Missing Data Estimation for Energy Resources Management in Tertiary Buildings

Antoine Garnier, Julien Eynard, Matthieu Caussanel, Stéphane Grieu

Missing data estimation for energy resources management in tertiary buildings

Antoine Garnier*†, Julien Eynard†, Matthieu Caussanel†, Stéphane Grieu†
*Pyrescom, Mas des Tilleuls, 66680 Canohès, France
Email: a.garnier@pyres.com
†PROMES-CNRS, Rambla de la thermodynamique, Tecnosud, 66100 Perpignan, France
Email: julien.eynard@univ-perp.fr, matthieu.caussanel@univ-perp.fr, grieu@univ-perp.fr

***Abstract*—The BATNRJ project, managed by Pyrescom, focuses on improving energy efficiency while preserving comfort in tertiary buildings. To this end, an open and cost-friendly monitoring solution based on instrumentation as well as analysis and control tools is being developed. Solar radiation and indoor temperature being key parameters, the present paper deals with estimating missing data in case of sensor failures. First, solar radiation is interpolated using as a basis the Gaussian or the Cosine function. Mean relative error is about 10%. Then, based on the concept of time series, feedforward artificial neural networks are used to estimate up to the next 24 hours missing data about indoor temperature. We obtained accurate results, especially for failures limited to 3 hours. The mean relative error does not exceed 6%, even in case of long sensor failures.**

I. Introduction

With the aim of optimizing energy management in tertiary buildings, the Pyrescom Company (www.pyres.com) has developed a monitoring system that carries out meteorological parameters and energy measurements. These data are used to find out possible ways to improve energy efficiency in such buildings and to develop an efficient HVAC (Heating, Ventilation and Air-Conditioning) control strategy. In case of sensor failures, measurements cannot be performed and/or sent to the storing database. These failures have several starting points. First, some sensors operate on batteries and these are not replaced in time despite proper warnings. Secondly, failures may be caused by the impossibility of sending the data or more rarely by a sensor malfunction. Because the development of an efficient energy management system requires real-time data, we propose tools for missing data estimation. Two parameter measurements require missing data assessment: solar radiation and indoor temperature. Solar radiation is related to the potential of the instrumented area for a future solar installation while temperature is linked to both the HVAC system control and thermal comfort.

Since 2010, many different sensors have been installed by the Pyrescom Company on several pilot sites, including about 120 temperature and many solar radiation sensors. After having analyzed every error that occurred on the different sensors (Figure 1), it has been noticed that about 2% of the measurements are actually missing. The duration of these failures varies, and its repartition is similar for each kind of sensor: 70% of failures last less than 3 hours and they rarely exceed 24 hours. In average, a sensor fails about

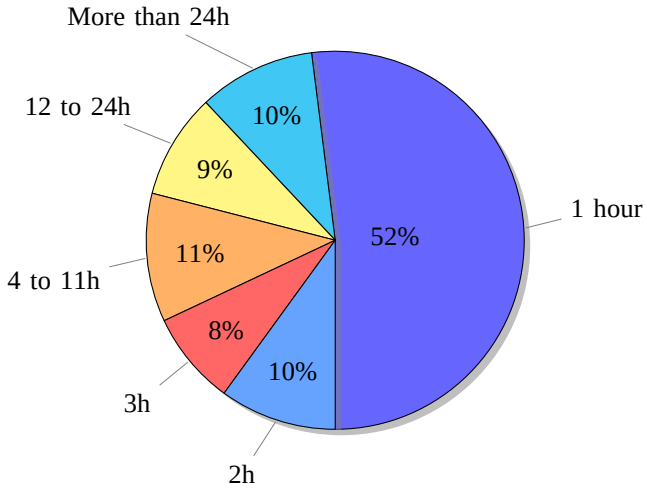

Fig. 1. Failure percent according to their duration in hours

12 hours per month. From this observation, our goal was to develop a correction system intended to estimate the missing data on a 24 hours time scale with an optimized estimation between 1 and 3 hours. Specially developed algorithms must be integrated into an existing server with a minimal calculation time in order to avoid penalizing the other installed applications.

In the literature, several methods have been developed to perform solar radiation or temperature estimation. Paulescu's work [1] is based on the Angstrom's equation that uses outdoor temperature instead of solar radiation duration. Dimas [2] first determined from outdoor humidity an atmospheric transmittance coefficient. With this coefficient and the temperature, he developed a statistical method to assess the solar radiation amplitude. Ruano [3] and Thomas [4] built an indoor temperature prediction algorithm based on artificial neural networks. This algorithm uses several outdoor parameters such as temperature, humidity, and solar radiation to quantify the indoor warming. These models can be used to estimate missing measurements of a failing sensor from data acquired by other working sensors of the monitoring system. In 1996, Khotanzad *et al.* [5] used artificial neural networks to forecast hourly temperature values for the next

seven days, using daily high and low temperatures and an adaptative daily update of the weights. In 2000, Chen and Hwang [6] used fuzzy time series to deal with temperature forecasting problems and overcome the drawback related to historical data represented by linguistic values. They proposed a new fuzzy time series model, called two-factor time-variant fuzzy time series model. Based on the proposed model, they developed two algorithms for temperature prediction and obtained good forecasting results. In 2002, Tassadduq *et al.* [7] used artificial neural networks to forecast the temperature at a given hour of the next day, only using the temperature at the same hour of the present day. In 2011, Eynard *et al.* [8] used a wavelet-based multi-resolution analysis as well as artificial neural networks to forecast outdoor temperature. The discrete wavelet transform allowed decomposing sequences of past data in subsequences according to different frequency domains while preserving their temporal characteristics. From these coefficients, artificial neural networks were used to estimate future subsequences. Future values of outdoor temperature were obtained by simply summing up the estimated coefficients.

In this paper, when a sensor fails, we use data anterior and/or posterior to the failure in order to rebuild the missing information. Two different approaches are presented : an interpolation technique is used for the estimation of missing solar radiation data whereas an extrapolation technique based on artificial intelligence deals with indoor temperature estimation. Using a feedforward neural network (a multi-layer Perceptron), we developed an estimation tool based on the concept of time series. With this concept, missing values about indoor temperature are estimated using observed values (data provided by the considered sensor just before its failure) only. In this paper, we focus on the developement phase of the tool. As a result, we highlight an optimal configuration based on the number of hidden neurons (related to the number of parameters of the developped model), the estimation support size and the number of examples used to train the network. The proposed tools will be implemented in embedded systems with low energy consumption requirement to manage energy resources in buildings.

II. Solar radiation estimation

This section of the paper deals with estimating missing solar radiation data. Since the latter has a lower priority than temperature data, its correction is simply done at the end of the day using all valid measurements available. Data acquisition takes place every 30 seconds and is then averaged over a period of 60 minutes. Equivalent to low-pass filtering the data, averaging decreases the influence of rapid clouds. However, data can be strongly altered by clouds staying in front of the sun for a long period of time.

In order to check the model against the different meteorological conditions, solar radiation data set has been divided in three categories depending on a subjectively defined perturbation level :

- Low: no significant impact of clouds,
- Medium: some radiation values are strongly attenuated,
- High: very cloudy day with a significant impact on solar radiation.

TABLE I
Occurence percent according to the perturbation level

Perturbation level	Occurrence percentage
Low	50%
Medium	30%
High	20%

Table I shows the repartition of days according to the perturbation level. Half of the measurements shows no trace of cloud perturbation and 80% allows to clearly observe a typical solar radiation bell curve. As a consequence, the approximation is achievable in the majority of cases.

A. Approach

An appropriate equation with a reasonnable number of parameters is needed to fit the bell shape with a non-linear regression. First, several distribution equations have been tested: Lorentzian, Lognormal and Gaussian ; the latter gave the best results. The equation of a Gaussian curve can be written as follows :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} * exp^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma^2}\right)^2} \quad (1)$$

We rewrite (1) to consider a potential offset (A_4) and to emphasize the fit parameters A_1, A_2, A_3, and A_4. These parameters are gathered in the vector $A = [A_1, A_2, A_3, A_4]$.

$$p(t, A) = A_1 * exp^{-\left(\frac{t-A_2}{A_3}\right)^2} + A_4 \quad (2)$$

A second solution is to use as template a sinusoidal function, based on the theoretical solar radiation equation [9]:

$$G_0 = \varepsilon_0 * (1 + 0.0334 * cos(w * (j - 2))) * sin(h) \quad (3)$$

G_0 : global horizontal of solar radiation on the upper limit of the atmosphere;
w : angular speed of sun above skyline;
j : day of the year;
h : solar height;
ε_0 : solar constant.

As for the Gaussian, we rewrite equation (3) by adding an offset and emphasizing fit parameters:

$$p(t, A) = A_1 * cos(A_3(t - A_2)) + A_4 \quad (4)$$

To interpolate missing values, we approach the curve of both functions by a non-linear least-square criterion [10]. This standard method is not detailed here. The partial derivatives equations required to complete the Jacobian matrix are specific

to our application and given below for both the Gaussian and Cosine fits.

$$\begin{cases} \frac{\partial r_i}{\partial A_1} = & -exp^{-\left(\frac{t_i-A_2}{A_3}\right)^2} \\ \frac{\partial r_i}{\partial A_2} = & -2\frac{A_1(t_i-A_2)}{A_3^2}exp^{-\left(\frac{t_i-A_2}{A_3}\right)^2} \\ \frac{\partial r_i}{\partial A_3} = & -2\frac{A_1(t_i-A_2)^2}{A_3^3}exp^{-\left(\frac{t_i-A_2}{A_3}\right)^2} \\ \frac{\partial r_i}{\partial A_4} = & -1 \end{cases} \quad (5)$$

$$\begin{cases} \frac{\partial r_i}{\partial A_1} = & -cos(A_3(t_i-A_2)) \\ \frac{\partial r_i}{\partial A_2} = & -A_1A_3 * sin(A_3(t_i-A_2)) \\ \frac{\partial r_i}{\partial A_3} = & -A_1(t_i-A_2) * sin(A_3(t_i-A_2)) \\ \frac{\partial r_i}{\partial A_4} = & -1 \end{cases} \quad (6)$$

An intial guess for the A vector is necessary for the first iteration. $A = [500, 12, 3, 10]$ has been chosen for the Gaussian algorithm and $A = [500, 12, 0.6, 10]$ for the Cosine algorithm. These coefficients define curves whose shapes are close to daily measurements. This minimizes the divergence risk, accelerates calculation convergence, and reduces the required computation time.

Using 4 coefficients requires a minimum of 4 measures to perform the fit. If too many values are missing, the least square algorithm cannot be executed. We assume that the solar radiation values are similar to those of the previous day and we use the previous day computed coefficients for the current day. To judge approximation's quality, mean relative error ε is computed by equation (7) once the algorithm is complete:

$$\varepsilon = \frac{1}{n}\sum_{i=0}^{n-1}\left|\frac{p_i - \hat{p_i}}{p_i}\right| \quad (7)$$

In (7), $\hat{p_i}$ is the solar radiation estimated value at the time t_i. Since values during the night are irrelevant, they are simply ignored. Day/night separation is defined by a threshold at $25W/m^2$, which eliminates low values of the regression.

B. Missing values estimation

The accuracy of both approaches (Gaussian and Cosine) has been compared for each of the three perturbation levels previously defined. Data used to perform the study have been measured from February 2010 to December 2011. To carry out consistent tests, daily failures have been simulated in the following manner. First, a one hour long failure is randomly introduced during the day. Then, it is successively increased by one hour steps up to a total duration of 24 hours. Two quality indicators are computed. The first one is a mean relative error ε that is determined only for interpolated values. The second one is a weighted mean relative error indicator ε_w that corresponds to ε for each failure duration. ε_w is calculated with equation (8), in which w_h is the failure percentage during h hours ($\sum_{h=1}^{24} w_h = 1$) and ε_h the error associated to this same duration:

$$\varepsilon_w = \sum_{h=1}^{24} w_h * \varepsilon_h \quad (8)$$

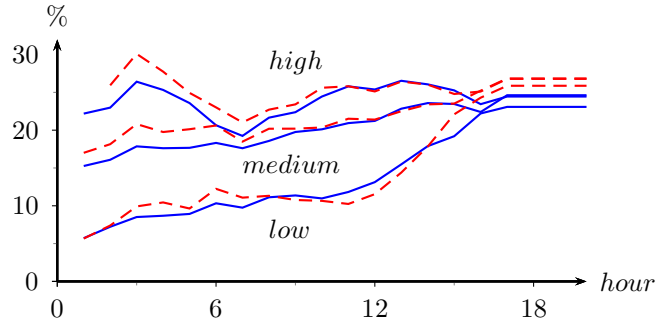

Fig. 2. Mean relative error, according to the duration of the failures, for each level of perturbation (low, medium, and high) and both proposed approaches: Gaussian (blue solid line) and Cosine (red dashed line)

C. Results

In a regression using all available measurements, the Cosine approach turns out to be a little more accurate than the Gaussian approach, assuming low or no cloud perturbation. In the other cases, the Gaussian approach is always more efficient with a reduced mean relative error of about 2 to 3%.

The same trend is observed when simulating missing values with the Gaussian approach being the most accurate (Figure 2). The mean relative error increases slightly as the duration of the failures increases from 1 to 12 hours: the algorithm can correctly estimate missing values. Beyond this threshold, the error tends towards 25%: the number of days with more than 4 valid measurements during daytime decreases, and the algorithm shall use the past day values to carry out the estimation. This method should be improved.

Table II summarizes performance for all type of perturbations, with the weighted mean relative error (8).

TABLE II
WEIGHTED MEAN RELATIVE ERROR ACCORDING TO THE PERTURBATION LEVEL

Perturbation Level	Gaussian fit error	Cosine fit error
Low	8.00%	8.33%
Medium	16.77%	18.58%
High	23.07%	26.22%

Respectively classified in 'low' and 'medium' perturbation levels, figures 3 and 4 show a fit of data from which values have been deliberately removed from 12 to 15 PM. For the day with low perturbation level (July 29^{th}, 2011), both approaches give similar results. For the second day (August 3^{rd}, 2011), the Gaussian approach is 2% better than the Cosine approach, whereas data are relatively spread.

Results are satisfying and the approach by the Gaussian algorithm will be preferred for its greater accuracy. Moreover using least square criterion allows to work with data sets presenting irregular steptimes due to missing points.

III. Indoor temperature estimation

Another tool is proposed to estimate missing indoor temperature data. To this end, a real-time estimation algorithm based on the concept of time series has been developed. A time series

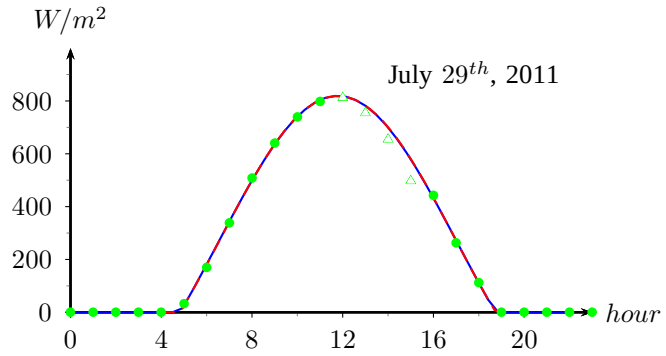

Fig. 3. Fit of incomplete solar radiation data (low perturbation). Green circles: Data used for fitting; Green triangles: Missing data; Blue solid line: Gaussian approach with ε = 7.89%; Red dashed line: Cosine approach with ε = 7.01%

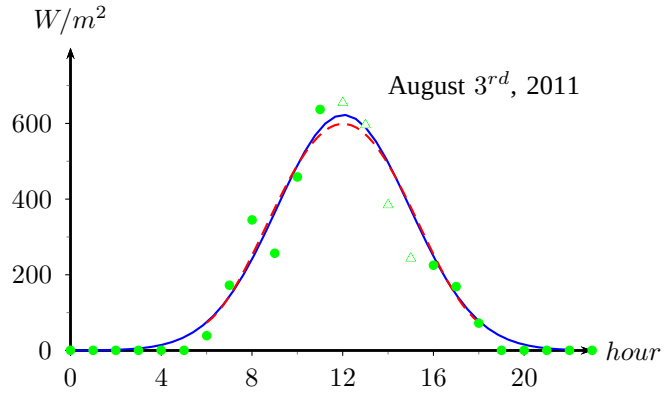

Fig. 4. Fit of incomplete solar radiation data (medium perturbation). Green circles: Data used for fitting; Green triangles: Missing data; Blue solid line: Gaussian approach with ε = 22.03%; Red dashed line: Cosine approach with ε = 24.00%

is a sequence of data points, measured typically at successive time instants and spaced at uniform time intervals. Time series estimation deals with the development of a model to estimate future values using observed values only (i.e. data provided by a given sensor before its failure). With an interpolation algorithm, the faulty sensor has to be working again to enable missing data estimation. In addition, the dysfunction period of time has to be short. Time series estimation allows flexibility.

A. Approach

Feedforward artificial neural networks are widely used in time series estimation [11][12][13]. That is why we decided for a Multi-Layer Perceptron (MLP) neural network to estimate missing indoor temperature data. Such a network is capable of approximating any nonlinear continuous function to any desired degree of accuracy with one hidden layer and an appropriate number of neurons. A too small (resp. large) number of hidden neurons leads to under-parametrized (resp. over-parametrized) models. In this case, approximation is bad what leads to a poor generalization ability. Hidden neurons use non-linear activation functions while output neurons use linear functions. With this topoloy, one can approximate any nonlinear function to any desired degree of accuracy if the number of hidden neurons is sufficiant. Feedforward neural networks are universal approximators [14].

We used the "GNU Octave" software [15] as well as the *nnet* package [16] to develop the model (Figure 5). First, the network has to be trained using examples to understand the dynamics of the indoor temperature measured by the considered sensor. The training phase allows the network parameters to be identified. The Levenberg-Marquardt algorithm [17][18] was used. It is a standard technique to solve non-linear least-square problems. This algorithm is a combination of two minimization approaches: the gradient descent method and the Gauss-Newton method. In the gradient descent method, the sum of the squared errors is reduced by updating the parameters (the synaptic weights and the biases when using artificial neural networks) in the direction of the greatest reduction of the least squares objective. In the Gauss-Newton method, the sum of the squared errors is reduced by assuming that the least squares function is locally quadratic and finding the minimum of the quadratic. So, the Levenberg-Marquardt method acts like a gradient-descent method when the parameters are far from their optimal values and like the Gauss-Newton method when these parameters are close to these values.

The validation phase allows highlighting the right topology of the network. As previously mentioned in the paper, failure duration is less than 3 hours in 70% of cases and less than 24 hours in 90%. That is why we set the estimation horizon to 24 hours. Moreover, a key point in time series estimation is to work out the observed values to be taken into account (named "estimation support").

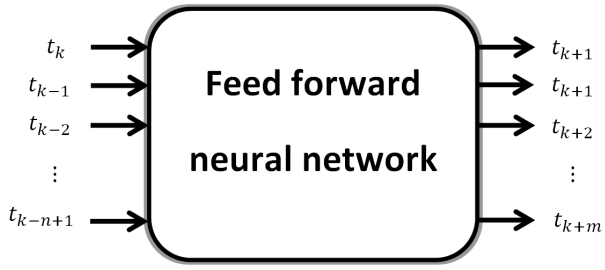

Fig. 5. Time series estimation using a feedforward neural network. k is the time index, n is the estimation support size and m is the horizon

B. Training and validation phases

A sliding window mechanism was used to obtain training and validation subsets. Examples are defined by an estimation support (the considered observed values) and an horizon (as previously mentioned, set to 24 hours). To favour efficiency during the learning process of the artificial neural network used, the training and validation examples were normalized between -1 and 1. The error goal and the maximum training epochs were set to 10% and 100, respectively. In this paper, we discuss about training examples and observed values. An observed value is only suitable when correlation with missing data is high. In addition, examples have to cover most of the possible cases the network will have to handle. As a

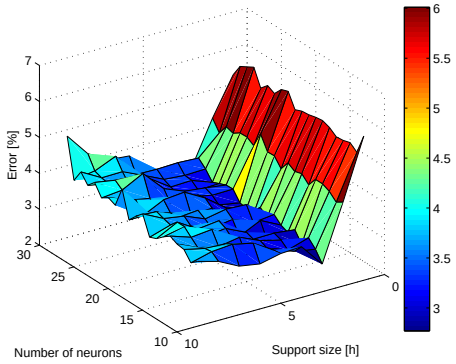

Fig. 6. Weighted mean relative error according to the number of hidden neurons and the estimation support size (100 training examples used to develop the model)

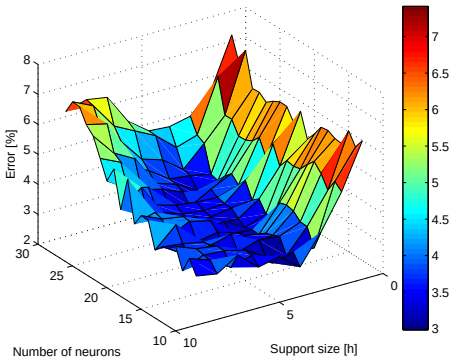

Fig. 8. Weighted mean relative error according to the number of hidden neurons and the estimation support size (300 training examples used to develop the model)

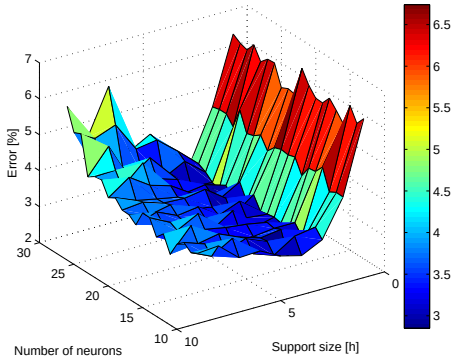

Fig. 7. Weighted mean relative error according to the number of hidden neurons and the estimation support size (200 training examples used to develop the model)

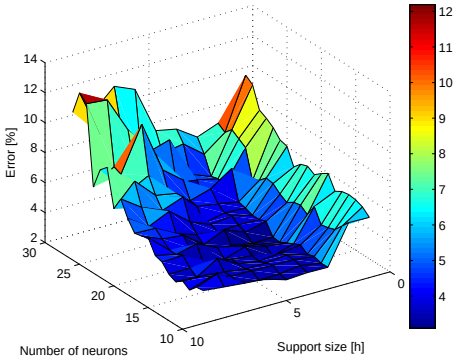

Fig. 9. Weighted mean relative error according to the number of hidden neurons and the estimation support size (400 training examples used to develop the model)

consequence, indoor temperature examples covering a whole year are mandatory to achieve a good training of the network. The validation phase allows checking if the trained network generalizes well and its ability to use learned information in new or partially unknown situations.

C. Optimal configuration and results

We considered training sets of 100, 200, 300 and 400 examples, respectively. Examples were picked randomly from indoor temperature data we collected during a period of height months (January to August, 2011). The number of hidden neurons and the estimation support size range from 10 to 30 and from 1 to 10 hours, respectively. To avoid non-significant results due to a bad convergence of the Levenberg-Marquardt algorithm or related to parameters initialization, the network was trained ten times for a given configuration. The models were validated using indoor temperature data from September 2011 to July 2012. Figures 6 to 9 depict the way the weighted (by the duration of the failures) mean relative error evolves, according to the just-mentioned parameters (the number of hidden neurons and the estimation support size), for the four training sets we considered.

First, the number of examples used to train the network has to be chosen carefully. With 100 examples, a large number of configurations provide good temperature estimations. When adding examples to the training subset, good configurations are less numerous. Overfitting seems to be more frequent. When using more than 200 examples to train the network, additional data do not provide pertinent information about indoor temperature behaviour. As a result, for most of the configurations, the weighted mean relative error increases. In addition, using more than three hours as estimation support is not recommended, because of low correlation levels between the observed data we consider and the (missing) data to be estimated. Generalization ability is clearly impacted by the estimation support size. Moreover, taking a look at the way the network's topoloy impacts on accuracy, one can highlight that 15 to 20 neurons are needed for a good estimation of missing data. More than 20 neurons leads to overfitting. So, the optimal configuration we obtained is the following one:

- One hidden layer with 15 to 20 neurons,
- An estimation support of 3 hours,

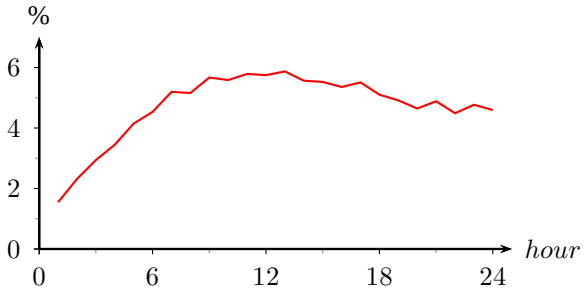

Fig. 10. Mean relative error, according to the estimation horizon, using the optimal network (indoor temperature)

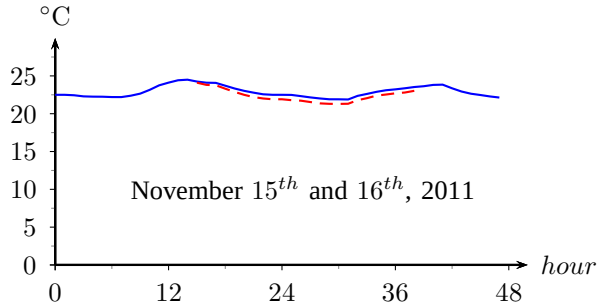

Fig. 11. Estimated indoor temperatures for a 24-hour failure, starting at 3 PM. Blue solid line: Measured values; Red dashed line: Estimated values

- 200 examples to train the feedforward network.

Figure 10 depicts the mean relative error we obtained during the validation phase of the optimal network (using September 2011 to July 2012 data), according to the estimation horizon. With an horizon shorter than 4 hours, the mean relative error is lower than 4%. This error reaches 6% with an horizon of 12 hours then deacreases slowly up to 5% with an horizon of 24 hours. Figure 11 shows estimated indoor temperatures for a 24-hour failure, starting at 3 PM (November 15^{th} and 16^{th}, 2011). The developed algorithm tends to underestimate the missing values of some tenth of degrees but accuracy is sufficient to manage energy resources in tertiary buildings efficiently, in case of sensor failure.

IV. Conclusion

The present paper deals with the development of estimation tools in case of missing data due to failures of sensors implemented in tertiary buildings. Solar radiation and indoor temperature are the two considered parameters. First, solar radiation missing data have been efficiently estimated using the Gaussian or the Cosine function. In case of sensor failures limited to a few hours only, mean relative error ranges between 3 and 8%. As a key point, the proposed algorithm needs at least 4 or 5 valid values of solar radiation in a day to preserve accuracy. As expected, and because of difficulties in modelling cloud phenomena, error increases for very cloudy days.

In a second time, based on the concept of time series, a feedforward artificial neural network is used to estimate missing data about indoor temperature up to the next 24 hours. A parametric study has been carried out to find the right topology of the network, an adequate number of training examples and, finally, the observed values to be taken into account. The results we obtained are very satisfactory. Using the proposed approach, the mean relative error observed does not exceed 6%, even in case of long sensor failures.

These estimation tools will be now implemented in a commercial embedded system with low energy consumption requirement, developed by Pyrescom. This open and cost-friendly monitoring solution will be used in tertiary buildings to control HVAC (Heating, Ventilation and Air-Conditioning) systems and manage energy resources while ensuring thermal comfort.

Acknowledgements

The authors want to thank the Pyrescom Company (www.pyres.com) for its financial support as well as Jean-Michel Cabanat and Cédric Calmon for their help.

References

[1] M. Paulescu, L. Fara, and E. Tulcan-Paulescu. Models for obtaining daily global solar irradiation from air temperature data. *Atmospheric Research*, 79:227 – 240, 2006.

[2] F. A. Dimas, S.I. Gilani, and M.S. Aris. hourly solar radiation estimation from limited meteorological data to complete missing solar radiation data. In *2011 international Conference on Environment Science and Engineering*, 2011.

[3] A.E. Ruano, E.M. Crispim, E.Z.E. Conceicao, and M.M.J.R. Lucio. Prediction of building's temperature using neural networks models. *Energy and Buildings*, 38(6):682 – 694, 2006.

[4] B. Thomas and S.M. Mohsen. Artificial neural network models for indoor temperature prediction: investigations in two buildings. *Neural Comput. Appl.*, 16:81–89, 2007.

[5] A. Khotanzad, M. Davis, A. Abaye, and D. Maratukulam. An artificial neural network hourly temperature forecaster with applications in load forecasting. *IEEE Transactions on Power Systems*, 11(2):870–876, 1996.

[6] S.M. Chen and J.R. Hwang. Temperature prediction using fuzzy time series. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cynernetics*, 30(2):263–275, 2000.

[7] I. Tassadduq, S. Rehman, and K. Bubshait. Application of neural networks for the prediction of hourly mean surface temperatures in saudi arabia. *Renewable Energy*, 25:545–554, 2002.

[8] J. Eynard, S. Grieu, and M. Polit. Wavelet-based multi-resolution analysis and artificial neural networks for forecasting temperature and thermal power consumption. *Engineering Applications of Artificial Intelligence*, 24(3):501 – 516, 2011.

[9] D. Delaunay and C. Sacre. *Années-types de données météorologiques pour 10 stations météorologiques*. CSTB, 1998.

[10] A. Bjõrck. *Numerical Methods for Least Squares Problems*. 1999.

[11] Z. Tang, C. de Almeida, and P.A. Fishwick. Time series forecasting using neural networks vs. box jenkins methodology. *Transactions of The Society for Modeling and Simulation International*, 57:303–310, 1991.

[12] K. Chakraborty, K. Mehrotra, C.K. Mohan, and S. Ranka. Forecasting the behavior of multivariate time series using neural networks. *Neural Networks*, 5(6):961 – 970, 1992.

[13] J.-J Kao. Forecasts using neural network versus box-jenkins methodology for ambient air quality monitoring data. *Journal of the Air & Waste Management Association*, 50:219–226, 2000.

[14] K. Hornik, M. Stinchcombe, and H. White. Multi-layer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[15] J.W. Eaton, D. Bateman, and S. Hauberg. *GNU Octave Manual Version 3*. 2008.

[16] M. D. Schmid. A neural network package for octave, 2009.

[17] C. Charalambous. Conjugate gradient algorithm for efficient training of artificial neural networks. *Circuits, Devices and Systems, IEE Proceedings G*, 139(3):301 –310, jun 1992.

[18] M.T. Hagan and M.B. Menhaj. Training feedforward networks with the marquardt algorithm. *Neural Networks, IEEE Transactions on*, 5(6):989 –993, nov 1994.