Energy and Buildings 83 (2014) 162–172

Contents lists available at ScienceDirect

Energy and Buildings

journal homepage: www.elsevier.com/locate/enbuild

On-line learning of indoor temperature forecasting models towards energy efficiency

F. Zamora-Martínez*, P. Romeu, P. Botella-Rocamora, J. Pardo

Embedded Systems and Artificial Intelligence Group, Escuela Superior de Enseñanzas Técnicas, Universidad CEU Cardenal Herrera, C/ San Bartolomé, 46115 Alfara del Patriarca, Valencia, Spain

ARTICLE INFO

Article history:
Available online 28 April 2014

Keywords:
Energy efficiency
Time series forecasting
Bayesian models
Gradient descent
Artificial neural networks

ABSTRACT

The SMLsystem is a house built at the Universidad CEU Cardenal Herrera (CEU-UCH) to participate in the Solar Decathlon 2012 competition. Several technologies have been integrated to reduce power consumption. A predictive module, based on artificial neural networks (ANNs), has been developed using data acquired in Valencia. The module produces short-term forecast of indoor temperature, using as input data captured by a complex monitoring system. The system expects to reduce the power consumption related to Heating, Ventilation and Air Conditioning (HVAC) system, due to the following assumptions: the high power consumption for which HVAC is responsible (53.9% of the overall consumption); and the energy needed to maintain temperature is less than the energy required to lower/increase it. This paper studies the development viability of predictive systems for a totally unknown environment applying on-line learning techniques. The model parameters are estimated starting from a totally random model or from an unbiased a priori knowledge. These forecasting measures could allow the house to adapt itself to future temperature conditions by using home automation in an energy-efficient manner. Experimental results show reasonable forecasting accuracy with simple models, and in relatively short training time (4–5 days).

1. Introduction

The School of Technical Sciences at the Universidad CEU-UCH constructed a solar-powered house, known as Small Medium Large System (SMLsystem), to participate in the 2012 Solar Decathlon Europe competition [1]. It is a house that integrates a whole range of different technologies to improve energy efficiency. The objective was to construct a near zero-energy house. Solar Decathlon Europe is an international competition among Universities that promotes research in the development of energy-efficient houses. The aim of the participating teams is to design and build houses that consume as few natural resources as possible while minimize the production of waste products during their lifecycle. Special emphasis is placed on reducing energy consumption and obtaining all the needed energy from the sun.

Thus, the status quo of the competition is about energy efficiency. The main reason is that nowadays, in European Union (EU), primary energy consumption in buildings represents about 40% of the total [2]. Recent studies indicate similar values worldwide; energy spent in buildings also represents a 40% rate of total energy consumed, where more than a half is used by Heating, Ventilation and Air Conditioning (HVAC) systems [3]. Additionally, the Spanish Institute for Diversification and Saving of Energy (IDAE) [4] of the Spanish Government reported that at the Spanish households it is consumed a 30% of total energy expenditure of the country. Such figures obligate the governments to think about how improve the energy consumption for the future. Energy is a scarce resource in nature, which has an important cost, which is finite and must be shared. It is becoming a precious asset of incalculable value that, converted into electricity, heat or fuel, makes easier and more comfortable the everyday life of people. Moreover, it is also a key factor to make feasible the progress of the industry and businesses.

Nevertheless, depending on the type of building, location and other factors, HVAC systems could represent up to 40% of the total energy consumption of a building [2,3]. The activation/deactivation of such systems depends on the comfort parameters that have been established, being one of the most important the indoor temperature, directly related with the notion of comfort. Several authors have been working on that idea; in [2] an excellent state of art can be found.

* Corresponding author. Tel.: +34 96 1369000x2361.
E-mail addresses: francisco.zamora@uch.ceu.es (F. Zamora-Martínez), pablo.romeu@uch.ceu.es (P. Romeu), pbotella@uch.ceu.es (P. Botella-Rocamora), juan.pardo@uch.ceu.es (J. Pardo).

http://dx.doi.org/10.1016/j.enbuild.2014.04.034
0378-7788/

Hence, it is mandatory or at least advisable to invest in the research of designing and implementing new systems at home, which should be able to produce and use energy efficiently and wisely, reaching a balance between consumption and streamlined comfort. A person could much easier realize its activities if its comfort is ensured and there are no negative factors (e.g. cold, heat, low light, noise, low air quality, etc.) to disturb him. With the evolution of technology new parameters or variables have become more controllable and the requirements for the people's comfort level have increased.

Systems that let us monitor and control such aspects make necessary to talk about the concept of Ambient Intelligence (AmI). This can be found in [5–7] and also by the ISTAG, European Commissions IST Advisory Group in [8]. AmI emphasizes the human viewpoint: i.e. user-friendliness, efficient service support, user-empowerment and support for human interactions. People, under the ambient intelligence, are surrounded by intelligent intuitive interfaces embedded in the environment and in different objects [8]. Thus, the environment is capable of recognizing and responding to the presence of people in a seamless, unobtrusive and often invisible way. The concept of a smart or intelligent environment emphasizes technical solutions such as pervasive or mobile computing, sensor networks, artificial intelligence, robotics, multimedia computing, middleware solutions, and agent-based software [9]. The result is a system that shows an active behavior (intelligent), anticipating possible solutions adapted to the context in which such system is located.

The term of *Home Automation* (HA) can be defined, as it is mentioned in [10], as the set of services provided by integrated technology systems to meet basic needs of security, communication, energy management and comfort of a person and its immediate environment. Thus, home automation can be understood as the discipline that studies the development of intelligent infrastructures and information technologies in buildings. In this paper, the concept of smart buildings is used in this way, as constructions that involve this kind of solutions.

To implement such intelligent systems, forecasting techniques at the area of artificial intelligence can be applied. Soft computing is widely used in real-life applications [11,12]. In fact, machine learning techniques have been widely used for a range of applications devoted to model of energy systems [13,14,2,15], through the estimation of artificial neural network (ANN) models by using historical records. The literature demonstrates their capabilities to work with time series or regression, over other conventional methods, on non-linear processes modeling such as energy consumption in buildings. Of special interest to this area is the use of ANN for forecasting the air temperature of a room, as a function of forecasted weather parameters (mainly solar radiation and air temperature) and actuators (as heating, ventilation and cooling systems) state are also manipulated variables. The aim is to obtain mid/long-range prediction models to be used for a more efficient temperature control, both in terms of regulation and energy consumption, as it can be read in [14]. The development of ANNs to predict such values could help us improve the overall energy consumption, balanced with a minimum affordable comfort of a home. Moreover, in case these values are well anticipated, it would be possible to define energy-efficient control actions.

Indoor temperature forecasting is an interesting problem that has been widely studied in the literature, as for example in [16,2,3,17,18]. The present paper is focused on multivariate forecasting using as input features different weather indicators, based on previous work [15]. Forecasting could be useful to reduce energy consumption of the HVAC system, but it is a complex task because of the multiple and different scenarios and placements where a house could be constructed. A forecasting model that obtains high performance in a location could be totally useless in another, not only because of geographical weather conditions, but also due to different constructive materials, energy transfer and HVAC systems. On-line learning algorithms [19] are needed to learn a model from *scratch* or to adapt a pre-trained model in such scenarios. Gradient Descent (GD) and Bayesian methods could be used to perform this learning and will be compared in this work.

The paper starts introducing the architecture of the overall domestic system and the different variables that are monitored and controlled. Next, it is presented how to tackle the problem of time series forecasting as the indoor temperature. Finally, experimental results are presented by comparing different models and estimation methods. At the conclusions, it is studied how the predicting results will be integrated within the energy consumption parameters and comfort levels of the SMLsystem house.

2. Environment setup

2.1. The Solar Decathlon Competition

The SMLhouse and SMLsystem solar houses[1] were built by the Universidad CEU Cardenal Herrera to participate in the Solar Decathlon 2010 and 2012 [1] editions respectively. The aim of these projects is to serve as prototypes for energy efficiency research. The Solar Decathlon Competition attempts to simulate the habits of people living in a house, requiring competitors to maintain conditions inside the house within a comfortable range (like temperature, CO_2 and humidity), while performing common tasks like using the oven, cooking with the hob, watching television (TV), having a shower, etc. using as less electrical power as possible.

The Embedded Systems and Artificial Intelligence (ESAI) group, who participated in both projects, focused its efforts on the development of a sensing and control framework, which might be able to forecast future values of indoor temperature. The aim of the prediction module is to reduce power consumption foreseeing future HVAC needs. As reference [20] states, due to thermal inertia, it is more efficient to maintain a temperature of a room or building than conditioning (heating or cooling) its temperature. Therefore, forecasting indoor temperature in the SMLsystem could reduce HVAC system consumption using future values of temperature and then decide whether to activate or not the HVAC system to maintain current temperature regardless its present value. This was subsequently confirmed by the competition measures in the SMLsystem prototype. In Table 1, two competition measuring scenarios are shown. Both scenarios show the behavior of indoor temperature in the SMLsystem and the HVAC system in competition. There is an initial conditioning phase in which the HVAC system lowers the SMLsystem temperature within a comfortable range (close to 24 °C) while outdoors temperature drops, and after that phase, there is a temperature maintenance phase. At the temperature maintenance phase the HVAC system maintains indoor temperature within a comfortable range (23.7–24.3 °C). Both phases duration are nearly identical but energy consumption of the maintenance phase represents up to 67.0% less than the energy needed to lower the SMLsystem temperature.

To evaluate the on-line indoor temperature prediction system a minimum of several weeks of sensing data are needed. Hence, the prediction module could not be evaluated using SMLsystem data, and it was trained using historical sensing data from SMLhouse prototype. The SMLhouse monitoring database is larger (40 days of data) and better to evaluate the feasibility of forecasting models following an on-line algorithm. This on-line module was integrated

[1] More info about both projects can be found here: http://sdeurope.uch.ceu.es/.

Table 1
SMLsystem temperature conditioning vs. temperature maintenance scenarios.

	Conditioning phase		Maintenance phase	
	Outdoors	Indoors	Outdoors	Indoors
Scenario 1[a]				
Initial	23.7 °C	28.3 °C	15.0 °C	23.7 °C
Final	15.0 °C	23.7 °C	12.3 °C	23.7–24.3 °C
Δ Temp.	−8.7 °C	−4.6 °C	−2.7 °C	–
Time elapsed	4.15 h		4.32 h	
HVAC power	4828 Wh		1593 Wh	
Scenario 2[b]				
Initial	25.2 °C	26.7 °C	19.7 °C	23.7 °C
Final	19.7 °C	23.7 °C	21.2 °C	23.7–24.3 °C
Δ Temp.	−5.5 °C	−3.0 °C	1.5 °C	–
Time elapsed	5.70 h		5.71 h	
HVAC power	6061 Wh		2359 Wh	

[a] Phase comparison: 67.0% less energy required for maintenance.
[b] Phase comparison: 61.1% less energy required for maintenance.

Fig. 1. SMLsystem roof, DHW and west facade solar panels.

at the SMLsystem house computer system, but the duration of the competition was not enough to allow data analysis.

2.2. The SMLsystem house

The SMLsystem is a modular house built using mainly wood and it was designed to be an energy self-sufficient house (near-zero energy building), using passive strategies and water heating systems to reduce the amount of electrical power needed to operate the house.

The energy supply of the SMLsystem shown in Fig. 1 is divided into solar power generation and Domestic Hot Water (DHW) system, and it might be connected to the grid to use electrical power when solar panels or batteries cannot supply enough power to the house. The photovoltaic solar system is responsible for generating electrical power by using 21 solar panels. These panels are installed on the roof and at the east and west facades. The energy generated by this system is managed by a device to inject energy into the house, or in case there is an excess of power, to the competition grid or a battery system. The batteries provide some autonomy to the house during low sun irradiance hours. The thermal power generation is achieved by using a solar panel that produces DHW which is also used for electric energy saving. This water circulates from a tank of 100 l to the panel when the control system detects that solar radiation is high enough to heat the water. When radiation is low, circulation stops to prevent water cooling.

The energy demand of the SMLsystem house is divided into three main groups: HVAC, house appliances and lighting and home electronics (HE). The HVAC system consists in a heat pump which is capable of heating or cooling water along with a rejector fan. Water pipes are installed inside the house to circulate the conditioned water and a fan coil system distributes the heat/cold using ventilation.

In the SMLsystem, the HVAC had a peak consumption of up to 3.6 kW when the heat pump was activated and, as shown in Table 2, it was the highest power consumption element of the SMLsystem in the contest with a 53.9% of total consumption. This is consistent with data from studies mentioned as the competition was held in Madrid (Spain) at the end of September and climate was hot (temperatures above 35 °C) during all competition days. The house has several energy-efficient appliances that are used during competition. Among them, there were a washing machine, refrigerator with freezer, an induction hob and a conventional oven. Regarding the consumption of washing machine and dishwasher, they can reduce the SMLsystem energy demand using hot water. The DHW system is capable of heating water to high temperatures, and then, when these appliances need hot water, the resistor must be activated just for a short time only to reach the desired temperature. The last energy-demanding group consists of several electrical outlets (e.g. TV, computer, Internet router, and others), the Energy Recovery Ventilation system (ERV) and lighting that consists of 16 LED light points controlled by the home automation system. The ERV ventilates the house reducing CO_2 levels and gaining comfort while maintaining indoor temperature by crossing the incoming and outgoing air pipes, and hence, conditioning the incoming air temperature. Although consumption of this subsystem do not exceed 0.3 kW peak power, it is a continuous energy demand, therefore, this group represents 11% of total energy demand.

Although the energy consumption of the house could be improved, the installed systems let the SMLsystem house to be a near-zero energy building, producing almost all the energy at the time the inhabitants need it. In a unfavorable hot environment for a wooden building like the SMLsystem, the house produced a 71.7% of energy excess which was returned to the competition grid as shown in Table 2. This good performance won the second place at the energy balance contest of the Solar Decathlon competition.[2]

2.3. The sensing and control system

A sensor and control framework shown in Fig. 2 has been used in the SMLsystem. It is operated by a Master Control Server (MCS)

[2] Classification of the energy balance contest can be found here: http://monitoring.sdeurope.org/index.php?action=scoring&scoring=S4.

Table 2
Energy consumption per subsystem, and balance between generation and consumption. H.E. indicates home electronics.

System	Subsystem	Peak (W)	Total (Wh)	Overall
HVAC	Heat pump	3497	22 423	31.8%
	Rest HVAC	–	15 564	22.1%
Total sum	–	3544	37 987	53.9%
Home appliances	Induction hob	3059	9285	13.2%
	Dish washer	2240	3070	4.3%
	Oven	2179	5042	7.2%
	Clothes washer	2089	2093	3.0%
Total sum	–	–	24 749	35.1%
Lighting &H.E.	–	300	7756	11.0%
All	–	–	70 492	100.0%

	Energy balance	
	Total (Wh)	Overall
Power generation	121 028	100.0%
Power consumption	70 493	58.2%
Energy balance (generation–consumption)	50 535	41.8%

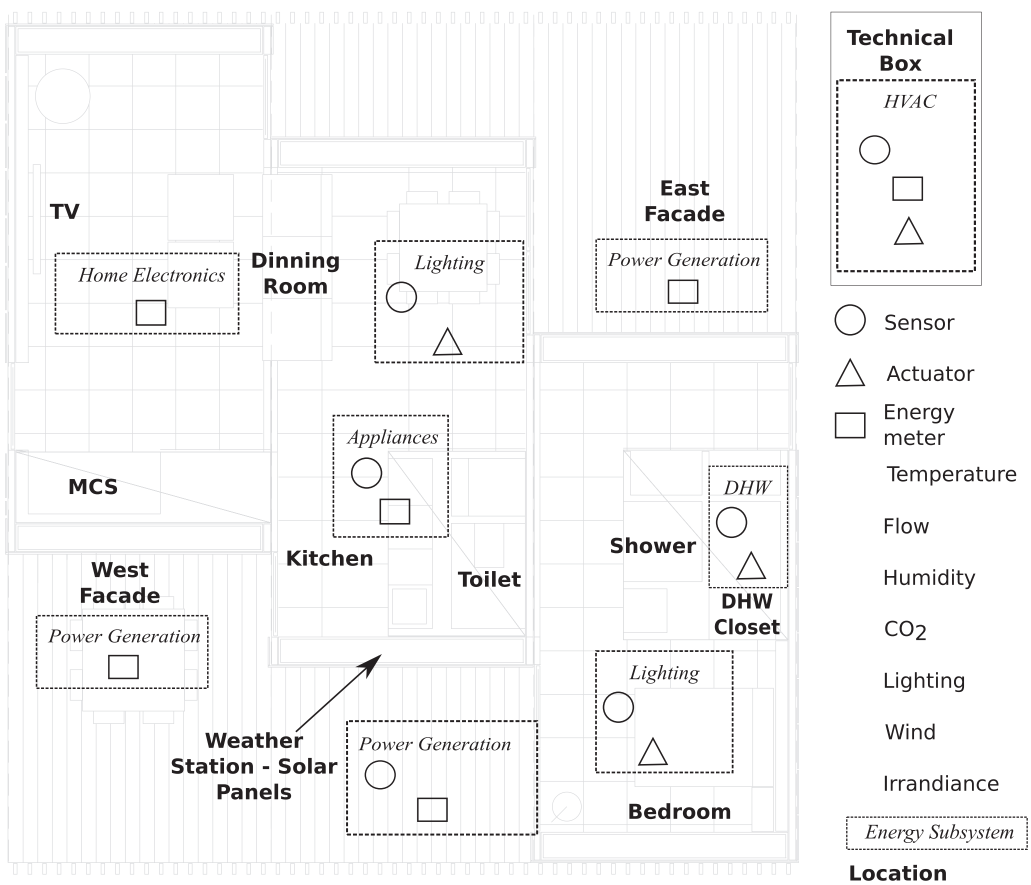

Fig. 2. SMLsystem sensors and actuators map.

using the European home automation standard protocol known as Konnex (KNX).[3] KNX modules are grouped by functionality: analog or binary inputs/outputs, gateways between transmission media, weather stations, CO_2 detectors, etc. The whole system uses 88 sensor values and 49 actuators. In the proposed system the immediate execution actions had been programmed to operate without the involvement of the MCS, such as controlling ventilation, the HVAC system and the DHW system. Beyond this basic level the MCS can read the status of sensors and actuators at any time and can perform actions on them via an Ethernet gateway.

A monitoring and control software was developed following a three-layered scheme. In the first layer, data is acquired from the KNX bus using a KNX-IP bridge device. The Open Home Automation Bus [21] (openHAB) software performs the communication between KNX and our software. At the second layer it is possible to find a data persistence module that has been developed to collect the values offered by openHAB with a sampling period of 60 seconds. Finally, the third layer is composed of different software

[3] Neither KNX nor Konnex are acronyms: http://ask.aboutknx.com/questions/430/abbreviation-knx.

Fig. 3. SMLsystem iOS App main status screen.

applications that are able to intercommunicate: a mobile application, shown in Fig. 3 has been developed to let the user watch and control the current state of domotic devices; and different intelligence modules are being also developed, for instance, the indoor temperature forecasting module.

Energy power generation systems described previously are monitored by a software controller. It is composed by multiple measurement sensors, including the voltage and current measurements of photovoltaic panels and batteries. Furthermore, the current, voltage and power of the grid is available. Also, the system power consumption of the house has sensors for measuring power energy values for each group element. The climate system has power consumption sensors for the whole system and individually for the heat pump. The HVAC system is composed of several actuators and sensors which are used for its operation. Among them are the inlet and outlet temperatures of the heat rejector and the inlet and outlet temperatures of the HVAC water in the SMLsystem. In addition, there are 14 switches for internal function valves, switches for the fan coil system, the heat pump and the heat rejector. The DHW system uses a valve and a pump for controlling water temperature. Some appliances have temperature sensors which are also monitored. The lighting system has 16 binary actuators that can be operated manually by using wall-mounted switches or by the MCS. The SMLsystem has indoor sensors for temperature, humidity and CO_2. Outdoor sensors are also available for lighting measurements, wind speed, rain, sun irradiance and temperature.

3. Time series forecasting

The interest of forecasting techniques in energy efficient environments relies in its easy integration into predictive control systems. A formal description of the forecasting modeling is introduced in this section. Time series are sequences of data sampled from any interesting process. Normally, time series show trend and pattern repetition through time, which are capable of being learned by statistical methods. They can be formalized as a sequence of scalars from a variable x obtained as output of the observed process:

$$\bar{s}(x) = s_0(x), s_1(x), \ldots, s_{i-1}(x), s_i(x), s_{i+1}(x), \ldots$$

A fragment beginning at position i and ending at position j will be denoted by $s_i^j(x)$.

It is possible to classify time series forecasting depending in the kind of data used as input, distinguishing between *univariate forecasting* when the system forecasts variable x using only past values of x, and *multivariate forecasting* when the system forecasts variable x using past values of x plus additional values of other variables. Multivariate approaches are useful when additional variables (covariates) cause variations on the variable x [15].

The number of past values, the size of the future window, and the position in the future of the prediction (future horizon), are the most important parameters in forecasting models. Following [22], it is possible to classify forecasting approaches depending in the future window size and how it is produced: *single-step-ahead forecasting* if the model forecasts only the next time step; *multi-step-ahead iterative forecasting* if the model forecasts only the next time step, producing longer windows by an iterative process; *multi-step-ahead direct forecasting* [23] if the model forecasts in one step a large future window of size Z. Following this last approach, two different major model types exist [22]:

- *Pure direct* which needs one forecasting model for each possible future horizon (Z models in total).
- *Multiple Input Multiple Output* (MIMO) which needs only one model to compute the full Z future window. ANNs could take big profit of this approach due to its discriminative basis. Additionally, ANNs with one or more hidden layers are able to learn non-linear input/output dependencies.

3.1. Formalization

A forecast model is a function F which receives as inputs a given number of past values for the interest variable (x_0), and a given number of past values for each of the available number C of covariates $(x_1, x_2, \ldots, x_C)$. The function produces as output a window with the Z predicted future values for variable x_0:

$$\langle \hat{s}_{t+1}(x_0), \hat{s}_{t+2}(x_0), \ldots, \hat{s}_{t+Z}(x_0) \rangle = F(\Omega(x_0), \Omega(x_1), \ldots, \Omega(x_C)), \quad (1)$$

being $\Omega(x) = s^t_{t-I(x)+1}(x)$ the $I(x)$ past values of variable/covariate x, and t the current time instant.

The number of past values $I(x)$ is important to ensure good performance of the model, however, it is not easy to estimate this number exactly. Different strategies are available in the literature, estimation of $I(x)$ by using linear or non-linear auto-correlation statistics, estimation of $I(x)$ by trial-and-error, and a combination of models varying the value of $I(x)$ [24,15].

3.2. Evaluation measures

The performance of forecast models is measured as the empirical error in a given validation or test set. This paper analyzes the model performance by using the mean absolute error (MAE) and the root mean square error (RMSE) functions. The error is computed comparing target values for the time series $s_{t+1}, s_{t+2}, \ldots, s_{t+Z}$ and its corresponding time series prediction $\hat{s}_{t+1}, \hat{s}_{t+2}, \ldots, \hat{s}_{t+Z}$ at current time t:

$$\mathrm{MAE}(t) = \frac{1}{Z}\sum_{z=1}^{Z} |\hat{s}_{t+z}(x_0) - s_{t+z}(x_0)| \quad (2)$$

$$\mathrm{RMSE}(t) = \sqrt{\frac{1}{Z}\sum_{z=1}^{Z} (\hat{s}_{t+z}(x_0) - s_{t+z}(x_0))^2} \quad (3)$$

3.3. Forecasting data description

The database used in this paper is a time series sampled with a period of T = 15 min, where each sample is a mean of the last quarter, reducing in this way the signal noise (for each hour it is computed this mean at 0 min, 15 min, 30 min, and 45 min). One time feature and five sensor signals were taken into consideration in a previous work [15], but the following three were found to be the most important:

- Indoor temperature in °C, denoted by variable $x = d$. This is the interest forecasted variable.
- Hour feature in universal time coordinated (UTC), extracted from the time-stamp of each pattern, denoted by variable $x = h$. The hour of the day is important to estimate the Sun position.
- Sun irradiance in W/m^2, denoted by variable $x = W$. It is correlated with temperature because more irradiance will mean more heat.

To evaluate the forecasting models performance, two datasets where used. The first, with 2764 instants (≈28 days), and the second with 1373 instants (≈14 days). They were captured during 2011 March, and 2011 June respectively. In total, 4136 time instants are available. The dataset is available for download at the UCI Machine Learning repository [25].

4. Forecasting methods

Two forecasting methods have been compared. In one hand, GD approach, where the model parameters are optimized following the direction of the gradient which minimizes a given loss function. In the other hand, a Bayesian approach which estimates the posterior probability distribution of the model parameters. Both approaches are implemented following the on-line learning paradigm, and the same data preprocessing.

4.1. On-line learning

Classically, the estimation of forecasting models is realized in a development phase where the given model is optimized by using iterative algorithms which traverse *several times* the called *training dataset*. In machine learning, every iteration through the whole dataset is known as *epoch*. Different updating strategy exists for the parameter estimation in one epoch. The selection of one of these strategies normally relies in the characteristics of the data. These strategies or learning modes are [26]:

- *Batch* mode: the algorithm iterates over all the training dataset and computing partial parameter updates, integrating all these partial updates in order to modify the parameters only once per epoch. This scheme allows to use accurate and fast matrix operations, but is not feasible with large datasets.
- *On-line* mode: the algorithm traverses the training dataset computing and updating the parameters with every pattern, modifying the parameters several times per epoch. This mode has faster convergence than batch, but could be more noisy.
- *Mini-batch* mode: a trade-off between both above strategies. The algorithm traverses the training dataset in batches with small size (hundreds of patterns), and integrating the parameters update for every mini-batch, modifying the parameters several times per epoch. This scheme has several computational advantages compared with on-line mode, allowing accurate and fast matrix operations.

However, in some real scenarios (e.g. indoor temperature forecasting), it is unsuccessful to train a laboratory model using previous algorithms, because the model needs to be adapted to the context or ambient characteristics. Besides, the data could vary during time, and this behavior could not be captured in the laboratory. The on-line training mode could be used in this kind of scenarios if, instead of using a training dataset, the model is estimated taken the patterns at real time from the given data source. In supervised learning, this training is only available for a small set of problems where the supervision is easy (or cheap in terms of time and resources) to be obtained. In unsupervised learning, this training is possible always, but not all the problems could be modeled in a unsupervised way.

In time series forecasting it is possible to follow an on-line training using real time scenario, allowing to place untrained systems in their target location, expecting to achieve better model predictions as the time it is running (and learning) goes on. This idea allows to abandon learning processes based on training sets [19], but it is not without problems. For instance, GD algorithms (typically used for ANN training) have a lot of tricky hyper-parameters which controls the learning and/or modify the model structure.[4] This hyper-parameters need to be set in order to train an accurate model, making harder the use of these models in on-line real scenarios, but opening a research line where all these problems need to be addressed. This paper presents empirical results of a direct implementation of on-line learning for time-series forecasting, using a GD for three different ANNs topologies, and a Bayesian estimation of a linear model.

4.2. Preprocessing of time series

Estimation of forecast models needs data preprocessing and normalization of input/output values in order to ensure better performance results. The indoor temperature variable ($x = d$) is the

[4] The number of hidden layers, number of neurons in each layer, the activation function, the learning rate, momentum and weight decay terms.

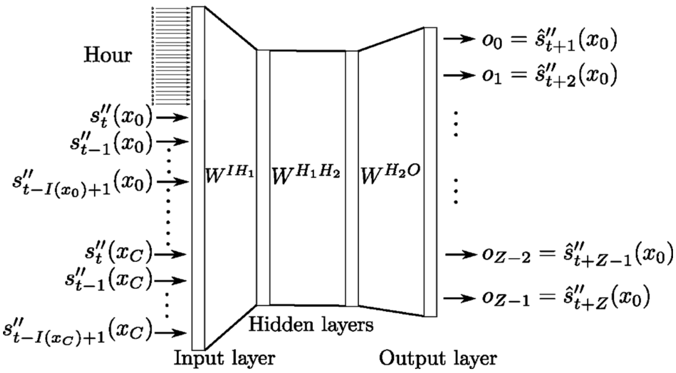

Fig. 4. Artificial neural network topology for time series forecasting.

interest forecasted variable. In order to increase model generalization, this variable is differentiated and a new $\bar{s}'(x=d)$ signal sequence is obtained following this equation:

$$s'_i(x=d) = s_i(x) - s_{i-1}(x), \tag{4}$$

being $s'_i(x=d)$ the component of instant ith. The differentiation of indoor temperature shows to be important to achieve good generalization results, and it is based on a previous work where not differentiated data had been used [27].

After differentiation, the time series are normalized to be zero-mean and standard deviation one (in case of indoor temperature d), or applying max-min normalization (for sun irradiance W). The values for this normalization where taken from the training data.

For the hour component ($x=h$), a different approach is followed. It is represented as a category locally-encoded, which consists in using a vector with 24 components where 23 components are set to 0, and the component which indicates the hour value is set to 1. This kind of encoding avoids the big jump between 23 and 0 at midnight, but forces the model to learn the relationship between adjacent hours. Other approaches for hour encoding could be done in future works.

4.3. Gradient descent approach: ANNs

For the GD approach, ANNs models has been used (from perceptron or linear models which are equivalent to the Bayesian linear models, up to more complex ANNs). ANNs has an impressive ability to learn complex mapping functions as they are an universal function approximator [28], and are widely used in forecasting [29,14,24,30].

ANNs are formed by one input layer, an output layer, and a few number of hidden layers. Fig. 4 is a schematic representation of an ANN with two hidden layers for time series forecasting. The input of the ANN are past values of covariates, and the output layer is formed by the Z future window predicted values, following the MIMO approach described in Section 3, which obtains better accuracy in previous experimentation [27,15].

The well-known GD algorithm, error Back-Propagation [31] (BP) has been used in its on-line version to estimate the ANN weights, adding a momentum term and a L2 regularization term (weight decay). Despite that theoretically algorithms more advanced than BP exists nowadays, at empirical level BP is easier to implement and a correct adjustment of momentum and weight decay helps to avoid bad local minima. Besides, BP has an on-line version which fits perfect with the approach followed in the present paper.

The BP minimizes the mean square error (MSE) function with the addition of the regularization term weight decay, denoted by ϵ, useful to avoid over-fitting and improve generalization. Therefore, the loss of the pattern at instant t follows:

$$E = \overbrace{\frac{1}{2}\sum_{i=1}^{Z}\left(\hat{s}_{t+i}(x_0) - s_{t+i}(x_0)\right)^2}^{\text{MSE}} + \frac{\epsilon}{2}\sum_{w_i\in\theta} w_i^2\,, \tag{5}$$

where θ is a set of all weights of the ANN (without the bias) and w_i is the value of weight ith. Any weight $w_j^{(e+1)}$ (including biases), at epoch $e+1$, is updated following the gradient of Eq. (5) plus momentum term:

$$w_j^{(e+1)} = w_j^{(e)} - \eta\frac{\partial \text{MSE}}{\partial w_j^{(e)}} + \mu(w_j^{(e)} - w_j^{(e-1)}) - \epsilon w_j^{(e)} \tag{6}$$

where η is the learning rate, μ is the momentum coefficient and ϵ is the weight decay coefficient.

The GD approach has been done with April-ANN [32], a toolkit for estimation of ANN models.

4.4. Bayesian approach

The Bayesian approach has been selected against other methods more standard in statistics as a practical decision. It is easier to develop an applicable theory of recursive inference under Bayesian framework than under, for example, maximum likelihood philosophy.

Bayesian inference [33,34] is a statistical method for model inference which allow to incorporate a prior distribution (prior knowledge) for each parameters in the model. That approach combines prior distribution with data for estimating the posterior distribution of each parameter in the model. It is especially useful in the dynamic analysis of a sequence of data (like that).

In this work, the parameters in different predictive models want to be estimated. When the algorithm starts, there are not any prior information for each parameter of the model. So, a non-informative prior distribution is proposed at the first time step for each of the parameters, and the first posterior distribution of them is computed. A non-informative prior distribution for each parameter represents a total lack of knowledge about them in that first step of time. With the mean value of the posterior distribution for each parameter, the model to predicts the following 12 future values.

Later, from the second time step in advance the same recursive process is repeated. The mean and standard deviation of the posterior distribution estimated for each parameter at the time step $t=k-1$ will be used as parameters for the normal prior distribution for each parameter at the next time step $t=k$. With that prior distribution and the new input in the time moment $t=k$ the posterior distribution of each parameter in the model is estimated.

In this Bayesian framework one simple linear model has been selected. It is a typical linear model with covariates, an order one auto-regressive model for indoor temperature with covariates based on indoor temperature in the last time moment and other covariates related plus a Gaussian error term.

It is possible to use some more difficult structures for recursive Bayesian estimation [35,36] and more sophisticated Bayesian hierarchical models [37,38], but in this work goal is to test an on-line forecasting model using the minimum possible resources.

The Bayesian approach has been done with WinBUGS [39], a toolkit for estimation of models following Monte Carlo Markov chain methods.

5. Experimental results

Several experiments were performed to decide which covariates optimize the forecasting accuracy. Multiple covariates

Table 3
Training parameters depending on the model (η is learning rate, μ is momentum term, ϵ is weight decay, model size is in floats).

	Input			Out.	Hyper-parameters				
Model	$I(d)$	$I(W)$	$I(h)$	Z	η	μ	ϵ	Hidden layers	Size
B-Lin	1	4	24	12	–	–	–		360
GD-Lin	13	4	24	12	0.003	0.09	10^{-4}		504
GD-D1	13	4	24	12	0.006	0.06	10^{-4}	16 *tanh*	876
GD-D2	15	4	24	12	0.08	0.02	10^{-4}	24 *tanh* – 16 *tanh*	1564

Table 4
Percentage of time instants (plus 99.9% confidence interval) where a model performs better than all the rest, computed comparing MAE and RMSE error measures. The first 100 time instants where ignored in order to avoid random model errors.

	B-Lin	GD-Lin	GD-D1	GD-D2
MAE wins	42.5 ± 2.1%	21.8 ± 1.7%	18.8 ± 1.6%	16.9 ± 1.5%
RMSE wins	44.2 ± 2.0%	20.9 ± 1.7%	17.7 ± 1.5%	17.2 ± 1.5%

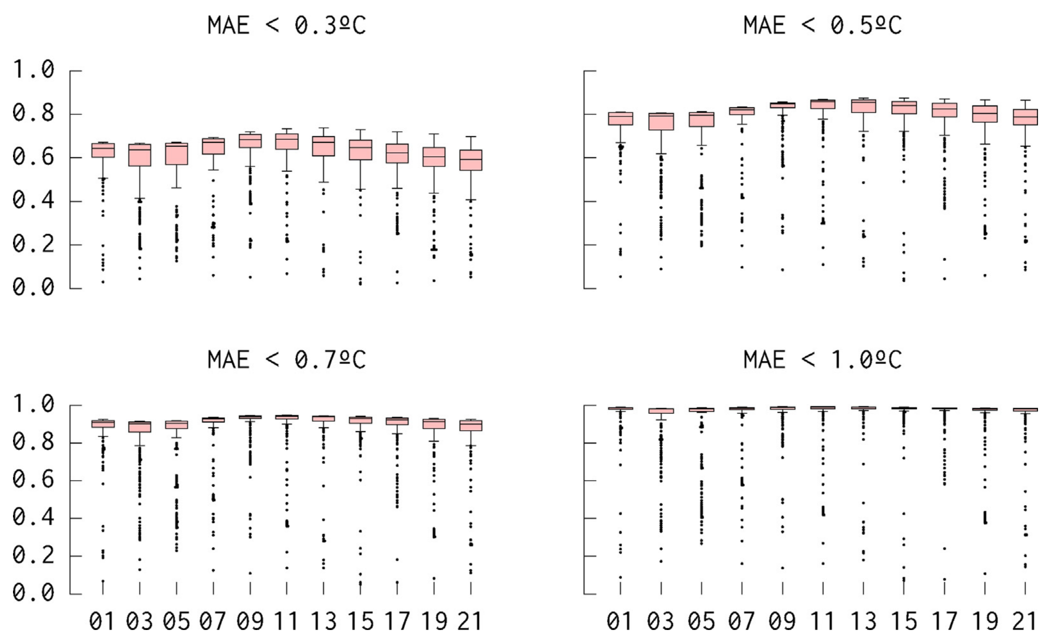

Fig. 5. Box-plot of the grid search performed to optimize hyper-parameters of GD algorithm. The boxes are computed taking the proportion of time instants where the MAE was less than a given value. The horizontal axes are different input sizes.

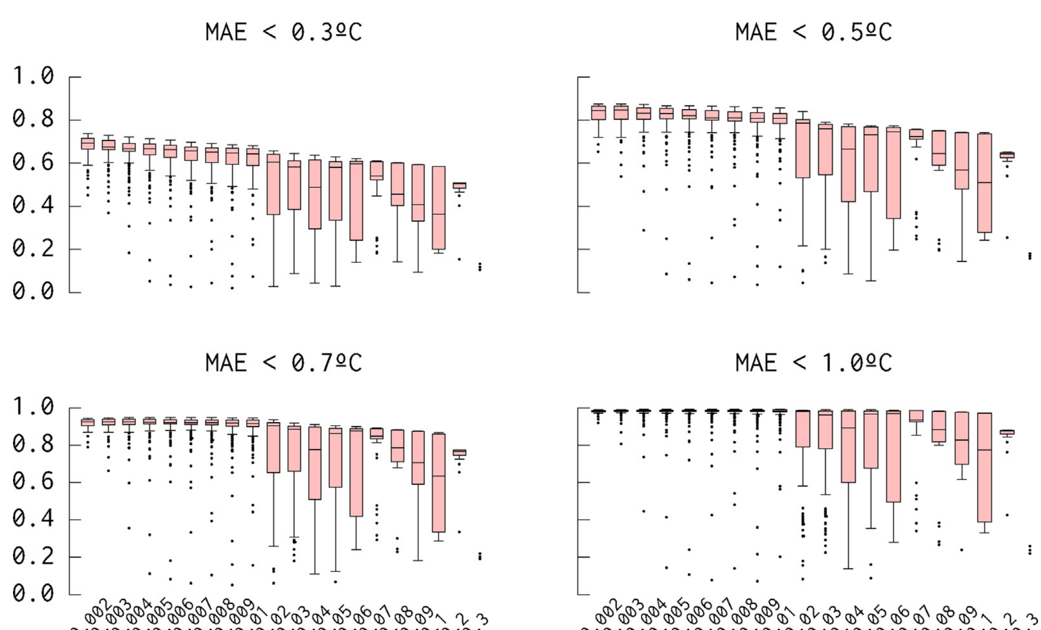

Fig. 6. Box-plot of the grid search performed to optimize hyper-parameters of GD algorithm. The boxes are computed taking the proportion of time instants where the MAE was less than a given value. The horizontal axes are different learning rates.

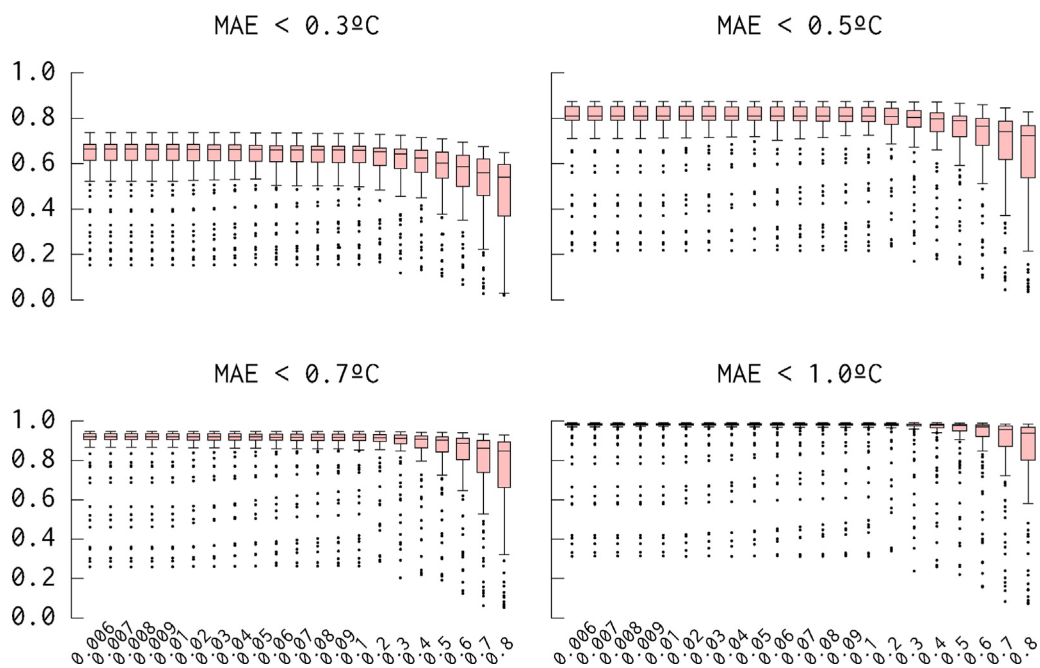

Fig. 7. Box-plot of the grid search performed to optimize hyper-parameters of GD algorithm. The boxes are computed taking the proportion of time instants where the MAE was less than a given value. The horizontal axes are different momentum term values.

combinations plus different ANN topologies were tested in a previous work [15]. The best performance was obtained using a model with indoor temperature d, current time hour h and sun irradiance W covariates, and two hidden layers with 24 and 16 neurons using *tanh* activation function.

5.1. Models architecture

Extending previous experimentation, this paper compares a Bayesian linear model (B-Lin, Eq. (7)) with three ANN topologies. The three ANN estimated by using GD are: a linear regression model (GD-Lin, Eq. (7)), a regression model with one hidden layer with 16 neurons and *tanh* activation function (GD-D1, Eq. (8)), and a regression model with two hidden layers of 24 and 16 neurons and *tanh* activation function (GD-D2, Eq. (9)). All the models have $Z=12$ output values. The Bayesian models receive as input the last indoor temperature, the last four sun irradiance values, and the 24 components of the current hour. ANN models receive in their input the last four sun irradiance values, the 24 components of the current hour, and different input sizes for indoor temperature were compared. Formally, the three models are written as:

$$\mathbf{y} = \alpha_1 + \mathbf{x} \cdot \beta_1 \quad (7)$$

$$\mathbf{y} = \alpha_2 + \beta_2 \cdot g(\alpha_1 + \mathbf{x} \cdot \beta_1) \quad (8)$$

$$\mathbf{y} = \alpha_3 + \beta_3 \cdot g(\alpha_2 + \beta_2 \cdot g(\alpha_1 + \mathbf{x} \cdot \beta_1)) \quad (9)$$

where $\mathbf{y} = s'_{t+1}(x=d)\ldots s'_{t+Z}(x=d)$ is the forecasted vector with size Z, the activation function g is the hyperbolic tangent, α_i are the vector of independent terms (biases in the ANNs), β_i are matrices of weights (the connections in the ANNs), and $\mathbf{x}$ is a vector with the input of the forecasted model: 13 past temperature values, 4 past sun irradiance values and 24 values for the locally encoded current hour. Details about hidden layers and model sizes are in Table 3.

5.2. Hyperparameters optimization

The GD approach needs the optimization of the hyper-parameters: learning rate, momentum and weight decay. A grid search was carried out in order to estimate them plus the indoor

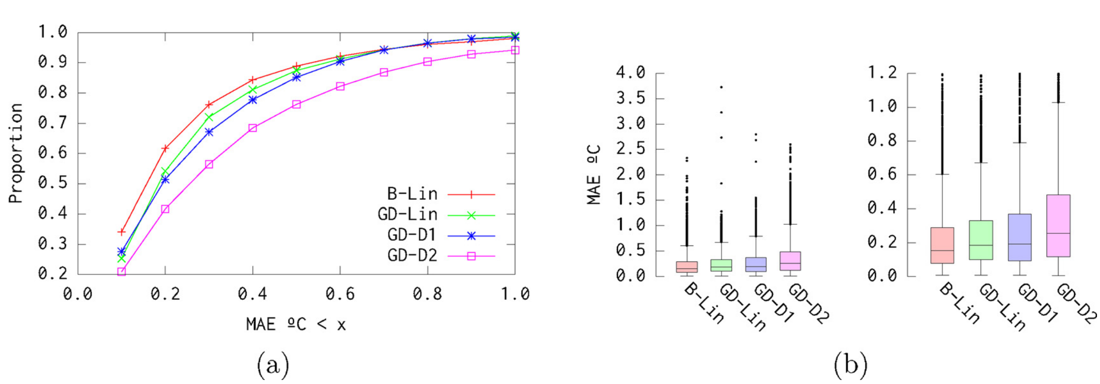

Fig. 8. Comparison between Bayesian and GD models, computed taken the best GD model, compared with the Bayesian estimation. (a) The proportion (y-axis) of time instants where the MAE was less than a given value (x-axis). (b) A box-plot of the distribution of all the time instants for each of the models. The first 100 time instants where ignored in order to avoid random model errors.

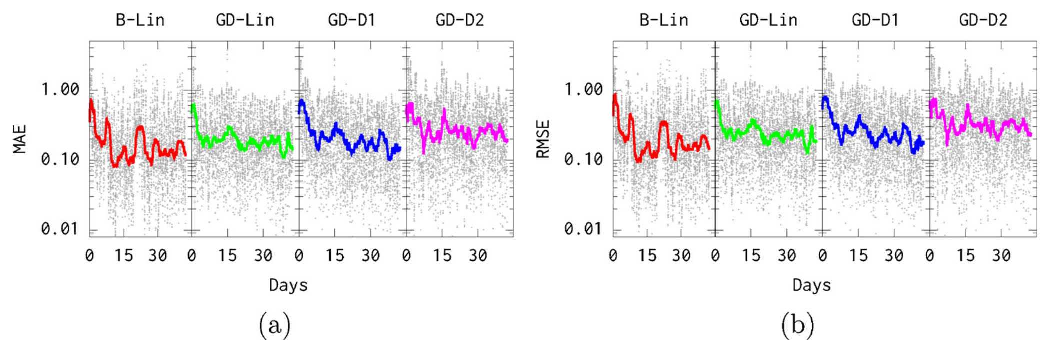

Fig. 9. Comparison between Bayesian and GD models. The plot represents the error (MAE or RMSE) at each time instant in dots, and the median in a window of the last 2 days (192 instants) with lines. The *y*-axis was log-scaled, in order to increase the resolution of small error values.

temperature input size. This grid search was repeated for the three ANN models, and the best combination was used to compare with the Bayesian approach. As example of the grid search, Figs. 5–7 show box-plots of the experiments for the linear ANN model estimated by GD algorithm. Intuitively, seems reasonable that a lower learning rate (<0.01), combined with a medium momentum (>0.01) would ensure good results. In general terms, a lower learning rate means that it is better to perform a smooth training, because the gradients are computed for only one pattern, and it is not the *real* gradient of the model. A large momentum helps to introduce a large inertia in the training of the models, capturing long term trends of the gradients. Looking to the grid-search, the dependencies between all these parameters make difficult to find this trend in all the cases. Table 3 shows the best combination of hyper-parameters.[5] The model GD-D2 is slightly out of these parameters because its optimization is harder due to the model complexity, and needs a large learning rate. Besides, the GD-D1 and GD-D2 models fail to converge in an important number of experiments.

5.3. Analysis of Bayesian and gradient descent approaches

A comparison between the best ANN models estimated with GD, and the linear model estimated by Bayesian methods, was performed. The on-line learning was executed for all the models, and the MAE and RMSE were computed for each possible forecast. Values of MAE < 1.0 °C were found after approximately 5 days of training (480 time steps). The trend is to improve the forecasting as the times goes by (see Fig. 9, the points are MAE and RMSE at each time instant).

Fig. 8 shows the comparison results of this training. The first 100 time steps were ignored, because the error is high due to the lack of a good initial point. The comparison relies in the computation of the number of time instants where the MAE < *x* (Fig. 8a), and in a box-plot of the MAE for the all the available time instants (Fig. 8b).

Comparing the GD models between themselves, the plots show better results using simpler models (GD-Lin), which differs from the results published in [15], where a model with two hidden layers was the best one for this covariates combination. However, this result makes sense because in the previous work, the models were trained following a dataset learning approach, instead of the on-line scheme presented here, and the model was trained doing several iterations with the whole dataset (several epochs). In this work, only one iteration is possible and complex models were not well estimated by GD approach.

Comparing GD with Bayesian approach, the GD-Lin and B-Lin models show similar results, slightly better for B-Lin. Both are the same linear regression model, but estimated following GD or Bayesian methods. Fig. 9 shows the evolution of MAE and RMSE for the best models, showing a noisy trend towards lower errors while time goes by. For comparison purposes, Table 4 shows the percentage of time instants where a model is considered the best because it achieves the minimum error. Clearly, the B-Lin model achieves the better result, it achieves a MAE which is the minimum in the 42.5 ± 2.1% of the time instants, front the 21.8 ± 1.7% of the GD-Lin model. The GD-1 and GD-2 perform slightly worst than the GD-Lin. These results are being reported with a confidence interval 99.9% computed using bootstrapping resampling [40], which is a statistical technique for estimation of variance in population samples.

5.4. Algorithm convergence

Fig. 9 shows the evolution of error for the on-line algorithm. An error reduction trend is observed looking into the two days error median. Besides, all tested models show a high slope at the beginning, suggesting fast model convergence from a totally random model to an *almost* usable model with a median of error below 0.5 °C, in approximately 5 days of learning. However, more data and experiments will be needed to make clear these points.

6. Conclusions and discussion

An overview of the monitoring and sensing system developed for the SMLsystem solar powered house has been described. This system was employed during the participation at the Solar Decathlon Europe 2012 competition. The research in this paper has been focused on how to predict the indoor temperature of a house, as it is directly related to HVAC system consumption. HVAC systems represents 53.89% of the overall power consumption of the SML-system house. Furthermore, performing a preliminary exploration of the SMLsystem competition data, energy used to maintain temperature was found to be 30–38.9% of the energy needed to lower it. Therefore, an accurate forecasting of indoor temperature could yield to an energy-efficient control.

An on-line learning approach was presented, which allows the integration of predictive models in totally unknown scenarios. Therefore, the model is trained continuously, expecting to achieve better performance during time. Different regression models, estimated following a Bayesian on-line algorithm and a Gradient Descent (GD) on-line algorithm, were compared. All of them achieve a promising MAE, with large proportions of data samples (almost 100% of data) below 1.0 °C. Simple models show better performance than complex models (ANNs with one or two hidden layers), even when complex models are more expressive. A reasonable explication is founded in the lack of data, complex models need more iterations to achieve good convergence, while simpler

[5] The best model is took as the one that maximizes the percentage of time instants where the MAE was <0.5 °C.

models converge faster. However, a deeper analysis of this issue is needed in order to state the dependence between the dataset size and the model complexity.

One interesting discussion appears when focusing into on-line learning issues. The on-line adaptation of predictive models could diverge during time, achieving its best performance in one exact time instant, but losing it if the training continues. Different strategies are possible, as adaptation of learning speed depending in the behavior of the model; the memorization of some patterns which could be used as a validation set; application of energy systems theory (like simulated annealing) to disturb the model an epsilon, helping to avoid local minima; among others ideas which need to be studied deeper as future work.

An important insight of this paper is the finding of simple models performing reasonably well with small training data. The models are simple enough to be implemented using low resources hardware. As it is shown in Table 3, the GD-Lin model, which performs better than other GD models, needs 504 float values to be implemented. The GD algorithm uses momentum (504 more floats) and needs to store auxiliary data for layers computation (41 inputs plus 12 outputs), increasing up to $2 \times 504 + 41 + 12 = 1061$ floats. Depending in the architecture, it is possible to implement this model using 32 bits floats, but 16 bits resolution could be enough for a task like this, requiring 2122 bytes. The on-line learning algorithm needs to store buffers.

Regarding to the Bayesian approach, the model is a linear model with only one past value of temperature, four past values of sun irradiance, and 24 inputs for the categorical hour representation, and produces 12 future forecasts. So, the model needs 360 floats to be represented, and 360 more for the training procedure (computation of the mean for each parameter). In this case, the auxiliary data needed to store the model computation is of 29 inputs and 12 outputs. At the end, the model requires $2 \times 360 + 29 + 12 = 761$ floats. Note that the input/output configuration of this model is simpler than for the GD-Lin model, a deeper analysis of the effect of more complex models in the Bayesian approach will be conducted in the future.

These low resources requirement makes possible to integrate such predictive models in almost any domestic device. Obviously, the indoor temperature forecasting is useful for the climate system, but other devices could also take profit of short-term forecasting.

Acknowledgements

This work has been supported by Banco Santander and Universidad CEU Cardenal Herrera through the project Santander-PRCEU-UCH07/12.

References

[1] United States Department of Energy, Solar Decathlon Europe Competition, 2012 http://www.solardecathlon.gov
[2] P. Ferreira, A. Ruano, S. Silva, E. Conceição, Neural networks based predictive control for thermal comfort and energy savings in public buildings, Energy and Buildings 55 (2012) 238–251, Cool Roofs, Cool Pavements, Cool Cities, and Cool World.
[3] J. Álvarez, J. Redondo, E. Camponogara, J. Normey-Rico, M. Berenguel, P. Ortigosa, Optimizing building comfort temperature regulation via model predictive control, Energy and Buildings 57 (2012) 361–372.
[4] Instituto para la diversificación y ahorro de la energía (IDAE), Practical Guide to Energy. Efficient and Responsible Consumption, Madrid, 2011.
[5] M. Weiser, Some computer science issues in ubiquitous computing, Communications of the ACM 36 (1993) 75–84.
[6] E. Zelkha, B. Epstein, From devices to 'Ambient Intelligence': the transformation of consumer electronics, in: Presentation at the Digital Living Room Conference, Philips, 1998.
[7] F. Sadri, Ambient intelligence: a survey, ACM Computing Surveys 43 (2011), 36:1-36:66.
[8] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, J. Burgelman, ISTAG Scenarios for Ambient Intelligence, European Commission, 2010.
[9] E. Kaasinen, T. Kymäläinen, M. Niemelä, T. Olsson, M. Kanerva, V. Ikonen, A user-centric view of intelligent environments: user experience and user role in building intelligent environments, Computers 2 (2013) 1–33.
[10] E. Sierra, A. Hossian, R. García, P. Marino, Expert system for intelligent control of environmental variables of a energy-efficient building, in: XI Workshop in Inf. Proc. and Ctrl, 2005.
[11] C.L. Wu, K.W. Chau, Y.S. Li, Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques, Water Resources Research 45 (2009).
[12] R. Taormina, K. wing Chau, R. Sethi, Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon, Engineering Applications of Artificial Intelligence 25 (2012) 1670–1676.
[13] S. Karatasou, M. Santamouris, V. Geros, Modeling and predicting building's energy use with artificial neural networks: methods and results, Energy and Buildings 38 (2006) 949–958.
[14] A. Ruano, E. Crispim, E. Conceição, M. Lúcio, Prediction of building's temperature using neural networks models, Energy and Buildings 38 (2006) 682–694.
[15] F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, J. Pardo, Towards energy efficiency: forecasting indoor temperature via multivariate analysis, Energies 6 (2013) 4639–4659.
[16] A.H. Neto, F.A.S. Fiorelli, Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption, Energy and Buildings 40 (2008) 2169–2176.
[17] F. Oldewurtel, A. Parisio, C.N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, M. Morari, Use of model predictive control and weather forecasts for energy efficient building climate control, Energy and Buildings 45 (2012) 15–27.
[18] F. Mateo, J.J. Carrasco, A. Sellami, M. Millán-Giraldo, M. Domínguez, E. Soria-Olivas, Machine learning methods to forecast temperature in buildings, Expert Systems with Applications 40 (2013) 1061–1068.
[19] S. Frandina, M. Gori, M. Lippi, M. Maggini, S. Melacci, Variational foundations of online backpropagation, in: Artificial Neural Networks and Machine Learning - ICANN 2013, volume 8131 of Lecture Notes in Computer Science, Springer, 2013, pp. 82–89.
[20] D. Pan, Y. Yuan, D. Wang, X. Xu, Y. Peng, X. Peng, P.-J. Wan, Thermal inertia: towards an energy conservation room management system, in: A.G. Greenberg, K. Sohraby (Eds.), INFOCOM, IEEE, 2012, pp. 2606–2610.
[21] K. Kreuzer, T. Eichstädt-Engelen, The OSGI-based Open Home Automation Bus, 2011 http://www.openhab.org
[22] S. Ben Taieb, G. Bontempi, A. Atiya, A. Sorjamaa, A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition, Expert Systems with Applications (2012), Preprint.
[23] H. Cheng, P. Tan, J. Gao, J. Scripps, Multistep-ahead time series prediction, Lecture Notes in Computer Science 3918 (2006) 765–774.
[24] L. Yu, S. Wang, K.K. Lai, Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm, Energy Economics 30 (2008) 2623–2635.
[25] K. Bache, M. Lichman, UCI Machine Learning Repository, 2013 http://archive.ics.uci.edu/ml
[26] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., John Wiley and Sons, New York, 2001.
[27] F. Zamora-Martínez, P. Romeu, J. Pardo, D. Tormo, Some empirical evaluations of a temperature forecasting module based on artificial neural networks for a domotic home environment, in: IC3K - KDIR, 2012.
[28] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, New York, 1995.
[29] G. Zhang, B.E. Patuwo, M.Y. Hu, Forecasting with artificial neural networks: the state of the art, International Journal of Forecasting 14 (1998) 35–62.
[30] G. Escrivá-Escrivá, C. Álvarez-Bel, C. Roldán-Blay, M. Alcázar-Ortega, New artificial neural network prediction method for electrical consumption forecasting based on building end-uses, Energy and Buildings 43 (2011) 3112–3119.
[31] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Neurocomputing: Foundations of Research (1988) 696–699.
[32] F. Zamora-Martínez, et al., April-ANN Toolkit, A Pattern Recognizer in Lua with Artificial Neural Networks, 2013 https://github.com/pakozm/april-ann
[33] A. Gelman, J. Carlin, H. Stern, D. Rubin, Bayesian Data Analysis, CRC Press, 2003.
[34] B. Carlin, T. Louis, Bayesian Methods for Data Analysis, CRC Press, 2011.
[35] W. Gilks, S. Richardson, D. Spiegelhalter, Markov Chain Monte Carlo in Practice, CRC Press, 1996.
[36] D. Gamerman, H. Lopes, Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, CRC Press, 2006.
[37] P. Congdon, Applied Bayesian Hierarchical Methods, CRC Press, 2010.
[38] N. Hjort, Bayesian Nonparametrics, Cambridge University Press, 2010.
[39] D.J. Lunn, A. Thomas, N. Best, D. Spiegelhalter, Winbugs—a Bayesian modelling framework: concepts, structure, and extensibility, Statistics and Computing 10 (2000) 325–337.
[40] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, New York, NY, 1993.