Short-Term Sensory Data Prediction in Ambient Intelligence Scenarios

Enrico Daidone and Fabrizio Milazzo

Abstract Predicting data is a crucial ability for resource-constrained devices like the nodes of a Wireless Sensor Network. In the context of Ambient Intelligence scenarios, in particular, short-term sensory data prediction becomes a key enabler for more difficult tasks such as prolonging network lifetime, reducing the amount of communication required and improving user-environment interaction. In this chapter we propose a software module designed for clustered wireless sensor networks, able to predict various environmental quantities, namely temperature, humidity and light. The software module is supported by an ontology that describes the topology of the AmI scenario and the effects of the actuators on the environment. We applied our module to real data gathered from a public office at our department and obtained significant results in terms of prediction error even in presence of environmental actuators.

1 Introduction

Ambient Intelligence (AmI) is an emergent field of AI aimed at developing smart, distributed pervasive systems able to support human-environment interaction [8]. The basic infrastructure of an AmI system is made up of sensors, actuators and reasoners [2, 4, 7, 9]. The sensory components monitor the environment by measuring physical phenomena like temperature, humidity and light, but also by acquiring digital images and sounds, detecting user presence and so on. The actuators are those elements able to affect the environment according to the users' needs. The reasoners are able to learn, recognize and infer users' needs as well as to predict environmental phenomena.

E. Daidone · F. Milazzo (✉)
University of Palermo, DICGIM-Viale delle Scienze, Edificio 6, 90128 Palermo, Italy
e-mail: fabrizio.milazzo@unipa.it

S. Gaglio and G. Lo Re (eds.), *Advances onto the Internet of Things*,
Advances in Intelligent Systems and Computing 260, DOI: 10.1007/978-3-319-03992-3_7,

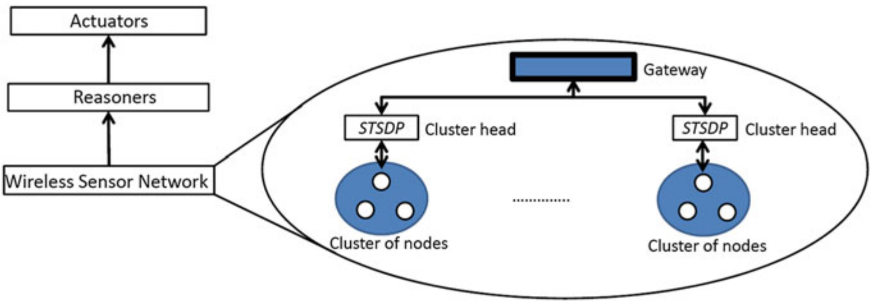

Fig. 1 Reference architecture of the AmI system and placement of STSDP software module

Over the last few years, AmI designers have begun to implement sensory infrastructure by using the so-called Wireless Sensor Network (WSN) technology, that is a network made up of sensor nodes able to sense physical phenomena and to perform small on-board computations. The growing use of such technology has been triggered by the presence of consolidated communication protocols like IEEE 802.15.4 and Wireless HART and a lightweight operating system like TinyOS. Nevertheless, wireless sensor nodes are limited by their scarce computational resources, storage and energy [1]. For these reasons, predicting data becomes a key enabler in improving WSN performance as it makes it possible to reduce communications and to prolong network lifetime. Moreover, in the context of Ambient Intelligence, predictions could be also exploited by reasoners to control actuators in order to satisfy user needs.

The main contribution of this chapter lies in the implementation of *Short-Term Sensory Data Prediction* (STSDP), i.e. a software module able to predict the physical phenomena monitored by a Wireless Sensor Network, with the combined effect of actuators.

Figure 1 shows our chosen reference AmI architecture and the placement of the STSDP module: we assume that the WSN is arranged as a set of clusters and that each cluster-head runs STSDP; the computational burden of the sensor nodes is kept as low as possible as they are only required to communicate the sensed readings to the cluster-head. The cluster heads relay their predictions to the gateway that is responsible for aggregating and communicating them to the upper layer. The intelligent modules (reasoners) are responsible for performing complicated tasks such as controlling the actuators and computing long-term data predictions. The context (sensors, actuators and physical phenomena) is modeled using *Ontology Web Language* (OWL), which allows STSDP to build an updated representation of the environment.

The remainder of this chapter is organised as follows: Sect. 2 depicts the current state of the art methods for predicting data in wireless sensor networks; Sect. 3 provides mathematical details of STSDP as well as the description of the OWL ontology that models the environmental context. Section 4 describes the experiments we carried out to validate our software module, and finally Sect. 5 presents our conclusions.

2 Related Works

Predicting data is a widely studied topic in wireless sensor networks as it makes it possible to prolong network lifetime by aggregating and compressing data, lowering network communications and reducing the amount of storage required.

Any prediction algorithm for wireless sensor networks is specified from three different perspectives: *scope*, *topology* and *methodology*. The temporal and spatial scope of a prediction algorithm are respectively short (minutes or hours/meters) or long (days or months/kilometers). Broadly speaking, short-term prediction algorithms fit the requirements of a WSN rather well, as they require a little knowledge of the past and few computational resources. Moreover their degree of precision within the temporal and spatial scopes designed is very high. Long-term prediction algorithms are very complex if compared to the short-term ones and are usually performed by devices with more computational resources (as an example the reasoners of the reference AmI architecture chosen here may be good candidates for performing such complex computations). The precision of such algorithms is quite constant over time, although it is less than that of short-term prediction algorithms designed only for limited scopes.

The topology perspective defines "who" is responsible for carrying out predictions. *Centralized* approaches assume that a central base station gathers readings from the surrounding nodes and then builds a global model of the monitoring field [3, 17], whereas *distributed* approaches [16] focus on local data processing and are very precise as compared to the former methods. Their main drawback is their elavated computational complexity, which makes them unsuitable for resource constrained devices like wireless sensor nodes.

The methodological perspective defines how to predict data; the *stochastic* approach models each physical phenomenon as a random process based on a set of observable and unobservable parameters. Such parameters are associated with previously learned prior probability distribution functions (PDF) and the predictions are drawn from the posterior PDF conditioned on the observed variables [11, 12, 15]. The *deterministic* approach assumes some kind of mathematical law that links past readings to future ones. Common implementations are "Time Series" [13, 16] and "Regression models" [10]. Stochastic approaches are suitable for all those applications that require long-term predictions, but their prediction error is very high as compared to deterministic approaches. On the other hand, deterministic approaches require a limited amount of storage and computation and produce smaller errors when used as short-term predictors.

We chose to implement STSDP using a centralized approach to minimize the computational burden on sensor nodes and to prolong their lifetime as much as possible. The decision to use a deterministic implementation was straightforward, as our module was specifically designed to perform short-term data prediction.

3 Proposed Approach

This Section discusses our implementation of the STSDP software module together with the OWL ontology adopted to describe the environmental context.

Let us assume that the WSN is arranged as a set of clusters and that the nodes behave as *cluster head* or *leaf*. Leaf nodes are characterized by their scarce computational resources and are responsible for gathering measurements and relaying them to the cluster head. Cluster heads are not limited by energy or computational resources and act as the so called *micro-servers* (e.g. Stargate nodes). They are responsible for building the spatio-temporal representation and prediction of the monitored phenomena.

Each cluster is associated with an *area of interest*—a spatial portion of the monitoring field—bounded by the convex-hull defined by its own sensor nodes. Moreover, we assume that the areas of interest do not overlap with each other.

Figure 2 shows the steps followed by STSDP to predict physical phenomena. The *Context Generation* submodule reads the Ontology and creates the description of the context: sensor nodes (e.g. sampling rate, position, status), actuators (e.g. affected phenomenon, position) and the phenomena (e.g. light, temperature or humidity). The *Prior estimates* submodule creates a rough representation of the phenomenon using the readings sensed by each node during the previous 24 hours; the current readings add fresh information and allow to build the more precise *Posterior estimators*; the *Fusion* step mixes the Posterior estimators to generate a spatio-temporal mesh of the area of interest from which predictions are collected. In order to make the approach suitable for ambient intelligence scenarios we added the *Actuators correction* module that integrates the effect of the actuators on the monitored phenomena. Finally, the *Context Update* module updates the ontology to keep track of possible changes in the context (e.g. dead nodes, active/inactive actuators).

The mathematical details of STSDP as well as the Ontology implemented will be described by referring to a single cluster of nodes. The generalization to more clusters is straightforward.

3.1 Context Generation and Update Modules

Ontologies are useful tools which enable us to describe a domain of interest (concepts and relationships between them) using a formal language (in our case we chose to adopt the Ontology Web Language, OWL).

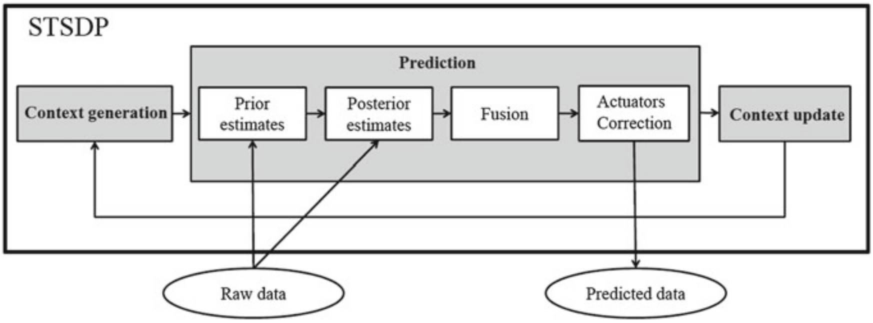

Fig. 2 Sequence diagram of the STSDP module

Classes represent categories of concepts classified by using "isa" hierarchies, whilst *properties* represent relationships among classes and are identified by *domain* and *range* class.

The Simple Web Rule Language (SWRL) implements logical inference and updates the OWL ontology as a function of changes in the environment. Each rule is written in the form $antecedent \rightarrow consequent$ where both *antecedent* and *consequent* are conjunctions of one or more atoms, where each atom is a property or a class.

The context generation module reads the content of the ontology and gathers all the information needed by the prediction module to carry out predictions (e.g. the number of nodes of the cluster, their sampling rate, the active actuators and so on). The context update module modifies the ontology as a function of any environmental changes. For example a node with a battery level under a given threshold should be excluded by the cluster head as its readings could be incorrect.

Table 1 represents in details the "isa" hierarchies we devised for the classes of interest. The root classes are the *Device* and the *Phenomenon*. Any device could be an *Actuator*, a *Reasoner* or a *Node*. Each *Node* in its turn is a *Gateway*, a *Cluster Head* or a *Sensor Node*. Finally, each *Phenomenon* could be *Light*, *Humidity* or *Temperature*.

Table 2 represents the ontology properties. Whenever the range class is a raw type (e.g. String, Float, Integer, Boolean and so on) it is conventionally named as a *data* property, i.e. an internal attribute of the class. Finally, a property is said to be *functional* when the mapping between domain and range is injective. For instance the "hasID" property is functional because each device has just one identifier; the "manages" property is not functional as a device could manage many devices (e.g. each cluster head manages many sensor nodes).

The context update submodule implements three SWRL rules that infer: (i) the observability of a phenomena, (ii) the working status of sensor nodes, and (iii) the manager device of each node (not codified at design time).

The formal representation of the three rules described is as follows:

Table 1 Ontology classes

Class name	Parent class
Device	–
Phenomenon	–
Actuator	Device
Reasoner	Device
Node	Device
Gateway	Node
Cluster head	Node
Sensor node	Node
Light	Phenomenon
Humidity	Phenomenon
Temperature	Phenomenon

Table 2 Ontology properties

Property Name	Domain	Range	Data	Funct.	Description
hasID	Device	Integer	✓	✓	The unique identifier
isAtX	Device		✓	✓	The X-position
isAtY	Device	Float	✓	✓	The Y-position
manages	Device	Device	✗	✗	Controller and controlled device
isManagedBy	Device	Device	✗	✓	Inverse of the manages property
hasSamplingPeriod	Sensor node	Float	✓	✓	Sampling period
hasBatteryLevel	Sensor node	Float	✓	✓	Current battery level
hasMinBatteryLevel	Sensor node	Float	✓	✓	Minimum battery level for considering a node as ”active”
hasStatus	Device	Boolean	✓	✓	Working status (active or inactive)
senses	Sensor node	Phenomenon	✗	✗	The phenomena sensed by a sensor node
affects	Actuator	Phenomenon	✗	✗	The phenomenon affected by the given actuator
isObservable	Phenomenon	Boolean	✓	✓	Observability of a phenomenon

Rule 1:
Phenomenon(?x) $\wedge$ senses(?y, ?x) $\wedge$ hasStatus(?y,True) $\rightarrow$ isObservable(?x,True)

Rule 2:
Node(?y) $\wedge$ hasBatteryLevel(?y,?t) $\wedge$ hasMinBatteryLevel(?y,?z) $\wedge$ lessOrEqual (?t,?z) $\rightarrow$

hasWorkingStatus(?y,False)

Rule 3:
manages(?x,?y) → isManagedBy(?y,?x)

3.2 Prediction Submodule

Let us assume that each sensor node i gathers readings with sampling rate Δt and that $r_i(t)$ is the reading at time t. Moreover the node location is (x_i, y_i).

The prior estimator $f^{prior}_{x_i,y_i}(t)$ roughly assumes that the current representation of the monitored phenomena is identical to that of the previous day; the past 24 h of readings are fitted using a Gaussian Mixture as follows:

$$f^{prior}_{x_i,y_i}(t) = \sum_{k=1}^{K} w_k N(t|\mu_k, \sigma_k^2) \tag{1}$$

where the parameters μ_k, σ_k^2 and w_k represent the mean, the variance and the importance weight of the k-th Gaussian. Such parameters minimize the square error between the fitted curve and the sensed readings and are recomputed with 24-h time steps using the Nelder-Mead optimization algorithm [14] as follows:

$$(\mu_1, \sigma_1, w_1, ..., \mu_K, \sigma_K, w_k) = \underset{\substack{\mu_k, \sigma_k, w_k \\ k\in[1..K]}}{argmin} \sum_t [r_i(t) - f^{prior}_{x_i,y_i}(t)]^2. \tag{2}$$

The prior estimator enables STSDP to learn the *shape* of the monitored phenomenon and to constrain the trend of the posterior estimator. Its intrinsic limitations are related to poor performance as a short-term predictor and to the covered space that is the point location x_i, y_i.

The posterior estimator integrates the trend of the current readings to the information provided by the above step. Let us consider the time window pinpointed by the last W sensed readings $r(t)$, $t \in [t_0 - (W-1)\Delta t, t_0]$; then the posterior estimator is computed as a geometrical transformation of the prior one as follows:

$$f^{post}_{x_i,y_i}(t) = \beta[f^{prior}_{x_i,y_i}(t) - \gamma] + \gamma, \tag{3}$$

where $\beta \in [\beta_l, \beta_u]$ and $\gamma \in [0, \gamma_u]$ represent respectively the scaling and translation parameters; the upper and lower bounds should be set by a field expert and limit the range of possible transformation of the prior estimator.

The geometrical parameters are computed for non-overlapping time windows of size W using the same approach as Eq. 2:

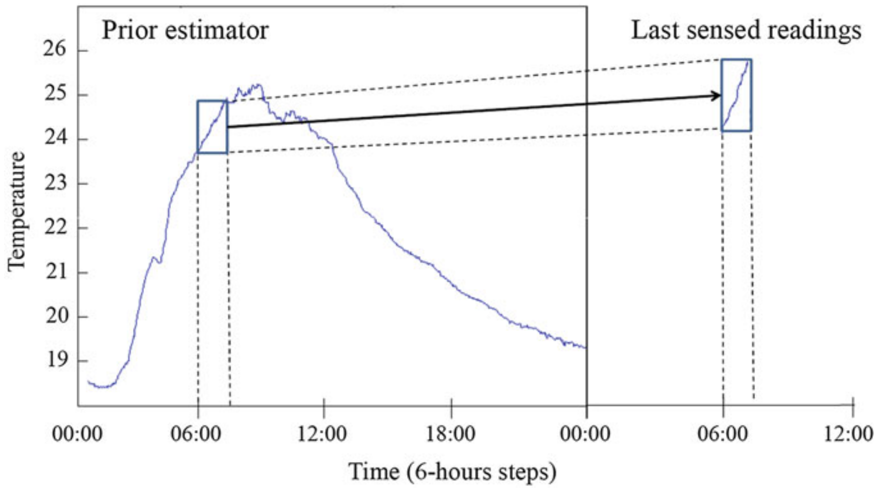

Fig. 3 Geometrical transformation of a prior estimator into a posterior one

$$(\beta, \gamma) = \underset{\beta, \gamma}{argmin} \sum_{t=t_0-(W-1)\Delta t}^{t_0} \{\beta[f_{x_i, y_i}^{prior}(t) - \gamma] + \gamma - r_i(t)\}^2 \tag{4}$$

Figure 3 shows how the computation works: the last W sensed readings identify the current time window (on the right); the same time instants also identify the portion of the prior estimator that is used to fit the current readings (on the left). Then, the geometrical transformation tries to compute the best match between prior estimator and current readings.

The posterior estimators perform better than prior ones as they integrate the knowledge from the previous day's readings and the current behavior of the phenomenon; however they are still limited by punctual spatial coverage.

The fusion step extends the spatial coverage of the posterior predictors to the entire area of interest. The continuous function $f^{fuse}(x, y, t)$ is computed as a normalized linear interpolation of the posterior estimators as follows:

$$f^{fuse}(x, y, t) = \frac{\sum_{i=1}^{I} w_i(x, y) f_{x_i, y_i}^{post}(t)}{\sum_{i=1}^{I} w_i(x, y)}, \tag{5}$$

where $w_i(x, y) = e^{-[(x-x_i)^2+(y-y_i)^2]}$ and I is the number of nodes within the cluster. The function covers the convex-hull of the cluster.

3.3 Modeling the Effect of Actuators

Environmental actuators are an important component of AmI applications that manipulate physical phenomena according to user needs. A crucial aspect of any predictor is its capability of integrating the effect of the actuators on the environment to draw more precise predictions.

The current implementation of the actuators correction submodule considers the effect of rolling shutters and neon lights on indoor environments. We are working to include the support for actuators affecting temperature and humidity such as air conditioners and radiators.

We assessed experimentally that the rolling shutter gives a multiplicative contribution to f^{fuse} and was modeled as a compression function $R : [0, 1] \rightarrow [0, 1]$ that accepts as input the value of $h \in [0, 1]$, i.e. the portion of the window that is not covered by the rolling shutter(0 means totally closed and 1 means totally open), and gives as output $c \in [0, 1]$ the compression factor. We did not include dependence on the space position in $R(h)$ as experimental results (Sect. 4) have shown that such knowledge only makes a negligible contribution to reduce prediction error.

The effect of neon lights is additive, location-dependent and was modeled as a function $N(x, y) : R^2 \rightarrow [0, \infty]$ that accepts as input the 2-D point of the area of interest and gives as output the increment in light exposure. The mathematical expression is as follows:

$$N(x, y) = \frac{\sum_{i=1}^{I} \sum_{l=1}^{L} w_i(x, y) s_l N_l(x_i, y_i)}{\sum_{i=1}^{I} \sum_{l=1}^{L} s_l w_i(x, y)}, \tag{6}$$

where l is the subscript that identifies the neon light, L is the number of neon lights, $w_i(x, y) = e^{-[(x-x_i)^2+(y-y_i)^2]}$ is the importance weight, s_l indicates whether the light is turned off/on, and $N_l(x_i, y_i)$ is the punctual light increment at the node location (x_i, y_i) due to the l-th neon light.

The output of the actuator correction submodule $f^{light}(x, y, t)$ is therefore computed accordingly to the previous considerations:

$$f^{light}(x, y, t) = R(h) \times f^{fuse}(x, y, t) + N(x, y). \tag{7}$$

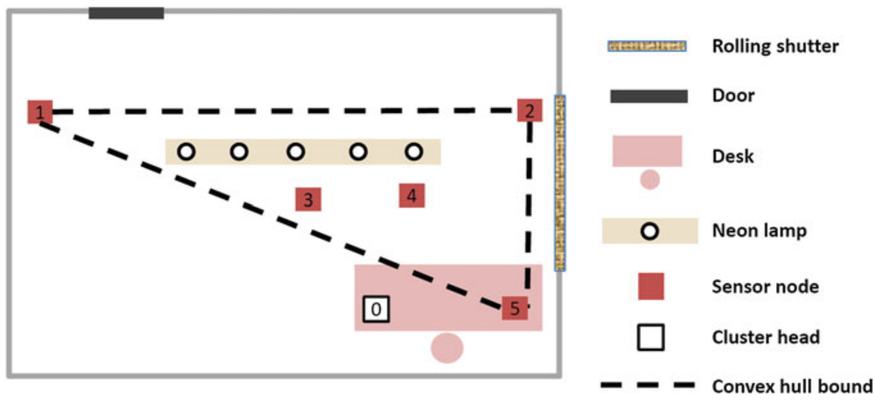

Fig. 4 The Ambient Intelligence scenario we used for evaluating the performance of STSDP

4 Experimental Results

The aim of this Section is to evaluate the performance of the STSDP module with respect to the spatial and temporal precision of the predicted data.

Figure 4 shows the Ambient Intelligence scenario we adopted to test the performance of STSDP. We deployed a single cluster within a 6 m $\times$ 5 m office at our department: the WSN was made up of five Mica2Dot sensor nodes equipped with light, temperature and humidity sensors while the cluster head was a FitPC2i (mini-computer). The monitoring field was bounded by the convex hull of the sensor nodes belonging to the cluster and is marked by a dashed line. Sensor nodes gathered measurements from 07-28-2013 to 08-02-2013 with a sampling rate of 30 s. The first day of readings was used to learn the prior models of the sensor nodes, while from 07-29-2013 to 07-31-2013 we assessed the performances of the predictor without the effect of the actuators. The days 08-01-2013 and 08-02-2013 were used to assess the performance of STSDP under the effect of the light exposure actuators. We set the number of mixing components of the prior estimators to $K = 5$.

4.1 *Prediction Performance*

In order to evaluate the ability of STSDP to predict data over space, we computed the function f^{fuse} by relying only on the measurements gathered from nodes 1, 2 and 5. We then compared the readings from nodes number 3 and 4 with the output provided by f^{fuse} and computed the mean and standard deviation of the prediction error.

Light, temperature and humidity were measured respectively in Lux, Celsius degrees and Percentage. Both the mean and standard deviation were normalized

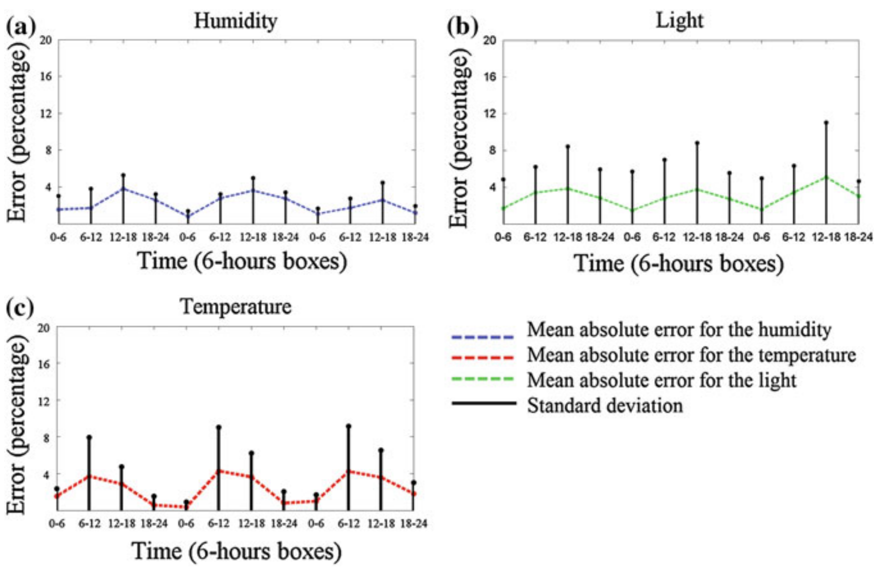

Fig. 5 Assessment of the performance of STSDP as spatial predictor

with respect to the minimum and maximum value of the observed phenomenon: the light ranged from 0 to 1600 Lux, the temperature from 17 to 35 °C, and humidity from 0 to 100 %. Figure 5 shows the results obtained: the x-axis ranges over 3 days and each step aggregates the errors of 6 h (military time). The y-axis contains the percentage mean absolute error and the standard deviation of the error. Experimental results show that the mean and standard deviation are very low and have peaks of about 4 % for all of the phenomena observed.

The temporal prediction performance were assessed by comparing the sensed readings and the value of the posterior predictor f^{post} for each sensor node with different sampling rates (from 0.5 to 60 mins).

Figure 6 shows the mean and standard deviation of the error: both the indicators are above 4 % for all the observed phenomena when the sampling rates range from 0.5 to 5 mins; the performance values are still acceptable for a sampling rate of 60 mins. In particular temperature/humidity and light have a mean error of about 8 and 12 % respectively, meaning that, in general, light exposure is less predictable and has a greater variance than humidity or temperature, a conclusion we had already reached in previous works [5, 6].

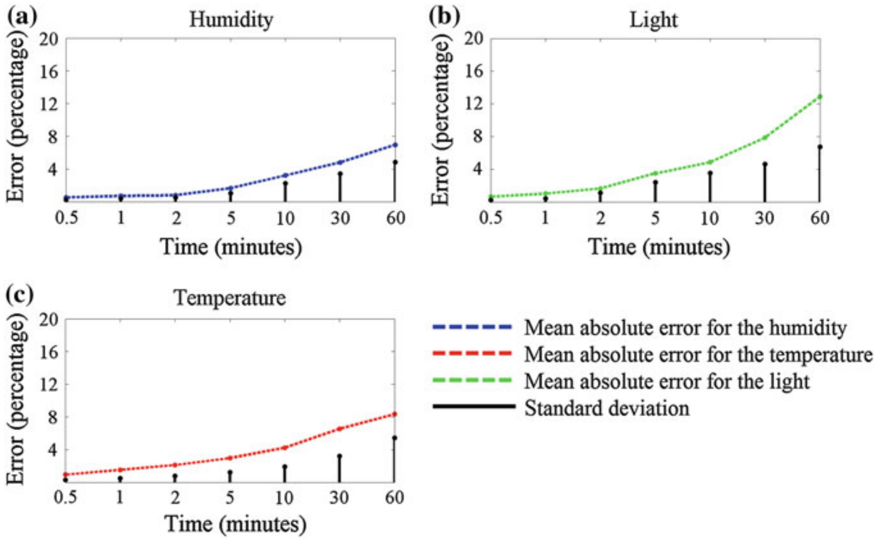

Fig. 6 Assessment of the performances of STSDP as temporal predictor

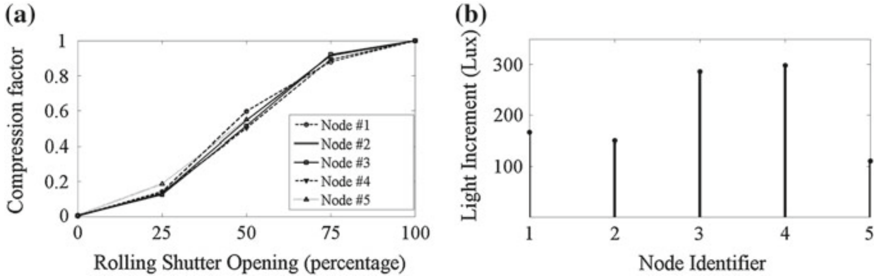

Fig. 7 Effects of the light exposure actuators on the sensor nodes

4.2 Effect of Light Exposure Actuators

The performance of STSDP under the effect of light exposure actuators (neon and rolling shutter) was evaluated during the day 08-02-2013. The functions $R(h)$ and $N(x, y)$ were learned using the readings gathered the previous day. At fixed steps of 1 h, we opened the rolling shutter and positioned it at five different locations (from 0 to 100 % opened) and let the nodes record the differences in the sensed light. We also carried out the same procedure for the neon light which was turned on and off. The results obtained were averaged for each node i over the 24 recorded values and the resulting curves $R_i(h)$ and $N(x, y)$ are reported in Fig. 7.

The compression functions are very similar to each other, so we excluded spatial dependence and computed $R(h)$ as the mean of the learned curves. The additive

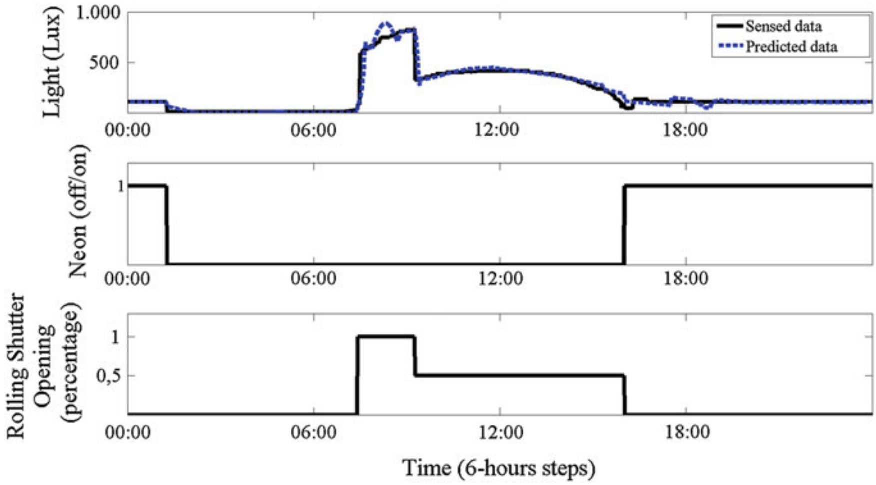

Fig. 8 Prediction of light exposure under the effect of a rolling shutter and a neon light

terms $N(x_i, y_i)$ are location dependent and based on the distance between the sensor node and the neon light; sensor 3 and 4 present two peaks as they are closer to the neon than the other nodes.

Figure 8 shows the performance of the actuators correction module for sensor node 5.

The rolling shutter was kept opened from 07:00 a.m. to 16:30 a.m. while the neon light was turned on for the remaining hours. The sensed and the predicted data are represented by the solid and dashed lines respectively.

The performance of the light exposure predictor appeared to be very encouraging and showed small errors even in correspondence of the transitions caused by actuators. At 08:20 am the predictions became unreliable, but recovered after about 20 mins. The problem was caused by a sequence of suboptimal solutions provided by the optimization algorithm that computes the geometrical transformation parameters (see Eq. 4).

5 Conclusions

This work proposes the implementation of Short-Term Sensory Data Prediction (STSDP), a software module for Ambient Intelligence scenarios. The module was able to predict common physical phenomena like temperature, humidity and light exposure even with the effect of environmental actuators.

The OWL ontology made it possible to describe the environmental context and the relationships among the components of the AmI reference architecture whilst keeping information about the state of the sensor network and actuators updated.

The experimental results were achieved using real data gathered in an office at our department and demonstrated that STSDP is able to provide reliable predictions both in space and time with ranges of meters and minutes respectively. We also assessed its capabilities in predicting light exposure with the effects of a neon light and a rolling shutter.

STSDP was implemented as a set of interconnected sub-modules that could be independently improved using more refined mathematical models. As a further development we are currently integrating the support for air conditioners and radiators to extend its applicability to more complex Ambient Intelligence scenarios.

Acknowledgments This work has been partially supported by the PO FESR 2007/2013 grant G73F11000130004 funding the SmartBuildings project.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Commun. Mag. **40**(8), 102–114 (2002)
2. Augusto, J.C., Nugent, C.D.: The use of temporal reasoning and management of complex events in smart homes. In: ECAI, vol. 16, p. 778 (2004)
3. Chong, S.K., Gaber, M.M., Krishnaswamy, S., Loke, S.W.: Energy-aware data processing techniques for wireless sensor networks: a review. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems III, pp. 117–137. Springer (2011)
4. De Paola, A., Gaglio, S.: Lo Re, G., Ortolani, M.: Sensor9k: A testbed for designing and experimenting with WSN-based ambient intelligence applications. Pervasive Mob. Comput. **8**(3), 448–466 (2012)
5. De Paola, A., Lo Re, G., Milazzo, F., Ortolani, M.: Adaptable data models for scalable ambient intelligence scenarios. In: International Conference on Information Networking (ICOIN), pp. 80–85 (2011)
6. De Paola, A., Lo Re, G., Milazzo, F., Ortolani, M.: Predictive models for energy saving in wireless sensor networks. In: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–6 (2011)
7. De Paola, A., Lo Re, G., Morana, M., Ortolani, M.: An intelligent system for energy efficiency in a complex of buildings. In: Sustainable Internet and ICT for Sustainability (SustainIT), pp. 1–5 (2012)
8. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: Scenarios for ambient intelligence in 2010. Office for official publications of the European Communities (2001)
9. Gaggioli, A.: Optimal experience in ambient intelligence. Ambient intelligence pp. 35–43 (2005)
10. Guestrin, C., Bodik, P., Thibaux, R., Paskin, M., Madden, S.: Distributed regression: an efficient framework for modeling sensor network data. In: IEEE Third International Symposium on Information Processing in Sensor Networks (IPSN), pp. 1–10 (2004)
11. Jain, A., Chang, E.Y., Wang, Y.F.: Adaptive stream resource management using kalman filters. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, pp. 11–22 (2004)
12. Kanagal, B., Deshpande, A.: Online filtering, smoothing and probabilistic modeling of streaming data. In: IEEE 24th International Conference on Data Engineering, pp. 1160–1169 (2008)
13. Le Borgne, Y.A., Santini, S., Bontempi, G.: Adaptive model selection for time series prediction in wireless sensor networks. Signal Process. **87**(12), 3010–3020 (2007)

14. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)
15. Oldewurtel, F., Parisio, A., Jones, C.N., Gyalistras, D., Gwerder, M., Stauch, V., Lehmann, B., Morari, M.: Use of model predictive control and weather forecasts for energy efficient building climate control. Energy Buildings **45**, 15–27 (2012)
16. Tulone, D., Madden, S.: Paq: Time series forecasting for approximate query answering in sensor networks. In: Wireless Sensor Networks, pp. 21–37. Springer (2006)
17. Ye, F., Zhong, G., Cheng, J., Lu, S., Zhang, L.: Peas: A robust energy conserving protocol for long-lived sensor networks. In: IEEE Proceedings of the 23rd International Conference on Distributed Computing Systems, pp. 28–37 (2003)