Heterogeneous data reduction in WSN: Application to Smart Grids

Jad Nassar, Karen Miranda, Nicolas Gouvy, Nathalie Mitton

Heterogeneous data reduction in WSN: Application to Smart Grids

Jad Nassar
HEI - Yncrea HdF, France
Inria, France
Jad.nassar@{yncrea,inria}.fr

Karen Miranda
UAM-Lerma, Mexico
k.miranda@correo.ler.uam.mx

Nicolas Gouvy
HEI - Yncrea HdF, France
Nicolas.gouvy@yncrea.fr

Nathalie Mitton
Inria, France
Nathalie.mitton@inria.fr

ABSTRACT

The transformation of existing power grids into Smart Grids (SGs) aims to facilitate grid energy automation for a better quality of service by providing fault tolerance and integrating renewable energy resources in the power market. This evolution towards a smarter electricity grid requires the ability to transmit in real time a maximum of data on the network usage. A Wireless Sensor Network (WSN) distributed across the power grid is a promising solution, given the reduced cost and ease of deployment of such networks. These advantages come up against the unstable radio links and limited resources of WSN. In order to reduce the amount of data sent over the network, and thus reduce energy consumption, data prediction is a potent solution of data reduction. It consists on predicting the values sensed by sensor nodes within certain error threshold, and resides both at the sensors and at the sink. The raw data is sent only if the desired accuracy is not satisfied, thereby reducing data transmission. We focus on time series estimation with Least Mean Square (*LMS*) for data prediction in WSN, in a Smart Grid context, where several applications with different data types and Quality of Service (QoS) requirements will exist on the same network. LMS proved its simplicity and robustness for a wide variety of applications, but the parameters selection (step size and filter length) can directly affect its global performance, choosing the right ones is then crucial. Having no clear and robust method on how to optimize these parameters for a variety of applications, we propose a modification of the original *LMS* that consists of training the filter for a certain time with the data itself in order to customize the aforementioned parameters. We consider different types of real data traces for the photo voltaic cells monitoring. Our simulation results provide a better data prediction while minimizing the mean square error compared to an existing solution in literature.

KEYWORDS

Smart Grids, Wireless Sensor Network, Data Reduction, Data Prediction, Time Series Estimation, Least Mean Square, Quality of Service

1 INTRODUCTION

Wireless Sensor Networks (WSNs) transmitting, monitoring and exchanging control command messages is a promising solution in the shift from the traditional electric grid into a "smarter" one in which the integration of renewable energy resources, energy reduction and auto-adaptation are the main benefits. This is done by equipping the electric grid with wireless sensors located at strategic measuring points to achieve remote monitoring, data collection and control of the grid. In a Smart Grid (SG), electricity and energy do exist, but connecting sensors to such high voltage with intermittent and ill-adapted energy levels is sometimes inappropriate. For that, battery-powered sensors must be deployed all over the grid alongside with the main-powered ones. Thus, reporting data measurements at specific intervals has a direct effect on the sensors battery lifetime since the communication task consumes most of their available energy [17]. In such context of continuous data reporting, data changes are limited between each reading, which will cause redundant information at the destination. To mitigate these energy losses, data reduction approaches are used [2]. Although data reduction techniques are widely used in literature, their adaptability is limited to specific applications. Thus, using these techniques for SG applications requires specific customization which is not addressed so far in the literature due to such applications are characterized by their diversity in terms of data types and QoS requirements. In this work, we focus on a time series forecasting technique, called Least Mean Squares (*LMS*), which is an adaptive algorithm with very low computational overhead and memory consumption, that despite its simplicity, provides satisfactory performances in terms of computational speed, robustness and precision [9]. *LMS* main drawback is the complex task of choosing the adequate step size and filter length for different applications with different data characteristics (e.g., maximum, minimum, data variation). This directly impacts the stability of the algorithm specially when using it with different data types as is the case in a SG context [18] [16]. We propose a modification for the *LMS* filter used for data prediction in WSN, which is introduced in [19], to adapt it to the different data types. We applied the algorithm to photo-voltaic cells monitoring data set. We tune the parameters of the filter by training it for a certain time with the real data values of every data set and choosing the values that minimizes the Mean Square Error (*MSE*). Different parameters are obtained after the training process in accordance with every data type. Our simulation results provide a better data prediction and a lower *MSE* compared to a solution from literature. The rest of the paper is organized as follows: Section 2 presents prior work on data reduction and *LMS* algorithm. Section 3 provides a brief overview about the aforementioned algorithm and its limitations. Section 4 describes our proposal. Section 5 shows the simulation parameters and environment used to validate our proposal. Section 6 presents the performance evaluation of our proposal. Section 7 discusses some relevant issues about our proposal. Finally, Section 8 concludes the paper.

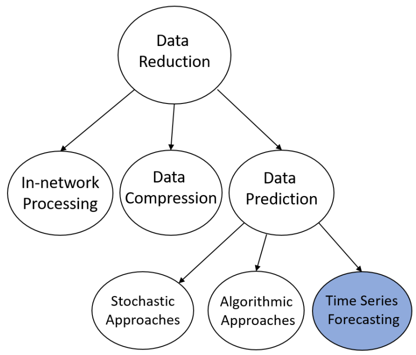


Figure 1: Categorization of energy saving in sensor networks

2 RELATED WORK

Data reduction approaches can be classified into three main categories: In-network processing, data compression and data prediction [2] (Figure 1). In-network processing consists of processing the data collected by the sensors nodes themselves between the source and the destination, in this way the amount of data is reduced while traversing the network. Unlike data compression, where the data is generally compressed/aggregated while performing data aggregation techniques [8] on specific nodes called "aggregators". Data prediction, that is the point of interest in the current work, aims for reducing the amount of data sent by each sensor by predicting the measured values both at the source and the sink node using specific algorithms, which will require sending the predicted information only if it is shifted from the sensed one by a certain threshold. Most of these algorithms work as follow. A model is constructed at the sensor node and sent to the sink node to keep track of the sensed phenomenon, then the sink node answers the user queries by using the predicted values from the model without communicating with the sensor node, thus reducing the energy consumption. This operation is valid only if the model at the sensor nodes is a valid representation of the phenomenon at a given instant. For that, the characteristics of a data prediction technique rely on the way the model is built. These techniques can be split into three categories [2]: stochastic, algorithmic and time series forecasting approaches. Stochastic approaches consist of a characterization of the sensed phenomenon as a random process. A probabilistic model can be used for data prediction. The main drawback of these approaches is their high computational overhead, which is not suitable for sensors with limited capabilities. Algorithmic approaches tend to be application specific, which may not be suitable to a SG with different applications having different characteristics running on the same network. Finally, time series estimations consist of the use of a model to predict future values based on previously observed ones. They provide satisfactory and accurate results even when simple and lightweight models are used which is the most beneficial in energy limited WSNs.

In literature, extensive work addressed time series forecasting techniques for WSNs [7] [3]. For example, in [15], a couple of autoregressive mechanisms were proposed to predict sensed samples in WSNs. The authors used Yule-Walker and Lattice-based approaches to estimate the model coefficients. Similarly, several works focused on *LMS* algorithm as well. In [13] a gradient adaptive step size (μ) algorithm with dual *LMS* adaptive filters was proposed, where the gradient is measured with two *LMS* filters. In [21] a new approach for updating the step size was proposed, by computing it in each iteration. The step size is dynamically re-chosen at each time point to minimize the sum of the squares of the estimation errors up to the current time, irrespective of the values of μ at all previous time points. In [19], an implementation of *LMS* algorithm for prediction in WSN is presented. The *LMS* algorithm uses a dual prediction scheme by running the instance of the filter on both the sink and the node. In [20], a variable step size is proposed to improve the initial adaptation of the data by switching to a new step size stable value after μ has sufficiently learned what kind of data the filter receives. Many other works have addressed the variable step size of *LMS* [4]. However, all of these proposals mostly require many adjustments of several parameters in order to optimize μ or update it on every iteration which is not suitable for a WSN with limited computation capabilities. Normalized Mean Square Error (*NLMS*) [10] is a modification of the *LMS* algorithm in which the step size is normalized with the power of the input data. In order to mitigate the variation of the latter the step size is updated automatically accordingly. Although *NLMS* offers a higher stability than *LMS*, the base value of the step size has to be chosen carefully. Moreover, computing the step size on every iteration is a costly task for WSN with retrained energy. The Recursive Least Square (*RLS*) [10] adaptive filter is another algorithm that recursively finds the filter coefficients in order to minimize the weighted linear least square cost function related to the input signals. *RLS* algorithm has excellent performance in time varying environments and exhibits fast convergence, but this comes at the cost of high computational complexity which is also inadequate to WSNs. The readers may refer to [6] for a comparison between *LMS*, *NLMS* and *RLS*.

Even though the time series estimation techniques have been successfully used in WSN applications, it is important to note that for each individual application the estimator parameters such as weights and order must be computed, moreover, a single time series estimation may do not fit for all different applications [14]. This is particularly noteworthy because in a SG network converges different data types with different QoS necessities. Thus, the proposed solution should handle those requirements and be as general as possible.

3 LMS ALGORITHM OVERVIEW

A sensor node generates a data stream $u[n]$ which consists of the previous n readings, which will form the input of the filter. The predicted values $y[n]$ at the output of the filter are such that [11]:

$$y[n] = w^T[n]u[n] \tag{1}$$

which is a linear combination of the previous n samples of the data stream weighted by a weight vector $w[n]$, where:

$$w(n) = [w_1, w_2, ..., w_M]^T \tag{2}$$

and

$$u(n) = [u(n\text{-}1), u(n\text{-}2), ..., u(n\text{-}M)]^T \tag{3}$$

where M is an integer corresponding to the memory of the filter also called filter length (how many previous samples it will use). We note that both w and u are of length M. The error between the output and the desired signal $d[n]$ that the filter tries to adapt to is computed by:

$$e[n] = y[n] - d[n] \tag{4}$$

This error is given as an input for the adaptation algorithm, which will update the weight coefficients at the next instant $n + 1$ by:

$$w[n + 1] = w[n] + \mu u[n]e[n] \tag{5}$$

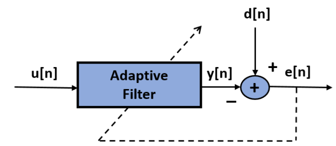
where μ is the step size parameter. The weight vector is modified at each step in order to minimize the *MSE*. With a simple modification for the *LMS* filter structure (figure 2), the *LMS* algorithm can be used for prediction by delaying the input signal by one step, using it as a the reference desired signal $d[n]$. The filter computes the estimated value $\hat{u}[n]$ of the input signal at time instance n, as a linear combination of the M previous readings. The step size μ and the filter length M are two important parameters that need to be defined in order to ensure the convergence and robustness of the algorithm. The former will tune the convergence of the algorithm and the latter impacts directly the computational load and memory consumption by considering more or less samples. A detailed explanation of the *LMS* filter can be found in [11]. The implementation of the *LMS* algorithm for data prediction in WSN is first presented in [19]. Here, identical filters are introduced at both the source and the sink referred as $LMS - DPS$ (dual prediction scheme). The algorithm consists of three modes of operation: Initialization, normal and stand-alone mode. In the initialization mode the data samples are collected and reported to the sink without prediction. In this phase the step size μ must be determined. Both the node and the sink compute the value of μ. It must satisfy the following condition[5]:

$$0 \leq \mu \leq \frac{2}{\lambda_{max}} \tag{6}$$

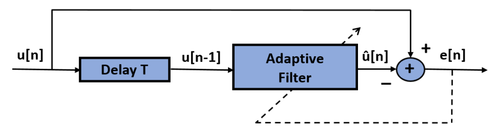
where λ_{max} is the greatest eigenvalue of the auto-correlation matrix **R**. In normal mode, both the sink and the node use the last M samples to compute the prediction for the upcoming measurement, and update the filter coefficients. When the error drops below e_{max} for M consecutive iterations, the node switches to stand-alone mode. We note that the default start values for the filter weights are assumed to be zero. In the stand alone mode, the node still collects data and makes predictions, but as long as the error is below e_{max}, the filter is fed with the prediction $y[n]$ instead of the reading value $u[n]$, and the sink receiving no reading from the node assumes that the predicted readings are below the error threshold. If the error exceeds e_{max}, the filter switches back to normal mode and reports the readings.

3.1 LMS Limitations

LMS adaptive algorithm is proved to be robust and accurate with a very low computation [9], yet showing features that perfectly fit WSN requirements. However, the choice of the step size and the filter length are essential in the convergence of the algorithm. Starting with a large step size gives a fast convergence of the filter but results in a larger *MSE*, and a too small step size degrades the capabilities of the algorithm. Varying the step size to a smaller value after a certain number of iterations is then beneficial. Concerning the filter length, the choice will indicate the computation load of the algorithm (how many samples we will consider on every iteration). We note that increasing the filter length does not necessarily improve the performance of the filter. Choosing the right parameters is then crucial. Many propositions to adapt and adjust these variables were proposed in literature, but having a direct mathematical analysis of the stability and steady-state performance is a very complicated task in *LMS* [11]. These adaptations may seem adequate for one application and kind of data set, but less efficient to other ones. Adaptive filters perform predictions generally without requiring a priori knowledge about the statistical properties of the phenomenon of interest. But due to the very complex task of selection of the optimal step size, when to increase/decrease it and the optimal filter length, specially in the case of multiple applications running on the network, we propose a modification to *LMS* that consists of collecting the data for every application for a specific time, storing them and performing a simple and straightforward training script to choose the optimal filter parameters for every application.



(a) Generic scheme



(b) Used as a prediction filter

Figure 2: Adaptive filter

4 OUR CONTRIBUTION: LMS_MOD

Our contribution, that we denoted by *LMS_MOD* (for modified), consists of adding another step to the initialization phase for the *LMS* prediction algorithm in [19] by training the filter with enough data. We vary the step size and filter length within specific intervals in order to optimize these values (by minimizing the *MSE*) for every specific application. We start by the upper bound of μ as per equation 6. In order to minimize the *MSE* we compute:

- The appropriate filter length M denoted as i
- The optimal time to switch to a smaller value denoted as j
- The new value of μ denoted as k

After we obtain the three aforementioned values, we execute our prediction algorithm with these parameters for the rest of the data. In this way and since the application data have different characteristics, every application will have distinct parameters achieving a minimal *MSE*. Concerning the energy load resulting from this adaptation, we run our adaptation script offload using numerical simulations in order to obtain the coefficients before running it on a WSN.

5 SIMULATION SETUP

In order to validate our proposition, we use real value traces from the NREL National Wind Technology Center [12] for voltaic cells. We considered the irradiance, temperature, humidity and average wind speed between 04/06/2017 and 06/30/2018 with data collected every minute between 4 am to 8 pm. A description of the traces characteristics is presented in Table 1. It is worth mentioning that each data type has different characteristics and ranges, tehrefore, the prediction task is even more challenging.

5.1 Parameters Determination

We calculate the upper bound of the step size λ_{max} using the first 60 values of $u[n]$, same as the number used to train the filter in the initialization phase. We consider four different thresholds for each data type (note this can be adjusted for specific needs). We consider a one hop communication environment with no loss in order to prove the efficiency of our proposal in an optimal case scenario. We test our algorithm by means of numeric simulation on Matlab. For the adaptation in the initialization phase, we execute a Matlab script for one day of collected data. We varied three parameters i, j and k corresponding to the filter length, the factor by which we will divide the old μ and after how many iterations simultaneously (the time we will switch to the new computed μ value) respectively. We vary i between 1 and 10, and j, k between 1 and 100 with a step of 5, and we choose the value that minimizes the *MSE*. We note that the choice of these intervals can be changed, but we realized after several tests that the optimal values always fall within these ranges. The obtained values are shown in Table 2 and then are use to feed the filter in order to predict the data for the whole previously mentioned duration. We compare *LMS_MOD* to *LMS_VSS* proposed in [20]. *LMS_VSS* respects the prediction phases as in [19] (Initialization, normal and stand-alone) but with a variable step size like our proposition. In [20] μ starts with the value:

$$\mu_{old} = 2\lambda_{max}^{-1}.10^{-2} \tag{7}$$

and switches to a stable value:

$$\mu_{new} = \mu_{old}/M \tag{8}$$

after n iterations, where M is the filter length and n is the number of consecutive readings in stand-alone mode. They chose $n = M^{3/2}$. M was initialized to 4 in [20] and to different values in [19] chosen arbitrarily. For the sake of fairness, we chose the same filter length and λ_{max} for *LMS_VSS* as the one used in *LMS_MOD* for every data set.

Data Type	Max. Value	Min. value	Std. dev.
Irradiance (w/m^2)	$1.4932 * 10^3$	$-2.3744 * 10^3$	360.41
Air Temp. (°F)	88.847	24.318	13.246
Humidity (%)	100	11.52	22.9013
Avg. Wind Speed (MPH)	54.60	0.693	6.518

Table 1: Data Traces Description

Data Type	Threshold	Filter Length	μ Div. Factor	Nbr. of Iter.
Irradiance (w/m^2)	1	1	96	76
	3	1	51	76
	5	1	96	81
	7	1	91	71
Air Temp. (°F)	0.5	4	26	16
	1	4	6	21
	2	4	11	41
	2.5	4	1	1
Humidity %	1	3	96	16
	2	3	36	31
	3	3	41	16
	4	3	16	31
Avg. Wind Speed (MPH)	0.5	2	6	71
	1	2	6	96
	1.5	1	26	96
	2	1	1	1

Table 2: Data Traces Obtained Parameters

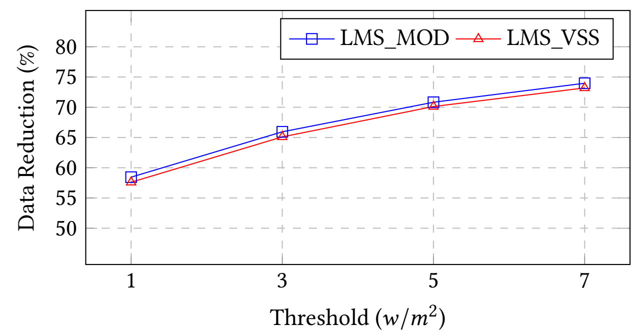


Figure 3: Data reduction for Irradiance

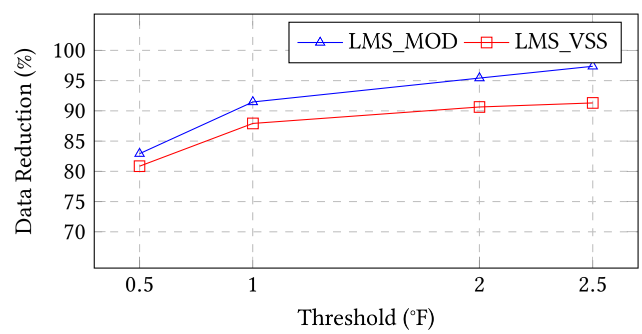


Figure 4: Data reduction for Temperature

6 PERFORMANCE EVALUATION

6.1 Root Mean Square Error

In order to reflect the overall performance of our proposition we compute the Root Mean Square Error (*RMSE*) for every data trace, which corresponds to the root of the MSE that is given by:

$$MSE = \frac{1}{n}\sum_{n}^{i=1}(y[n] - u[n])^2 \tag{9}$$

Where n is the number of samples.

Figures 7 $\longrightarrow$ 10 show the *RMSE* for *LMS_MOD* and *LMS_VSS*

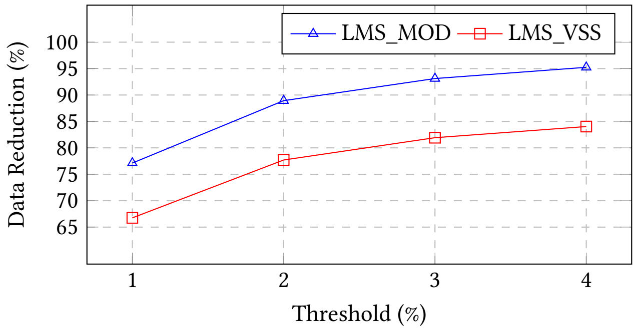


Figure 5: Data reduction for Humidity

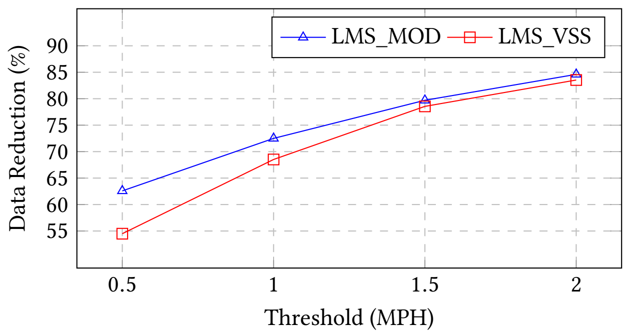


Figure 6: Data reduction for Average Wind Speed

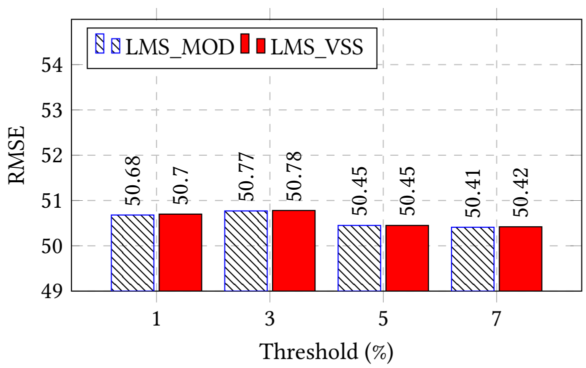


Figure 7: RMSE for Irradiance

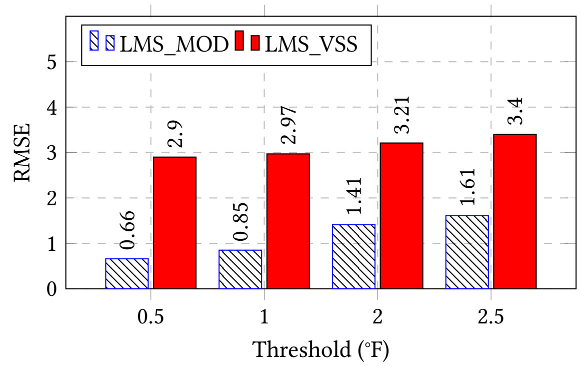


Figure 8: RMSE for Temperature

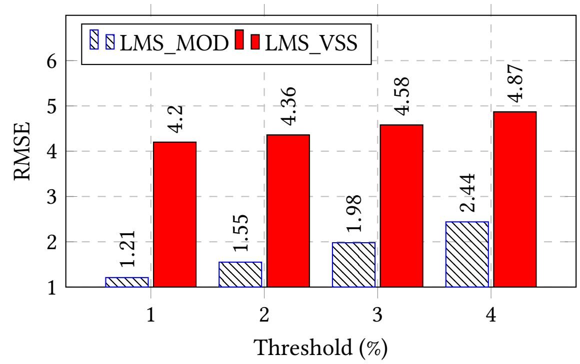


Figure 9: RMSE for Humidity

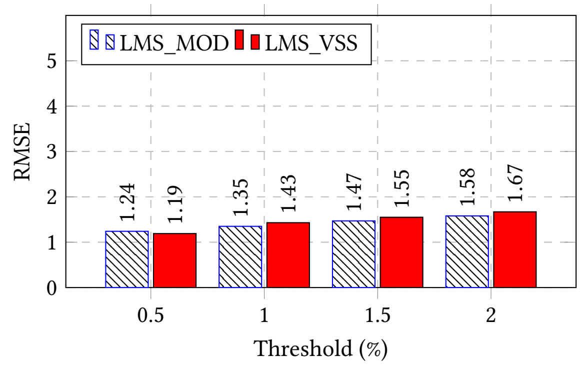


Figure 10: RMSE for Average Wind Speed

for the different data types We observe that for the temperature, humidity and average wind speed *LMS_MOD* has a lower *RMSE* than *LMS_VSS*, this is mainly due to the choice of the parameters (table 2) that minimizes the *MSE*. For the irradiance, the *RMSE* is quite close for *LMS_MOD* and *LMS_VSS* with a slight improvement for *LMS_MOD*. Here, the filter length chosen is equal to one (table 2). This is mostly due to the high deviation of the collected data as we observe from the high value of the standard deviation, 360.41, in table 2. In this case, the step size has a relative small value (the data values for the irradiance have a strong variance between negative and positive values). Then, the dividing factor has less effect on the step size variation. Hence the close values of *RMSE*.

6.2 Data Reduction Percentage

Figures 3 $\longrightarrow$ 6 show the data reduction percentage achieved for both methods. The latter corresponds to the number of predicted packets whose values fall within the range of the chosen threshold. Thus, were not sent to the sink. We can see that our proposition presents higher reduction percentage for the temperature, humidity and average wind speed. Between 2 and 6% for the temperature, between 10 and 12% for the humidity and between 1 and 8% for the average wind speed. Concerning the irradiance, the reduction percentage is again close between *LMS_MOD* and *LMS_VSS* with a slight improvement for *LMS_MOD*. This is due as already mentioned for the similarity of the chosen parameters between the two propositions. It's worth to note, and as already mentioned in section 4 that the existing solutions, and in particularly *LMS_VSS* in this case may perform well in some applications but less efficient for others, which is shown in our results.

7 DISCUSSION

Before coming to our conclusions, we discuss some relevant issues in our proposition. While *LMS_MOD* proved to be efficient for several data types by reducing the *MSE* and ensuring a high data reduction percentage, our straightforward training may misbehave in some conditions, i.e., in environments where the data may become incoherent from one season to another, or when one day of data training is not enough. A possible improvement could be to investigate the variations of every data set (e.g, maximum and minimum values, standard deviation) and train the filter for every data type accordingly by taking these variations into consideration. Moreover, in *LMS_MOD* we optimize the parameters so as we minimize the *MSE*, which might result in a lower data reduction percentage in some cases. Same way if we trained it in the opposite way. Further improvement on how to optimize these two metrics should be studied and considering new metrics as well.

8 CONCLUSION AND FUTURE WORKS

In this paper, we presented a modification of the *LMS* prediction algorithm for WSN to adapt it to different applications with different characteristics as per a SG environment. We trained the filter for one day with the data traces corresponding to each application in order to optimize the parameters that minimize the *MSE*. We tested our approach with real data traces for photo-voltaic cells, and performed simulations considering one hop communication networks. Our numerical results show a better performance than *LMS_VSS* in terms of *RMSE* and percentage of data economy. As future work, we will continue investigating our approach with more tests on different data sets and considering more metrics. Later on, we will implement our algorithm on a WSN testbed [1] to evaluate its performance in a real scenario with interference and losses.

ACKNOWLEDGMENTS

This work was partially funded by a grant from the MEL (Métropole Européenne de Lille), the SoMel SoConnected project (Ademe, PIA2), HEI Yncréa Haut-de-France, Collège Doctoral Lille Nord de France and CPER Data. Part of this work was performed when Jad Nassar was a visiting student at the Metropolitan Autonomous University, Mexico.

REFERENCES

[1] Cedric Adjih, Emmanuel Baccelli, Eric Fleury, Gaetan Harter, Nathalie Mitton, Thomas Noel, Roger Pissard-Gibollet, Frederic Saint-Marcel, Guillaume Schreiner, Julien Vandaele, et al. 2015. FIT IoT-LAB: A large scale open experimental IoT testbed. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE, 459–464.

[2] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. 2009. Energy conservation in wireless sensor networks: A survey. *Ad hoc networks* 7, 3 (2009), 537–568.

[3] Siddhartha Bhandari, Neil Bergmann, Raja Jurdak, and Branislav Kusy. 2017. Time Series Data Analysis of Wireless Sensor Network Measurements of Temperature. *Sensors* 17, 6 (2017), 1221.

[4] Dariusz Bismor, Krzysztof Czyz, and Zbigniew Ogonowski. 2016. Review and comparison of variable step-size LMS algorithms. *International Journal of Acoustics and Vibration* 21, 1 (2016), 24–39.

[5] Hans-Jürgen Butterweck. 1995. A steady-state analysis of the LMS adaptive algorithm without use of the independence assumption. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 2. IEEE, 1404–1407.

[6] Jyoti Dhiman, Shadab Ahmad, and Kuldeep Gulia. 2013. Comparison between Adaptive filter Algorithms (LMS, NLMS and RLS). *International Journal of Science, Engineering and Technology Research (IJSETR)* 2, 5 (2013), 1100–1103.

[7] Gabriel Martins Dias, Boris Bellalta, and Simon Oechsner. 2016. A survey about prediction-based data reduction in wireless sensor networks. *ACM Computing Surveys (CSUR)* 49, 3 (2016), 58.

[8] Michel Grabisch, Jean-Luc Marichal, Radko Mesiar, and Endre Pap. 2011. Aggregation functions: means. *Information Sciences* 181, 1 (2011), 1–22.

[9] Babak Hassibi, Ali H Sayed, and Thomas Kailath. 1996. H-infinity optimality of the LMS algorithm. *IEEE Transactions on Signal Processing* 44, 2 (1996), 267–280.

[10] S.S. Haykin. 2002. *Adaptive Filter Theory*. Prentice Hall. https://books.google.fr/books?id=eMcZAQAAIAAJ

[11] Simon Haykin and Bernard Widrow. 2003. *Least-mean-square adaptive filters*. Vol. 31. John Wiley & Sons.

[12] David Jager and Afshin Andreas. 1996. *NREL National Wind Technology Center (NWTC): M2 Tower; Boulder, Colorado (Data)*. Technical Report DA-5500-56489. National Renewable Energy Lab.(NREL), Golden, CO (United States). http://dx.doi.org/10.5439/1052222

[13] Yuzhong Jiao, Rex YP Cheung, Winnie WY Chow, and Mark PC Mok. 2013. A novel gradient adaptive step size LMS algorithm with dual adaptive filters. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*. IEEE, 4803–4806.

[14] Karen Miranda and Victor Ramos. 2016. Improving data aggregation in Wireless Sensor Networks with time series estimation. *IEEE Latin America Transactions* 14, 5 (May 2016), 2425–2432.

[15] Karen Miranda, Tahiry Razafindralambo, and Victor Ramos. 2013. Using efficiently autoregressive estimation in wireless sensor networks. In *International Conference on Computer, Information and Telecommunication Systems (CITS)*. IEEE, 1–5.

[16] Jad Nassar, Nicolas Gouvy, and Nathalie Mitton. 2017. Towards Multi-instances QoS Efficient RPL for Smart Grids. In *Fourteenth ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN 2017)*.

[17] Usman Raza, Alessandro Camerra, Amy L Murphy, Themis Palpanas, and Gian Pietro Picco. 2012. What does model-driven data acquisition really achieve in wireless sensor networks?. In *International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 85–94.

[18] Mouna Rekik. 2016. *Routage géographique multi-chemin basé sur l'intelligence d'essaim pour réseaux de capteurs et d'actionneurs sans fil : application aux Smart Grids*. Ph.D. Dissertation. Univ. Lille 1.

[19] Silvia Santini and Kay Romer. 2006. An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proceedings of the 3rd international conference on networked sensing systems (INSS)*. 29–36.

[20] Biljana Stojkoska, Dimitar Solev, and Danco Davcev. 2011. Data prediction in WSN using variable step size LMS algorithm. In *Proceedings of the 5th International Conference on Sensor Technologies and Applications*.

[21] Peijie Wang, Pooi Yuen Kam, and Meng Wah Chia. 2009. A novel automatic step-size adjustment approach in the LMS algorithm. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*. IEEE, 867–871.