from [3], for each wireless technology and up-time duration. Figure 4c shows the difference in energy consumption between a given policy and *in situ* training. For each curve, three parameters are given: 1) $s$, the slope obtained by linear regression (in J/day) 2) $c_p$, the average coverage per day, for a given policy 3) $c_t$, the average coverage per day of *in situ* training.**

For each curve, three parameters are given: (i) $s$ represents the linear regression's slope of the curve in J/day; (ii) $c_p$ (for coverage policy) represents the average coverage achieved by the fixed policy, per day; (iii) $c_t$ (for coverage training) is the average coverage of *in situ* training, per day.

For every nodes configuration, the results highlight the importance of using the correct class if we are not performing *in situ* training. In fact, the difference in energy consumption between each class and *in situ* training increases significantly over time. With LoRa and an up-time of 180s, *Hint* consumes more energy compared to *in situ* training per day with $s = 2\,343$J/days. Over time, after 20 months, this difference reaches 1500kJ. In the same scenario, *in situ* training consumes $1\,276J$ more per day compared to *Extended* while achieving a greater coverage ($c_p = 10.9$) close to *Hint* ($c_p = 11.2$).

In most nodes configurations, there is a class that achieves a coverage at least as good as *in situ* training ($c_p >= c_t$) or close to it ($|c_p - c_t| < 1$) while consuming less energy ($s < 0$). This is not the case for LoRa with 60s up-time since, as explained by

related work [3], *Hint* and *Baseline* do not extend the nodes up-time which prevents the data from being transmitted. With this node configuration, the *Hint* and *Extended* are the two classes that consume the highest amount of energy. But, they are able to achieve a coverage greater than *in situ* training by reaching at least two more nodes, in average. All classes that consume less energy than *in situ* training, have a coverage equal to 0. Thus, performing *in situ* training in these cases would always be beneficial, when achieving coverage is crucial.

## 4.3 Discussion

The analysis of *in situ* training approach for a resource constrained environment provides interesting conclusions. It shows that, *in situ* training for the classification models used in this work, under resource constrained environments (i.e short up-time duration in a LPWAN context, even with relaxed assumption) is a challenge. The duration of the training period ranges from several months to years. But different methods can be used to leverage this duration. As seen in the results, the nodes configuration (chosen wireless technology or up-time duration) has a significant impact on the learning curves and can be used as a leverage.

Designing policies with very distinct behaviors helps in reducing the duration of the training period. Other training schemes should also be investigated such as reducing the turnover of policies (e.g going from 24h per policy to 12h). The results also show that, KNN and DT have different learning curves. Hence, studying other classification algorithms such as Random Forest or multi-class Support Vector Machine could be interesting and impact the learning period.

On the energy consumption perspective, performing *in situ* training of the model has a cost in energy consumption. For all node configurations, there is a policy that consumes less energy than *in situ* training while still maintaining a good coverage. However, using the appropriate policy instead of *in situ* training is important as the energy consumption adds up rapidly over time. But choosing the policy is not trivial as it depends on the nodes configuration and the trade-offs between the coverage and the energy consumption.

# 5 Offline Approach Analysis

This section presents an analysis of the *offline* approach, which consists in predicting the communication policy to use (using KNN and DT), prior to perform real nodes deployment.

## 5.1 Models Performance

**Table 2: F1-Score and accuracy of KNN and DT**

| | F1-Score | | | | |
|---|---|---|---|---|---|
| **Model** | *Baseline* | *Hint* | *Extended* | *Hint+Extended* | **OAcc** |
| *KNN* | 0.83 | 0.73 | 0.90 | 0.79 | 0.69 |
| *DT* | 0.90 | 0.75 | 0.86 | 0.79 | 0.70 |

Table 2 summarize the prediction performance of the KNN and DT, for the *offline* approach. F1-Score of each class and the overall model accuracy for both models are shown. For our use case, we consider that having a F1-Score of at least 0.8 is sufficient. This

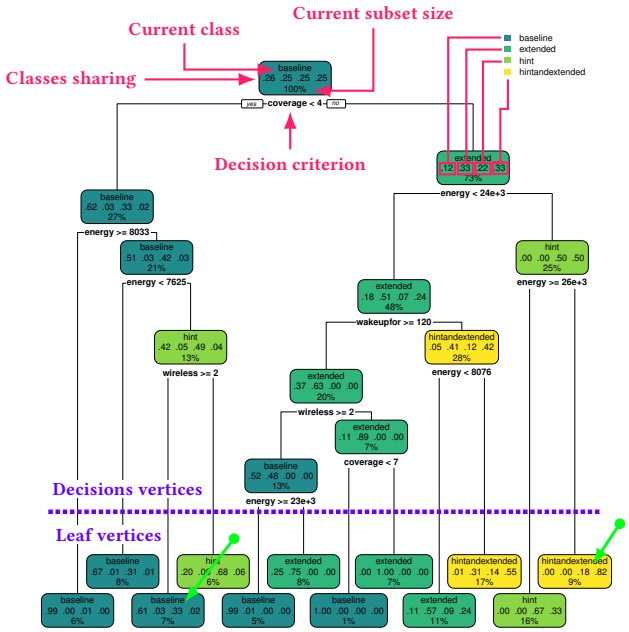



**Figure 5: Overview of the decision tree used by the DT model.**

table shows that both, KNN and DT, have similar overall accuracy (0.69 and 0.7 respectively). F1-Score for *Baseline* and *Extended* are accurate (greater than 0.8, using both models). For *Baseline*, F1-Score for DT is greater than KNN. For *Extended*, F1-Score of the KNN is greater than DT. However, both models have a lower F1-Score on *Hint*.

In fact, Figure 3 shows that both *Hint* and *Hint+Extended* provide similar coverage in the scenarios with NbIoT and 180*s* or 60*s* up-time. For the scenario using LoRa and 60*s* up-time duration, *Hint* and *Baseline* have similar results since neither of them is able to improve the coverage. For this scenario, even if, in most cases, *Hint* increases the energy consumption compared to *Baseline*, there is still a significant overlap in terms of coverage and energy consumption between both classes. These observations suggest that *Hint*, *Baseline* and *Hint+Extended* have similar behavior in many cases which contributes to reducing models' prediction accuracy.

Figure 5 depicts the complete decision tree produced by the DT model. This tree represents the set of rules followed by the model to perform its predictions. It contains two types of vertices. The decision vertices and the leaf vertices. Starting from the top of the tree, the decision vertices determine the next decision vertex based on a decision criteria. In our case these criteria are the coverage, the energy consumed by the nodes, the up-time duration and the wireless technology used. This process continues until a leaf vertex is reached. The class of this leaf vertex determined the predicted class.

This figure shows that *Baseline*, *Extended* and *Hint+Extended* have distinct distributions in the tree compared to *Hint*, that is part of both main branches of the tree. As shown by the leaf vertices, *Hint* is part of at least one of the leaf nodes that are classified as *Baseline*, *Extended* or *Hint+Extended*. Its proportion is significant

in *Baseline* (up to 0.33) and *Hint+Extended* (up to 0.18) as pointed by the green arrows of Figure 5. Consequently, to improve the prediction accuracy of both classification models, we removed *Hint* from the data set. Training models offline allows to perform this type of optimizations. Two reasons justify this approach: (i) *Hint* provides similar coverage than *Hint+Extended* in most cases, while increasing the energy consumption of nodes; (ii) In the scenario using LoRa with 60*s* as up-time duration, *Hint* is an overhead in terms of energy consumption compared to *Baseline*, while not improving the coverage.

**Table 3: F1-Score and accuracy of KNN and DT (no *Hint*)**

| | F1-Score | | | | |
|---|---|---|---|---|---|
| **Model** | *Baseline* | *Hint* | *Extended* | *Hint+Extended* | **OAcc** |
| *KNN* | 0.88 | NA | 0.89 | **0.91** | **0.81** |
| *DT* | 0.93 | NA | 0.86 | **0.92** | **0.83** |

The Table 3 presents the F1-Score and the accuracy of both models when *Hint* is removed from the data set. These results show a significant improvement in F1-Score of each class and the accuracy of the models. The lowest F1-Score is 0.86 (*Extended* class of the DT model) and the lowest accuracy is 0.81 (KNN). Given the better performance of KNN and DT using this new data set, the analysis of this section assumes that *Hint* is not taken into account.

## 5.2 Evaluation Through Simulations

**Table 4: Offline Learning Simulation Results**

| **Wireless** | **Up-time** | **Model** | $\overline{\Delta}$ $E_{total}$ **(J)** | $\overline{\Delta}$ **Network Coverage** |
|---|---|---|---|---|
| LoRa | 60s | KNN | -171.89(120) | -0.78(0.88) |
| | | DT | -207.11(123) | -1.05(0.90) |
| | 180s | KNN | -2629.47(203) | 0.11(0.44) |
| | | DT | **-2924.29(173)** | **-1.44(0.38)** |
| NbIoT | 60s | KNN | -560.44(68) | -0.53(0.38) |
| | | DT | -521.77(62) | 0.19(0.35) |
| | 180s | KNN | -1543.86(378) | **1.51(0.43)** |
| | | DT | **-1874.18(357)** | 1.36(0.41) |

As presented in Section 3, the data used to train and test our KNN and DT models are simulation results extracted from related work [3]. To evaluate both models with the *offline* training approach in a similar context, the simulator proposed in [3] is reused. The simulation's parameters are identical to [3] and reported in Table 1.

For each wireless technology tuple (LoRa or NbIoT) and up-time duration (60*s* and 180*s*), 100 random energy consumption and coverage targets are uniformly selected between their respective minimum and maximum values from the results of Figure 3. The goal is to replicate the selection of an energy consumption and a coverage target for a real node deployment. For a given target, a prediction using one model (KNN or DT) is derived and used by all nodes of the simulation. Similarly to [3], each simulation is run 200 times with the aim of evaluating each scenario with 200 random node up-time schedules.

The simulation results are aggregated in Table 4. This table presents the average difference between the simulated metric (energy consumption of the nodes and coverage) to random target values. Negative values means that the simulated metric is lower than the chosen target and positive values means that the simulated metric is greater than the chosen target. Standard deviations over the 200 runs are reported in parenthesis.

The results show that the node's up-time duration has a major impact on the model predictions. When using LoRa and the KNN model, the average $\overline{\Delta}$ for the energy consumption varies from $-171.89$J with 60*s* up-time to $-2629.47$J with 180*s* up-time (a small increase in the standard deviation is also visible). This variation of the average $\overline{\Delta}$ for the energy consumption is also visible for the DT model and, similar trends are shown by the results with NbIoT.

Regarding the average delta in coverage, it never exceeds more than $\pm 1.51$ nodes with a small standard deviation (lower than 1). Half of the scenarios cover more nodes than the chosen target.

From these observations, two major conclusions can be derived. First, there are parameters that impact significantly the differences between the target and the results obtained by simulation. In this case, the wireless technology and the up-time duration have an important impact. This difference is visible on Figure 3, where the data distribution is significantly different according to these two parameters. Second, the performance of the two models at predicting the target energy consumption and coverage is dependent on whether the target can be reached. The targeted energy consumption and coverage must be chosen within reachable ranges, to ensure comparable and meaningful predictions.

Concerning the comparison between the KNN and DT, there is no significant difference in using one or the other. As an example, when using LoRa with an up-time of 180*s* the KNN predictions leads to an increase in energy consumption (greater average $E_{total}$) compared to the DT, but with a better coverage. In most cases, KNN predicts classes with a greater coverage compared to DT while DT predicts classes with lower energy consumption. Thus, in scenarios where energy consumption is the priority, DT is a better choice compared to KNN whereas in scenario where the coverage is more important, KNN should be used.

Both models consume less energy than the target, with a coverage close to the target. In scenarios such as NbIoT with an up-time of 180*s*, the KNN model predictions achieve a better coverage compared to the target, while providing a lower average $E_{total}$. These results show that these methods can be a leverage to extend nodes lifespan, while maintaining a coverage close to the target.

## 5.3 Discussion

Choosing the correct data dissemination policy to use in deployment with the *offline* approach is promising. It allows an energy and coverage aware node deployment. It permits optimisations that are not easy to implement using *in situ* (e.g removing *Hint* from the dataset). As the model outcomes are based on existing

data, it is crucial that it reflects the behavior of real deployments. Data can be collected through simulations, test-beds experiments, real deployments or from related works. In addition, the energy consumption and the coverage target must be reachable based on the collected data to ensure accurate predictions of the models.

## 6 Conclusion

Disseminating data in a extremely constrained environment is a challenging task. The design of loosely coupled data dissemination policies in such context is proposed in [3] and it is a first attempt to answer this challenge. However, choosing the appropriate policy to match a given coverage and energy consumption target is an unanswered challenge. To solve this problem, this paper proposes to use classification algorithms and two training methodologies.

A study of an *in situ* learning approach of two classification models is provided. This study reveals that, even with relaxed assumptions (concerning the costs of communications during the training period), models can take several months to reach a threshold of 0.8 accuracy and F1-Score of 0.8. *In situ* learning can still be interesting in terms of coverage and energy consumption, when compared to fixed policies in given scenarios (e.g *Hint+Extended* using NbIoT). However, *in situ* has a significant energy consumption overhead, compared to other policies such as *Baseline* or *Extended* policy.

An *offline* approach, based on the same classification models, is proposed. Its analysis shows that classification models are able to predict the data dissemination policy that can match a given energy consumption budget and coverage target. For constrained environments, using *offline* trained models is a better trade-off, as *in situ* training has significant energy consumption overhead, even with relaxed assumptions concerning the costs of communications during the training period.

Future works include the investigation of the evolution of model learning curves on various network scales and compare them against real deployments. Learning curves of un-evaluated classification models should be studied, as it is a potential leverage to reduce the models training duration.

## Acknowledgments

The DAO project is supported by the Research Council of Norway (RCN) IKTPluss program, project number 270672. Thank you very much to the COAT ecologists, UiT.

## References

[1] Silvia Liberata Ullo and G. R. Sinha. Advances in smart environment monitoring systems using iot and sensors. *Sensors*, 20(11), may 2020.

[2] Issam Rais, Otto Anshus, John Markus Bjorndalen, Daniel Balouek-Thomert, and Manish Parashar. Trading Data Size and CNN Confidence Score for Energy Efficient CPS Node Communications. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, Melbourne, Australia, May 2020. IEEE.

[3] Issam Raïs, Loic Guegan, and Otto Anshus. Impact of loosely coupled data dissemination policies for resource challenged environments. In *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 524–533, 2022. doi: 10.1109/CCGrid54584.2022.00062.

[4] Muhammad Usman Younus, Muhammad Khurram Khan, Muhammad Rizwan Anjum, Sharjeel Afridi, Zulfiqar Ali Arain, and Abdul Aleem Jamali. Optimizing the Lifetime of Software Defined Wireless Sensor Network via Reinforcement Learning. *IEEE Access*, 9, 2021.

[5] Neng-Chung Wang and Wei-Jung Hsu. Energy Efficient Two-Tier Data Dissemination Based on Q-Learning for Wireless Sensor Networks. *IEEE Access*, 8, 2020.

[6] Saleh M. Altowaijri. Efficient Next-Hop Selection in Multi-Hop Routing for IoT Enabled Wireless Sensor Networks. *Future Internet*, 14(2), January 2022.

[7] S. V. N. Santhosh Kumar, Yogesh Palanichamy, M. Selvi, Sannasi Ganapathy, Arputharaj Kannan, and Sankar Pariserum Perumal. Energy efficient secured K means based unequal fuzzy clustering algorithm for efficient reprogramming in wireless sensor networks. *Wireless Networks*, 27(6), August 2021.

[8] Nabajyoti Mazumdar, Amitava Nag, and Sukumar Nandi. HDDS: Hierarchical Data Dissemination Strategy for energy optimization in dynamic wireless sensor network under harsh environments. *Ad Hoc Networks*, 111, February 2021.

[9] Hung Nguyen-Duy, Thu Ngo-Quynh, Fumihide Kojima, Tien Pham-Van, Toan Nguyen-Duc, and Sonxay Luongoudon. RL-TSCH: A Reinforcement Learning Algorithm for Radio Scheduling in TSCH 802.15.4e. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea (South), October 2019. IEEE.

[10] Moussa Aboubakar, Pierre Roux, Mounir Kellil, and Abdelmadjid Bouabdallah. An Efficient and Adaptive Configuration of IEEE 802.15.4 MAC for Communication Delay Optimisation. In *2020 11th International Conference on Network of the Future (NoF)*, Bordeaux, France, October 2020. IEEE.

[11] Claudio Savaglio, Pasquale Pace, Gianluca Aloi, Antonio Liotta, and Giancarlo Fortino. Lightweight Reinforcement Learning for Energy Efficient Communications in Wireless Sensor Networks. *IEEE Access*, 7, 2019.

[12] Francesco Fraternali, Bharathan Balaji, Yuvraj Agarwal, and Rajesh K. Gupta. ACES: Automatic Configuration of Energy Harvesting Sensors with Reinforcement Learning. *ACM Transactions on Sensor Networks*, 16(4), November 2020.

[13] Faycal Ait Aoudia, Matthieu Gautier, and Olivier Berder. RLMan: An Energy Manager Based on Reinforcement Learning for Energy Harvesting Wireless Sensor Networks. *IEEE Transactions on Green Communications and Networking*, 2 (2), June 2018.

[14] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Athens, March 2018. IEEE.

[15] Technical Marketing Workgroup 1.0. A technical overview of LoRa and LoRaWAN, November 2015.

[16] Jeff Geerling. Raspberry Pi Power Consumption Benchmarks. https://www.pidramble.com/wiki/benchmarks/power-consumption.

[17] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. A survey on LPWA technology: LoRa and NB-IoT. *ICT Express*, 3(1), March 2017.

[18] Ritesh Kumar Singh, Priyesh Pappinisseri Puluckul, Rafael Berkvens, and Maarten Weyn. Energy Consumption Analysis of LPWAN Technologies and Lifetime Estimation for IoT Application. *Sensors*, 20(17), August 2020.

[19] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for Multi-Class Classification: An Overview, August 2020.

[20] Sumana De and Baisakhi Chakraborty. An energy-efficient wireless sensor network construction algorithm for air quality condition detection system. *Computers & Electrical Engineering*, 91, May 2021.

[21] Imran Zualkernan, Salam Dhou, Jacky Judas, Ali Reza Sajun, Brylle Ryan Gomez, and Lana Alhaj Hussain. An IoT System Using Deep Learning to Classify Camera Trap Images on the Edge. *Computers*, 11(1), January 2022.

[22] Claudia Perlich. Learning Curves in Machine Learning.