Impacts of GPS module on energy consumption and machine-learning based battery lifetime estimation

André Teixeira de Aquino, José Ailton Leão Barboza Júnior, Nicolas de Araujo Moreira, Paulo Peixoto Praça

Impacts of GPS module on energy consumption and machine-learning based battery lifetime estimation

André Teixeira de Aquino[1]*, José Ailton Leão Barboza Júnior[1]†, Nícolas de Araújo Moreira[1]‡, and Paulo Peixoto Praça[1]§

Federal University of Ceara, Fortaleza, Brazil
andret.aquino@gmail.com, ailton.junior@ufersa.edu.br, nicolas.araujom@gmail.com, paulopp@dee.ufc.br

Abstract

The maintenance of a Wireless Sensor Network may represent a logistic challenge. Battery replacement on applications involving huge numbers of sensors spread over a wide and distant area may be expensive and difficult. So, good planning of the maintenance schedule is necessary. This work discusses the impact of the GPS module for bovine tracking on farms on current consumption and its estimation. It compares Long-Short Term Memory (LSTM) networks and Decision Trees with AdaBoost to estimate this consumption. The results show that the activation of the GPS module increases 114.36% the current consumption and Decision Trees with AdaBoost using 300 estimators and with a depth equal to 20 outperforms LTSM with Root Mean Square (RMS) error of 0.00015.
Keywords: Machine Learning, Internet of Things, Wireless Sensor Networks

1 Introduction

The Internet of Things (IoT) is the connection between different types of sensors, devices, objects, machines, vehicles, and buildings using Wireless Sensor Networks (WSNs). IoT is a key concept in the evolution of communication, industry, and agriculture, for example, [19, 21, 4].

The nodes of this network are self-organized and decentralized intelligent sensors able to communicate with each other through radio links [10, 1]. WSN are characterized by complexity and heterogeneity: it integrates different analog and digital blocks, such as Analog-Digital Converters (ADCs), Digital Signal Processors (DSPs), and Radio-Frequency Circuits [10]. Usually, the communication of WSNs is multi-path asymmetric, presenting a many-to-one data flow, i.e., the nodes send data to a monitoring station [10].

Wireless Sensor Networks (WSNs) have been used widely in many scientific, industrial, and commercial applications, such as ecological control (forest fire detection, pollution control), intelligent houses, etc., acquiring environmental parameters (e.g. temperature, illumination, etc.) [10]. WSN must be low cost and have strong energy constraints: depending on the scenario, it is quite difficult to recharge or replace the battery, as for example, applications with a huge number of nodes spread over a large area (forest or farm monitoring, for instance) or applications in distant and dangerous locations (such as volcano monitoring) [10].

The uneven use of livestock pastures has a significant impact on productivity, the ecosystem, biodiversity, and function. The use of collars with Global Positioning System (GPS) allows for obtaining a large amount of data about the position and activity of the animals, giving

*Responsible for experimental results
†Responsible for experimental results
‡Co-Advisor, and responsible for theoretical results
§Advisor

new possibilities to researchers [12]. Similarly to the previously described scenarios, one of the limitations related to using these devices is their autonomy, more precisely, the battery lifetime. Battery replacement or recharging impacts the logistics of animal handling. So, it is important to have an estimate of battery lifetime to have a more adequate maintenance schedule for these networks. A second relevant point is related to the way the data is accessed. The current work is inserted in the context of a project of a collar for a location using satellite, data acquisition, and transmission using LoRa for remote data access for real-time surveying of pastures.

This paper has discussed the impact of the GPS module, to track bovines on farms, activation on current consumption, and two machine-learning-based estimation techniques are used to estimate the current consumed and then compared. The remainder of this paper is as follows: the next section (2) gives an overview of existing works related to this research. The following section (3) presents the two machine-learning techniques used: Long-Short Term Memory (LSTM) Networks and Decision Trees with AdaBoost. Section 4 presents the used hardware. Then, section 5 presents and discusses the results of measurements and estimation.

2 Literature Review and State of the Art

As discussed before, IoT has applications in a wide variety of areas. A considerable number of researches discusses also how to increase the battery lifetime and study factors that affect it, such as the impact of interference on energy consumption. Applications of IoT in precision irrigation for agriculture are explored in [8, 4]. The author in [15] explores the use of IoT in wildlife tracking.

The author in [16] presents a platform named SYNERGIE to measure the energy consumption on nodes of WSNs and quantify the impact of interference on energy consumption. It measures the consumption of microcontroller, memory, sensor, and radio-frequency circuit at a rate of 1,500 samples per second through General Purpose Input-Output (GPIO) lines and a resistor interface. The measurements are transmitted to a computer through a serial link. This platform is based on commercially available low-cost and low-power components: AT-mega328p microcontroller, which contains an embedded 10-bit Analog-to-Digital-Converter (ADC) and five operational amplifiers XCT1086 able to acquire up to five independent measurements. A similar power meter system is presented in [23]: the authors present a real-time power metering system for a wireless sensor network called Nemo, a noninvasive and plug-and-play device that can be easily installed without wires. Nemo has a circuit that dynamically adjusts the resistance of shunt resistors according to the current load (shunt resistor switch). Nemo presents a high measurement accuracy (average measurement error of 1.34%) on TelosB devices.

The author in [6], in addition to the introduction of SYNERGIE platform, also explores the problem of energy efficiency and discusses a model of energy consumption on WSN nodes, presenting software that allows generating automatically an energy consumption model based on Principal Component Analysis (PCA) and an algorithm for node lifetime estimation based on Markov chain.

The authors in [2] also present a model for global energy consumption for WSN nodes using a simulator called Ns-2 and iMote2 hardware. The results show energy consumption simulated close to measured values on real devices under the same experimental conditions for radio-frequency circuits. A similar work is shown in [22], however, based on an event-trigger mechanism for each component (processor, radio-frequency circuit, sensors). Energy modeling in sensor networks is also discussed in [14].

3 Theoretical Introduction: Machine Learning Algorithms

In this section, the two algorithms used to estimate the current consumption will be introduced: Long-Short-Term Memory (LSTM) Networks and Decision Trees with AdaBoost.

3.1 Long-Short-Term Memory (LSTM) Networks

Long Short-Term Memory Network (LSTM) is a kind of deep learning algorithm called Recurrent Neural Networks (RNN) - dense neural networks connected to themselves: this feedback loop allows the information to be stored [11, 7]. LSTM networks have a high capability of predicting time series and it differs from other types of neural networks due to the introduction of a forget gate, that controls which states are remembered or forgotten.

A gate is a structure that adds, removes, or transmits data, being composed by a sigmoid neural network layer (here denoted by σ), that outputs a number within the interval $[0, 1]$ - where 1 denotes that all information is re-transmitted and 0 denotes to not allow the transmission of any information, and a Hadamard product (denoted by $\circ$). Let's denote by U and V the matrices connecting inputs and recurrent outputs, respectively, and $, x_t, h_{t-1}$, the input and the output of the previous cell, thus:

$$h_g = \sigma(Ux_t + Vh_{t-1}). \tag{1}$$

Each cell is composed of: (i) cell state (c_t), the internal memory of the cell that stores long and short-term memories; (ii) hidden state (c_h), which decides to retain short and/or long-term information on state cell to predict; (iii) input gate (i_t), which conditionally decides which input values will be updated on state memory; (iv) forget gate (f_t), which decides how much information of current input and from previous state cell is forwarded to current state cell; and, finally, (v) the output gate (o_t), which conditionally decides which will be the output according to input and memory block.

A piece of data is forgotten only when new data replaces it. LSTM starts deciding which data will be forgotten and then which new data will be stored.

The previous cell c_{t-1} is updated into a new cell c_t and then the old state is multiplied by a forgetting factor f_t. A new possible value is generated, so it is necessary to decide what will be transmitted to output based on the cell state applying a sigmoid layer. The tanh layer forces the value to be mapped in the interval $[-1, 1]$. Denoting U_g, V_g the weights for input and previous cell output, respectively, and b_g an input bias, we have:

$$g = \tanh(b_g + x_t U_g + h_{t-1} V_g). \tag{2}$$

The input gate is a hidden layer of sigmoid activation nodes with weighted inputs x_t, h_{t-1}. Then, the expression for the input gate can be written as:

$$i = \sigma(b_i + x_t U_i + h_{t-1} V_i). \tag{3}$$

Denoting by s_t the internal state of an LSTM cell, the forget gate is a set of sigmoid activation nodes multiplied by s_{t-1} to determine what must be remembered (output of forget gate close to 1) and what must be forgotten (output of forget gate close to 0), allowing LSTM to learn according to the scenario. So, the mathematical expression for the forget gate is given by:

$$f = \sigma(b_f + x_t U_f + h_{t-1} V_f). \tag{4}$$

So, the output of the forget gate works as a weight for the internal states. Thus, the output of this state s_t is expressed by:

$$s_t = s_{t-1} \circ f + g \circ i. \tag{5}$$

The last stage of the LSTM cell is the output gate, which is composed of a sigmoid layer and hyperbolic tangent (tanh) layer, responsible to create a vector with new possible values to be added to a $\tilde{c}_t$ state. The output is given by:

$$o = \sigma(b_o + x_t U_o + h_{t-1} V_o). \tag{6}$$

thus, cell output is expressed by:

$$h_t = \tanh(s_t) \circ o. \tag{7}$$

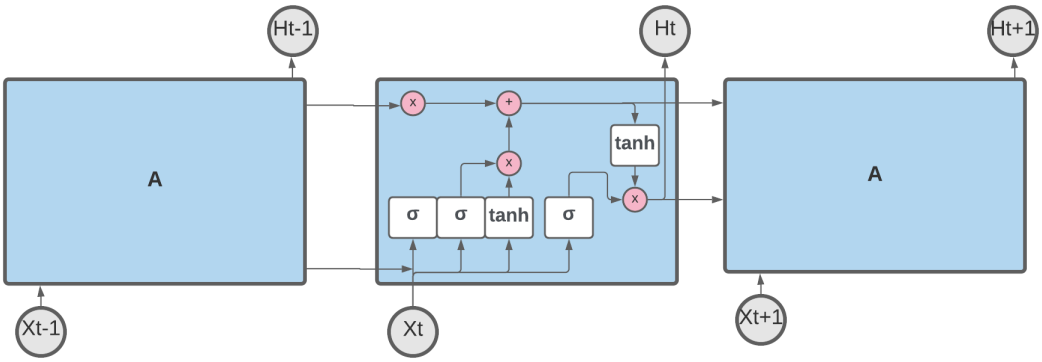

Figure 1: LSTM Architecture.

3.2 Decision Tree

A tree is a set T of positive integers with two functions $l(\cdot), r(\cdot) \geq 0$ denoting the left and right nodes respective with equality indicating a terminal node and from T to $T \cup \{0\}$ where each element of T denotes a node in the three. There is a unique parent of each node [18].

Decision trees are capable of modeling complex nonlinear decision boundaries. Associated with each internal node of the tree are a variable and a threshold and with each leaf (terminal node), is a class label. The decision tree divides the decision into simpler decisions at each node.

A sub-tree is a non-empty subset ($T_k \subset T$) associated to two functions l_k, r_r. Denoting $\{u(t), t \in \tilde{T}\}$ a partition of the data space $\mathbb{R}^p$ ($u(t)$ is a subspace of $\mathbb{R}^p$) and denoting $\omega_{j(t)} \in \{\omega_1, ..., \omega_C\}$ denote one of the class labels [18] then, a classification tree is a tree T associated with the class labels $\{\omega_{j(t), t \in \tilde{T}}\}$ and the partition $\{u(t), t \in \tilde{T}\}$ [18].

A classification tree is built using a labeled data set, $\mathcal{L} = (\mathbf{x}_i, y_i), i = 1, ..., n$ where $\mathbf{x}_i$ are the data samples and y_i the corresponding class labels [18]. Example: Properties {taste,color,shape,size}, pattern $\mathbf{x}$={sweet,yellow,long,medium} is classified as banana.

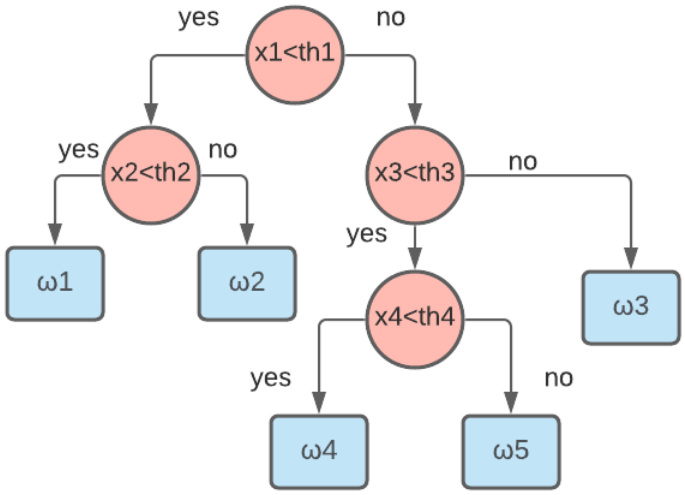

Figure 2: Generic example of a decision tree.

3.3 AdaBoost

Boosting is a technique for improving the accuracy of learning algorithms [3] and is a classifier in the form [20]:

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \tag{8}$$

where each f_t is a weak learner that takes an object $\mathbf{x}$ and returns the class of the object. At each iteration of the training process, a weight $w_{i,t}$ is assigned to each sample in the training set equal to the current error on that sample.

AdaBoost (Adaptive Boosting) is a machine learning algorithm [20], an important ensemble method, and is a particular method of training a boosted classifier. It is very accurate, simple to implement, and presents good generalization [20]. AdaBoost often tends to be empirically resistant to overfitting [5], however, presents a sub-optimal solution. The AdaBoost algorithm was introduced by Freund and Schapire in 1995 [5, 20]. AdaBoost training process selects only features known to improve the prediction, reducing dimension and improving computation time once irrelevant features are not used [20]. The AdaBoost algorithm aims to build a strong classifier by doing a linear combination of weak learners (classifier) $h_t(x) : \xi \to \{-1, 1\}$ [5]:

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{9}$$

AdaBoost is capable of reducing bias and variance of weak classifiers, has a good generalization property, and its output converges to the algorithm of the likelihood ratio. AdaBoost can be seen as a feature selector and is close to sequential decision-making (it produces a sequence of gradually more complex classifiers).

Taking as input a training set $(x_i, y_i), i = 1, ..., m$ where x_i belongs to some domain and y_i is a label on some label set Y. For instance, assume $Y = \{-1, 1\}$. Let's consider $t = 1, ..., T$ the number of rounds. One of the ideas of the algorithm is to maintain a distribution or a set of weights over the training set. The weight of this distribution on training example i on round t is denoted $D_t(i)$. Initially, all weights are set equally, but on each round, the weights of incorrectly classified examples are increased to force the focus on the examples harder to learn. The objective of the learner is to find a weak hypothesis $h_t : X \to \{-1, 1\}$ appropriate for the distribution D_t. The goodness of a weak hypothesis is measured by its error [3]:

$$\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i:h_t(x_i) \neq y_i} D_t(i) \tag{10}$$

First, it assigns equal weights to all the training examples. Denoting D_t the distribution of the weights at the t-th learning round. From the training set and D_t, the algorithm generates a weak learner $h_t : \mathcal{X} \to \mathcal{Y}$ and then uses the training examples to test h_t. The weights of the incorrect classified examples will be increased. Such a process is repeated for T rounds. The final model is derived by weighted majority voting of the T weak learners [20]. The steps of the algorithm are shown below [3, 5, 17, 13, 20]:

Algorithm 1 AdaBoost

procedure AdaBoost($(x_i, y_i), i = 1, ..., N, \_i \in X, y_i \in Y = \{-1, 1\}$)
  Initialize $D_1(i) = 1/m$
  for $t = 1, ..., T$ **do**:
    Train the weak learner using distribution D_t
    Get weak hypothesis $h_t : X \to \{-1, 1\}$ with error $\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$
    Choose $\alpha_t = 0,5 \ln(\epsilon_t^{-1}(1 - \epsilon_t))$
    Update: $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$, where Z_t is a normalization factor
  end for
Output the final hypothesis: $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$
end procedure

4 Hardware

The system is composed of a microcontroller module, a WiFi Lora 32 (V2) module, a u-blox Gy-neo6mv2 GPS module, and an ADS 1115 16 analog-digital (A/D) converter. To activate and deactivate the GPS module, a BC 548 transistor was used to cut the ground (GND) of GPS module when a logic low level is applied to its base terminal that is connected to the microcontroller (See Fig. 3).

The current measurement is done through the voltage drop on a 1 Ohm resistor connected between the battery and the system. Two channels of A/D converter are connected between the terminals of the resistor. The microcontroller receives data from the converter and calculates the voltage between terminals. The obtained value is the current, once the resistance is equal to 1 ohm.

The system operates in cycles. The microcontroller activates the GPS and waits until the reception of a valid location. After receiving the valid location, GPS is deactivated and the coordinates are sent using the radio and the system waits five minutes to restart the cycle. The current reading is done each second.

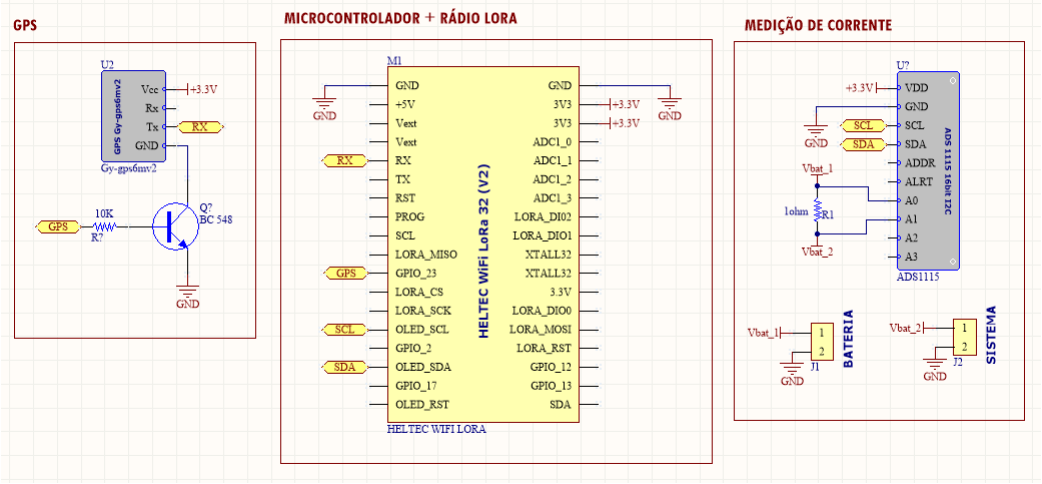

Figure 3: Circuit implemented.

5 Results and Discussion

The consumed current was estimated using LSTM networks (Figure 4) and a Decision Tree with AdaBoost (Figure 5). As is possible to see, 11,784 current samples were measured. 67% of the data were used for training and the 33% remaining were used for testing. The cycles of GPS activation/deactivation are quite evident on the graphics, showing the 5min period to restart the GPS data acquisition. When the GPS is activated, the circuit consumes 151.48 mA (average), while when deactivated, the circuit consumes 70.666 mA(average). It means that the activation of GPS module increases 114.36% the energy consumption.

In Figure 4, the blue curve represents the measured current, the orange one is the result of the estimation of LSTM during the training period and the green one is during the test period. The loss function used was Mean Squared Error and the Adam algorithm was used as the optimization function. Adam optimization is a stochastic gradient descent method adequate for large data and parameters problems based on adaptive estimation of first and second-order moments and which is computationally efficient, that consumes low memory, and which is invariant to diagonal re-scaling of gradients [9]. The calculated Root Mean Square (RMS) error for LSTM was 9.88. In figure 5, is possible to see, the blue curve represents the training samples (real data), the green one is the estimation with 1 estimator, and the red one with 300 estimators. The maximum depth of the tree was set to 10. Clearly, the accuracy when using only one estimator is much better, having an RMS error 0.0086.

As is possible to see, both algorithms can predict the behavior (profile of curve) of current consumption. So, the Decision Tree with AdaBoost with 1 estimator has better accuracy than LSTM and LSTM has better accuracy than the Decision Tree with AdaBoost with 300 estimators considering a depth equal to 10. However, if we increase the depth of the Decision Tree with AdaBoost to 20, the curve fits almost perfectly (see figures 6 and 7 for 1 estimator and 300 estimators, respectively) with a small advantage for the case using 300 estimators, presenting and RMS error equal to 0.00015.

While the architecture presented in [6, 16] is based on an AT-mega328p microcontroller, which contains an embedded 10-bit ADC, the present work is based on an ESP32 microcontroller with a 16-bit ADC. None of the papers mentioned in section 2 focuses specifically on GPS

consumption or on energy consumption forecasting. The models presented in [6] (PCA and Markov Chain) have errors between the experimental and the model varying between 1.09% and 15.19%.

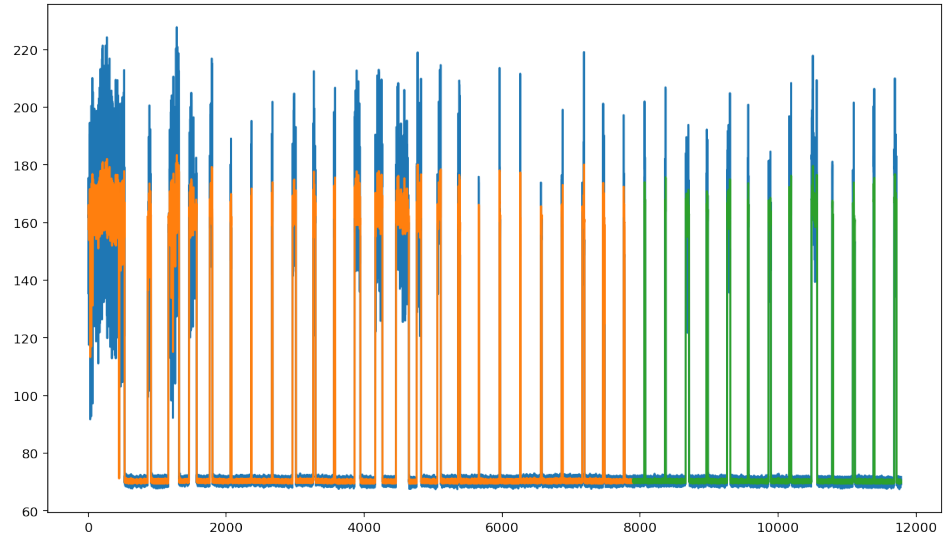

Figure 4: Current consumed estimated using LSTM.

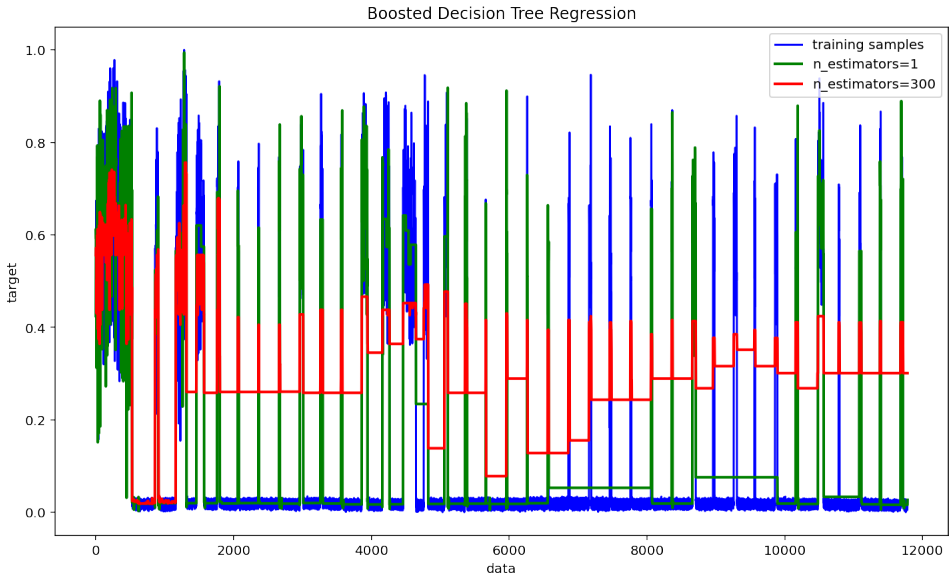

Figure 5: Current consumed estimated using Decision Tree with AdaBoost considering depth equal to 10.

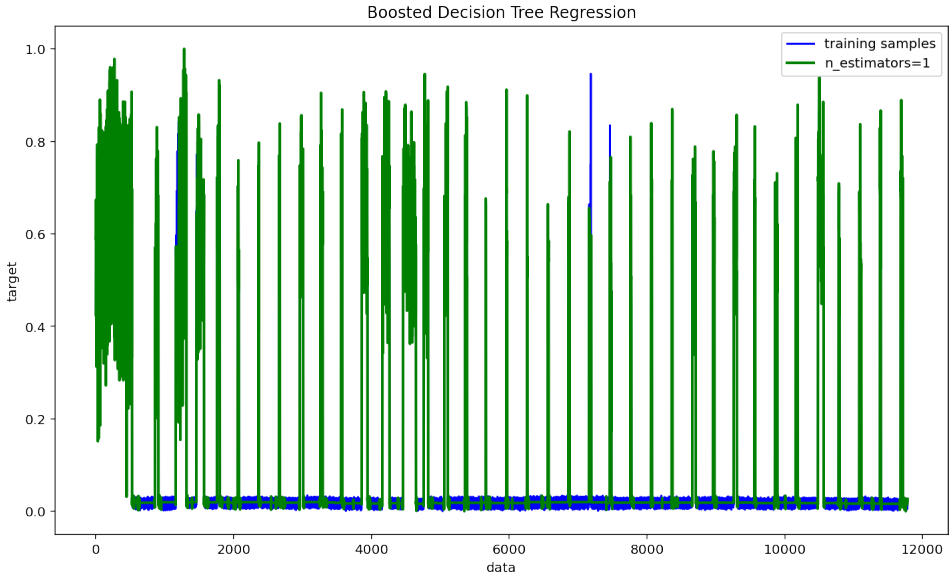

Figure 6: Current consumed estimated using Decision Tree with AdaBoost considering depth equal to 20 and 1 estimator only.

Figure 7: Current consumed estimated using Decision Tree with AdaBoost considering depth equal to 20 and 300 estimators.

6 Conclusion and Future Works

WSN has a wide number of applications. In our study, we used GPS to have a near real-time bovines location on large farms. The maintenance of the devices in distant and hard access areas may represent an important challenge for WSNs. In this context, is important to have an estimation of the energy consumption of WSN devices to predict their lifetime and have better maintenance schedules.

This paper showed the impact of the GPS module on current consumption and compared two approaches to predict the current consumption of WSN devices, LSTMs and Decision Tree with AdaBoost, with the last one presenting a better accuracy, mainly considering a depth equal to 20.

Possible interesting approaches to be explored in future works are RANdom SAmple Consensus (RANSAC) and Support Vector Machines (SVM).

References

[1] Jean Carle and David Simplot Ryl. Energy-efficient area monitoring sensor networks. *Computer*, pages 40–46, 2004.

[2] A. Coutray, A. Pegatoquet, M. Auguin, and C. Chabaane. Wireless sensor network node global energy consumption modeling. In *Conference on Design and Architectures for Signal and Image Processing (DASIP) 2010*, 10 2010.

[3] Yoav Freund and Robert E Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771 – 780, 9 1999.

[4] Nidia G. S. Campos, Atslands R. Rocha, Rubens Gondim, Ticiana L. Coelho da Silva, and Danielo G. Gomes. Smart and green: An internet-of-things framework for smart irrigation. *Sensors*, 20(1), 2020.

[5] Wei Gao and Zhi Hua Zhou. On the doubt about margin explanation of boosting. *Artificial Intelligence Journal*, 203:1–18, 7 2013.

[6] Román José Igual-Pérez. *Adaptive MAC layer for interference limited WSN*. PhD thesis, Université de Lille, 2015.

[7] Rafal Josefowic, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[8] Carlos Kamienski, Juha Pekka Soininen, Markus Taumberger, Ramide Dantas, Attilio Toscano, Tullio Salmon Cinotti, Rodrigo Filev Maia, and André Torre Neto. Smart water management platform: Iot based precision irrigation for agriculture. *Sensors*, (19), 2019.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[10] Abdelbasset Massouri. *Modélisation comportamentale SystemC-AMS d'interfaces RF et liaisons radio multipoint pour réseaux de capteurs*. PhD thesis, Université de Lille, 2012.

[11] L.R. Medsker and L.C. Jain. *Recurrent neural networks: design and applications*. CRC Press, 2001.

[12] Humberto Lauro Perotto Baldivieso, Susan Margaret Cooper, Andres Francisco Cibilis, Manuel Figueroa Pagan, Karen Udaeta, and Christina Michelle Black Rubio. Detecting autocorrelation problems from gps collar data in livestock studies. *Applied Animal Behavior Science*, (136):117–125, 2012.

[13] Raúl Rojas. Adaboost and the super bowl of classifiers: A tutorial introduction to adaptive boosting., 2009.

[14] D. Schmidt, M. Kramer, T. Kuhn, and N. Wehn. Energy modelling in sensor networks. *Advances in Radio Science*, (5):347–351, 2007.

[15] Viktor Toldov. *Adaptive MAC layer for interference limited WSN*. PhD thesis, Université de Lille, 2017.

[16] Viktor Toldov, Román Igual-Pérez, Rahul Vyas, Alexandre Boé, Laurent Clavier, and Nathalie Mitton. Experimental evaluation of interference impact on the energy consumption in wireless sensor networks. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, 2016.

[17] Liwei Wang, Massahi Sugiyama, Cheng Yang, ZhiHua Zhou, and Jufu Feng. On the margin explanation of boosting algorithms. In *21st Annual Conference on Learning Theory*, 7 2008.

[18] Andrew Webb. *Statistical Pattrern Recognition*. John Wiley and Sons, 2 edition, 2002.

[19] Gu Wei. *Robustness against interference in Internet of Things*. PhD thesis, Université de Lille, 2012.

[20] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Bing Ng, Angus amd Liu, Phillip S. Yu, ZhiHua Zhou, Michael Steinbach, David J. Hand, and Dan. Steinberg. Top 10 algorithms in data mining. *Knowledge Information Systems*, 14, 2008.

[21] Xin Yan. *Robustness against interference in sensor networks*. PhD thesis, Université de Lille, 2015.

[22] Hai Ying Zhou, Dan Yan Luo, Yan Gao, and De Cheng Zuo. Modeling of node energy consumption for wireless sensor networks. *Wireless Sensor Newtork*, (3):18–23, 2011.

[23] Ruogu Zhou and Guoliang Xing. Nemo: a high fidelity noninvasive power meter system for wirelss sensor networks. In *2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 141–152, 2013.